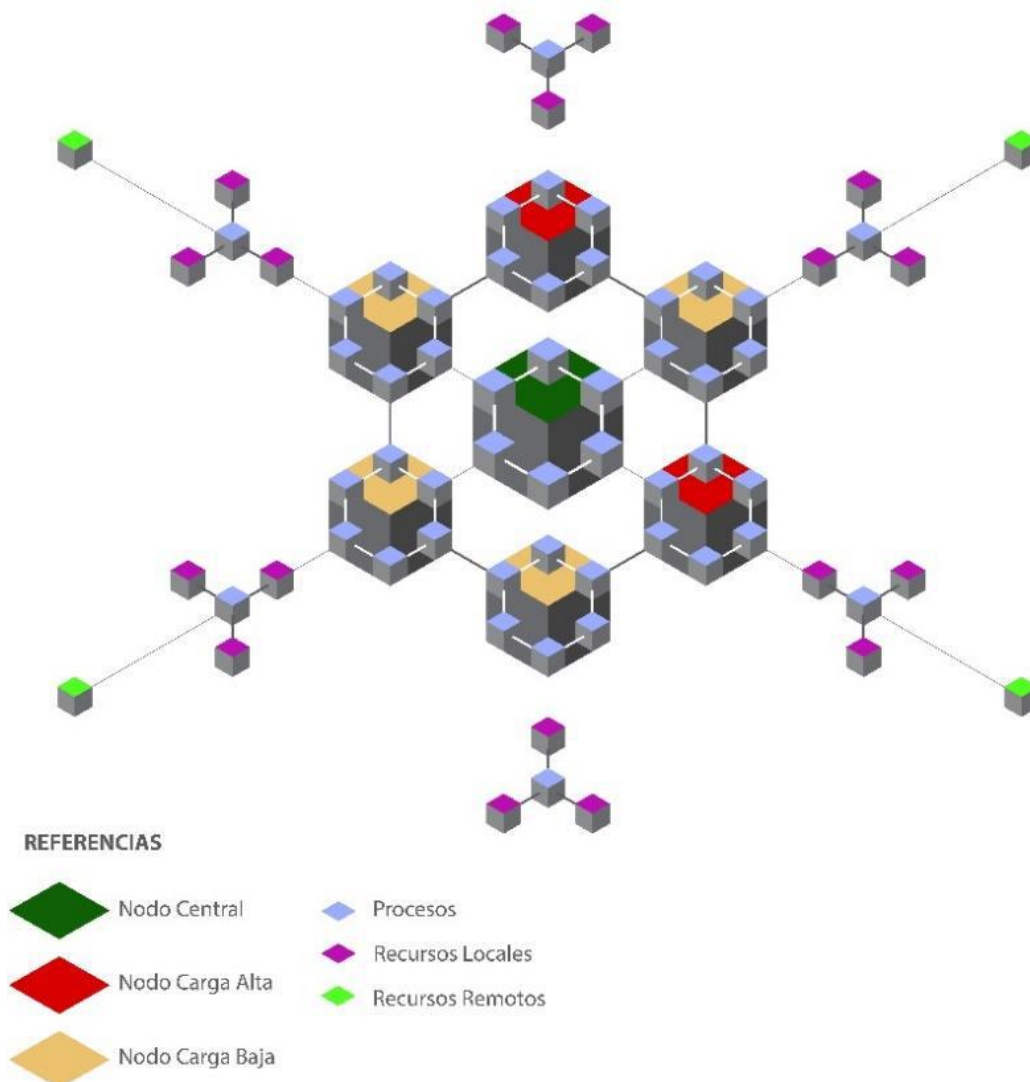


UNIVERSIDAD NACIONAL DE PILAR – FACULTAD DE CIENCIAS APLICADAS
GRUPO DE INVESTIGACION DE SISTEMAS DISTRIBUIDOS

Tesis de Maestría en Informática y Computación



UNIVERSIDAD NACIONAL DE PILAR

FACULTAD DE CIENCIAS APLICADAS

Prof. Mgtr. Jorge Forneron Martínez, Decano

Prof. Mgtr. Miguel Angel Delpino Aguayo, Vicedecano

Miembro del Consejo Directivo

Dra. Luisa Gamarra, consejera Docente

Lic. Luis Alberto Fossati Dávalos, consejero Docente

Ing. Rubén Darío Fornerón Portillo, consejero Docente

Lic. Juan Mora, consejero Docente

MSc. Miguel Ángel Benítez, consejero Docente

Lic. Sonia Escobar, consejera no Docente

Est. Rolando Concepción Vázquez Cuenca, representante Estudiantil

Est. Malena del Pilar Gamarra Gaona, representante Estudiantil

Editor Responsable

Dra. Nélica Soria Rey

Comité Editorial

Dra. Lourdez Sánchez de Velaustegui

MSc. María Elena López de Silva

Mgtr. Miguel Ángel Delpino Aguayo



Presentación

Continuamos el proceso de publicación de los trabajos de las tesis aprobadas en el contexto del programa de Maestría en Informática y Computación, presentadas por integrantes que conforman el Grupo de Investigación en Sistemas Distribuidos de la Facultad de Ciencias Aplicadas.

En el presente número especial, se publica la tesis del Magíster **Terecio Diosnel Marecos Brizuela**, referido al tema “Migración controlada de procesos en sistemas distribuidos”, desarrollado con el propósito de contribuir al conocimiento y/o aportar soluciones innovadoras al problema planteado.

A handwritten signature in blue ink, reading "Nélide Soria". The signature is fluid and cursive, with a horizontal line underlining the name.

Dra. Nélide Soria Rey

Editora

Editorial

En la presente edición especial, me enorgullece presentar la Tesis de Maestría de Terecio Diosnel Marecos Brizuela, que aborda uno de los desafíos más críticos en el ámbito de los sistemas distribuidos: la migración de procesos. En el trabajo denominado “Migración controlada de procesos en sistemas distribuidos” se ofrece una visión innovadora y meticulosamente desarrollada sobre cómo gestionar los recursos compartidos en entornos complejos de procesamiento distribuido.

Los sistemas distribuidos son el corazón de muchas infraestructuras tecnológicas modernas, permitiendo que múltiples procesos operen de manera simultánea y coordinada a través de diferentes nodos. Sin embargo, a medida que estos sistemas se vuelven más complejos, la sincronización y la operatividad conjunta de los procesos se convierten en un reto cada vez más difícil de superar. Esta Tesis se enfoca en la creación de modelos de decisión que no solo consideran la migración de procesos entre nodos, sino que también lo hacen de una manera controlada y eficiente, priorizando el balance de carga, el consumo energético y las necesidades técnicas específicas de cada entorno.

Uno de los aportes más significativos de este estudio es la propuesta de un modelo de migración inteligente de procesos. Este modelo tiene como objetivo principal corregir los desequilibrios en la distribución de cargas entre los nodos, ya sean estos homogéneos o heterogéneos. Mediante una metodología cuantitativa, se establecen relaciones claras y generalizables entre los procesos y los recursos que estos requieren, permitiendo una toma de decisiones fundamentada en datos empíricos y simulaciones de alto rigor técnico.

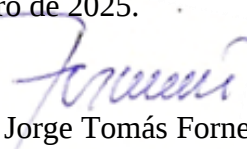
La simulación de modelos de decisión y el uso de operadores de agregación para analizar la información empírica son técnicas clave que respaldan la validez de este enfoque. Los resultados obtenidos a través de estas técnicas muestran que el modelo propuesto supera a los modelos clásicos en términos de eficiencia y equilibrio de la carga de trabajo, confirmando su potencial para mejorar la gestión de los recursos en sistemas distribuidos.

Además, este trabajo no solo presenta soluciones a los problemas de sobrecarga, sino que también sugiere caminos para futuras investigaciones. La posibilidad de integrar algoritmos basados en aprendizaje automático para afinar aún más la precisión en la toma de decisiones es una línea de investigación prometedora que podría llevar a desarrollos aún más robustos en este campo.

En un mundo donde la demanda por sistemas distribuidos más rápidos, eficientes y confiables sigue creciendo, estudios como este son esenciales, donde, con enfoques innovadores y bien fundamentados, es posible avanzar hacia un futuro en el que la tecnología no solo responda a las necesidades actuales, sino que también esté preparada para los desafíos futuros.

Invitamos a nuestros lectores a explorar a fondo el trabajo de Marecos Brizuela y a considerar las implicaciones que tiene no solo para la academia, sino también para la industria y el desarrollo tecnológico global. Sin duda, las contribuciones aquí presentadas marcarán un hito importante en el campo de los sistemas distribuidos y servirán de base para otras investigaciones que continúen mejorando la capacidad para gestionar eficientemente los recursos en entornos distribuidos.

Pilar, febrero de 2025.



Prof. Mgtr. Jorge Tomás Fornerón Martínez

UNIVERSIDAD NACIONAL DE PILAR

FACULTAD DE CIENCIAS APLICADAS



Migración controlada de procesos en sistemas distribuidos

Terecio Diosnel Marecos Brizuela

Orientador: **Prof. Mgter. Federico Agostini**

Coorientador: **Prof. Mgter. Jorge Tomás Fornerón Martínez**

Tesis presentada a la Facultad de Ciencias Aplicadas de la Universidad Nacional de Pilar, para la obtención del Grado de Magíster en Informática y Computación, correspondiente al Programa de Maestría en Informática y Computación.

PILAR – PARAGUAY

DICIEMBRE 2023



Migración controlada de procesos en sistemas distribuidos

Terecio Diosnel Marecos Brizuela

Orientador: **Prof. Mgter. Federico Agostini**

Coorientador: **Prof. Mgter. Jorge Tomás Fornerón Martínez**

Tesis presentada a la Facultad de Ciencias Aplicadas de la Universidad Nacional de Pilar, para la obtención del Grado de Magíster en Informática y Computación, correspondiente al Programa de Maestría en Informática y Computación.

PILAR – PARAGUAY

DICIEMBRE 2023

Terecio Diosnel Marecos Brizuela. (2023). **Migración controlada de procesos en sistemas distribuidos**. Terecio Diosnel Marecos Brizuela. 216 p.: il.

Orientador: Prof. Mgter. Federico Agostini

Coorientador: Prof. Mgter. Jorge Tomás Fornerón Martínez

Tesis académica de Magíster en Informática y Computación - Facultad de Ciencias Aplicadas -
Universidad Nacional de Pilar



MIGRACIÓN CONTROLADA DE PROCESOS EN SISTEMAS DISTRIBUIDOS

TERECIO DIOSNEL MARECOS BRIZUELA

Aprobado en fecha: 28/12/2023 (Res. N° 345/2023 CD-FCA).

Tribunal Examinador: Mgtr. Liliana Cuenca Pletch

Mgtr. Carlos Brys

Dra. Sonia I. Mariño

Prof. Mgter. FEDERICO AGOSTINI

Orientador

Prof. Mgter. JORGE TOMÁS FORNERÓN MARTÍNEZ

Coorientador

Prof. Mgter. JORGE TOMÁS FORNERON MARTÍNEZ

Decano

Dedicatoria

El presente trabajo de investigación lo dedico a mi esposa Adriana Benítez que me brindó su apoyo y comprensión en todo momento. A mis hijos, Benjamín Ezequiel y Gonzalo Ariel, porque cedieron su valioso tiempo de juegos y travesuras para que papá estudie.

A mis padres y hermanos que me brindaron su apoyo incondicional en todo momento.

Agradecimiento

A Dios, por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional.

Al Prof. Mgter. Federico Agostini (UNNE), Orientador de Tesis, por la confianza hacia mi persona, el esfuerzo y por la predisposición de trabajar sin importar el horario para poder realizar este trabajo de investigación.

Al Prof. Dr. David Luis la Red Martínez (UNNE), director del Programa Maestría en Informática y Computación de la Facultad de Ciencias Aplicadas de la UNP, por su tiempo, por compartir sus experiencias, conocimientos, sus consejos e indicaciones.

Al Decano Prof. Mgter. Jorge Tomás Fornerón y en su persona a toda la familia de la Facultad de Ciencias Aplicadas de la Universidad Nacional de Pilar por la oportunidad y el espacio educativo brindado.

A todos los Docentes que integran el programa de la Maestría en Informática y Computación de la Facultad de Ciencias Aplicadas de la UNP.

A todas las personas que de alguna u otra manera hicieron posible lograr este objetivo.

Índice

Dedicatoria.....	v
Agradecimiento	vi
Listado de tablas	x
Lista de figuras	xvii
Resumen.....	xviii
Capítulo I - Introducción.....	1
1.1.Introducción.....	1
1.2.El problema	3
1.3.Objetivos.....	7
1.4.Justificación.....	7
1.5.Hipótesis y objeto	8
1.6.Antecedentes	8
1.7.Encuadre o Marco Teórico	10
1.8.Relevancia	17
1.9.Metodología.....	19
1.10.Estructura de la tesis	24
1.11.Discusiones y comentarios	25
Capítulo II – Escenario 1. Migración de procesos en condiciones de homogeneidad	28
2.1.Trabajos previos	28
2.2.Premisas	30
2.3.Propuesta de solución	32
2.4.Descripción del operador de agregación.....	37
2.4.1.Cálculo de la carga computacional actual de los nodos. Escenario 1 (E1)	40
2.4.2.Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos (E1).....	43
2.4.3.Selección de los posibles nodos destino mediante la evaluación de su carga (E1).....	45
2.4.4.Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E1).....	46
2.4.5.Determinación del orden de migración de procesos y de los nodos destino correspondientes (E1)	52

2.4.6.Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema (E1).....	58
2.4.7.Métrica propuesta para evaluar el balanceo de carga en el sistema distribuido (E1).....	62
2.5.Ejemplo	63
2.5.1.Cálculo de la carga computacional actual de los nodos (E1).....	65
2.5.2.Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos (E1).....	74
2.5.3.Selección de los posibles nodos destino mediante la evaluación de su carga (E1).....	76
2.5.4.Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E1).....	76
2.5.5.Determinación del orden de migración de procesos y de los nodos destino correspondientes (E1).....	84
2.5.6.Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema (E1).....	93
2.5.7.Métrica propuesta para evaluar el balanceo de carga en el sistema distribuido (E1).....	96
2.6.Comentarios y discusiones	98
Capítulo III – Escenario 2. Migración de procesos en condiciones de heterogeneidad.....	100
3.1.Trabajos previos	100
3.2.Premisas	101
3.3.Propuesta de solución	101
3.4.Descripción del operador de agregación.....	103
3.4.1.Cálculo de la carga computacional actual de los nodos. Escenario 2 (E2)	107
3.4.2.Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos (E2)	108
3.4.3.Selección de los posibles nodos destino mediante la evaluación de compatibilidad y la carga (E2).....	109
3.4.4.Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E2).....	111
3.4.5.Establecimiento de las categorías nodales según sus prestaciones (E2)	117
3.4.6.Determinación del orden de migración de procesos y de los nodos destino correspondientes (E2).....	120
3.4.7.Evaluación de nuevos valores de sobrecarga de los procesos inmigrantes según categoría del nodo receptor (E2)	122
3.4.8.Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento	

global del sistema (E2).....	124
3.4.9.Evaluación del consumo de energía de los nodos una vez migrados los procesos (E2)	125
3.4.10.Métrica propuesta para evaluar el balanceo de carga en el sistema distribuido (E2)	125
3.5.Ejemplo	127
3.5.1.Cálculo de la carga computacional actual de los nodos (E2).....	129
3.5.2.Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos (E2)	136
3.5.3.Selección de los posibles nodos destino mediante la evaluación de compatibilidad y la carga (E2).....	136
3.5.4.Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E2).....	140
3.5.5.Establecimiento de las categorías nodales según sus prestaciones (E2)	148
3.5.6.Determinación del orden de migración de procesos y de los nodos destino correspondientes (E2).....	153
3.5.7.Evaluación de nuevos valores de sobrecarga de los procesos inmigrantes según categoría del nodo receptor (E2)	164
3.5.8.Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema (E2).....	173
3.5.9.Evaluación del consumo de energía de los nodos una vez migrado los procesos (E2).....	174
3.5.10.Métrica propuesta para evaluar el balanceo de carga y consumo de energía en el sistema distribuido (E2)	176
3.6.Comentarios y discusiones	178
Capítulo IV. Comparación de resultados.....	180
Capítulo V - Conclusiones y futuras líneas de investigación	184
Bibliografía	187
Apéndice de publicaciones.....	195

Listado de tablas

Tabla 1. Procesos en cada nodo (E1)	33
Tabla 2. Recursos compartidos disponibles en cada nodo (E1)	34
Tabla 3. Recursos solicitados por los procesos (E1)	34
Tabla 4. Promedio de prioridad de procesos (E1)	44
Tabla 5. Nodos con carga baja (E1)	45
Tabla 6. Matriz general de procesos (E1)	52
Tabla 7. Matriz primaria (E1)	53
Tabla 8. Función de asignación para migración en sistemas distribuidos (<i>FAMSD</i>) primaria (E1).....	55
Tabla 9. Matriz secundaria (E1).....	55
Tabla 10. Función de asignación para migración en sistemas distribuidos (<i>FAMSD</i>) secundaria (E1)	56
Tabla 11. Matriz de procesos no migrables (E1).....	57
Tabla 12. Concatenación de las tablas de asignación (<i>FAMSD</i> primaria y <i>FAMSD</i> secundaria) (E1).....	58
Tabla 13. Criterios para medir la carga computacional en cada nodo (E1).....	60
Tabla 14. Procesos en cada nodo (E1)	64
Tabla 15. Recursos en cada nodo (E1).....	64
Tabla 16. Recursos solicitados por los procesos (E1)	64
Tabla 17. Valores de los criterios de carga computacional (E1).....	66
Tabla 18. Valores de las categorías para medir la carga computacional (E1)	66
Tabla 19. Pesos asignados a los criterios para calcular la prioridad (Nº Procesos, %CPU, %Memoria, %MV, Prioridad Proceso) (E1)	67
Tabla 20. Pesos asignados a los criterios para calcular la prioridad (Peso del Recurso, Impacto Transferencia, Sobrecarga Memoria, Sobrecarga Procesador, Sobrecarga E/S) (E1).....	68
Tabla 21. Valoraciones asignadas a los criterios para calcular la prioridad (Nº proceso, %CPU, %Memoria, %MV, Prioridad Proceso) (E1).....	68
Tabla 22. Valoraciones asignadas a los criterios para calcular la prioridad (Peso del Recurso, Impacto Transferencia, Sobrecarga Memoria, Sobrecarga Procesador, Sobrecarga E/S) (E1).....	71
Tabla 23. Lista de procesos con prioridad de procesos superior al límite (E1)	75

Tabla 24. Vector de migración del proceso p_{11} (E1)	77
Tabla 25. Vector de migración del proceso p_{12} (E1)	77
Tabla 26. Vector de migración del proceso p_{13} (E1)	78
Tabla 27. Vector de migración del proceso p_{14} (E1)	79
Tabla 28. Vector de migración del proceso p_{15} (E1)	79
Tabla 29. Vector de migración del proceso p_{17} (E1)	80
Tabla 30. Vector de migración del proceso p_{31} (E1)	80
Tabla 31. Vector de migración del proceso p_{32} (E1)	81
Tabla 32. Vector de migración del proceso p_{34} (E1)	81
Tabla 33. Vector de migración del proceso p_{35} (E1)	82
Tabla 34. Vector de migración del proceso p_{36} (E1)	83
Tabla 35. Vector de migración del proceso p_{37} (E1)	83
Tabla 36. Matriz general de procesos (MGP) (E1)	83
Tabla 37. Matriz primaria (MP). Primera ronda (E1)	84
Tabla 38. Matriz secundaria (MS). Primera ronda (E1)	85
Tabla 39. Función de Asignación para Migración en Sistemas Distribuidos ($FAMSD$) primaria. Primera ronda (E1)	85
Tabla 40. Matriz primaria (MP). Segunda ronda (E1)	86
Tabla 41. Matriz secundaria (MS). Segunda ronda (E1)	86
Tabla 42. Matriz de procesos no migrables. Segunda ronda (E1)	86
Tabla 43. Función de Asignación para Migración en Sistemas Distribuidos ($FAMSD$) primaria. Segunda ronda (E1)	87
Tabla 44. Matriz primaria (MP). Tercera ronda (E1)	87
Tabla 45. Matriz secundaria (MS). Tercera ronda (E1)	87
Tabla 46. Matriz de procesos no migrables. Tercera ronda (E1)	87
Tabla 47. Función de Asignación para Migración en Sistemas Distribuidos ($FAMSD$) primaria. Tercera ronda (E1)	88
Tabla 48. Matriz primaria (MP). Cuarta ronda (E1)	88
Tabla 49. Matriz secundaria (MS). Cuarta ronda (E1)	89

Tabla 50. Matriz de procesos no migrables. Cuarta ronda (E1)	89
Tabla 51. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) primaria. Cuarta ronda (E1)	89
Tabla 52. Matriz primaria (<i>MP</i>). Quinta ronda (E1)	90
Tabla 53. Matriz secundaria (<i>MS</i>). Quinta ronda (E1)	90
Tabla 54. Matriz de procesos no migrables. Quinta ronda (E1)	90
Tabla 55. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) primaria. Quinta ronda (E1)	91
Tabla 56. Matriz secundaria (<i>MS</i>). Sexta ronda (E1)	91
Tabla 57. Matriz de procesos no migrables. Sexta ronda (E1)	91
Tabla 58. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) secundaria. Sexta ronda (E1)	92
Tabla 59. Función de Asignación para Migración en Sistemas Distribuidos concatenada (<i>FAMSDC</i>) (E1)	92
Tabla 60. Evaluación de promedios de sobrecarga del nodo 1 (E1)	93
Tabla 61. Evaluación de promedios de sobrecarga del nodo 2 (E1)	93
Tabla 62. Evaluación de promedios de sobrecarga del nodo 3 (E1)	94
Tabla 63. Evaluación de promedios de sobrecarga del nodo 4 (E1)	94
Tabla 64. Evaluación de promedios de sobrecarga del nodo 5 (E1)	94
Tabla 65. Evaluación de promedios de sobrecarga del nodo 6 (E1)	95
Tabla 66. Evaluación de promedios de sobrecarga del nodo 7 (E1)	95
Tabla 67. Evaluación de promedios de sobrecarga del nodo 8 (E1)	95
Tabla 68. Carga computacional sin migración (E1)	96
Tabla 69. Carga computacional con migración (E1)	96
Tabla 70. Componentes de hardware. (E2)	109
Tabla 71. Componentes de software (E2)	110
Tabla 72. Nodos compatibles con carga baja (E2)	111
Tabla 73. Matriz general de pesos de migración (E2)	116
Tabla 74. Características específicas del componente de hardware x (E2)	117
Tabla 75. Vector de peso para características específicas de cada componente (E2)	118
Tabla 76. Potencia del componente de hardware X . (E2)	118
Tabla 77. Potencia de componentes de hardware. (E2)	119

Tabla 78. Vector de pesos para el índice ponderado de potencia. (E2).....	119
Tabla 79. Potencia nodal (E2).....	120
Tabla 80. Matriz general de impacto de migración (E2).....	121
Tabla 81. Valores de sobrecarga en nodo origen (E2)	122
Tabla 82. Tabla de conversión de categorías nodales (E2)	123
Tabla 83. Valores de sobrecarga en nodo destino (E2)	124
Tabla 84. Procesos en cada nodo (E2)	128
Tabla 85. Recursos compartidos disponibles en cada nodo (E2)	128
Tabla 86. Recursos solicitados por los procesos (E2)	128
Tabla 87. Valores de los criterios de carga computacional (E2)	129
Tabla 88. Valores de las categorías para medir la carga computacional (E2)	130
Tabla 89. (E2) Valoraciones asignadas a los criterios para calcular la prioridad (N° proceso, %CPU, %Memoria, %MV, Prioridad Proceso) (E2)	130
Tabla 90. (E2) Valoraciones asignadas a los criterios para calcular la prioridad (Peso del Recurso, Impacto Transferencia, Sobrecarga Memoria, Sobrecarga Procesador, Sobrecarga E/S) (E2).....	133
Tabla 91. Lista de procesos con prioridad de procesos superior al límite (E2)	136
Tabla 92. Componentes de hardware (E2)	137
Tabla 93. Componentes de software (E2)	137
Tabla 94. Vector de migración del proceso p_{11} (E2)	141
Tabla 95. Vector de migración del proceso p_{12} (E2)	142
Tabla 96. Vector de migración del proceso p_{13} (E2)	142
Tabla 97. Vector de migración del proceso p_{14} (E2)	143
Tabla 98. Vector de migración del proceso p_{15} (E2)	143
Tabla 99. Vector de migración del proceso p_{17} (E2)	144
Tabla 100. Vector de migración del proceso p_{31} (E2)	145
Tabla 101. Vector de migración del proceso p_{32} (E2)	145
Tabla 102. Vector de migración del proceso p_{35} (E2)	146
Tabla 103. Vector de migración del proceso p_{36} (E2)	146

Tabla 104. Vector de migración del proceso p_{37} (E2)	147
Tabla 105. Matriz general de pesos de migración ($MGPM$) (E2)	147
Tabla 106. Componentes de hardware (E2).....	148
Tabla 107. Valores de atributos del componente CPU (E2)	148
Tabla 108. Vector de peso CPU (E2).....	149
Tabla 109. Potencia del componente CPU (E2).....	149
Tabla 110. Valores de atributos del componente memoria RAM (E2)	150
Tabla 111. Vector de peso memoria RAM (E2)	150
Tabla 112. Potencia del componente memoria RAM (E2)	150
Tabla 113. Valores de atributos del componente disco duro (E2).....	151
Tabla 114. Vector de peso Disco Duro (E2)	151
Tabla 115. Potencia del componente memoria disco duro (E2).....	151
Tabla 116. Potencia de componentes de hardware. (E2).....	152
Tabla 117. Vector de pesos para el índice ponderado de potencia (E2).....	152
Tabla 118. Potencia nodal (E2).....	152
Tabla 119. Clasificación de nodos según sus prestaciones (E2)	153
Tabla 120. Matriz general de pesos de migración normalizado ($MGPMN$) (E2).....	154
Tabla 121. Potencia nodal invertido y normalizado (E2).....	154
Tabla 122. Matriz general de impacto de migración (MGI) (E2).....	155
Tabla 123. Matriz primaria (MP). Primera ronda (E2)	156
Tabla 124. Matriz secundaria (MS). Primera ronda (E2)	156
Tabla 125. Función de Asignación para Migración en Sistemas Distribuidos ($FAMSD$) primaria. Primera ronda (E2)	156
Tabla 126. Matriz primaria (MP). Segunda ronda (E2)	157
Tabla 127. Matriz secundaria (MS). Segunda ronda (E2)	157
Tabla 128. Matriz de procesos no migrables. Segunda ronda (E2).....	157
Tabla 129. Función de Asignación para Migración en Sistemas Distribuidos ($FAMSD$) primaria. Segunda ronda (E2)	158

Tabla 130. Matriz primaria (<i>MP</i>). Tercera ronda (E2).....	158
Tabla 131. Matriz secundaria (<i>MS</i>). Tercera ronda (E2).....	159
Tabla 132. Matriz de procesos no migrables. Tercera ronda (E2)	159
Tabla 133. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) primaria. Tercera ronda (E2)	159
Tabla 134. Matriz secundaria (<i>MS</i>). Cuarta ronda (E2)	160
Tabla 135. Matriz de procesos no migrables. Cuarta ronda (E2)	160
Tabla 136. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) secundaria. Cuarta ronda (E2)	160
Tabla 137. Matriz secundaria (<i>MS</i>). Quinta ronda (E2)	161
Tabla 138. Matriz de procesos no migrables. Quinta ronda (E2).....	161
Tabla 139. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) secundaria. Quinta ronda (E2)	161
Tabla 140. Matriz secundaria (<i>MS</i>). Sexta ronda (E2)	162
Tabla 141. Matriz de procesos no migrables. Sexta ronda (E2).....	162
Tabla 142. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) secundaria. Sexta ronda (E2)	162
Tabla 143. Matriz secundaria (<i>MS</i>). Séptima ronda (E2).....	163
Tabla 144. Matriz de procesos no migrables. Séptima ronda (E2).....	163
Tabla 145. Función de Asignación para Migración en Sistemas Distribuidos (<i>FAMSD</i>) secundaria. Séptima ronda (E2)	164
Tabla 146. Función de Asignación para Migración en Sistemas Distribuidos concatenada (<i>FAMSDC</i>) (E2).....	164
Tabla 147. Valores de sobrecarga en nodo origen (E2)	165
Tabla 148. Tabla de conversión de categorías nodales (E2)	168
Tabla 149. Evaluación de promedios de sobrecarga del nodo 1 (E2)	168
Tabla 150. Evaluación de promedios de sobrecarga del nodo 2 (E2)	169
Tabla 151. Evaluación de promedios de sobrecarga del nodo 3 (E2)	169
Tabla 152. Evaluación de promedios de sobrecarga del nodo 4 (E2)	170

Tabla 153. Evaluación de promedios de sobrecarga del nodo 5 (E2)	170
Tabla 154. Evaluación de promedios de sobrecarga del nodo 7 (E2)	171
Tabla 155. Evaluación de promedios de sobrecarga del nodo 8 (E2)	171
Tabla 156. Evaluación de promedios de sobrecarga del nodo 9 (E2)	172
Tabla 157. Evaluación de promedios de sobrecarga del nodo 10 (E2)	172
Tabla 158. Carga computacional sin migración (E2).....	173
Tabla 159. Carga computacional con migración (E2).....	174
Tabla 160. Vector de conversión consumo de energía. (E2).....	174
Tabla 161. Consumo de energía sin migración (E2)	175
Tabla 162. Consumo de energía con migración (E2)	175
Tabla 163. Comparativa de carga computacional con migración de procesos	180
Tabla 164. Comparativa del consumo de energía en entornos heterogéneo.....	181
Tabla 165. Etapas del operado de agregación para cada escenario	183

Lista de figuras

Fig. 1. Representación de un sistema distribuido con cargas nodales.	31
Fig. 2. Etapas del cálculo de migración de procesos (E1).....	40
Fig. 3. Módulo de nodo (E1).....	41
Fig. 4. Módulo proceso (E1)	43
Fig. 5. Recursos locales vs. Recursos remotos (E1).....	46
Fig. 6. Módulo nodo destino (E1).	48
Fig. 7. Vector de migración de proceso (E1)	50
Fig. 8. Función de Asignación para la Migración en Sistemas Distribuidos Concatenada (E1)	58
Fig. 9. Etapas del cálculo de migración de procesos (E2).....	106
Fig. 10. Módulo de nodo (E2).....	107
Fig. 11. Módulo proceso (E2).	108
Fig. 12. Módulo nodo destino (E2)	112
Fig. 13. Vector de migración de procesos (E2).....	116



MIGRACIÓN CONTROLADA DE PROCESOS EN SISTEMAS DISTRIBUIDOS

AUTOR: TERCIO DIOSNEL MARECOS BRIZUELA

ORIENTADOR: PROF. MGTER. FEDERICO AGOSTINI

COORIENTADOR: PROF. MGTER. JORGE TOMÁS FORNERON MARTÍNEZ

Resumen

En entornos de sistemas de procesamiento distribuido, la toma de decisiones respecto al acceso a los recursos compartidos es crucial para asegurar la sincronización y operatividad conjunta de los procesos. Este estudio se enfoca en desarrollar modelos de decisión que consideren la migración controlada de procesos, abordando aspectos como el balanceo de carga, el consumo energético y las necesidades técnicas específicas. Su objetivo principal radica en resolver los desafíos asociados a la sobrecarga en los nodos de sistemas distribuidos.

La investigación propone un modelo innovador de migración inteligente de procesos, priorizando el equilibrio de la carga entre los nodos, tanto en entornos homogéneos como heterogéneos. La estrategia adoptada se basa en una metodología cuantitativa que enfoca la relación entre los procesos y los recursos requeridos. Se seleccionan casos específicos alineados con los objetivos del estudio, buscando establecer relaciones causa-efecto generalizables mediante una lógica que relaciona la teoría con la evidencia empírica.

Las técnicas utilizadas para obtener y analizar información empírica se sustentan en la simulación de modelos de decisión y operadores de agregación. Estos datos permiten comparar el rendimiento de los modelos propuestos con los modelos clásicos, evaluando su desempeño en términos de carga de trabajo y consumo de recursos, con la meta continua de mejorar los modelos implementados.

La evaluación periódica del estado de las cargas, utilizando la desviación estándar como

indicador, señala la necesidad de ajustes para equilibrar la carga de trabajo entre los nodos, especialmente en entornos heterogéneos donde la compatibilidad en las características de los nodos es esencial para la migración. Los resultados obtenidos confirman la eficacia del modelo en la gestión de recursos, corrigiendo desequilibrios en los nodos distribuidos. Las conclusiones resaltan la utilidad de este enfoque novedoso para abordar problemas de sobrecarga y distribución desequilibrada de carga en sistemas distribuidos. Se sugiere la necesidad de investigaciones futuras para desarrollar algoritmos más avanzados, posiblemente utilizando enfoques basados en aprendizaje automático para mejorar la precisión en la toma de decisiones.

Palabras clave: Sistemas Operativos, Migración de Procesos en Sistemas Distribuidos, Balanceo de Carga, Consumo de Energía, Operadores de Agregación, Modelo de Decisión.



MIGRACIÓN CONTROLADA DE PROCESOS EN SISTEMAS DISTRIBUIDOS

AUTOR: TERCIO DIOSNEL MARECOS BRIZUELA

ORIENTADOR: PROF. MGTER. FEDERICO AGOSTINI

COORIENTADOR: PROF. MGTER. JORGE TOMÁS FORNERON MARTÍNEZ

Abstract

In distributed processing systems environments, decision-making regarding access to shared resources is critical to ensure synchronization and joint operation of processes. This study focuses on developing decision models that consider controlled process migration, addressing aspects such as load balancing, energy consumption, and specific technical requirements. Its main objective is to address challenges associated with overload in distributed systems nodes.

The research proposes an innovative model for intelligent process migration, prioritizing load balancing among nodes, both in homogeneous and heterogeneous environments. The adopted strategy is based on a quantitative methodology that focuses on the relationship between processes and required resources. Specific cases aligned with the study's objectives are selected, aiming to establish generalizable cause-and-effect relationships through a logic that connects theory with empirical evidence.

The techniques used to obtain and analyze empirical information are based on simulating decision models and aggregation operators. These data allow for comparing the performance of proposed models with classical models, assessing their performance in terms of workload and resource consumption, with the ongoing goal of improving the implemented models.

Regular evaluation of the state of workloads, using the standard deviation as an indicator, highlights the need for adjustments to balance the workload among nodes, especially in heterogeneous environments where compatibility in node characteristics is essential for migration.

The results obtained confirm the model's effectiveness in resource management, correcting imbalances in distributed nodes. Conclusions highlight the usefulness of this novel approach in addressing overload and uneven load distribution issues in distributed systems. Further research is suggested to develop more advanced algorithms, possibly utilizing machine learning-based approaches to enhance decision-making accuracy.

Keywords: Operating Systems, Process Migration in Distributed Systems, Load Balancing, Energy Consumption, Aggregation Operators, Decision Model.

Capítulo I - Introducción

1.1. Introducción

Algunos trabajos de investigación sobre comunicación en sistemas distribuidos se presentan en (Tanenbaum et al., 1996; Tanenbaum, 2009), (Tanenbaum & Van Steen, 2008) y en (La Red Martínez, 2004), en los cuales se describen los principales algoritmos de comunicación en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

Una investigación relacionada a la sincronización en sistemas distribuidos se presenta en (Agrawal & El Abbadi, 1991) donde se propone una solución eficiente y tolerante a fallas para el problema de la exclusión mutua distribuida. Desde la óptica de las ciencias de la computación, en (Ricart & Agrawala, 1981), (Cao & Singhal, 2001) y en (Lodha & Kshemkalyani, 2000), se presentan algoritmos para gestionar la exclusión mutua en redes de computadoras. Conforme a las ciencias de la computación, en (Stallings, 2005), se detallan los principales algoritmos de las ciencias de la computación para la gestión distribuida de procesos, los estados globales distribuidos y la exclusión mutua distribuida. En (La Red Martínez, 2017), (La Red Martínez et al., 2017) y (La Red Martínez et al., 2018) se desarrollan operadores de agregación para asignaciones de recursos en sistemas distribuidos. En (Agostini & La Red Martínez, 2019) y (Agostini et al., 2019), se presentan modelos de decisión innovadores para la gestión de recursos en sistemas distribuidos.

Investigación relacionada a migración de procesos en sistemas distribuidos se presenta en (Eskicioglu, 1995), (Milojić et al., 2000) y en (Sharma et al., 2012) donde se analiza el campo de la migración de procesos resumiendo los conceptos clave y ofreciendo una visión general de las implementaciones más importantes, sus problemas y beneficios. En (Rathore & Chana, 2014), (Beirut & Ganjali, 2020) y en (Chang et al., 2020) se han propuesto y discutido nuevas técnicas

de migración de procesos. En (Thakkar & Pandya, 2013) se presenta un mecanismo de migración de procesos en un sistema distribuido heterogéneo. En (Wang et al., 2012) se desarrolla un novedoso mecanismo de migración en vivo a nivel de proceso que soporta la ejecución continuada de aplicaciones durante gran parte de la migración de procesos.

Investigaciones relacionadas al balanceo de carga se presentan en (Alakeel, 2010) y en (Maurya & Ahmad, 2013), en este último se desarrolla además, un algoritmo para el equilibrio de carga que calcula dinámicamente la carga de cada nodo. En (Shah & Kapadia, 2012) se desarrollan dos algoritmos para el equilibrio de carga, uno iniciado por el emisor y otro iniciado por el receptor. En (Sánchez Dams, 2014) se propone un algoritmo de balanceo de carga para sistemas homogéneos. En (Chen et al., 2020), (Junaidi et al., 2020) y en (Kim et al., 2021) se proponen algoritmos de aprendizaje automático que utilizan los recursos del servidor, como la CPU y la memoria, para predecir el futuro de las cargas del servidor. En (Banerjee et al., 2019) se describe el diseño de un sistema que aborda el reto de programar cargas de trabajo de análisis de datos distribuidos en redes de procesamiento heterogéneas que incluyen CPUs (Central Processing Unit - Unidad Central de Procesamiento), GPUs (Graphics Processing Unit - Unidad de Procesamiento Gráfico), FPGAs (Field Programmable Gate Arrays - Matriz de Puertas Lógicas Programable en Campo) y ASICs (Application Specific Integrated Circuit - Circuito Integrado para Aplicaciones Específicas) en entornos basados en la nube, dinámicos y de múltiples usuarios.

En investigaciones relacionadas al consumo de energía, en (Duolikun et al., 2014) se propone un algoritmo de migración energéticamente eficiente para que cada proceso decida si el proceso permanece en el servidor actual o es migrado a otro servidor. En (Naghash Asadi et al., 2019a), (Naghash Asadi et al., 2019b), y en (Asadi et al., 2020) se proponen modelos analíticos, basados en las redes de recompensa estocástica. En (Liu et al., 2013) se estima el coste de la

migración en vivo de las máquinas virtuales en términos de rendimiento y energía con un enfoque cuantitativo.

1.2. El problema

Planteamiento del problema

En los sistemas de procesamiento distribuido es frecuentemente necesario coordinar la asignación de recursos compartidos que deben ser asignados a los procesos en la modalidad de exclusión mutua; en tales casos se debe decidir el orden en que los recursos compartidos serán asignados a los procesos que requieren de los mismos (La Red Martínez, 2017).

En muchas ocasiones los nodos pueden generar sobrecarga cuando intentan resolver múltiples asignaciones de recursos a procesos, provocando gran consumo de energía. Otros nodos pueden estar ociosos durante gran parte del tiempo. También existen procesos que requieren ciertas características de poder de cómputo, o recursos especiales. Por lo mencionado anteriormente se considera la necesidad de lograr un balanceo de carga de trabajo, menor consumo de energía y una migración controlada de procesos para mejorar el rendimiento. Para cubrir estas necesidades se plantean las siguientes propuestas:

- **Escenario 1:** migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo homogeneidad (respecto de compatibilidad de código, SO (Sistema Operativo), arquitectura y set de instrucciones).
- **Escenario 2:** migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo heterogeneidad (respecto de compatibilidad de código, SO, arquitectura y set de instrucciones).

En ambos escenarios se consideran el acceso a datos distribuidos con alto impacto en E/S

(entrada/salida), tráfico de red (consumo de ancho de banda) y la posible migración de proceso consecuencia de los cambios en el estado global del sistema y de los requerimientos variables de los procesos a lo largo del tiempo, esta verificación se hará cada cierto Δt , también se tendrá en cuenta el impacto en el consumo de energía que generaría la migración.

En base a lo expuesto, se propone generar un modelo de decisión en un Sistema Distribuido agregando la migración controlada de procesos a un operador de agregación, que permita la redistribución de la carga de trabajo entre los nodos del sistema distribuido, para mejorar, la utilización de los recursos, el tiempo de respuesta de los procesos, la mejora del rendimiento y verificar el impacto en el consumo de energía.

Enunciado preciso del eje de la situación problemática

Como se ha mencionado anteriormente en la descripción de la situación problemática, el Runtime gestiona los procesos y recursos compartidos y define el escenario específico correspondiente a cada situación. Los Runtime interactúan entre sí para intercambiar información, y mediante un nodo central se recopila la información de todos los nodos, se aplica el proceso de agregación y se obtiene la lista de asignaciones de recursos a procesos.

Durante el funcionamiento de los sistemas de procesamiento distribuido, en cada ciclo se obtiene una imagen global del sistema (macro imagen), para poder verificar los estados de los nodos del SD (Sistema Distribuido). Una vez verificado e identificado la existencia de nodos accesibles y no accesibles, se atienden las solicitudes de los procesos y recursos correspondientes a los nodos accesibles. Aquellas asignaciones que no han podido ser resueltas, se agrupan con las nuevas solicitudes y se vuelve a tener una macro imagen (entendiéndose por macro imagen a la información referida al estado de los nodos respecto de la carga computacional actual y la demanda

de asignación de recursos a procesos) para verificar si el nodo que estaba inaccesible ahora ya no lo está.

En base a la información recogida del sistema se puede detectar la presencia de nodos muy cargados con respecto a nodos que están inactivos o con baja carga de trabajo, una inadecuada distribución de la carga de trabajo, provocando un aumento en el tiempo de ejecución de los procesos que, a su vez, puede generar un alto consumo de energía y perjudica al rendimiento general del sistema.

Fuentes de interés

A continuación, se enumeran las principales fuentes que han sido fundamentales para el desarrollo de este trabajo:

- a) Los enfoques tradicionales para la asignación de recursos en sistemas distribuidos, respetando la exclusión mutua en el acceso a los recursos compartidos, como se exponen en los estudios de (Ricart & Agrawala, 1981), (Cao & Singhal, 2001), (Lodha & Kshemkalyani, 2000).
- b) Los enfoques innovadores presentados en las investigaciones de (La Red Martínez, 2017), (La Red Martínez et al., 2017), (La Red Martínez et al., 2018), (Agostini & La Red Martínez, 2019), (Agostini et al., 2019), (Agostini, 2019).
- c) Los principales métodos relacionados con la migración de procesos, el equilibrio de carga y la gestión del consumo de energía, discutidos en los trabajos de (Beiruti & Ganjali, 2020), (Upadhyay & Lakkadwala, 2015), (Deshmukh & Deshmukh, 2015), (Junaidi et al., 2020), (Banerjee et al., 2019), (Liu et al., 2013) y (Asadi et al., 2020).
- d) Los modelos de toma de decisiones que utilizan operadores de agregación pertinentes,

como se evidencia en las investigaciones de (Chiclana et al., 2001), (Peláez et al., 2004, 2007), (Dong et al., 2016) y otros.

Formulación del problema

¿Cuáles son los nuevos modelos de decisión que habrá que desarrollar incorporando migración controlada de procesos para la toma de decisiones en grupos de procesos, que trasciendan el enfoque tradicional de las ciencias de la computación?

Sistematización del problema

¿Cuáles son los nuevos modelos de decisión que habrá que desarrollar incorporando migración controlada de procesos para la toma de decisiones en grupos de procesos, que trasciendan el enfoque tradicional de las ciencias de la computación, en las siguientes circunstancias?

- a) Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua, incorporando mecanismos de migración controlada de procesos para lograr un mejor balanceo de carga de trabajo suponiendo homogeneidad (respecto de compatibilidad de código, SO, arquitectura y set de instrucciones).
- b) Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua, incorporando mecanismos de migración controlada de procesos suponiendo heterogeneidad (respecto de compatibilidad de código, SO, arquitectura y set de instrucciones).

En ambos escenarios se considera el acceso a datos distribuidos con alto impacto en E/S y tráfico de red (consumo de ancho de banda) y la migración de procesos consecuencia de cambios en el estado global del sistema y de los requerimientos variables de los procesos a lo largo del

tiempo, esta verificación se hará cada cierto Δt .

1.3. Objetivos

Objetivo General

Desarrollar un modelo de decisión que permita la migración controlada de procesos, considerando aspectos de balanceo de carga, consumo de energía y requisitos de poder de cómputo de los procesos.

Objetivos específicos

- Obtener una macroimagen del estado del sistema con las solicitudes de recursos a procesos a través del operador de agregación.
- Crear un modelo de decisión considerando la migración del proceso y balanceo de carga de trabajo.
- Generar un operador que permita evaluar los procesos migrables, las características de los nodos en cuanto a sus prestaciones y la carga de trabajo de cada nodo, e integrar dicho operador en el modelo de decisión.

1.4. Justificación

En los sistemas de procesamiento distribuido es frecuentemente necesario coordinar la asignación de recursos compartidos que deben ser asignados a los procesos en la modalidad de exclusión mutua; en tales casos se debe decidir el orden en que los recursos compartidos serán asignados a los procesos que requieren de los mismos.

La migración controlada de procesos es la reubicación de un proceso desde su ubicación actual a otra ubicación. La característica más importante que se espera conseguir al implementar

una migración controlada de procesos es la mejora del rendimiento general del sistema, logrando un balanceo de carga de trabajo, verificando el impacto en el consumo de energía.

El trabajo de investigación tiene como objetivo desarrollar un modelo de decisión que incorpore la migración controlada de procesos a un operador. Este modelo permitirá analizar situaciones en las que el nodo central recibe información de los nodos del sistema distribuido. Posteriormente, se determinarán los procesos a migrar con el fin de redistribuir la carga de trabajo entre los nodos, buscando mejorar la utilización de los recursos, los tiempos de respuesta de los procesos, el rendimiento general del sistema, y evaluar el impacto en el consumo de energía.

1.5. Hipótesis y objeto

Hipótesis

La implementación de modelos de decisión junto a sus operadores de agregación, considerando la migración controlada de procesos en sistemas distribuidos, puede tener un efecto positivo en diversos aspectos del sistema, incluyendo la optimización de recursos, la reducción de tiempos de respuesta, la disminución del consumo de energía y el mejoramiento general del rendimiento del sistema.

Objeto

Asignación de recursos en sistemas distribuidos considerando la migración controlada de proceso.

1.6. Antecedentes

Para poder realizar la investigación se han buscado y analizado materiales de gran importancia referentes a:

Comunicación en sistemas distribuidos.

Sincronización en sistemas distribuidos.

Sistemas de soporte de decisión.

Uso compartido de recursos y exclusión mutua en sistemas distribuidos.

Toma de decisiones en grupo y modelos de decisión.

Migración de procesos.

Balanceo de Carga.

Consumo de energía.

A pesar de contar con una cantidad considerable de antecedentes en la bibliografía, no se ha encontrado un trabajo que aborde de manera integral el esquema de asignación de recursos a procesos descrito en (Agostini & La Red Martínez, 2019) y (La Red Martínez et al., 2017). Este enfoque global contempla el estado de los nodos y el impacto potencial en estos debido a las asignaciones de recursos a procesos. Es decir, no se ha hallado un estudio que considere simultáneamente la toma de decisiones relacionada con la gestión de recursos compartidos para conjuntos de procesos en sistemas distribuidos, bajo diversas condiciones, abarcando la migración controlada de procesos, el balanceo de carga y la eficiencia en el consumo de energía.

Por esta razón se considera que el trabajo de investigación propuesto reúne las condiciones de originalidad necesarias, siendo además relevante académicamente por los aspectos que se mencionan a continuación.

Se espera producir conocimiento científico específico no disponible actualmente.

El desarrollo de modelos de decisión que permita la migración controlada de procesos para la toma de decisiones en grupos de procesos, en el contexto de sistemas complejos y esquemas de autorregulación, constituye un aporte significativo a las ciencias de la computación, en especial en relación con la gestión del acceso a recursos compartidos desde procesos distribuidos, mejorando la gestión de los mismos por parte de los sistemas operativos.

Se podría utilizar el modelo de decisión y sus operadores de agregación para la asignación de recursos a procesos, contribuyendo así a mejorar el desempeño de los sistemas distribuidos considerando información global del sistema para realizar una migración controlada de procesos y así mejorar el balanceo de carga, el rendimiento global del sistema a reducción del consumo de energía, aspectos no considerados en los métodos tradicionales.

1.7. Encuadre o Marco Teórico

En este apartado se describen los principales aportes a la ciencia con relación al presente trabajo.

Operadores de agregación y Modelos de decisión

En (La Red Martínez & Acosta, 2014) se propone el uso de uno o varios operadores de agregación capaces de proporcionar una relación de preferencia colectiva para llegar a una cierta conclusión o tomar una determinada decisión. Estos operadores se deben elegir según criterios específicos teniendo en cuenta las propiedades características de cada operador.

En (La Red Martínez, 2017) se propone un operador de agregación (que podría ser utilizado por un módulo administrador de recursos compartidos) que considerando los requerimientos de los procesos (recursos compartidos) y el estado de los nodos distribuidos donde operan los procesos (su carga computacional), decidirá el orden de asignación de los recursos a los procesos.

En (Agostini et al., 2018) se propone un mecanismo de consenso para establecer el orden de asignación de recursos a los procesos mediante un operador de agregación. El consenso se entiende como el acuerdo que debe lograrse para la asignación de todos los recursos solicitados por cada proceso.

En (Leyva-vázquez et al., 2012) se propone un operador de agregación que tiene en cuenta la presencia de valores extremos y una variante para el modelo de la fiabilidad de las fuentes. Se presenta un caso de estudio vinculado a la construcción de mapas cognitivos difusos.

En (Fullér, 1996) se presenta un breve estudio de los operadores OWA e ilustra su aplicabilidad mediante un ejemplo de la vida real.

En (La Red Martinez & Acosta, 2015) se hace una breve revisión del problema de la expresión de preferencias en los modelos de decisión; se aportan abundantes referencias bibliográficas relacionadas con los principales conceptos.

En (Andrade et al., 2020) se propone un marco de decisión que sirve de apoyo a los ingenieros a la hora de refactorizar sistemas de software para adaptarlos a plataformas heterogéneas. Incluye la evaluación de factores importantes para minimizar el riesgo de problemas recurrentes en el proceso.

Migración de procesos en sistemas distribuidos

En (Sharma et al., 2012) se estudia el campo de la migración de procesos resumiendo los conceptos clave involucrados en él y destacando los beneficios y desafíos que enfrenta la migración de procesos. También incluye la experiencia de migración de procesos en sistema operativo distribuido.

En (Milojčić et al., 2000) se revisa el campo de la migración de procesos resumiendo los conceptos clave y ofreciendo una visión general de las implementaciones más importantes. Los problemas de diseño e implementación de la migración de procesos se analizan en general, y luego se revisan para distintos casos de estudios.

En (Rathore & Chana, 2014) se presenta un amplio estudio de las técnicas existentes de equilibrio de carga y migración de trabajos. Además, se ha propuesto y discutido una nueva técnica de equilibrio de carga, junto con el enfoque de migración de trabajos para cumplir con las lagunas de investigación existentes.

En (Smith, 1988) se examinan los mecanismos propuestos y aplicados para la migración de procesos, prestando especial atención a los objetivos del diseño, como el rendimiento, el equilibrio de la carga y la fiabilidad.

En (Eskicioglu, 1995) se analiza el diseño de los mecanismos de migración de procesos en los sistemas distribuidos con respecto a cuestiones clave, como los modelos de sistema en los que se implementan los mecanismos, las plataformas de hardware en las que se ejecutan, los métodos utilizados para mover un proceso de una máquina a otra, las políticas de distribución de carga adoptadas y la transparencia de la red.

En (Sohrabi & Mousavi Khaneghah, 2020) se examinan los desafíos de la ocurrencia de eventos dinámicos e interactivos en los migradores de procesos basados en máquinas virtuales.

En (Chang et al., 2020) se propone un esquema de migración de procesos de borde que considera los recursos que es capaz de optimizar la decisión de migración de procesos. Para llevar a cabo este sistema y validar el esquema, han desarrollado un marco denominado Edge Process-enabled Internet of Thing (EPIoT) host; en (Elleuch & Muntean, 1994) se trata la construcción de

mecanismos de migración de procesos para sistemas paralelos masivos (MPS). Propone tres protocolos de migración para mecanismos de paso de mensajes correctos, para lograr corrección, transparencia, escalabilidad y eficiencia mediante la minimización de la sobrecarga introducida por el uso de los mecanismos de migración en tiempo de ejecución.

En (Beiruti & Ganjali, 2020) se presenta un nuevo y eficiente protocolo de migración de carga, denominado ERC, para desplazar la carga de entrada asociada a instancias de controlador sobrecargadas hacia instancias poco cargadas. Puede utilizarse para una amplia gama de aplicaciones de red, como el equilibrio de carga, el ahorro de energía y la optimización de recursos.

En (Wang et al., 2012) se complementa la tolerancia a fallos reactiva con la proactiva a nivel de proceso. Se desarrolla un novedoso mecanismo de migración en vivo a nivel de proceso que soporta la ejecución continuada de aplicaciones durante gran parte de la migración de procesos. El trabajo también proporciona un novedoso enfoque de migración hacia atrás para eliminar el desequilibrio de carga o los cuellos de botella causados por las tareas migradas.

En (Upadhyay & Lakkadwala, 2015) se ha diseñado dos algoritmos distintos para diferentes tareas en un entorno de nube creado. En el primer algoritmo, crea un algoritmo de migración en el que migrar los procesos sobrecargados de una máquina a otra máquina en el mismo entorno de la nube. En el segundo algoritmo, crea un algoritmo de programación que programa los procesos teniendo en cuenta la utilización de los recursos, que se basa en el uso de la memoria, el uso de la CPU, el rendimiento y el tiempo de procesamiento.

En (Deshmukh & Deshmukh, 2015) se adopta un enfoque modificado para crear un nuevo equilibrador de carga dinámico y distribuido para un sistema de archivos distribuido. El algoritmo tiene en cuenta restricciones como la carga de la red, la carga de IO del disco, la capacidad del

disco y la carga que no se consideraba en los enfoques anteriores.

En (Patel, 2015) se propone la idea de balancear la carga usando la migración de procesos y restaurando el proceso desde el punto en que lo dejó en el nodo de destino seleccionado. Utiliza la migración de procesos para controlar el reparto de la carga, la disponibilidad de los procesos largos y la utilización de algunos recursos especiales.

En (Thakkar & Pandya, 2013) se presenta un mecanismo de migración de procesos en un sistema distribuido heterogéneo. Utiliza un mecanismo de representación de datos externos para ayudar a reducir la complejidad del software en el proceso de traducción, considerando el formato del nodo de origen con respecto al formato de datos del nodo de destino.

Balanceo de carga

En (Maurya & Ahmad, 2013) se presenta un algoritmo para el equilibrio de carga que calcula dinámicamente la carga de cada nodo y migra la tarea sobre la base de una restricción predefinida que reduce la sobrecarga de la red.

En (Alakeel, 2010) se presenta y analiza las cuestiones para el desarrollo de un algoritmo eficaz de balanceo de carga dinámico que implica: estimación de la carga, comparación de los niveles de carga rendimiento, la estabilidad del sistema, la cantidad de información entre nodos, estimación de los requisitos de recursos de los trabajos, selección de trabajos para su transferencia, selección de nodos remotos, etc.

En (Shah & Kapadia, 2012) se desarrolla una técnica de migración de procesos para el equilibrio de la carga, teniendo en cuenta dos algoritmos, uno iniciado por el emisor y otro iniciado por el receptor.

En (Hac & Johnson, 1986) se centra su estudio en el equilibrio de la carga aplicado a la colocación óptima de procesos y sitios de lectura. Se propone un algoritmo que aumenta el rendimiento del sistema mediante el equilibrio de la carga. Este algoritmo utiliza los datos recogidos por el sistema para basar sus decisiones.

En (Sánchez Dams, 2014) se propone un algoritmo de balanceo de carga para sistemas homogéneos distribuidos cliente servidor con gran cantidad de peticiones. Utiliza como parámetro de medición el tiempo de respuesta y el número de paquetes perdidos.

En (Chen et al., 2020) se explora la aplicación del aprendizaje automático al algoritmo de equilibrio de carga del sistema operativo de un sistema multiprocesador. Se utiliza el aprendizaje por imitación para incorporar un modelo de aprendizaje automático como componente del sistema en el núcleo.

En (Junaidi et al., 2020) se propone un algoritmo de aprendizaje automático que utiliza los recursos del servidor, la CPU y la memoria, para predecir el futuro de las cargas del servidor; en (Kim et al., 2021) se propone una técnica de equilibrio de carga para el entorno distribuido de aprendizaje automático. El método propuesto configura el servidor distribuido en una estructura de tabla hash compuesta por diferentes niveles y asigna tareas de aprendizaje automático al servidor considerando el rendimiento de cada servidor.

En (Banerjee et al., 2019) se describe el diseño de un sistema que aborda el reto de programar cargas de trabajo de análisis de datos distribuidos en redes de procesamiento heterogéneos (que incluyen CPUs, GPUs, FPGAs y ASICs) en entornos basados en la nube, dinámicos y de múltiples usuarios.

Consumo de energía

En (Duolikun et al., 2014) se considera un enfoque de migración de procesos que se realice de manera energéticamente eficiente y fiable. Se discute las condiciones en las que un proceso se migra de un servidor actual a otro servidor mediante la estimación del consumo de energía de un servidor y el tiempo de finalización de los procesos actuales.

En (Asadi et al., 2020) se proponen modelos analíticos, basados en las redes de recompensa estocástica, para analizar el impacto de los algoritmos de asignación de recursos y los métodos de migración de procesos en el consumo de energía y el rendimiento de los sistemas virtualizados.

En (Naghash Asadi et al., 2019a) se modela la infraestructura de los sistemas de computación en la nube para evaluar el consumo de energía y las medidas de rendimiento, utilizando redes de actividad estocástica. En el modelo propuesto, los servidores ejecutan diferentes números de máquinas virtuales y las tareas se dividen en dos categorías, demostrando sus prioridades de ejecución.

En (Naghash Asadi et al., 2019b) se menciona que la potencia y el rendimiento de las infraestructuras de la nube se modelan analíticamente y se evalúan numéricamente utilizando redes de recompensa estocásticas. El objetivo principal del modelo propuesto es analizar el impacto de la temperatura y el tráfico de red en la potencia y el rendimiento de los sistemas de computación en nube.

En (Liu et al., 2013) se estima el coste de la migración en vivo de las máquinas virtuales en términos de rendimiento y energía con un enfoque cuantitativo. Construye modelos de predicción de costes basados en la aplicación, utilizando el conocimiento aprendido sobre las cargas de trabajo a nivel del hipervisor.

En (Kansal & Chana, 2012) se analizan las técnicas de equilibrio de carga existentes en la computación en nube y se comparan en función de varios parámetros como el rendimiento, la escalabilidad, la sobrecarga asociada, etc., que se tienen en cuenta en las diferentes técnicas. El objetivo del equilibrio de carga es minimizar el consumo de recursos, lo que reducirá aún más el consumo de energía y la tasa de emisión de carbono, que es la necesidad imperiosa de la computación en nube.

En (Mallikarjuna et al., 2018) habla del propósito de la computación ecológica que es maximizar la eficiencia energética y el equilibrio de carga en los centros de datos y el servidor verde. Se demuestra que el modelo matemático mejora la eficiencia de los servidores y la transmisión efectiva de datos, se describe el equilibrio de carga efectiva que proporciona el consumo de energía en los servidores de datos en la computación en nube.

En (Thapliyal & Dimri, 2020) se incluye una revisión detallada de varias estrategias de equilibrio de carga como algoritmo que son aplicables en cualquier entorno con respecto a la nube y la variedad de parámetros para el equilibrio de carga.

1.8. Relevancia

Académica

Se trata de un proyecto que conlleva en sí una fuerte innovación y que llama constantemente a la reflexión sobre las formas de encarar la solución a problemas de las ciencias de la computación desde una óptica no convencional, proponiéndose la utilización de distintas disciplinas en las que ya se han realizado trabajos y publicaciones, y que ahora se tiene previsto integrar en la búsqueda de nuevas soluciones a los problemas planteados de gestión de recursos compartidos en grupos de procesos operando en sistemas distribuidos, contemplando migración

de procesos y balanceo de carga de trabajo.

Social

El propósito del presente plan de investigación es el de desarrollar modelos de decisión que permitan la toma de decisiones basadas en la migración controlada de procesos, aplicables a grupos de procesos computacionales; en tal sentido podría considerarse que la relevancia social está dada, ya que la informática y las comunicaciones son hoy parte de la vida diaria de millones de personas alrededor del mundo, considerándose que la informática se ha vuelto ubicua y pervasiva, ubicua en el sentido de estar presente de manera muchas veces inadvertida en gran cantidad de actos de la vida cotidiana, y pervasiva al permitir el acceso a los sistemas informáticos casi desde cualquier lugar y en cualquier momento, haciendo uso de redes de comunicaciones muchas veces inalámbricas.

Es un hecho además que actualmente la mayoría de los sistemas informáticos utilizados operan de manera distribuida sobre redes de comunicaciones, debiendo realizar permanentemente operaciones de coordinación y toma de decisiones en grupos de procesos.

Los sistemas antes muy brevemente descriptos son masivamente utilizados por la población en general, de manera directa o indirecta, razón por la cual se considera que cualquier forma de mejorar el desempeño de dichos sistemas redundaría en beneficio de la sociedad, en el contexto de la llamada Sociedad de la Información y el Conocimiento, con aplicaciones relacionadas, por ejemplo, con el e-government (gobierno electrónico), el e-commerce (comercio electrónico), el e-learning (aprendizaje electrónico), la e-democracy (democracia electrónica) y la m-cognocracy (gobierno electrónico móvil en la sociedad del conocimiento).

Relevancia Científica

El desarrollo de modelos de decisión para toma de decisiones en grupos de procesos en el contexto de sistemas complejos y esquemas de autorregulación, constituye un aporte significativo de las ciencias cognitivas a las ciencias de la computación, en especial en relación con la gestión del acceso a recursos compartidos desde procesos distribuidos, mejorando la gestión de los mismos por parte de los sistemas operativos.

1.9. Metodología

Dimensión de la Estrategia General

a) Tipo de estrategia y fundamentación

Considerando el objeto, problema y objetivos, se ha decidido que el tipo de investigación que se utilizará será la cuantitativa, con medición de variables, formulación de hipótesis, validación de hipótesis.

b) Universo

El universo estará formado por recursos y procesos que se encuentran en los nodos que compone el sistema distribuido, donde se originan requerimientos de los accesos compartidos en la modalidad de exclusión mutua.

c) Unidad de análisis

La unidad de análisis estará integrada por la relación proceso-recurso requerido, donde los procesos y recursos podrán estar en diferentes nodos del sistema distribuido.

d) Selección de casos

Se trabajará con los escenarios indicados en los objetivos específicos y detallados en el Problema.

e) Énfasis en el proceso lineal

Conforme a las características de la investigación propuesta, se ha optado por la lógica cuantitativa, basada en la búsqueda y verificación de relaciones causa-efecto, que se puedan generalizar; es decir, hay un proceso lineal de relación teoría-empiría, con diferentes espacios y momentos para la definición de las hipótesis, la obtención de los datos, su análisis y su posterior interpretación.

Dimensión de las técnicas de obtención y análisis de información empírica

Técnicas de obtención y análisis

La información que se obtendrá y habrá que analizar, surgirá de los procesos de simulación de los modelos de decisión y operadores de agregación que se desarrollarán. A continuación, se describen las estructuras de datos con las que se trabajará y el tratamiento y análisis de datos que se tiene previsto efectuar.

Estructuras de datos

El sistema de matrices de datos que se utilizará contemplará las siguientes premisas y estructuras de datos. Se trata de grupos de procesos distribuidos en nodos de procesos que acceden a recursos críticos compartidos en la modalidad de exclusión mutua distribuida, debiendo decidirse, ante la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar los recursos a los procesos que los requieren (sólo intervendrán como alternativas de

asignación a los procesos aquellos recursos disponibles, es decir, no asignados ya a determinados procesos):

- El permiso de acceso a los recursos compartidos propios de un nodo no dependerá sólo de si los nodos los están utilizando o no, sino del valor de agregación de las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas).
- Las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas) dependerá de la consideración del valor de variables que representen el estado de cada uno de los distintos nodos. Cada nodo deberá expresar sus prioridades para la asignación de los distintos recursos compartidos respecto de los requerimientos de recursos de cada proceso de cada grupo.

Nodos que alojan procesos: $1, \dots, n$.

Procesos alojados en cada uno de los n nodos: $1, \dots, p$.

Grupos de procesos distribuidos: $1, \dots, g$.

Tamaño de cada uno de los g grupos de procesos: $1, \dots, t$.

Recursos críticos compartidos en la modalidad de exclusión mutua distribuida disponibles en cada uno de los n nodos: $1, \dots, r$.

Estados posibles de cada uno de los p procesos:

- Grupo al que pertenece el proceso (0 significa proceso independiente).
- Requiere sincronización con los demás procesos del grupo al que pertenece (los procesos del grupo deben estar activos en sus respectivos procesadores en un mismo lapso de tiempo), (0 significa que no requiere sincronización, 1 significa que sí la requiere).

- En espera de un recurso compartido con el grupo de procesos al que pertenece.
- En espera de un recurso no compartido con el grupo de procesos al que pertenece.
- En ejecución con permiso de acceso a un recurso compartido con el grupo de procesos al que pertenece.
- En ejecución sin permiso de acceso a un recurso compartido con el grupo de procesos al que pertenece.
- Inactivo.

Estado posible de cada uno de los n nodos:

- Número de procesos.
- Prioridades de los procesos.
- Uso de CPU.
- Uso de memoria principal.
- Uso de memoria virtual.
- Estado de cada uno de los r recursos críticos compartidos en la modalidad de exclusión mutua distribuida existentes en el nodo:
 - Asignado a un proceso local.
 - Asignado a un proceso remoto.
 - Disponible.
- Predisposición (prioridad nodal) para otorgar el acceso a cada uno de los r recursos críticos compartidos (alternativas) en la modalidad de exclusión mutua distribuida (resultará de la consideración de las variables representativas del estado del nodo, para cada recurso crítico compartido existente). Se obtendrá una tupla por cada uno de los n nodos, cada tupla contendrá r valores (prioridades nodales) para compartir los recursos críticos.

Estado global del sistema:

- Número de grupos de procesos y tamaño (número t de procesos) de cada uno de los g grupos.
- Porcentajes de consenso requeridos para otorgar el acceso a cada uno de los r recursos críticos disponibles en cada uno de los n nodos.
- Predisposición (prioridad global) para otorgar el acceso a cada uno de los r recursos críticos compartidos (alternativas) en la modalidad de exclusión mutua distribuida (resultará de la agregación de las prioridades nodales para cada recurso crítico compartido existente (alternativas). Se obtendrá una tupla con r valores normalizados (prioridades globales) para compartir los recursos críticos.
- Decisión de acceso a los r recursos críticos en función de contrastar las prioridades globales normalizadas para compartir los mismos con los porcentajes de consenso requeridos para otorgar los respectivos accesos.
- El estado global del sistema deberá actualizarse reiteradamente mientras haya alguno o algunos de los p procesos que requieran acceder a alguno o algunos de los r recursos compartidos.

El sistema se auto regula reiteradamente en función del estado local de los n nodos y del estado global del sistema, produciéndose una actualización de los estados locales de los nodos como consecuencia de la evolución de sus respectivos procesos y de las decisiones de acceso a los recursos críticos producidas teniendo en cuenta el estado global del sistema: el sistema distribuido en el que se ejecutan grupos de procesos que acceden a recursos críticos, se observa a sí mismo y produce decisiones de accesos a recursos que modifican el estado del sistema y lo reajustan reiterativamente.

Tratamiento y análisis de datos

Se efectuarán simulaciones con los modelos de decisión propuestos y los modelos de decisión computacionales clásicos, a los efectos de analizar el comportamiento de estos para las mismas condiciones de carga de trabajo y de consumo de recursos, dando lugar a un esquema iterativo de modificación de los modelos propuestos para intentar lograr un desempeño de los mismos al menos equivalente al de los modelos computacionales clásicos.

1.10. Estructura de la tesis

Para la elaboración de la presente investigación se procedió a presentar el problema del cual se inició el trabajo, como también los antecedentes principales y los conceptos teóricos. Además, se mencionó la metodología a utilizar y posteriormente se indican los capítulos restantes con los cuales se ha estructurado esta tesis.

Capítulo II – Escenario 1. Migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo homogeneidad (respecto de compatibilidad de código, SO, arquitectura y set de instrucciones). En este capítulo se han efectuado los siguientes pasos:

- Estudiado el escenario general E1.
- Formalizado el operador de agregación para el escenario E1.
- Validado empíricamente el operador de agregación para el escenario E1.
- Definido y validado teóricamente el modelo global de decisión del escenario E1.
- Definido detalladamente las etapas, niveles o capas del modelo de decisión E1 y validado empíricamente.
- Evaluado y comparado los resultados del modelo de decisión del escenario E1.

Capítulo III – Escenario 2. Migración de procesos para lograr un mejor balanceo de carga

de trabajo suponiendo heterogeneidad (respecto de compatibilidad de código, SO, arquitectura y set de instrucciones). En este capítulo se han efectuado los siguientes pasos:

- Estudiado el escenario general E2.
- Formalizado el operador de agregación para el escenario E2.
- Validado empíricamente el operador de agregación para el escenario E2.
- Definido y validado teóricamente el modelo global de decisión del escenario E2.
- Definido detalladamente las etapas, niveles o capas del modelo de decisión E2 y validado empíricamente.
- Evaluado y comparado los resultados del modelo de decisión del escenario E2.

Capítulo IV - Comparación de resultados: Se han comparado los resultados al aplicar el modelo de decisión al conjunto de datos migrado y no migrado para ambos escenarios propuesto (sistema distribuido homogéneo y heterogéneo), así también al conjunto de datos sobre consumo de energía de los nodos del sistema distribuido aplicando la migración y sin migración.

Capítulo V – Conclusiones y futuras líneas de investigación: se han comentado las principales conclusiones y se han indicado las posibles líneas futuras de investigación.

1.11. Discusiones y comentarios

Se elaboró un despliegue de los antecedentes de investigaciones y el marco teórico donde se especifican aspectos relevantes y actualizados del estado del arte en las diferentes áreas con relación a la presente tesis (sistemas distribuidos, operadores de agregación y modelos de decisión, migración de procesos, balanceo de carga e impacto en el consumo de energía). Se definió la hipótesis y los objetivos de la investigación donde se pueden apreciar las situaciones en las que los procesos puedan acceder a recursos compartidos en la modalidad de exclusión mutua.

También se describieron las estructuras de datos, los elementos que intervienen dentro del sistema, los nodos, recursos y procesos, con prioridades y cargas de trabajo y que en diferentes situaciones permiten que el sistema se autorregule de manera reiterada en función del estado local de los n nodos y del estado global del sistema.

Para finalizar se describen los capítulos conformados en la estructura de la tesis. A parte de lo expuesto anteriormente se prosigue con cuatro capítulos referentes a los distintos escenarios propuestos, un capítulo sobre migración de procesos en un entorno de sistema distribuido homogéneo, otro sobre migración de procesos en un entorno de sistema distribuido heterogéneo, y uno más con las comparaciones de los resultados, finalizando con las conclusiones y líneas futuras de investigación.

En este trabajo se ha generado un método para migración controlada de procesos a un operador de agregación, que permita la redistribución de la carga de trabajo entre los nodos del sistema distribuido para mejorar la utilización de los recursos, el tiempo de respuesta de los procesos, la mejora del rendimiento y verificar el impacto en el consumo de energía. El algoritmo verifica la carga de los nodos dentro del SD, identificado los nodos con carga alta se analizan sus procesos y los recursos que solicita, en un entorno de SD homogéneo se enumeran todos los procesos candidatos a ser migrados y los posibles nodos receptores que puede ser cualquier nodo con carga baja, en un entorno de SD heterogéneo el algoritmo verifica que el nodo capaz de alojar al proceso debe tener carga baja y debe ser compatible en cuanto a características de hardware y software, también se tiene en cuenta el consumo de ancho de banda que genera transferir un recurso de un nodo a otro.

Se lista todos los procesos candidatos a ser migrado con sus posibles nodos destino, se migra primero el proceso que posee recursos en el nodo destino y que genere menor peso de

transferencia, si no posee recursos en el nodo destino se toma el proceso que genere menor peso de transferencia al nodo elegido.

El impacto en el consumo de energía se mide aplicando el promedio de la sobrecarga de memoria, CPU y E/S al promedio carga general antes de la migración y una vez realizada la migración se vuelve a hacer el mismo calculo aplicando la sobrecarga al nodo que recibe al proceso.

La propuesta es presentar un método innovador para la gestión de recursos compartidos en sistemas distribuidos, basado en (La Red Martínez, 2017) y en (Agostini et al., 2019), donde se desarrollan operadores de agregación para asignar recursos en sistemas distribuidos, pero que no consideran la posibilidad de migración de procesos para el balanceo de carga de trabajo. Este trabajo consiste en resolver esta problemática, agregando una capa de migración controlada de procesos, a un modelo de decisión existente para gestión de recursos y procesos en sistemas distribuidos, considerando como punto de inicio las premisas y las estructuras de datos mencionadas en esas publicaciones.

Lo mencionado anteriormente inicia en el siguiente capítulo.

Capítulo II – Escenario 1. Migración de procesos en condiciones de homogeneidad

2.1. Trabajos previos

Entre los métodos considerados tradicionales para la asignación de recursos en sistemas distribuidos respetando la exclusión mutua en el acceso a los recursos compartidos, se destacan los mencionados en (Ricart & Agrawala, 1981), (Cao & Singhal, 2001) y (Lodha & Kshemkalyani, 2000).

Métodos innovadores con respecto a la asignación de recursos a procesos se mencionan en (La Red Martínez, 2017) y (La Red Martínez et al., 2018), y una alternativa con respecto a la asignación de recursos, considerando niveles estrictos de consenso y ordenamiento por grupos de procesos se puede ver en (Agostini & La Red Martínez, 2019), (La Red Martínez et al., 2017), (Agostini et al., 2019) y (Agostini, 2019).

Existen diferentes estudios sobre los métodos de migración de procesos, balanceo de carga y consumo de energía. En (Beiruti & Ganjali, 2020), desarrollan un nuevo protocolo que se utiliza para una amplia gama de aplicaciones de red, como el equilibrio de carga, el ahorro de energía y la optimización de recursos.

En (Upadhyay & Lakkadwala, 2015) desarrollan un algoritmo de migración para gestionar procesos en máquinas sobrecargadas derivándolos a otra en el mismo entorno de la nube teniendo en cuenta la utilización de los recursos.

Este adopta un enfoque modificado para crear un nuevo equilibrador de carga dinámico distribuido para el sistema de archivos distribuido, el algoritmo tiene en cuenta restricciones como la carga de la red, la carga de IO (entrada/salida) del disco, la capacidad del disco y la carga que no se consideraba en otros enfoques.

En (Junaidi et al., 2020) se propone un algoritmo de aprendizaje automático que utiliza los recursos del servidor, CPU y la memoria, para predecir el futuro de las cargas del servidor.

En (Liu et al., 2013) se investigan metodologías de diseño para predecir cuantitativamente el rendimiento de la migración y el consumo de energía.

En (Asadi et al., 2020) se proponen modelos analíticos, basados en las redes de recompensa estocástica (SRN), para analizar el impacto de los algoritmos de asignación de recursos y los métodos de migración de procesos en el consumo de energía y el rendimiento de los sistemas virtualizados.

Algunos modelos de decisión utilizando operadores de agregación específicos se muestran en (Chiclana et al., 2001) y (Dong et al., 2016).

En (Sohrabi & Mousavi Khaneghah, 2020) se examinan los desafíos de la ocurrencia de eventos dinámicos e interactivos en los migradores de procesos basados en máquinas virtuales, mediante el análisis de la función de rendimiento de los migradores.

En (Rathore & Chana, 2014) se presenta un amplio estudio de las técnicas existentes de equilibrio de carga y migración de trabajos, donde se incluye una clasificación detallada basada en diferentes parámetros que dependen del análisis de las técnicas existentes, y se propone una nueva técnica de equilibrio de carga.

En (Chang et al., 2020) se propone un esquema de migración de procesos de borde consciente de los recursos (REM) que es capaz de optimizar la decisión de migración de procesos.

En (Marecos Brizuela et al., 2022) se propone la aplicación de un modelo basado en competencias y en el aprendizaje centrado en el alumno para el estudio de la migración controlada

de procesos, y donde el estudiante debe analizar el comportamiento de los recursos y procesos bajo diferentes cargas de trabajo, teniendo en cuenta el balanceo de carga como objetivo principal.

En (Fornerón Martínez et al., 2020) se propone un operador de agregación para gestión de recursos compartidos en sistemas distribuidos utilizando 2-tuplas asociadas a etiquetas lingüísticas.

2.2. Premisas

Se trata de grupos de procesos distribuidos en nodos de procesamiento que acceden a recursos críticos compartidos en la modalidad de exclusión mutua distribuida, debiendo decidirse, ante la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar los recursos a determinados procesos o grupos de procesos que los requieren (sólo intervendrán como alternativas de asignación a los procesos aquellos recursos disponibles, es decir, no asignados aún a determinados procesos):

- El permiso de acceso a los recursos compartidos propios de un nodo no dependerá sólo de si los nodos los están utilizando o no, sino del valor de agregación de las preferencias (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas).
- Las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas) dependerá de la consideración del valor de variables que representen el estado de cada uno de los distintos nodos. Cada nodo deberá expresar sus prioridades para la asignación de los distintos recursos compartidos respecto de los requerimientos de recursos de cada proceso.

En un entorno de sistemas distribuidos, representado en la Fig. 1, es común encontrarse con múltiples nodos, algunos de ellos con alta carga de trabajo, con carga media y otros con carga

baja. Cada nodo se encarga de ejecutar procesos que requieren acceso a ciertos recursos, ya sea de manera local o remota.

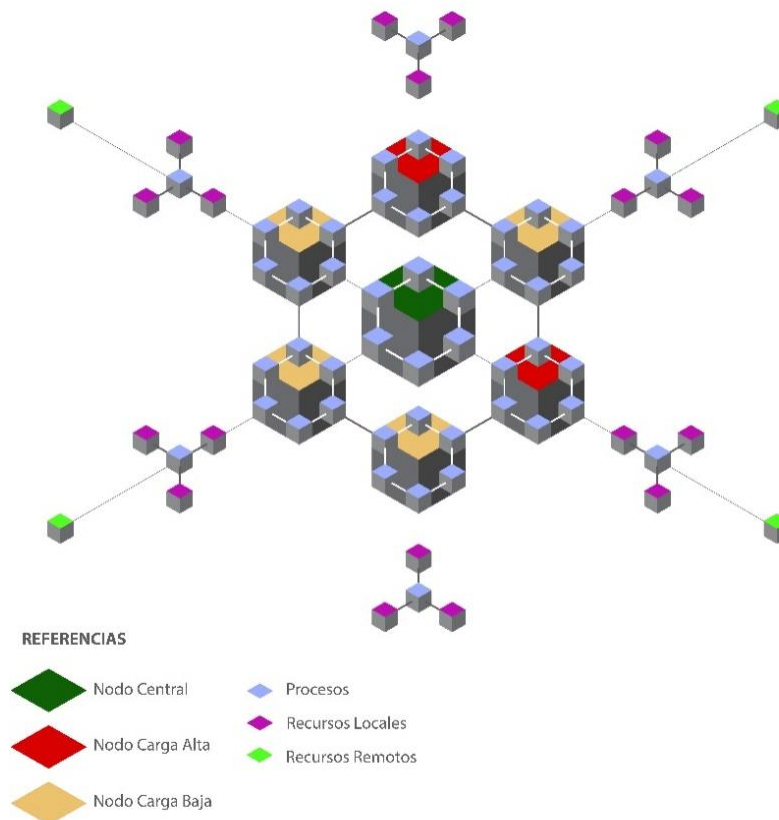


Fig. 1. Representación de un sistema distribuido con cargas nodales.

Fuente: Elaboración propia

El escenario planteado consiste en que el Runtime (software tiempo de ejecución) del Nodo Central, reciba la información proveniente de los distintos nodos y a través de un operador de agregación determine el orden de asignación de recursos a procesos y de procesos a nodos, según la siguiente premisa:

- Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua sin constituir grupos de procesos, los procesos no requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y sin exigencias estrictas de consenso para

lograr el acceso (no se requiere un consenso para asignar de manera consecutiva los recursos solicitados por un proceso o grupo de procesos, es decir, que iniciada la secuencia de asignación de recursos a un proceso, la misma puede ser interrumpida para asignar recursos a otro proceso).

- Bajo las circunstancias mencionadas anteriormente se tendrá en cuenta la aplicación de la migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo homogeneidad (respecto de compatibilidad de código, SO (Sistema Operativo), arquitectura y set de instrucciones).

El algoritmo de balanceo de carga corresponde a una serie de cálculos que se realizan para lograr la migración de procesos de nodos cargados a nodos más aliviados, partiendo de la estructura de datos utilizados en (La Red Martínez, 2017) y (La Red Martínez et al., 2018).

2.3. Propuesta de solución

Considerando lo desarrollado en (La Red Martínez, 2017) y (La Red Martínez et al., 2018) se presentará una variante de un enfoque innovador para la gestión de recursos compartidos en sistemas distribuidos, donde se desarrolla un operador de agregación para la asignación de recursos en un sistema distribuido.

Las premisas, estructuras de datos y operadores mencionados en la publicación anterior se utilizan como punto de partida para este trabajo. Para afrontar la problemática planteada, se define un Runtime (software en tiempo de ejecución) central que gestiona los procesos y recursos compartidos e interactúa con sus similares de los nodos distribuidos para intercambiar información. En el nodo central se recopila la información de todos los nodos, se aplica el proceso de agregación y se obtiene la lista de asignaciones de recursos a procesos.

Durante el funcionamiento del sistema distribuido, cada cierto tiempo se reúne y evalúa

información de control referida a las cargas nodales para obtener una imagen global del sistema (macro imagen), que refleja el estado de los nodos, recursos y procesos en un determinado momento, lo que permite evaluar posibles migraciones para balancear la carga de trabajo de los distintos nodos del sistema. A continuación se mencionan las estructuras de datos utilizadas:

Nodos que alojan procesos: $1, \dots, n$. El conjunto de nodos se representa de la siguiente manera:

$$nodos = \{n_1, \dots, n_n\}$$

Procesos alojados en cada uno de los n nodos: $1, \dots, p$. El conjunto de procesos se representa de la siguiente manera:

$procesos = \{p_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, p$ (n° máximo de procesos en cada nodo), lo que se puede expresar mediante la Tabla 1.

Tabla 1. Procesos en cada nodo (E1)

Nodos		Procesos			
1	p_{11}	p_{12}		p_{1p}	
....			...		
i	p_{i1}	p_{i2}		p_{ip}	
....			...		
n	p_{n1}	p_{n2}		p_{np}	

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Grupos de procesos distribuidos: $1, \dots, g$. El conjunto de grupos de procesos distribuidos se representa de la siguiente manera:

$$grupos = \{p_{ij}\} \text{ con } i \text{ indicando el nodo y } j \text{ el proceso en dicho nodo.}$$

Tamaño de cada uno de los g grupos de procesos. El n° de procesos en cada grupo indica la cardinalidad del grupo y se representa de la siguiente manera:

$$card = \{card(g_i)\} \text{ con } i = 1, \dots, g \text{ indicando el grupo.}$$

Prioridad grupal de cada uno de los g grupos de procesos. Estas prioridades se pueden fijar según distintos criterios; en esta propuesta se considerará que es función de la cardinalidad de cada grupo y se representa de la siguiente manera:

$$prg = \{prg_i = card(g_i)\} \text{ con } i = 1, \dots, g \text{ indicando el grupo.}$$

Recursos compartidos en la modalidad de exclusión mutua distribuida disponibles en los n nodos: $1, \dots, r$, El conjunto de recursos se representa de la siguiente manera:

$recursos = \{r_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, r$ (n° máximo de recursos en cada nodo), lo que se puede expresar mediante la Tabla 2.

Tabla 2. Recursos compartidos disponibles en cada nodo (E1)

Nodos		Recursos		
1	r_{11}	r_{12}		r_{1r}
....			...	
i	r_{i1}	r_{i2}		r_{ir}
....			...	
n	r_{n1}	r_{n2}		r_{nr}

Fuente: Elaboración propia. Basado (La Red Martínez, 2017).

Estos recursos compartidos disponibles alojados en distintos nodos del sistema distribuido podrán ser requeridos por los procesos (agrupados o independientes) en ejecución en los nodos, estas solicitudes de recursos por parte de los procesos se muestran en la Tabla 3.

Estados posibles de cada uno de los p procesos:

Tabla 3. Recursos solicitados por los procesos (E1)

Recursos			Procesos		
r_{11}	p_{11}		p_{kl}		p_{np}
....			...		
r_{ij}	p_{11}		p_{kl}		p_{np}
....			...		
r_{nr}	p_{11}		p_{kl}		p_{np}

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

- Proceso independiente.
- Proceso perteneciente a un grupo de procesos.

Estado posible de cada uno de los n nodos:

- Número de procesos.
- Prioridades de los procesos.
- Uso de CPU.
- Uso de memoria principal.
- Uso de memoria virtual.
- Memoria adicional necesaria para cada recurso solicitado por cada proceso (si se dispone del dato).
- Carga adicional de procesador estimada necesaria para cada recurso solicitado por cada proceso (si se dispone del dato).
- Carga adicional de E/S (entrada/salida) estimada necesaria para cada recurso solicitado por cada proceso (si se dispone del dato).
- Estado de cada uno de los r recursos compartidos en la modalidad de exclusión mutua distribuida existentes en el nodo:
 - Asignado a un proceso local o remoto.
 - Disponible.
 - Predisposición (prioridad nodal) para otorgar el acceso a cada uno de los r recursos compartidos en la modalidad de exclusión mutua distribuida (resultará de la consideración de las variables representativas del estado del nodo, de la prioridad de los procesos y de la carga computacional adicional que significaría asignar el recurso al proceso solicitante).

- Carga actual del nodo, que se podrá calcular como el promedio de los porcentajes de uso de CPU, memoria y entrada/salida en un momento dado (estos indicadores de carga podrán variar según los casos, pudiendo agregarse otros o cambiarse algunos de los puestos como ejemplo); también habrán de definirse las categorías de carga actual, por ejemplo, Alta, Media y Baja, señalándose los rangos de valores para cada categoría.

Una vez verificado e identificado la existencia de nodos accesibles y no accesibles, se atienden las solicitudes de los procesos y recursos correspondientes a los nodos accesibles. Aquellas asignaciones que no han podido ser resueltas, se agrupan con las nuevas solicitudes y se vuelve a tener una macro imagen (entendiéndose por macro imagen a la información referida al estado de los nodos respecto de la carga computacional actual y la demanda de asignación de recursos a procesos) para verificar si el nodo que estaba inaccesible ahora ya no lo está.

En base a la información recogida del sistema se puede detectar la presencia de nodos muy cargados con respecto a nodos que están inactivos o con baja carga de trabajo, una inadecuada distribución de la carga de trabajo, provocando un aumento en el tiempo de ejecución de los procesos que, a su vez, podría generar un incremento en el consumo de energía y perjudicar al rendimiento general del sistema.

En el siguiente apartado, se considerarán y evaluarán los estados referentes a los nodos, procesos y recursos y las opciones disponibles. Se establecerán las configuraciones iniciales del algoritmo, que podrán ser distintas para cada caso en particular y que permitirán adecuar una solución en base a la carga de trabajo de los nodos distribuidos y la locación de los procesos y recursos.

Para cubrir estas necesidades se plantea un conjunto de propuestas que consideran la

migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo homogeneidad de los nodos (respecto de compatibilidad de código, SO, arquitectura, set de instrucciones, etc.), incluyendo migración de procesos que significarán el acceso a datos distribuidos con alto impacto en E/S y tráfico de red (consumo de ancho de banda), y una posible migración de proceso consecuencia de cambios en el estado global del sistema y de los requerimientos variables de los procesos a lo largo del tiempo, esta verificación se hará cada cierto lapso Δt .

2.4. Descripción del operador de agregación

En cada nodo se define una interfaz entre las aplicaciones y el sistema operativo, que a través de un *Runtime* (software en tiempo de ejecución complementario del sistema operativo) incluido en esa interfaz, gestiona los procesos y recursos compartidos y define el escenario correspondiente. Además, los *Runtime* interactúan entre sí para intercambiar información y en uno de los nodos hay un coordinador global de *Runtime* que evalúa y ejecuta el modelo de decisión y el operador de agregación correspondiente.

Se plantea una propuesta que considera la migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo homogeneidad de los nodos (respecto de compatibilidad de código, sistema operativo, arquitectura, set de instrucciones, etc.).

El operador de agregación propuesto utiliza información proveniente de (La Red Martínez, 2017), relacionada con la carga computacional actual de los nodos, las categorías de carga computacional y de los vectores de pesos asociados a las mismas, las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo y las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles.

Los cálculos relacionados con la obtención de esta información, considerados preliminares,

son el punto de partida para otros que determinarán los procesos que serán migrados.

Al considerar aspectos relacionados con el balanceo de carga, se considera una política efectiva de migración, que tiene en cuenta la prioridad de los procesos, limitaciones en la migración (para determinar el número de procesos que pueden pasar de un nodo a otro), el peso de los recursos cuyo estado puede ser local respecto del proceso que lo requiere, pero podría cambiar a remoto al realizar la migración de procesos y el impacto de ésta en función de las prestaciones del nodo receptor según características de hardware.

Para un buen diseño en el balanceo de carga se consideran algunas políticas muy utilizadas para esta tarea, entre las cuales se pueden mencionar, políticas de:

- Locación, para determinar el destino del proceso seleccionado.
- Intercambio de información, que permita intercambiar información entre los nodos.
- Asignación, que determine la prioridad de ejecución de los procesos locales y remotos en un determinado nodo.
- Migración, que limite la cantidad de procesos migrables por nodo, y la cantidad de procesos que puede recibir un nodo destino.

Los procesos pueden ser locales o remotos, los segundos se diferencian en que se procesarán en un nodo distinto al original.

Los recursos serán considerados locales si están en el mismo nodo donde se aloja el proceso que los utilizará, y remotos si están en nodos distintos.

Para obtener información de la carga de nodos y estado global del sistema distribuido se deben realizar diferentes observaciones, de las que surgen varios escenarios que se representan mediante un diagrama de flujo que muestra una serie de situaciones en las que se deben tomar

decisiones que derivan en dos opciones posibles: "Sí" y "No". Cada opción se define como "Primera alternativa" y "Segunda alternativa" respectivamente.

Dependiendo del resultado de la evaluación, el flujo del diagrama se dirigirá hacia otra situación, donde se tomará otra decisión. Este proceso se repetirá hasta que se alcance una solución final, cuando ya no existan evaluaciones de recursos, procesos y nodos pendientes.

Se considerará la aplicación de las siguientes etapas adicionales para lograr balanceo de carga aplicando migración de procesos:

1. Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos.
2. Selección de los posibles nodos destino mediante la evaluación de su carga.
3. Evaluación del peso de los recursos y su posible impacto ante la migración de procesos.
4. Determinación del orden de migración de procesos y de los nodos destino correspondientes.
5. Evaluación del estado de las cargas computacionales una vez migrados los procesos.

Se presentará una variante innovadora al operador de agregación utilizado en (La Red Martínez, 2017) y (La Red Martínez et al., 2018), incluyendo la migración inteligente de procesos en sistemas distribuidos homogéneo.

El modelo se desarrolla en cuatro etapas bien definidas como se muestra en la Fig. 2.



Fig. 2. Etapas del cálculo de migración de procesos (E1).

Fuente: Elaboración propia

Las etapas corresponden a la evaluación de los nodos, clasificándolos según su nivel de carga (alta, media, baja), seguidamente se evalúan los procesos de los nodos con carga alta y los recursos vinculados a ellos. Se priorizan los nodos remotos que contengan mayor peso de recursos para cada proceso. Se obtiene un listado de los procesos candidatos a ser migrados que serán ordenados por su impacto en el ancho de banda de red, priorizándose los procesos con menor peso de transferencia.

2.4.1. Cálculo de la carga computacional actual de los nodos. Escenario 1 (E1)

En la se muestra un diagrama que contempla la evaluación a nivel nodal y se describen a continuación.

En el punto (a) de la Fig. 3 se realiza la configuración de valores iniciales, en donde se establecen los parámetros para indicar los porcentajes de carga y prioridades que se tendrán en cuenta para la migración. Para los nodos con carga alta se calcula el promedio de carga del porcentaje de uso de CPU, memoria y entrada/salida. Se consideran cargados los nodos cuyo promedio sea mayor a 70, carga media entre 40 y 70, y como carga baja a los nodos que como valor promedio tenga menor que 40.

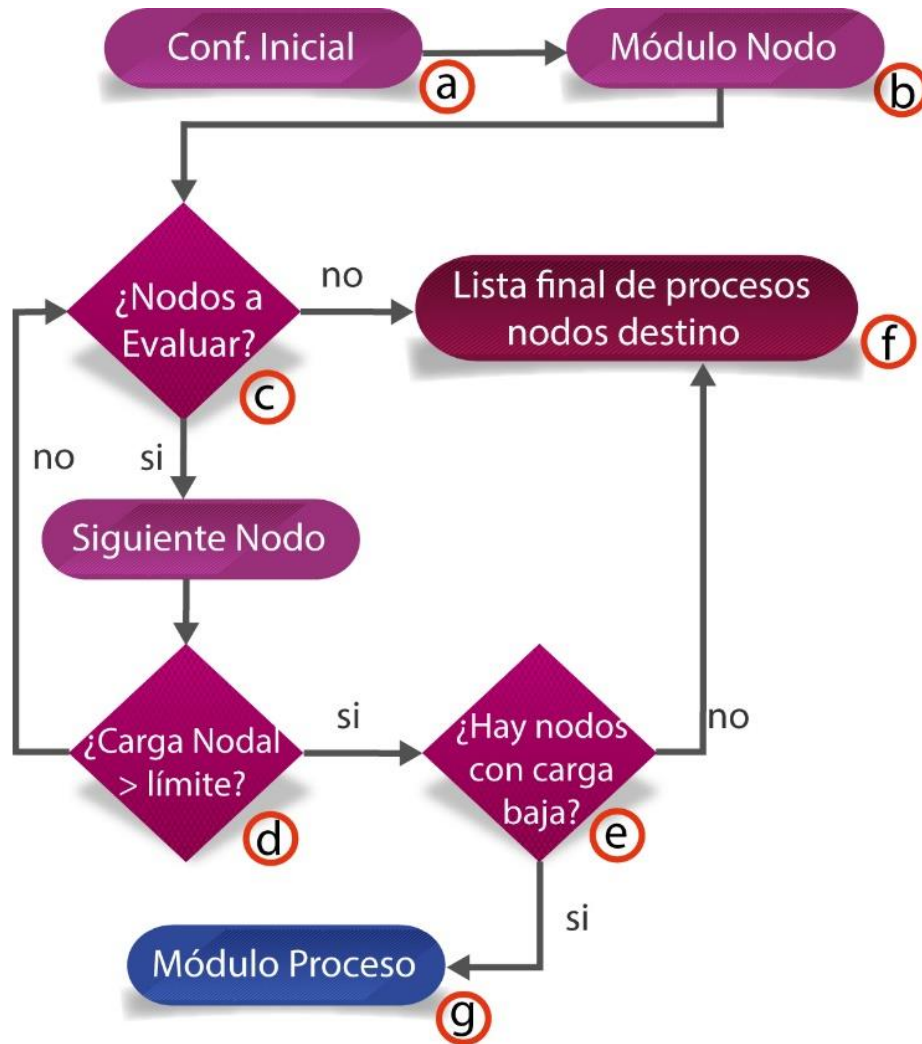


Fig. 3. Módulo de nodo (E1).
Fuente: Elaboración propia

(b) El primer paso consiste en analizar el escenario a nivel de nodos del sistema, considerando que cada uno puede tener diferentes procesos y recursos. Estos procesos pueden competir no solo con el acceso exclusivo a recursos locales (*RL*) sino también por aquellos que se encuentran en otros nodos, es decir, que se encuentran de manera remota. Para cada situación se considerarán diferentes alternativas. Esta evaluación se repetirá siempre y cuando existan nodos por evaluar, una vez terminada la evaluación de todos los nodos, se obtendrá un listado final de procesos y los nodos destinos.

(c) Situación: se analiza si existen nodos a evaluar dentro del sistema distribuido.

- *Primera alternativa* (sí): dado que existen nodos a evaluar se pasa a evaluar la carga nodal en el punto (d).
- *Segunda alternativa* (no): al no existir nodos a evaluar se pasa a evaluar el listado final de procesos y nodos destinos en el punto (f).

(d) Situación: se verifica si la carga del nodo supera un cierto límite, éste se refiere al valor mínimo de carga que debe superar el nodo para considerarse como cargado (a modo de ejemplo el valor mínimo de carga podría ser un 70%).

- *Primera alternativa* (sí): dado que la carga nodal supera el límite establecido se pasa al punto (e) para verificar si dentro del SD existen nodos con carga baja.
- *Segunda alternativa* (no): dado que la carga nodal no supera el límite se vuelve al punto (c) para evaluar otro nodo.

(e) Situación: se verifica si existen nodos con carga baja a donde puedan migrarse procesos.

- *Primera alternativa* (sí): si existen nodos con carga baja se pasa al módulo de procesos en (g) para analizar cada proceso con que cuenta el nodo. Esto corresponde a la etapa B., que se describe posteriormente.
- *Segunda alternativa* (no): si no quedan nodos con carga baja se vuelve al punto (f).

(f) Situación: se guarda la tabla de los procesos candidatos a ser migrados con sus posibles nodos destino e impacto correspondiente.

2.4.2. Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos (E1)

La información resultante de la evaluación del estado de los nodos ayuda a identificar aquellos con carga alta, sus procesos y los recursos tanto locales como remotos vinculados con dichos procesos. En el punto (g) de la Fig. 4, se da inicio al módulo de procesos, y es en donde se analiza el escenario a nivel de procesos, dentro de cada nodo.

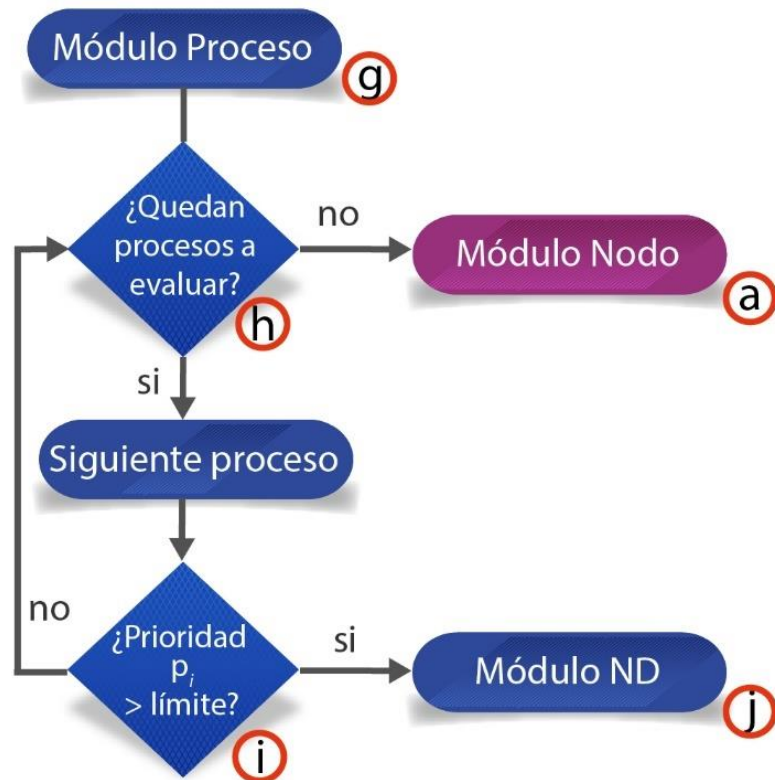


Fig. 4. Módulo proceso (E1)

Fuente: Elaboración propia

(h) Situación: se verifica dentro del nodo si existen procesos que aún no han sido evaluados.

- *Primera alternativa (sí):* si existen procesos por evaluar se pasa a (i) para verificar la prioridad del proceso.
- *Segunda alternativa (no):* si ya no quedan procesos por evaluar se vuelve al módulo nodos

en el punto (a).

(i) Situación: para el cálculo del promedio de prioridades de cada proceso se tendrán en cuenta el criterio de prioridad de procesos para todas las asignaciones de recursos (La Red Martínez, 2017). Por cada solicitud de recursos de cada proceso, se establece un criterio de prioridad, que es asignado por el Runtime del nodo al momento de hacer la solicitud. La sumatoria de este valor de prioridad, para cada recurso solicitado, dividido el total de solicitudes de cada proceso, indicará la prioridad promedio del proceso.

Se consideran candidatos a ser migrados los procesos cuya prioridad supere un límite fijado arbitrariamente; se consideran solo aquellos procesos cuyo promedio de prioridad supere 0.6.

$$\text{promedio de prioridad de proceso } (p_{ij}) = (\sum cp_{ijk} / o) \quad (1)$$

con p_{ij} indicando los procesos, donde $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, m$ (n° de procesos de cada nodo),

con cp_{ijk} indicando el criterio prioridad de proceso relacionado a cada recurso, donde $i = 1, \dots, n$ (n° de nodos en el sistema distribuido), $j = 1, \dots, m$ (n° de procesos de cada nodo), y $k = 1, \dots, o$ (n° de recursos para cada proceso),

con o = cantidad de asignaciones asociadas a cada proceso, lo que se puede expresar mediante la Tabla 4.

Tabla 4. Promedio de prioridad de procesos (E1)

Procesos	r_1	...	r_k	...	r_o	Promedio
p_{11}	cp_{111}	...	cp_{11k}	...	cp_{11o}	ppp_{11}
...	...	...	...	...	...	...
p_{ij}	cp_{ij1}	...	cp_{ijk}	...	cp_{ijo}	ppp_{ij}
...	...	...	...	...	...	...
p_{nm}	cp_{nm1}	...	cp_{nmk}	...	cp_{nmo}	ppp_{nm}

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Con r_k indicando los recursos, donde $k = 1, \dots, o$ (nº de recursos para cada proceso), con ppp_{ij} indicando los promedios de prioridad de los procesos, donde $i = 1, \dots, n$ (nº de nodos en el sistema distribuido) y $j = 1, \dots, m$ (nº de procesos de cada nodo).

Para todos los procesos de los nodos con carga alta se utilizará el mismo método para seleccionar los procesos candidatos a ser migrados.

- *Primera alternativa* (sí): si supera el límite establecido se pasa a evaluar a los posibles nodos destinos en el punto (**j**).
- *Segunda alternativa* (no): si el promedio de prioridad de proceso no supera el límite establecido se vuelve a evaluar otro proceso en el punto (**h**).

2.4.3. Selección de los posibles nodos destino mediante la evaluación de su carga

(E1)

En esta etapa se evalúa si existen nodos con carga baja que puedan alojar procesos de otros nodos. Los procesos que serán evaluados son aquellos que clasificaron en la etapa anterior. Se denotan los nodos que se encuentran dentro de la categoría de carga baja:

$nodos\ destino\ con\ carga\ baja = \{NDB\} = \{n_1, \dots, n_n\}$ con n indicando el nº total de posibles nodos destino según se muestra en la Tabla 5.

Tabla 5. Nodos con carga baja (E1)

Nodos con carga baja
n_1
....
n_i
....
n_n

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Para cada proceso puede haber varios nodos con carga baja disponibles. Teniendo en cuenta el listado de la Tabla 5 se verifican los recursos solicitados por cada proceso p_{ij} para determinar el nodo candidato.

2.4.4. Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E1)

Una vez identificado los procesos del nodo con carga alta y los posibles nodos destinos, se realiza la comparación de los recursos locales en relación con los remotos, con el fin elaborar la lista preliminar de los procesos migrables y sus posibles nodos destinos.

En la Fig. 5 se puede apreciar la distribución de los recursos de un proceso en un determinado momento. La opción a) representa al nodo origen con sus procesos y recursos, la opción b) representa al nodo remoto, que no tiene recursos asignados, la opción c) también representa al nodo remoto, pero al tener recursos asignados en ese nodo se lo denomina nodo destino.

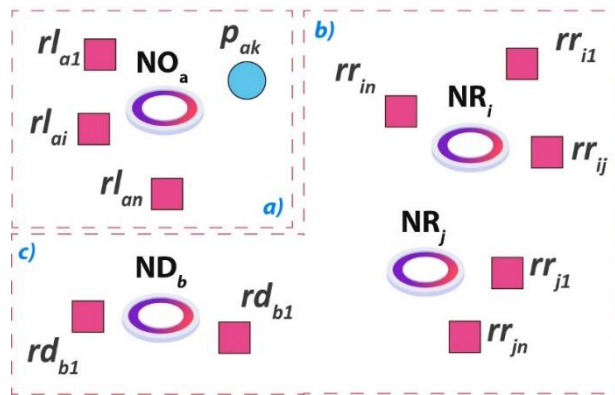


Fig. 5. Recursos locales vs. Recursos remotos (E1)

Fuente: Elaboración propia

NO_a , NO representa el nodo origen, a el número de nodo origen.

p_{ak} , p representa el proceso, a el nodo origen y k el número de proceso.

rl_{ai} , donde rl representa al recurso local, con a indicando el nodo origen, con i de 1 a n indicando el número de recursos.

NR_i , NR representa los nodos remotos con carga baja, con i el número de nodo remoto.

rr_{ij} , donde rr representa el recurso remoto que solicita el proceso, con i indicando el nodo remoto, con j de 1 a n indicando el número de recursos.

ND_b , ND representa el posible nodo destino que cumple con los requisitos de carga baja, compatibilidad y peso mínimo de transferencia de recursos (entendiéndose dicho peso como el impacto en el consumo de ancho de banda); b es el número de nodo destino.

rd_{bg} , donde rd representa el recurso remoto solicitado por un proceso, alojado en un posible nodo destino, con b indicando el nodo destino, con g de $1, \dots, n$, indicando el número de recurso.

En el punto **(j)** de la Fig. 6, se activa el módulo de nodos destino, donde se buscan nodos remotos que puedan albergar recursos solicitados por cada proceso evaluado. Se busca generar el menor peso de transferencia al pasar los recursos locales (que pueda tener en un momento dado un proceso), a remotos (al migrar el proceso). Cualquier nodo remoto compatible con el nodo origen será evaluado, y podrá ser candidato a convertirse en un nodo destino.

(k) Situación: se verifica dentro del sistema distribuido si existen posibles nodos destino (ND) que aún no han sido evaluados.

- *Primera alternativa* (sí): si existen nodos destino que aún no han sido evaluados se selecciona el siguiente nodo destino y se pasa al punto **(l)**.

Segunda alternativa (no): si no existen nodos destino para evaluar entonces se vuelve al módulo de procesos en el punto **(g)**.

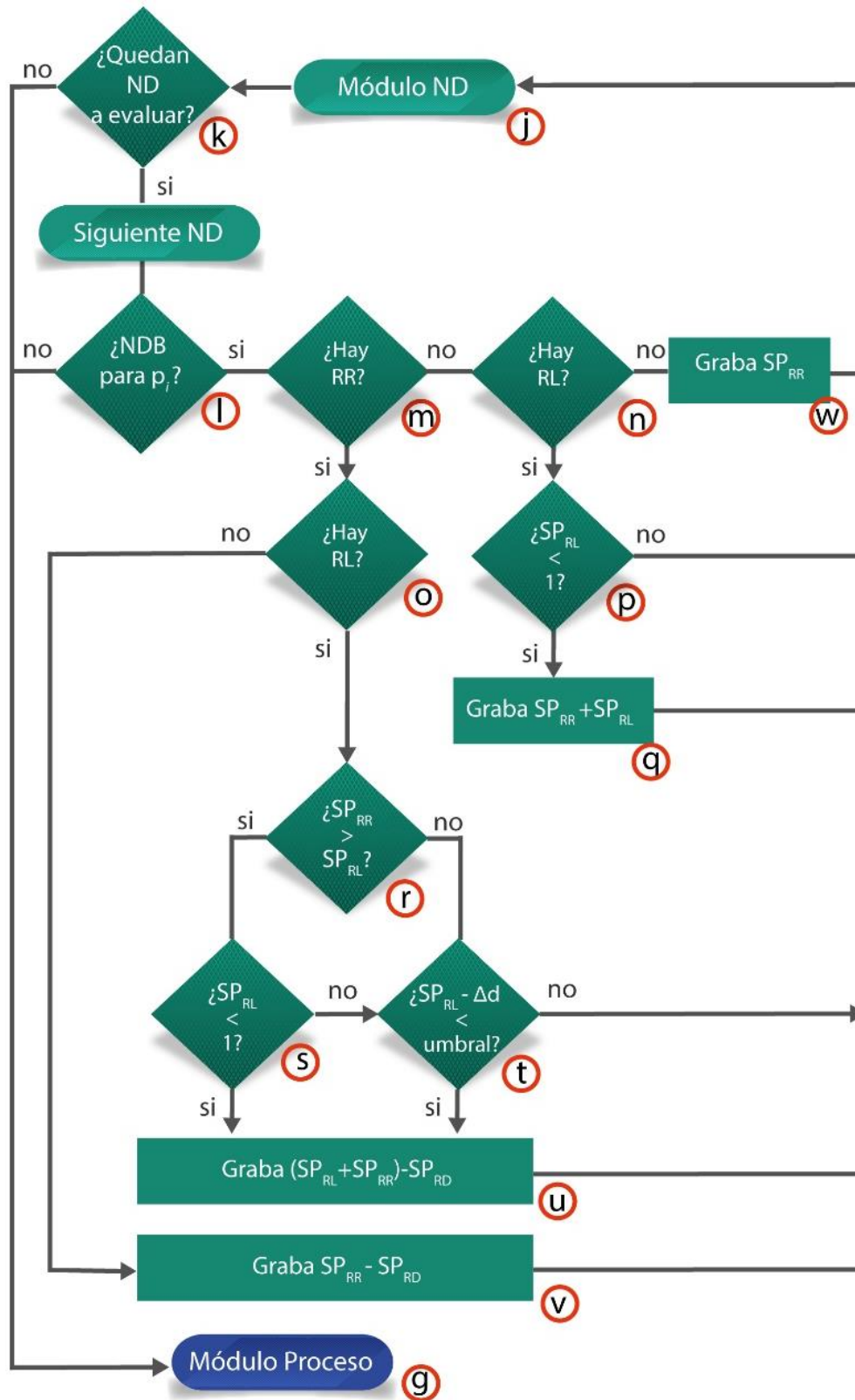


Fig. 6. Módulo nodo destino (E1).

Fuente: Elaboración propia

(**l**) Situación: se evalúa si para el proceso en cuestión existen nodos destino con carga baja (**NDB**) considerando compatibilidad de códigos, arquitectura, set de instrucciones, etc.

- *Primera alternativa* (sí): si para el proceso existen nodos con carga baja se pasa al punto (**m**).
- *Segunda alternativa* (no): si no existen nodos con carga baja entonces se vuelve al punto (**g**).

(**m**) Situación: se verifica si el proceso tiene asociados recursos remotos (**RR**)

- *Primera alternativa* (sí): si el proceso tiene recursos remotos se realiza (**o**).
- *Segunda alternativa* (no): si el proceso no tiene recursos remotos entonces se evalúa (**n**).

(**n**) Situación: se verifica si el proceso en cuestión tiene recursos locales.

- *Primera alternativa* (sí): si el proceso tiene recursos de forma local se evalúa (**p**).
- *Segunda alternativa* (no): si el proceso no tiene recursos de forma local entonces se vuelve a (**j**).

(**o**) Situación: se verifica si el proceso en cuestión tiene recursos locales.

- *Primera alternativa* (sí): si el proceso tiene recursos de forma local se realiza (**r**).
- *Segunda alternativa* (no): si el proceso no tiene recursos de forma local entonces se ejecuta el punto (**v**).

(**p**) Situación: se verifica si la sumatoria de pesos de los recursos locales (**SP_{RL}**) es menor a un límite; sin pérdidas de generalidad, en este caso se considera que el valor límite es 1 (uno).

- *Primera alternativa* (sí): si la sumatoria de pesos de los recursos locales es menor a 1 (uno) entonces se ejecuta el punto (**q**).

- *Segunda alternativa* (no): si la sumatoria de pesos de los recursos locales es no menor a 1 (uno) entonces se vuelve al punto (j).

(q) Situación: en el vector de migración de procesos (Fig. 7) se almacena el valor resultante de la suma de pesos de recursos locales y los pesos de recursos remotos, peso de transferencia que generaría la migración del proceso al nodo destino.

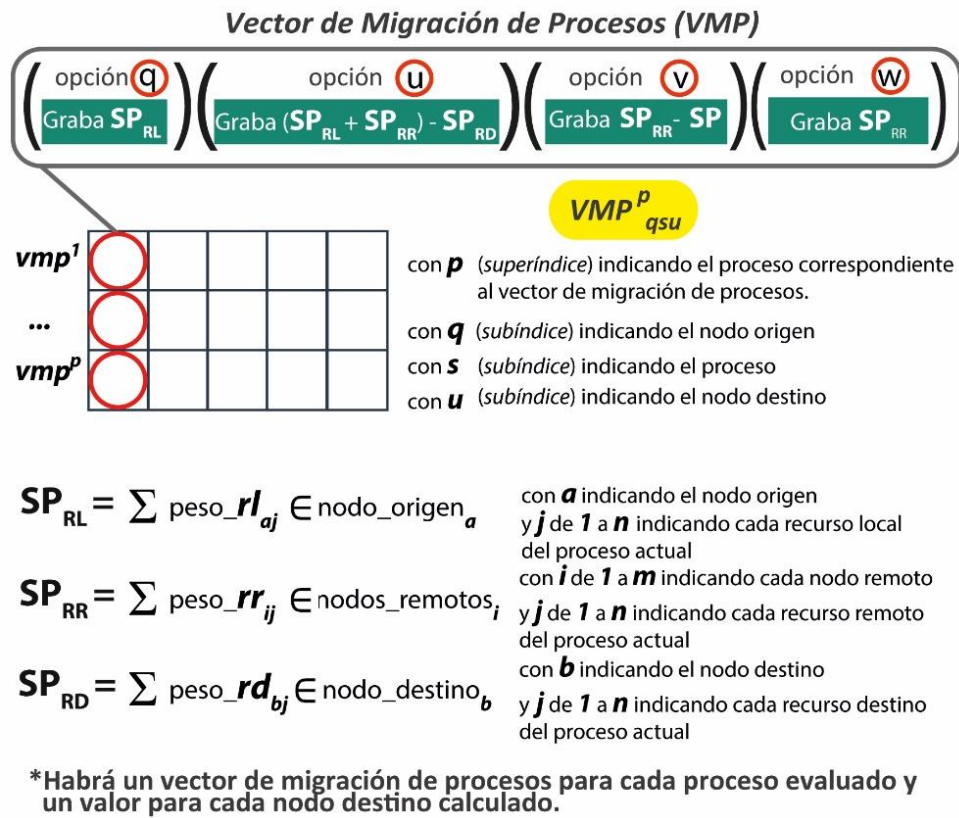


Fig. 7. Vector de migración de proceso (E1)

Fuente: Elaboración propia

(r) Situación: Se verifica si la suma de los pesos de los recursos remotos (SP_{RR}) es mayor a la suma de los pesos de recursos locales.

- *Primera alternativa* (sí): Si la sumatoria de los pesos de recursos remotos es mayor al peso de los recursos locales se realiza (s).
- *Segunda alternativa* (no): Si la sumatoria de los pesos de recursos remotos es menor al

peso de los recursos locales se realiza (**t**).

(**s**) Situación: Se verifica si la suma de los pesos del recurso local es menor a un límite (para este caso límite es igual a uno en base a (La Red Martínez, 2017)).

- *Primera alternativa* (sí): si la suma de los pesos del recurso local es menor al límite entonces se realiza (**u**).
- *Segunda alternativa* (no): si la suma de los pesos del recurso local es mayor al límite se evalúa (**t**).

(**t**) Situación: se verifica si la suma de los pesos de recursos locales menos Δd (Δd es el peso de los recursos destino más un 20% del total de pesos de recursos del proceso) es menor a un umbral (sin pérdidas de generalidad, en este caso se considera que el umbral es 1 (uno)).

- *Primera alternativa* (sí): si la suma de los pesos de recursos locales menos Δd es menor a un umbral, entonces se realiza el punto (**u**).
- *Segunda alternativa* (no): si la suma de los pesos de recursos locales menos Δd es mayor a un umbral, se vuelve a (**j**).

(**u**) Situación: En el vector de migración de procesos (Fig. 7) se almacena el valor resultante de la suma de pesos de recursos locales más la suma de los pesos de los recursos remotos menos la suma de pesos de recursos destino (SP_{RD}), peso de transferencia que generaría la migración del proceso al nodo destino.

(**v**) Situación: En el vector de migración de procesos (Fig. 7) se almacena el valor resultante de la suma de pesos de recursos locales menos la suma de pesos de recursos destino, peso de transferencia que generaría la migración del proceso al nodo destino.

(w) Situación: En el vector de migración de procesos (Fig. 7) se almacena el valor resultante de la suma de pesos de recursos remotos que tiene asociado el proceso, peso de transferencia que generaría la migración del proceso al nodo destino.

2.4.5. Determinación del orden de migración de procesos y de los nodos destino correspondientes (E1)

En la Tabla 6 se observa la matriz general de procesos (*MGP*), resultado de la evaluación de las diferentes situaciones que pueden presentarse dentro de un sistema distribuido para la migración inteligente de procesos. Se verifica qué procesos migrar y adónde migrar, considerando el peso de transferencia que se generaría en la migración. Para cada proceso se guarda el conjunto de nodos con carga baja y para cada nodo evaluado el valor del peso de transferencia del vector *vmp* obtenido en la Fig. 7.

Tabla 6. Matriz general de procesos (E1)

<i>MGP</i>	n_1	...	n_k	...	n_m
mgp_1	vmp^1_{111}	...	vmp^1_{11u}	...	vmp^1_{11m}
...	...	...	...	...	...
mgp_i	vmp^p_{qs1}	...	vmp^p_{qsu}	...	vmp^p_{qsm}
...	...	...	...	...	...
mgp_o	vmp^n_{rt1}	...	vmp^n_{rtu}	...	vmp^n_{rtm}

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Con i en mgp_i variando de 1, ..., o , indicando el proceso correspondiente en la matriz general de procesos.

Con p en vmp (*superíndice*) variando de 1, ..., n , indicando el proceso correspondiente al vector de migración de procesos.

Con q,s,u en vmp (*subíndices*) variando q de 1, ..., r , indicando el nodo origen, s de 1, ..., t , indicando el número de proceso y u de 1, ..., m , indicando el nodo destino, del vector de

migración de procesos.

Con k en n variando de $1, \dots, m$, indicando el nodo con carga baja para cada proceso.

El primer paso que habrá que realizar a partir de los datos de la matriz general es dividirla en dos matrices, una primaria (primera ronda de asignación) y otra secundaria (segunda ronda de asignación). La diferencia radica en que primero se intentará favorecer a los procesos que tienen recursos en los nodos destinos, es decir, cuya migración implicaría una reducción en el impacto de entrada salida (transferencia) con respecto a aquellos que no tengan recursos en nodos destino; para ello se utiliza la matriz primaria. La segunda matriz se analizará luego de haber evaluado la matriz primaria.

Matriz Primaria

En la Tabla 7 se observa la matriz primaria (MP) que se obtiene de la matriz general de procesos. Esta matriz contiene datos solamente de aquellos procesos que tienen asignados recursos en los nodos destino.

Tabla 7. Matriz primaria (E1)

MP	n_1	...	n_k	...	n_m
mp_1	vmp^1_{111}	...	vmp^1_{11u}	...	vmp^1_{11m}
...	...	...	...	...	...
mp_i	vmp^p_{qs1}	...	vmp^p_{qsu}	...	vmp^p_{qsm}
...	...	...	...	...	...
mp_o	vmp^n_{rt1}	...	vmp^n_{rtu}	...	vmp^n_{rtm}

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Con i en mp_i variando de $1, \dots, o$, indicando el proceso correspondiente en la matriz general de procesos.

Con p en vmp (*superíndice*) variando de $1, \dots, n$, indicando el proceso correspondiente al vector de migración de procesos.

Con q, s, u en vmp (subíndices) variando q de 1, ..., r , indicando el nodo origen, s de 1, ..., t , indicando el número de proceso y u de 1, ..., m , indicando el nodo destino, del vector de migración de procesos.

Con k en n variando de 1, ..., m , indicando el nodo con carga baja para cada proceso.

Se considerará para cada proceso el menor peso de transferencia en la matriz primaria, es decir, se verifica todos los pesos de la matriz primaria y se ordena de menor a mayor según el peso de transferencia, el proceso que esté asociado al menor peso tendrá prioridad en la selección del nodo destino. La evaluación continúa con los procesos pendientes. Por cada ronda de evaluación se verifica que el porcentaje de migración del nodo origen sea menor al límite establecido, ya que el nodo puede migrar procesos mientras no supere ese límite. Este cálculo se realiza sobre la cantidad total de procesos por nodo.

Se inicia la elección del nodo destino para el primer proceso con menor peso de transferencia. Al terminar cada ronda, se eliminan de la matriz primaria el proceso y el nodo asignado recientemente. Los procesos cuyos recursos destino estaban únicamente en el nodo seleccionado pasan a formar parte de la matriz secundaria (MS) (Tabla 8).

Los procesos que se encuentran en nodos origen que hayan alcanzado el límite de migración establecido, son descartados y pasan a formar parte de una matriz de procesos no migrables ($MPNM$) (Tabla 11). Se continúa la evaluación mientras existan procesos a evaluar. El orden de asignación de los procesos a nodos destino de la matriz primaria se indica mediante la Función de Asignación para la Migración en Sistemas Distribuidos ($FAMSD$) primaria.

$$FAMSD^p \text{ primaria} = vmp^p_{qsu} \text{ del mínimo}(MP^p) \quad (2)$$

Con p indicando el proceso correspondiente a cada proceso de la matriz primaria.

Calculando la *FAMSD* primaria para todos los procesos de la matriz primaria se obtendrá un vector que indica el orden de migración de procesos a nodos destino. Esto se indica en la Tabla 8.

Tabla 8. Función de asignación para migración en sistemas distribuidos (*FAMSD*) primaria (E1)

Peso de transferencia del proceso	Proceso seleccionado
1º: vmp_{qsu}^p del Mínimo(MP^p)	p_{ij} del <i>MP</i> seleccionado
2º: vmp_{qsu}^p del Mínimo(MP^p) para los p_{ij} no asignados	p_{ij} del <i>MP</i> seleccionado
...	...
último: vmp_{qsu}^p del Mínimo(MP^p) del p_{ij} no asignado	p_{ij} del <i>MP</i> seleccionado

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Matriz Secundaria

En la Tabla 9 se observa la matriz secundaria que resulta de la matriz general de procesos. Esta tabla contiene datos de procesos que no tienen asociados recursos (requerimientos de accesos) en nodos destino.

Tabla 9. Matriz secundaria (E1)

<i>MS</i>	n_1	...	n_k	...	n_m
ms_1	vmp_{111}^1	...	vmp_{11u}^1	...	vmp_{11m}^1
...	...	...	...	...	...
ms_i	vmp_{qs1}^p	...	vmp_{qsu}^p	...	vmp_{qsm}^p
...	...	...	...	...	...
ms_o	vmp_{rt1}^n	...	vmp_{rtu}^n	...	vmp_{rtm}^n

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Con i en ms_i variando de 1, ..., o , indicando el proceso correspondiente en la matriz general de procesos.

Con p en vmp (*superíndice*) variando de 1, ..., n , indicando el proceso correspondiente al vector de migración de procesos.

Con q,s,u en vmp (*subíndices*) variando q de 1, ..., r , indicando el nodo origen, s de 1, ..., t , indicando el número de proceso y u de 1, ..., m , indicando el nodo destino, del vector de

migración de procesos.

Con k en n variando de $1, \dots, m$, indicando el nodo con carga baja para cada proceso.

La matriz secundaria se analiza una vez que la primera ronda de evaluación correspondiente a la matriz primaria haya terminado. Se aplica el mismo criterio de evaluación que en la matriz primaria, pero para procesos que no tienen requerimientos de acceso a recursos de otros nodos. El orden de asignación de los procesos a nodos destino de la matriz secundaria se indica a continuación mediante la Función de Asignación para la Migración en Sistemas Distribuidos (*FAMSD*) secundaria.

$$FAMSD^p \text{ secundaria} = vmp_{qsu}^p \text{ del mínimo } (MS^p) \quad (3)$$

Calculando la *FAMSD* para todos los procesos de la matriz secundaria se obtendrá un vector que indica el orden de asignación de procesos a nodos destino. Esto se indica en la Tabla 10.

Tabla 10. Función de asignación para migración en sistemas distribuidos (*FAMSD*) secundaria (E1)

Peso de transferencia del proceso	Proceso seleccionado
1º: vmp_{qsu}^p del Mínimo (MS^p)	p_{ij} del MS seleccionado
2º: vmp_{qsu}^p del Mínimo (MS^p) para los p_{ij} no asignados	p_{ij} del MS seleccionado
....	
último: vmp_{qsu}^p del Mínimo (MS^p) del p_{ij} no asignado	p_{ij} del MS seleccionado

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Matriz de procesos no migrables

En la Tabla 11 se observa la matriz de procesos no migrables, que resulta de los procesos que se encuentran en nodos origen, que hayan alcanzado el límite de migración establecido y de procesos cuyos nodos destino fueron seleccionados por otros procesos.

Tabla 11. Matriz de procesos no migrables (E1)

$MPNM$	n_1	...	n_k	...	n_m
$mpnm_1$	$mpnm^1_{111}$	...	$mpnm^1_{11u}$	...	$mpnm^1_{11m}$
...	...	...	...	...	...
$mpnm_i$	$mpnm^p_{qs1}$	...	$mpnm^p_{qsu}$	...	$mpnm^p_{qsm}$
...	...	...	...	...	...
$mpnm_o$	$mpnm^n_{rt1}$	...	$mpnm^n_{rtu}$	...	$mpnm^n_{rtm}$

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Con i en $mpnm_i$ variando de 1, ..., o , indicando el proceso correspondiente en la matriz general de procesos.

Con p en $mpnm$ (*superíndice*) variando de 1, ..., n , indicando el proceso correspondiente al vector de migración de procesos.

Con q,s,u en $mpnm$ (*subíndices*) variando q de 1, ..., r , indicando el nodo origen, s de 1, ..., t , indicando el número de proceso y u de 1, ..., m , indicando el nodo destino, del vector de migración de procesos.

Con k en n variando de 1, ..., m , indicando el nodo con carga baja para cada proceso.

De la evaluación de las matrices primaria y secundaria se obtiene la Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) primaria y la Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) secundaria.

La unión de estas dos asignaciones conforma lo que se denomina Función de Asignación para la Migración en Sistemas Distribuidos Concatenada (*FAMSDC*), como se muestra en la Fig. 8 y en la Tabla 12, que contiene la lista final de las asignaciones de cada ronda, proceso, nodo origen, nodo destino y peso de transferencia.

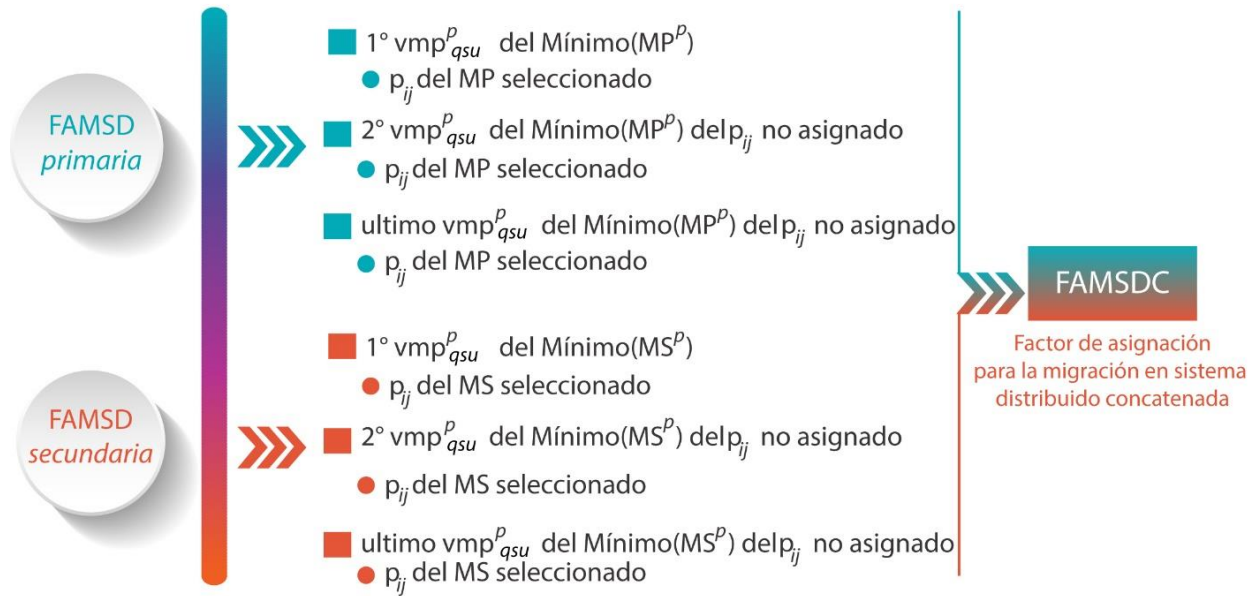


Fig. 8. Función de Asignación para la Migración en Sistemas Distribuidos Concatenada (E1)
Fuente: Elaboración propia

Tabla 12. Concatenación de las tablas de asignación (*FAMSD* primaria y *FAMSD* secundaria) (E1)

<i>FAMSDC</i>	Iteraciones
<i>FAMSD primaria</i>	Filas de 1 a n n = número de filas de la <i>FAMSD primaria</i> en la primera iteración
<i>FAMSD secundaria</i>	Filas de $n+1$ a m m = número de filas de la <i>FAMSD secundaria</i> en la segunda iteración

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

2.4.6. Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema (E1)

Para obtener un indicador de la carga computacional actual de cada nodo se pueden adoptar distintos criterios; en esta propuesta los criterios serán el % de uso de la CPU, el % de uso de la memoria y el % de uso de operaciones de entrada / salida.

Se aplica el operador de agregación desarrollado en (La Red Martínez, 2017) para obtener carga computacional actual de los nodos, las categorías de carga computacional y de los vectores de pesos asociados a las mismas, las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo y las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles.

Para evaluar y comparar los nuevos valores de cargas computacionales se realiza en dos etapas, la primera es aplicar la sobrecarga sin realizar la migración de procesos y la segunda es aplicar la sobrecarga una vez migrado los procesos, se utilizarán los siguientes criterios asociado a cada proceso migrado: sobrecarga de memoria (memoria adicional que tendrá el nodo solicitado), sobrecarga de procesador (uso adicional de procesador que tendrá el nodo solicitado) y sobrecarga de entrada / salida (entrada / salida adicional que tendrá el nodo solicitado).

La carga computacional de cada nodo se calculará de dos maneras.

a. Cálculo de sobrecarga sin migración

- Establecimiento del n° de criterios para determinar la carga de los nodos:

$$card(\{criterios\}) = c$$

- Establecimiento de los criterios que se aplicarán (podrán diferir de un nodo a otro):

$criterios = \{c_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, c$ (n° máximo de criterios para cada nodo), lo que se puede expresar mediante la Tabla 13.

Al promedio de carga computacional que se obtiene mediante el operador de agregación desarrollado en (La Red Martínez, 2017) se le agrega un criterio adicional que es la sobrecarga, que se obtiene con la suma de los promedios de sobrecargas de cada proceso del nodo.

Cálculo de la carga computacional actual de cada nodo:

$carga_i = (\text{valor_criterio}(c_{i1}) + \dots + \text{valor_criterio}(c_{ic})) / m$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, c$ (n° de criterios para cada nodo), con m cantidad de criterios del nodo. (4)

Cálculo de la sobrecarga estimada que generaría la nueva asignación de recursos a procesos:

$$sobrecarga_i = \sum(\text{prom_sobrec}(p_{ij})) \quad (5)$$

con $i = 1, \dots, n$ (n° de nodos) y $j = 1, \dots, k$ (n° de procesos en el nodo).

Cálculo de la carga computacional de cada nodo contemplando la sobrecarga de las nuevas asignaciones:

$$carga_final_sin_migración_i = carga_i + sobrecarga_i \quad (6)$$

Tabla 13. Criterios para medir la carga computacional en cada nodo (E1)

Nodos		Criterios			
1	C_{11}	C_{12}	...	C_{1c}	
...	...	...	...	...	
i	C_{i1}	C_{i2}	...	C_{ic}	
...	...	...	...	...	
n	C_{n1}	C_{n2}	...	C_{nc}	

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

b. Cálculo de sobrecarga con migración

Se realiza el mismo procedimiento que para el cálculo de sobrecarga sin migración, pero se tiene en cuenta la sobrecarga que generan los procesos que no se migran del nodo actual y la sobrecarga que generan los procesos migrados que éste recibe.

En relación con la migración de procesos, estos pueden tener diferentes estados. Los procesos migrados entrantes son aquellos procesos que fueron migrados desde un nodo origen hacia el nodo actual. Los procesos migrados salientes son los procesos que fueron migrados del nodo actual hacia un nodo destino. Los procesos no migrados son aquellos procesos que permanecen en el nodo actual.

Cálculo de la carga computacional actual de cada nodo:

$carga_i = (\text{valor_criterio}(c_{i1}) + \dots + \text{valor_criterio}(c_{ic})) / m$ con $i = 1, \dots, n$ (nº de nodos en el sistema distribuido) y $j = 1, \dots, c$ (nº de criterios para cada nodo), con m cantidad de criterios del nodo.

(7)

Cálculo de la sobrecarga estimada que generaría la nueva asignación de recursos a los procesos que no se migran del nodo actual:

$sobrecarga_no_migrados_i = \sum(\text{prom_sobrec}(p_{ij} \text{ no migrados})) = \sum(\text{prom_sobrec}(p_{ij})) - \sum(\text{prom_sobrec}(p_{ij} \text{ salientes}))$

(8)

Cálculo de la sobrecarga estimada que generaría la nueva asignación de recursos a los procesos que son migrados hacia el nodo actual:

$sobrecarga_inmigrados_i = \sum (\text{prom sobrec}(p_{ij} \text{ inmigrados}))$

(9)

Cálculo de la carga computacional de cada nodo contemplando la sobrecarga de las nuevas asignaciones:

$carga_final_con_migración = carga_i + sobrecarga_no_migrados_i + sobrecarga_inmigrados_i$

(10)

2.4.7. Métrica propuesta para evaluar el balanceo de carga en el sistema distribuido (E1)

La aplicación de una métrica es esencial para la evaluación de la investigación. Al tener una medida objetiva de los resultados, se pueden comparar los resultados de una investigación con otros de investigaciones similares, lo que permite evaluar su relevancia.

Para evaluar la eficacia del modelo propuesto, es necesario medir los resultados para determinar si se comporta según lo deseado.

Para evaluar el impacto de la migración de procesos y cómo esta afecta el estado global del sistema, se utilizará la desviación estándar. Esta medida considerará la carga final de los nodos del sistema distribuido para evaluar la variabilidad o dispersión de los datos.

La desviación estándar se aplicará a los promedios finales obtenidos de la aplicación de sobrecarga sin migración y de la aplicación de sobrecarga con migración. Si la desviación estándar de los valores de los promedios finales de sobrecarga sin migración es mayor que la desviación estándar de los mismos con migración, esto indica que hay una mayor variabilidad en los resultados. Por lo tanto, los valores de sobrecarga después de la migración estarán más balanceados porque estarán más cerca de la media.

Para el cálculo de la desviación estándar sin considerar migración se procede de la siguiente manera:

$$desvest_sin_migración (carga_final_sin_migración) = s \quad (11)$$

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}}$$

Donde $\{x_1, x_2, \dots, x_n\}$ son los valores de carga final de cada uno de los nodos (sin contemplar migración), es el valor medio de todos estos elementos y N es el número total de nodos en el sistema distribuido.

Para el cálculo de la desviación estándar considerando la migración se procede de manera similar considerando los valores de sobrecarga luego de la migración de procesos, aplicando la fórmula (11).

2.5. Ejemplo

En esta sección se explicará en detalle un ejemplo de aplicación del operador de agregación propuesto. El sistema de procesamiento distribuido, las estructuras de datos, los recursos y los procesos que se ejecutan en los diferentes nodos, grupos, cardinalidades, criterios y categorías para evaluar las diferentes cargas y cálculos necesarios, son los mencionados en (La Red Martínez, 2017), (Agostini & La Red Martínez, 2019), (Agostini et al., 2019) y (Agostini et al., 2018).

Para generar la entrada de datos para el ejemplo desarrollado se ha utilizado un simulador (La Red Martínez et al., 2022) haciendo posible la representación de los procesos, recursos y nodos en un modelo dinámico, simulándose diferentes cargas de trabajo para el sistema distribuido.

El sistema de procesamiento distribuido tiene diez nodos:

$$nodos = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Los procesos que se ejecutan en los nodos son los siguientes: siete procesos en el nodo 1, cinco procesos en el nodo 2 y siete procesos en el nodo 3. Los nodos del 4 al 10 son nodos que se agrega al modelo original mencionados en (La Red Martínez, 2017), (Agostini & La Red Martínez, 2019), (Agostini et al., 2019) y (Agostini et al., 2018).

$procesos = \{p_{ij}\}$ con i indicando el nodo y j indicando el proceso, lo que se puede expresar mediante la Tabla 14.

Tabla 14. Procesos en cada nodo (E1)

Nodos		Procesos					
1	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}
2	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}		
3	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Los recursos compartidos disponibles en los nodos son los siguientes: cuatro recursos en el nodo 1, cuatro en el nodo 2, tres en el nodo 3, tres en el nodo 4, tres en el nodo 5, dos en el nodo 6, cuatro en el nodo 7 y tres en el nodo 8.

$recursos = \{r_{ij}\}$ con i indicando el nodo y j indicando el recurso, lo que se puede expresar mediante la Tabla 15.

Tabla 15. Recursos en cada nodo (E1)

Nodos		Recursos			
1	r_{11}	r_{12}	r_{13}	r_{14}	
2	r_{21}	r_{22}	r_{23}	r_{24}	
3	r_{31}	r_{32}	r_{33}		
4	r_{41}	r_{42}	r_{43}		
5	r_{51}	r_{52}	r_{53}		
6	r_{61}	r_{62}			
7	r_{71}	r_{72}	r_{73}	r_{74}	
8	r_{81}	r_{82}	r_{83}		

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Las solicitudes de recursos por parte de los procesos se muestran en la Tabla 16.

Tabla 16. Recursos solicitados por los procesos (E1)

Recursos	Procesos
r_{11}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{16}, p_{17}, p_{24}, p_{32}, p_{33}, p_{36}, p_{37}$
r_{12}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{16}, p_{17}, p_{21}, p_{23}, p_{24}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$
r_{13}	$p_{13}, p_{14}, p_{16}, p_{17}, p_{21}, p_{31}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}$
r_{14}	p_{16}

Recursos	Procesos
r_{21}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{22}, p_{25}, p_{33}, p_{36}, p_{37}$
r_{22}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{16}, p_{21}, p_{22}, p_{33}, p_{34}, p_{35}, p_{36}$
r_{23}	$p_{11}, p_{21}, p_{24}, p_{32}, p_{33}, p_{34}$
r_{24}	$p_{11}, p_{23}, p_{24}, p_{34}, p_{35}, p_{36}$
r_{31}	$p_{12}, p_{13}, p_{16}, p_{21}, p_{22}, p_{23}, p_{31}, p_{34}, p_{35}, p_{36}$
r_{32}	$p_{13}, p_{15}, p_{23}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$
r_{33}	$p_{12}, p_{13}, p_{15}, p_{16}, p_{21}, p_{22}, p_{23}, p_{31}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$
r_{41}	p_{14}
r_{42}	p_{14}
r_{43}	p_{14}
r_{51}	p_{15}
r_{52}	p_{15}
r_{53}	p_{15}
r_{61}	p_{15}
r_{62}	p_{15}
r_{71}	p_{31}
r_{72}	p_{31}
r_{73}	p_{35}
r_{74}	p_{35}
r_{81}	p_{36}
r_{82}	p_{36}
r_{83}	p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

A continuación, se describirá cada una de las etapas de cálculo.

2.5.1. Cálculo de la carga computacional actual de los nodos (E1)

Para obtener un indicador de la carga computacional actual de cada nodo se adoptarán los mismos tres criterios en los tres nodos:

$$card(\{criterios\}) = 3$$

$criterios = \{\% \text{ de uso de la CPU, el \% de uso de la memoria, \% de uso de operaciones de entrada / salida}\}.$

Los valores que se asumirán para los indicadores de carga computacional de los nodos y el cálculo de carga promedio para cada nodo se muestran en la Tabla 17.

Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas

En esta propuesta las categorías serán las mismas para todos los nodos: Alta (si la carga es mayor al 70%), Media (si la carga está entre el 40% y el 70% inclusive) y Baja (si la carga es menor al 40%).

$$\text{card}(\{\text{categorías}\}) = 3$$

$$\text{categorías} = \{\text{Alta, Media, Baja}\}.$$

Los valores que se obtienen para las categorías de carga en base a los promedios mostrados en la Tabla 17, se indican en la Tabla 18.

Tabla 17. Valores de los criterios de carga computacional (E1)

Nodos	Valores de los Criterios			Promedio
1	80	90	75	81.67
2	30	30	50	36.67
3	77	49	85	70.33
4	6	7	9	07.33
5	5	5	6	05.33
6	7	9	7	07.67
7	7	10	15	10.67
8	9	6	7	07.33
9	10	9	15	11.33
10	8	7	8	07.67

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Tabla 18. Valores de las categorías para medir la carga computacional (E1)

Nodos	Valores de las Categorías
1	Alta
2	Baja
3	Alta
4	Baja
5	Baja
6	Baja
7	Baja
8	Baja
9	Baja
10	Baja

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Para establecer los vectores de pesos asociados a las categorías de carga computacional actual de cada nodo se utilizarán, para todos los nodos y para todas las categorías de carga, los siguientes criterios: N° de procesos en el nodo, % de uso de CPU, % de uso de memoria, % de uso de memoria virtual, prioridad del proceso (prioridad del proceso en el nodo donde se ejecuta), peso del recurso (peso de transferencia que genera sobre el ancho de banda), sobrecarga de memoria (memoria adicional que requerirá disponer el recurso solicitado, si el dato está disponible), sobrecarga de procesador (uso adicional de procesador que requerirá disponer el recurso solicitado, si el dato está disponible), sobrecarga de entrada / salida (entrada / salida adicional que requerirá disponer el recurso solicitado, si el dato está disponible).

$$\text{card}(\{\text{critpref}\}) = 9$$

criterios para preferencias = {N° de procesos en el nodo, % de uso de CPU, % de uso de memoria, % de uso de memoria virtual, prioridad del proceso, peso del recurso, impacto transferencia sobre ancho de banda, sobrecarga de memoria, sobrecarga de procesador, sobrecarga de entrada / salida}.

Seguidamente se deben establecer los valores correspondientes a los criterios constituyendo así los vectores de pesos para las distintas categorías de carga, que serán iguales para todos los nodos, lo cual se indica en la Tabla 19 que se continúa en la **Tabla 20**.

Tabla 19. Pesos asignados a los criterios para calcular la prioridad (N° Procesos, %CPU, %Memoria, %MV, Prioridad Proceso) (E1)

Categorías	N° Procesos	% CPU	Pesos		Prioridad Proceso
			% Memoria	% MV	
Alta	0.05	0.1	0.1	0.3	0.1
Media	0.04	0.08	0.4	0.1	0.03
Baja	0.1	0.2	0.1	0.2	0.1

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Tabla 20. Pesos asignados a los criterios para calcular la prioridad (Peso del Recurso, Impacto Transferencia, Sobrecarga Memoria, Sobrecarga Procesador, Sobrecarga E/S) (E1)

Categorías	Pesos				
	Peso Recurso	Impacto Transfer.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
Alta	0.1	0.05	0.1	0.05	0.05
Media	0.2	0.05	0.05	0.025	0.025
Baja	0.2	0.01	0.025	0.025	0.04

Nota: Pesos asignados a los criterios para calcular la prioridad o preferencia que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo.

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

La sumatoria de los pesos asignados a los distintos criterios es 1 para cada una de las categorías, o lo que es lo mismo, que la suma de elementos del vector de pesos de cada categoría es 1.

Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo

Los vectores de valoraciones se aplican para cada requerimiento de un recurso hecho por un proceso, según los criterios establecidos para la determinación de la prioridad que en cada caso y momento fija el nodo en el cual se produce el requerimiento; cada vector de valoraciones de cada requerimiento se multiplica escalarmente por el vector de pesos correspondiente a la categoría de carga actual del nodo para obtener la prioridad según cada criterio y la prioridad nodal otorgada a cada requerimiento; esto se muestra en la Tabla 21 que se continúa en la Tabla 22.

Tabla 21. Valoraciones asignadas a los criterios para calcular la prioridad (N° proceso, %CPU, %Memoria, %MV, Prioridad Proceso) (E1)

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
$p_{11}r_{11}$	0.7	0.5	0.7	0.9	0.8
$p_{11}r_{12}$	0.8	0.7	0.4	0.5	0.3
$p_{11}r_{21}$	0.3	0.4	0.5	0.2	0.9
$p_{11}r_{22}$	0.5	0.5	0.7	0.4	0.8
$p_{11}r_{23}$	0.5	0.6	0.8	0.8	0.9
$p_{11}r_{24}$	0.3	0.5	0.9	0.2	0.6
$p_{12}r_{11}$	0.4	0.7	0.5	0.9	1.0

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
$p_{12}r_{12}$	0.2	0.7	0.3	0.7	0.8
$p_{12}r_{21}$	0.7	0.4	0.3	0.7	0.8
$p_{12}r_{22}$	0.9	0.6	0.7	0.7	0.8
$p_{12}r_{31}$	0.2	0.5	0.7	0.7	0.3
$p_{12}r_{33}$	0.4	0.5	0.7	0.9	0.3
$p_{13}r_{11}$	0.5	0.7	0.7	0.8	0.6
$p_{13}r_{12}$	0.7	0.8	0.7	0.4	0.9
$p_{13}r_{13}$	0.7	0.6	0.7	0.8	0.9
$p_{13}r_{21}$	0.7	0.4	0.9	0.3	0.5
$p_{13}r_{22}$	0.5	0.9	0.8	0.3	0.5
$p_{13}r_{31}$	0.5	0.7	0.3	0.6	0.8
$p_{13}r_{32}$	0.6	0.9	0.3	0.6	0.4
$p_{13}r_{33}$	0.6	0.2	0.4	0.6	0.9
$p_{14}r_{11}$	0.5	0.7	0.6	0.7	0.7
$p_{14}r_{12}$	0.6	0.7	0.6	0.0	0.8
$p_{14}r_{13}$	0.6	0.5	0.6	3.0	0.8
$p_{14}r_{21}$	0.6	0.3	0.8	0.3	0.4
$p_{14}r_{22}$	0.4	0.8	0.7	0.3	0.4
$p_{14}r_{41}$	0.4	0.6	0.4	0.5	0.7
$p_{14}r_{42}$	0.6	0.8	0.2	0.5	0.8
$p_{14}r_{43}$	0.5	0.3	0.3	0.7	0.7
$p_{15}r_{21}$	0.6	0.8	0.8	0.9	0.7
$p_{15}r_{32}$	0.8	0.9	0.8	0.5	0.8
$p_{15}r_{33}$	0.7	0.3	0.5	0.7	0.6
$p_{15}r_{51}$	0.8	0.4	0.8	0.4	0.6
$p_{15}r_{52}$	0.6	0.8	0.9	0.4	0.6
$p_{15}r_{53}$	0.8	0.7	0.8	0.9	0.8
$p_{15}r_{61}$	0.6	0.8	0.4	0.7	0.8
$p_{15}r_{62}$	0.7	0.8	0.4	0.7	0.5
$p_{16}r_{11}$	0.4	0.7	0.5	0.9	1.0
$p_{16}r_{12}$	0.2	0.7	0.3	0.7	0.8
$p_{16}r_{13}$	0.3	0.6	0.3	0.6	0.7
$p_{16}r_{14}$	0.4	0.7	0.2	0.5	0.6
$p_{16}r_{21}$	0.7	0.4	0.3	0.7	0.8
$p_{16}r_{22}$	0.9	0.6	0.7	0.7	0.8
$p_{16}r_{31}$	0.2	0.5	0.7	0.7	0.3
$p_{16}r_{33}$	0.4	0.5	0.7	0.9	0.3
$p_{17}r_{11}$	0.3	0.6	0.5	0.8	0.6
$p_{17}r_{12}$	0.4	0.6	0.4	0.6	0.8
$p_{17}r_{13}$	0.5	0.7	0.6	0.9	0.7
$p_{21}r_{12}$	0.2	0.1	0.3	0.8	0.7
$p_{21}r_{13}$	0.2	0.4	0.8	0.9	0.7
$p_{21}r_{22}$	0.3	0.5	0.8	0.9	0.5

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
$p_{21}r_{23}$	0.7	0.5	0.6	0.8	0.5
$p_{21}r_{31}$	0.7	0.5	0.8	0.6	0.9
$p_{21}r_{33}$	0.7	0.4	0.9	0.8	0.4
$p_{22}r_{21}$	0.5	0.4	0.7	0.8	0.6
$p_{22}r_{22}$	0.5	0.9	0.6	0.8	0.9
$p_{22}r_{31}$	0.5	0.4	0.5	0.2	0.4
$p_{22}r_{33}$	0.8	0.4	0.3	0.2	0.8
$p_{23}r_{12}$	0.5	0.6	0.8	0.3	0.5
$p_{23}r_{24}$	0.6	0.2	0.1	0.7	0.3
$p_{23}r_{31}$	0.3	0.1	0.4	0.8	0.7
$p_{23}r_{32}$	0.4	0.4	0.8	0.2	0.4
$p_{23}r_{33}$	0.3	0.6	0.8	0.2	0.9
$p_{24}r_{11}$	0.4	0.6	0.7	0.9	0.5
$p_{24}r_{12}$	0.3	0.6	0.7	0.8	0.9
$p_{24}r_{23}$	0.4	0.9	0.4	0.8	0.8
$p_{24}r_{24}$	0.5	0.8	0.7	0.9	0.3
$p_{25}r_{21}$	0.2	0.8	0.8	0.9	0.4
$p_{31}r_{13}$	0.6	0.9	0.6	0.9	0.7
$p_{31}r_{71}$	0.8	0.3	0.9	0.5	0.7
$p_{31}r_{72}$	0.4	0.7	0.7	0.5	0.9
$p_{32}r_{11}$	0.6	0.9	0.8	0.5	0.9
$p_{32}r_{12}$	0.7	0.4	0.6	0.9	0.8
$p_{32}r_{13}$	0.8	0.9	1.0	0.7	0.9
$p_{32}r_{21}$	0.8	0.8	1.0	0.9	0.6
$p_{33}r_{11}$	0.2	0.7	0.9	0.8	0.6
$p_{33}r_{12}$	0.9	1.0	0.9	0.7	0.3
$p_{33}r_{13}$	0.9	0.5	0.7	0.9	0.3
$p_{33}r_{21}$	0.4	0.6	0.7	0.8	0.8
$p_{33}r_{22}$	0.9	0.4	0.7	0.8	0.7
$p_{33}r_{23}$	0.4	0.6	0.9	1.0	0.6
$p_{33}r_{32}$	0.6	0.6	0.7	0.8	0.4
$p_{33}r_{33}$	0.6	1.0	0.7	0.4	0.6
$p_{34}r_{12}$	0.8	1.0	0.3	0.5	0.7
$p_{34}r_{13}$	0.2	1.0	0.8	0.5	0.8
$p_{34}r_{22}$	0.9	0.8	0.6	0.8	0.9
$p_{34}r_{23}$	0.4	0.6	0.7	0.8	0.9
$p_{34}r_{24}$	0.9	0.4	0.7	0.4	0.7
$p_{34}r_{31}$	0.5	0.6	0.7	0.9	0.7
$p_{34}r_{32}$	0.8	0.6	0.9	0.5	0.6
$p_{34}r_{33}$	0.4	0.6	0.8	0.5	0.9
$p_{35}r_{31}$	0.9	0.8	0.7	0.4	0.8
$p_{35}r_{32}$	0.9	0.7	0.8	0.6	0.4
$p_{35}r_{33}$	0.5	0.7	0.6	0.9	0.8

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
<i>p₃₅r₆₁</i>	0.2	0.4	0.7	0.4	0.9
<i>p₃₅r₆₂</i>	0.8	0.9	0.7	0.3	0.9
<i>p₃₅r₇₂</i>	0.3	0.8	0.9	0.4	0.8
<i>p₃₅r₇₄</i>	0.5	0.8	0.9	0.4	0.6
<i>p₃₆r₁₁</i>	0.8	0.9	0.6	0.5	0.8
<i>p₃₆r₁₂</i>	0.7	0.9	0.4	0.9	0.8
<i>p₃₆r₁₃</i>	0.9	0.9	0.8	0.7	0.8
<i>p₃₆r₂₁</i>	0.8	0.9	0.5	0.3	0.7
<i>p₃₆r₂₂</i>	0.8	0.2	0.1	0.3	0.8
<i>p₃₆r₂₄</i>	0.4	0.7	0.9	0.3	0.8
<i>p₃₆r₈₁</i>	0.5	0.9	0.9	0.7	0.8
<i>p₃₆r₈₂</i>	0.9	0.8	0.6	0.7	0.8
<i>p₃₆r₈₃</i>	0.2	0.7	0.4	0.8	0.8
<i>p₃₇r₁₁</i>	0.9	0.6	0.8	0.6	0.9
<i>p₃₇r₁₂</i>	0.7	0.9	0.8	0.7	0.5
<i>p₃₇r₃₁</i>	0.9	0.7	0.8	0.6	0.6
<i>p₃₇r₄₂</i>	0.8	0.9	0.5	0.3	0.8
<i>p₃₇r₄₃</i>	0.8	0.4	0.6	0.8	0.8

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Tabla 22. Valoraciones asignadas a los criterios para calcular la prioridad (Peso del Recurso, Impacto Transferencia, Sobrecarga Memoria, Sobrecarga Procesador, Sobrecarga E/S) (E1)

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
<i>p₁₁r₁₁</i>	0.3	0.5	0.2	0.3	0.4
<i>p₁₁r₁₂</i>	0.3	0.5	0.7	0.2	0.4
<i>p₁₁r₂₁</i>	0.8	0.6	0.2	0.5	0.7
<i>p₁₁r₂₂</i>	0.4	0.3	0.3	0.5	0.6
<i>p₁₁r₂₃</i>	0.7	0.5	0.9	0.7	0.6
<i>p₁₁r₂₄</i>	0.2	0.2	0.6	0.7	0.4
<i>p₁₂r₁₁</i>	0.7	0.6	0.9	0.8	0.8
<i>p₁₂r₁₂</i>	0.3	0.2	0.3	0.8	0.9
<i>p₁₂r₂₁</i>	0.3	0.3	0.9	0.5	0.2
<i>p₁₂r₂₂</i>	0.9	0.7	0.2	0.5	0.4
<i>p₁₂r₃₁</i>	0.8	0.3	0.2	0.7	0.8
<i>p₁₂r₃₃</i>	0.5	0.9	0.4	0.5	0.8
<i>p₁₃r₁₁</i>	0.5	0.3	0.5	0.7	0.8
<i>p₁₃r₁₂</i>	0.8	0.9	0.2	0.9	0.7
<i>p₁₃r₁₃</i>	0.8	0.9	0.4	0.8	0.7
<i>p₁₃r₂₁</i>	0.4	0.9	0.7	0.2	0.3
<i>p₁₃r₂₂</i>	0.6	0.4	0.4	0.9	0.3
<i>p₁₃r₃₁</i>	0.6	0.4	0.9	0.9	0.5
<i>p₁₃r₃₂</i>	0.6	0.6	0.8	0.7	0.8

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
$p_{13}r_{33}$	0.8	0.7	0.8	0.5	0.8
$p_{14}r_{11}$	0.6	0.2	0.4	0.6	0.7
$p_{14}r_{12}$	0.7	0.8	0.2	0.8	0.6
$p_{14}r_{13}$	0.9	0.8	0.3	0.9	0.6
$p_{14}r_{21}$	0.5	0.8	0.8	0.3	0.2
$p_{14}r_{22}$	0.7	0.5	0.5	0.8	0.2
$p_{14}r_{41}$	0.5	0.3	0.8	0.8	0.6
$p_{14}r_{42}$	0.7	0.6	0.9	0.6	0.7
$p_{14}r_{43}$	0.7	0.5	0.9	0.6	0.7
$p_{15}r_{21}$	0.6	0.4	0.6	0.8	0.7
$p_{15}r_{32}$	0.9	0.8	0.3	0.8	0.8
$p_{15}r_{33}$	0.9	0.8	0.9	0.6	0.9
$p_{15}r_{53}$	0.9	0.8	0.5	0.9	0.8
$p_{15}r_{51}$	0.5	0.8	0.8	0.3	0.4
$p_{15}r_{52}$	0.7	0.5	0.5	0.8	0.4
$p_{15}r_{61}$	0.7	0.5	0.8	0.8	0.6
$p_{15}r_{62}$	0.7	0.7	0.9	0.8	0.9
$p_{16}r_{11}$	1.0	0.6	0.8	0.7	0.7
$p_{16}r_{12}$	1.0	0.2	0.4	0.7	0.8
$p_{16}r_{13}$	0.7	0.3	0.3	0.5	0.4
$p_{16}r_{14}$	0.8	0.4	0.5	0.6	0.7
$p_{16}r_{21}$	0.3	0.3	0.9	0.5	0.2
$p_{16}r_{22}$	0.9	0.7	0.2	0.5	0.4
$p_{16}r_{31}$	0.8	0.3	0.2	0.7	0.8
$p_{16}r_{33}$	0.5	0.9	0.4	0.5	0.8
$p_{17}r_{11}$	0.3	0.5	0.8	0.9	0.9
$p_{17}r_{12}$	0.4	0.3	0.4	0.7	0.8
$p_{17}r_{13}$	0.2	0.3	0.3	0.8	0.6
$p_{21}r_{12}$	0.8	0.7	0.7	0.5	0.8
$p_{21}r_{13}$	0.4	0.3	0.4	0.5	0.2
$p_{21}r_{22}$	0.6	0.2	0.6	0.2	0.4
$p_{21}r_{23}$	0.4	0.8	0.2	0.9	0.4
$p_{21}r_{31}$	0.2	0.8	0.4	0.9	0.9
$p_{21}r_{33}$	0.7	0.7	0.8	0.6	0.6
$p_{22}r_{21}$	0.8	0.8	0.4	0.6	0.9
$p_{22}r_{22}$	0.3	0.6	0.4	0.2	0.1
$p_{22}r_{31}$	0.5	0.7	0.8	0.2	0.9
$p_{22}r_{33}$	0.2	0.7	0.9	0.5	0.8
$p_{23}r_{12}$	0.2	0.7	0.7	0.5	0.5
$p_{23}r_{24}$	0.8	0.7	0.8	0.9	0.4
$p_{23}r_{31}$	0.5	0.3	0.9	0.4	0.6

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
$p_{23}r_{32}$	0.5	0.6	0.6	0.6	0.6
$p_{23}r_{33}$	0.6	0.8	0.6	0.4	0.2
$p_{24}r_{11}$	0.7	0.9	0.7	0.9	0.5
$p_{24}r_{12}$	0.4	0.5	0.7	0.6	0.9
$p_{24}r_{23}$	0.4	0.8	0.7	0.6	0.3
$p_{24}r_{24}$	0.6	0.4	0.4	0.8	0.7
$p_{25}r_{21}$	0.9	0.7	0.5	0.6	0.8
$p_{31}r_{13}$	0.7	0.8	0.4	0.9	0.8
$p_{31}r_{71}$	0.6	0.9	0.3	0.9	0.7
$p_{31}r_{72}$	0.9	0.5	0.9	0.7	0.7
$p_{32}r_{11}$	0.2	0.6	0.7	0.3	0.7
$p_{32}r_{12}$	0.4	0.3	1.0	0.9	0.7
$p_{32}r_{13}$	0.3	0.5	0.3	0.5	0.9
$p_{32}r_{21}$	0.6	0.6	0.8	0.4	0.9
$p_{33}r_{11}$	0.4	0.2	0.9	1.0	0.3
$p_{33}r_{12}$	0.8	0.5	0.5	0.7	0.3
$p_{33}r_{13}$	0.7	0.5	0.4	0.5	0.8
$p_{33}r_{21}$	0.4	0.9	0.4	0.8	0.8
$p_{33}r_{22}$	0.9	0.5	0.4	0.9	0.7
$p_{33}r_{23}$	0.3	0.8	0.4	0.7	0.8
$p_{33}r_{32}$	0.3	0.2	0.5	0.8	0.8
$p_{33}r_{33}$	0.2	0.2	0.8	0.8	0.9
$p_{34}r_{12}$	0.5	0.6	0.3	0.8	0.7
$p_{34}r_{13}$	0.7	0.7	0.6	0.9	0.7
$p_{34}r_{22}$	0.5	0.6	0.8	0.5	0.6
$p_{34}r_{23}$	0.3	0.3	0.5	0.9	0.6
$p_{34}r_{24}$	0.4	0.6	0.5	0.9	0.7
$p_{34}r_{31}$	0.4	0.5	0.9	0.6	0.7
$p_{34}r_{32}$	0.4	0.2	0.9	0.3	0.9
$p_{34}r_{33}$	0.7	0.8	0.9	0.4	0.3
$p_{35}r_{31}$	0.2	0.5	0.3	0.4	0.8
$p_{35}r_{32}$	0.7	0.7	0.9	0.4	0.7
$p_{35}r_{33}$	0.6	0.7	0.5	0.9	0.7
$p_{35}r_{61}$	0.3	0.7	0.5	0.7	0.9
$p_{35}r_{62}$	0.9	0.9	0.8	0.7	0.4
$p_{35}r_{72}$	0.9	0.4	0.6	0.3	0.6
$p_{35}r_{74}$	0.9	0.5	0.9	0.4	0.7
$p_{36}r_{11}$	0.7	0.6	0.9	0.9	0.6
$p_{36}r_{12}$	0.8	0.9	0.9	0.4	0.7
$p_{36}r_{13}$	0.3	0.5	0.7	0.9	0.7
$p_{36}r_{21}$	0.5	0.4	0.7	0.9	0.9
$p_{36}r_{22}$	0.5	0.7	0.7	0.6	0.9

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
p_{36r24}	0.6	0.2	0.5	0.8	0.9
p_{36r81}	0.7	0.7	0.8	0.8	0.7
p_{36r82}	0.3	0.6	0.5	0.6	0.7
p_{36r83}	0.2	0.4	0.9	0.4	0.5
p_{37r11}	0.4	0.9	0.7	0.9	0.8
p_{37r12}	0.7	0.5	0.8	0.8	0.6
p_{37r31}	0.8	0.5	0.9	0.5	0.4
p_{37r42}	0.5	0.9	0.7	0.4	0.6
p_{37r43}	0.8	0.3	0.6	0.9	0.3

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

2.5.2. Determinación de procesos migrables mediante el cálculo de prioridad

promedio de los procesos (E1)

Por cada solicitud de recursos de cada proceso, se establece un criterio de prioridad, que es asignado por el Runtime al momento de hacer la solicitud. La sumatoria de este valor de prioridad, para cada recurso solicitado, dividido el total de solicitudes de cada proceso, indicará la prioridad promedio del proceso. Los nodos con carga alta son el 1 y 3, y sus procesos se evalúan según la siguiente fórmula:

$$\text{promedio de prioridad de proceso } (p_{ij}) = (\sum cp_{ijk} / o) \quad (12)$$

Nodo 1

$$\text{promedio de prioridad de proceso } (p_{11}) = (0.8+0.3+0.9+0.8+0.95+ 0.6) /6 = \mathbf{0.725}$$

$$\text{promedio de prioridad de proceso } (p_{12}) = (1+0.8+0.8+0.8+0.3+0.3) /6 = \mathbf{0.667}$$

$$\text{promedio de prioridad de proceso } (p_{13}) = (0.6+0.9+0.9+0.5+0.5+ 0.8+0.4+0.9) /8 = \mathbf{0.687}$$

$$\text{promedio de prioridad de proceso } (p_{14}) = (0.7+0.8+0.8+0.4+0.4+ 0.7+0.8+0.7) /8 = \mathbf{0.662}$$

$$\text{promedio de prioridad de proceso } (p_{15}) = (0.7+0.8+0.6+0.8+0.6+ 0.6+0.8+0.5) /8 = \mathbf{0.675}$$

Nodo 3

promedio de prioridad de proceso (p_{31}) = $(0.7+0.7+0.9) / 3 = \mathbf{0.767}$

promedio de prioridad de proceso (p_{32}) = $(0.9+0.8+0.9+0.6) / 4 = \mathbf{0.800}$

promedio de prioridad de proceso (p_{33}) = $(0.6+0.3+0.3+0.8+0.7+ 0.6+0.4+0.6) / 5 = 0.538$

promedio de prioridad de proceso (p_{34}) = $(0.7+0.8+0.9+0.9+0.7+ 0.7+0.6+0.9) / 8 = \mathbf{0.775}$

promedio de prioridad de proceso (p_{35}) = $(0.9+0.9+0.8+0.6+0.8+ 0.4+0.8) / 7 = \mathbf{0.743}$

promedio de prioridad de proceso (p_{36}) = $(0.8+0.8+0.8+0.7+0.8+ 0.8+0.8+0.8+0.8) / 9 = \mathbf{0.789}$

promedio de prioridad de proceso (p_{37}) = $(0.9+0.5+0.6+0.8+0.8) / 5 = \mathbf{0.720}$

El proceso p_{33} no se evalúa en este ciclo porque para el modelo propuesto sólo se consideran aquellos procesos cuyo promedio de prioridad de proceso supere el límite establecido, para este caso 0.6. Los procesos que clasifican para evaluación para posible migración a otro nodo se listan en la Tabla 23.

Tabla 23. Lista de procesos con prioridad de procesos superior al límite (E1)

Nodos	Procesos	Promedio de prioridad de proceso
1	p_{11}	0.725
1	p_{12}	0.667
1	p_{13}	0.687
1	p_{14}	0.662
1	p_{15}	0.675
3	p_{31}	0.767
3	p_{32}	0.800
3	p_{34}	0.775
3	p_{35}	0.743
3	p_{36}	0.789
3	p_{37}	0.720

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

2.5.3. Selección de los posibles nodos destino mediante la evaluación de su carga (E1)

En esta etapa se evalúa si existen nodos con carga baja que puedan alojar procesos de otros nodos. De la Tabla 18 que contiene los valores de la carga computacional, se filtran solamente los nodos que se encuentran dentro de la categoría de carga baja.

$$\text{nodo destino con carga baja} = \{2, 4, 5, 6, 7, 8, 9, 10\}$$

Para cada proceso puede haber varios nodos con carga baja disponible. Teniendo en cuenta el conjunto de nodos con carga baja se verifican los recursos solicitados por los procesos p_{ij} para determinar el nodo candidato para una posible migración.

2.5.4. Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E1)

Una vez identificados los nodos con carga alta, sus procesos y los posibles nodos destinos, se realiza la comparación de los recursos locales en relación con los remotos, con el fin de elaborar la lista preliminar de los procesos migrables y sus posibles nodos destinos.

En la Fig. 7 se muestra el vector de migración de procesos donde se almacenan valores del peso de migración que generaría el proceso si se migrara a un determinado nodo destino, habrá un vector de migración de procesos para cada proceso en relación con el nodo destino evaluado.

Para cada nodo con carga alta se realiza la evaluación de sus procesos para las diferentes situaciones que pueden presentarse según la Fig. 6 y la Tabla 22.

En la Tabla 24 se muestra el vector de migración de proceso para el proceso p_{11} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 0.6 y como peso de

recursos remotos en el nodo 2 es de 2.1. Analizando la Fig. 6 en relación con el nodo 2 su valor se graba dada la situación (u), y en relación con los nodos 4 al 10 su valor se graba dada la situación (q).

Tabla 24. Vector de migración del proceso p_{11} (E1)

vmp¹			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	1	2	-	0.6	-	-
1	1	4	2.7	-	-	-
1	1	5	2.7	-	-	-
1	1	6	2.7	-	-	-
1	1	7	2.7	-	-	-
1	1	8	2.7	-	-	-
1	1	9	2.7	-	-	-
1	1	10	2.7	-	-	-

Fuente: Elaboración propia

En la Tabla 25 se muestra el vector de migración de proceso para el proceso p_{12} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 1.0, el peso de recursos remotos está distribuido en dos nodos, en el nodo 2 es de 1.2 y en el nodo 3 es de 1.3. Analizando la Fig. 6 en relación con el nodo 2 su valor se graba dada la situación (u). El proceso p_{12} podría migrar únicamente al nodo 2 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 25. Vector de migración del proceso p_{12} (E1)

vmp²			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	2	2	-	2.3	-	-
1	2	4	-	-	-	-
1	2	5	-	-	-	-
1	2	6	-	-	-	-
1	2	7	-	-	-	-
1	2	8	-	-	-	-
1	2	9	-	-	-	-
1	2	10	-	-	-	-

Fuente: Elaboración propia

En la Tabla 26 se muestra el vector de migración de proceso para el proceso p_{13} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 2.1, el peso de recursos remotos está distribuido en dos nodos, en el nodo 2 es de 1.0 y en el nodo 3 es de 2.0. Analizando la Fig. 6 en relación con el nodo 2 su valor se graba dada la situación (u). El proceso p_{13} podría migrar únicamente al nodo 2 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 26. Vector de migración del proceso p_{13} (E1)

q	vmp³		Opciones			
	s	u	(q)	(u)	(v)	(w)
1	3	2	-	4.1	-	-
1	3	4	-	-	-	-
1	3	5	-	-	-	-
1	3	6	-	-	-	-
1	3	7	-	-	-	-
1	3	8	-	-	-	-
1	3	9	-	-	-	-
1	3	10	-	-	-	-

Fuente: Elaboración propia

En la Tabla 27 se muestra el vector de migración de proceso para el proceso p_{14} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 2.2, el peso de recursos remotos está distribuido en dos nodos, en el nodo 2 es de 1.2 y en el nodo 4 es de 1.9. Analizando la Fig. 6 en relación con los nodos 2 y 4 su valor se graba dada la situación (u). El proceso p_{14} podría migrar únicamente al nodo 2 o nodo 4 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

En la Tabla 28 se muestra el vector de migración de proceso para el proceso p_{15} que tiene los siguientes pesos de recursos asociados: el peso de recursos remotos está distribuido en cuatro nodos, en el nodo 2 es de 0.6, en el nodo 3 es de 1.8, en el nodo 5 es de 2.1 y en el nodo 6 es de 1.4. Analizando la Fig. 6 se observan dos grupos de nodos posibles, el primer grupo integrados por los

nodos 2, 5, 7, 8, 9 y 10, su valor se graba dada la situación (v), y el segundo grupo integrados por los nodos 4 y 6, su valor se graba dada la situación (q).

Tabla 27. Vector de migración del proceso p_{14} (E1)

vmp⁴			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	4	2	-	4.1	-	-
1	4	4	-	3.4	-	-
1	4	5	-	-	-	-
1	4	6	-	-	-	-
1	4	7	-	-	-	-
1	4	8	-	-	-	-
1	4	9	-	-	-	-
1	4	10	-	-	-	-

Fuente: Elaboración propia

Tabla 28. Vector de migración del proceso p_{15} (E1)

vmp⁵			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	5	2	-	-	5.3	-
1	5	4	5.9	-	-	-
1	5	5	-	-	3.8	-
1	5	6	4.5	-	-	-
1	5	7	-	-	5.9	-
1	5	8	-	-	5.9	-
1	5	9	-	-	5.9	-
1	5	10	-	-	5.9	-

Fuente: Elaboración propia

El proceso p_{16} se descarta de la posibilidad de migración porque no cumple las condiciones para migrarse a otros nodos ya que hacerlo generaría un alto impacto en el ancho de banda.

En la Tabla 29 se muestra el vector de migración de proceso para el proceso p_{17} , el peso de recursos locales de 0.9, no posee recursos remotos y dado que su peso no supera el límite establecido en la situación (p) de la Fig. 6, cualquiera de los nodos con carga baja del sistema distribuido es candidato para una posible migración y su valor se graba dada la situación (q).

Tabla 29. Vector de migración del proceso p_{17} (E1)

vmp⁷			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	7	2	0.9	-	-	-
1	7	4	0.9	-	-	-
1	7	5	0.9	-	-	-
1	7	6	0.9	-	-	-
1	7	7	0.9	-	-	-
1	7	8	0.9	-	-	-
1	7	9	0.9	-	-	-
1	7	10	0.9	-	-	-

Fuente: Elaboración propia

En la Tabla 30 se muestra el vector de migración de proceso correspondiente al proceso p_{31} del nodo 3 que es el siguiente con carga alta.

Respecto del proceso p_{31} , su peso de recursos remotos está distribuido en dos nodos, en el nodo 1 es de 0.7 y en el nodo 7 es de 1.5. Analizando la Fig. 6 y teniendo en cuenta que el proceso p_{31} no posee recursos locales, en relación con los nodos 2, 4, 5, 6, 8, 9, 10, su valor se graba dada la situación (w) y en relación con el nodo 7 su valor se graba dada la situación (v).

Tabla 30. Vector de migración del proceso p_{31} (E1)

vmp¹			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	1	2	-	-	-	2.2
3	1	4	-	-	-	2.2
3	1	5	-	-	-	2.2
3	1	6	-	-	-	2.2
3	1	7	-	-	0.7	-
3	1	8	-	-	-	2.2
3	1	9	-	-	-	2.2
3	1	10	-	-	-	2.2

Fuente: Elaboración propia

En la Tabla 31 se muestra el vector de migración de proceso para el proceso p_{32} que tiene los siguientes pesos de recursos asociados: el peso de recursos remotos está distribuido en dos nodos, en el nodo 1 es de 0.9 y en el nodo 2 es de 0.6. Analizando la Fig. 6 y teniendo en cuenta que el

proceso p_{32} no posee recursos locales, en relación con los nodos 4, 5, 6, 7, 8, 9, 10 su valor se graba dada la situación (w) y en relación con el nodo 2 el valor se graba dada la situación (v).

Tabla 31. Vector de migración del proceso p_{32} (E1)

vmp²			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	2	2	-	-	0.9	-
3	2	4	-	-	-	1.5
3	2	5	-	-	-	1.5
3	2	6	-	-	-	1.5
3	2	7	-	-	-	1.5
3	2	8	-	-	-	1.5
3	2	9	-	-	-	1.5
3	2	10	-	-	-	1.5

Fuente: Elaboración propia

En la Tabla 32 se muestra el vector de migración de proceso para el proceso p_{34} que tiene los siguientes pesos de recursos asociados, el peso de recursos locales es de 1.5, el peso de recursos remotos está distribuido en dos nodos, en el nodo 1 es de 1.2 y en el nodo 2 es de 1.2. Analizando la Fig. 6 en relación con el nodo 2 su valor se graba dada la situación (u). El proceso p_{34} podría migrar únicamente al nodo 2 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 32. Vector de migración del proceso p_{34} (E1)

vmp⁴			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	4	2	-	2.7	-	-
3	4	4	-	-	-	-
3	4	5	-	-	-	-
3	4	6	-	-	-	-
3	4	7	-	-	-	-
3	4	8	-	-	-	-
3	4	9	-	-	-	-
3	4	10	-	-	-	-

Fuente: Elaboración propia

En la Tabla 33 se muestra el vector de migración de proceso para el proceso p_{35} que tiene los siguientes pesos de recursos asociados: el peso de los recursos locales es de 1.5, el peso de recursos remotos está distribuido en dos nodos, en el nodo 6 es de 1.2 y en el nodo 7 es de 1.8. Analizando la Fig. 6 en relación con el nodo 6 y nodo 7, su valor se graba dada la situación (u). El proceso p_{35} podría migrar únicamente al nodo 6 o nodo 7 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 33. Vector de migración del proceso p_{35} (E1)

vmp⁵			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	5	2	-	-	-	-
3	5	4	-	-	-	-
3	5	5	-	-	-	-
3	5	6	-	3.3	-	-
3	5	7	-	2.7	-	-
3	5	8	-	-	-	-
3	5	9	-	-	-	-
3	5	10	-	-	-	-

Fuente: Elaboración propia

En la Tabla 34 se muestra el vector de migración de proceso para proceso p_{36} que tiene los siguientes pesos de recursos asociados: el peso de recursos remotos está distribuido en tres nodos, en el nodo 1 es de 1.8, en el nodo 2 es de 1.6 y en el nodo 8 es de 1.8. Analizando la Fig. 6 se observan dos grupos de nodos posibles, el primer grupo integrado por los nodos 2 y 8, su valor se graba mediante la situación (v), y el segundo grupo integrados por los nodos 4, 5, 6, 7, 9 y 10, su valor se graba dada la situación (w).

En la Tabla 35 se muestra el vector de migración de proceso para el proceso p_{37} que tiene los siguientes pesos de recursos asociados, el peso de recursos locales es de 0.8, el peso de recursos remotos está distribuido en dos nodos, en el nodo 1 es de 1.1 y en el nodo 4 es de 1.3. Analizando

la Fig. 6 en relación con los nodos 2, 5, 6, 7, 8, 9 y 10, su valor se graba dada la situación (q), y en relación con el nodo 4, su valor se graba dada la situación (u).

Tabla 34. Vector de migración del proceso p_{36} (E1)

q	vmp⁶		Opciones			
	s	u	(q)	(u)	(v)	(w)
3	6	2	-	-	3	-
3	6	4	-	-	-	4.6
3	6	5	-	-	-	4.6
3	6	6	-	-	-	4.6
3	6	7	-	-	-	4.6
3	6	8	-	-	3.4	-
3	6	9	-	-	-	4.6
3	6	10	-	-	-	4.6

Fuente: Elaboración propia

Tabla 35. Vector de migración del proceso p_{37} (E1)

q	vmp⁷		Opciones			
	s	u	(q)	(u)	(v)	(w)
3	7	2	3.2	-	-	-
3	7	4	-	1.9	-	-
3	7	5	3.2	-	-	-
3	7	6	3.2	-	-	-
3	7	7	3.2	-	-	-
3	7	8	3.2	-	-	-
3	7	9	3.2	-	-	-
3	7	10	3.2	-	-	-

Fuente: Elaboración propia

Una vez evaluadas todas las situaciones de cada proceso en relación con cada nodo destino posible, se genera la matriz general de procesos (MGP) que se observa en la Tabla 36.

Tabla 36. Matriz general de procesos (MGP) (E1)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{11}	0.6	2.7	2.7	2.7	2.7	2.7	2.7	2.7
p_{12}	2.3	-	-	-	-	-	-	-
p_{13}	4.1	-	-	-	-	-	-	-
p_{14}	4.1	3.4	-	-	-	-	-	-
p_{15}	5.3	5.9	3.8	4.5	5.9	5.9	5.9	5.9

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{17}	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
p_{31}	2.2	2.2	2.2	2.2	0.7	2.2	2.2	2.2
p_{32}	0.9	1.5	1.5	1.5	1.5	1.5	1.5	1.5
p_{34}	2.7	-	-	-	-	-	-	-
p_{35}	-	-	-	3.3	2.7	-	-	-
p_{36}	3	4.6	4.6	4.6	4.6	3.4	4.6	4.6
p_{37}	3.2	1.9	3.2	3.2	3.2	3.2	3.2	3.2

Fuente: Elaboración propia

2.5.5. Determinación del orden de migración de procesos y de los nodos destino correspondientes (E1)

En la matriz general de procesos se almacenan datos de procesos y los pesos de transferencia (migración) que se generarían al ser migrado cada proceso. Las celdas verdes indican que el proceso tiene asignado algún recurso en ese nodo, las celdas en rosa que no lo tiene.

El primer paso que habrá que realizar a partir de los datos de la matriz general de procesos es dividirla en dos matrices, una primaria (ver Tabla 37) y otra secundaria (ver Tabla 38). La diferencia radica en que primero se intenta favorecer a los procesos que tienen recursos en los posibles nodos destino, es decir, cuya migración implicaría una reducción en el impacto de transferencia (consumo de ancho de banda) con respecto a aquellos que no tengan recursos en nodos destinos. La segunda matriz se analizará luego de concluir las evaluaciones de solicitudes de la primera matriz.

Para el modelo propuesto, cada nodo con una carga alta podrá migrar hasta el 40% de sus procesos, con el fin de evitar migrar todos sus procesos y sobrecargar otros nodos.

Tabla 37. Matriz primaria (MP). Primera ronda (E1)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{11}	0.6	2.7	2.7	2.7	2.7	2.7	2.7	2.7
p_{12}	2.3	-	-	-	-	-	-	-

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{13}	4.1	-	-	-	-	-	-	-
p_{14}	4.1	3.4	-	-	-	-	-	-
p_{15}	5.3	5.9	3.8	4.5	5.9	5.9	5.9	5.9
p_{31}	2.2	2.2	2.2	2.2	0.7	2.2	2.2	2.2
p_{32}	0.9	1.5	1.5	1.5	1.5	1.5	1.5	1.5
p_{34}	2.7	-	-	-	-	-	-	-
p_{35}	-	-	-	3.3	2.7	-	-	-
p_{36}	3	4.6	4.6	4.6	4.6	3.4	4.6	4.6
p_{37}	3.2	1.9	3.2	3.2	3.2	3.2	3.2	3.2

Fuente: Elaboración propia

Tabla 38. Matriz secundaria (MS). Primera ronda (E1)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{17}	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9

Fuente: Elaboración propia

El orden de asignación de los procesos a nodos destino de la matriz primaria se indica mediante la Función de Asignación para la Migración en Sistemas Distribuidos (FAMSD) primaria.

El primer proceso asignado es aquel que tiene menor peso de transferencia. En este caso el proceso p_{11} es el que implica menor peso de transferencia, que corresponde al nodo 2.

Lo anterior define la primera ronda de asignación (Ver Tabla 39); se continua la evaluación mientras existan procesos a evaluar.

Tabla 39. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Primera ronda (E1)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.6	p_{11}	2

Fuente: Elaboración propia

El siguiente paso es eliminar el proceso p_{11} y el nodo 2 de la matriz primaria porque fueron asignados en la primera ronda. Aquellos procesos que tenían recursos destino en el nodo 2 (ya

eliminado) pero tienen como opciones otros nodos destinos (que no poseen recursos destino de ese proceso) pasan a formar parte de la matriz secundaria (Ver Tabla 40 y Tabla 41).

Tabla 40. Matriz primaria (MP). Segunda ronda (E1)

Procesos	Nodos						
	4	5	6	7	8	9	10
p_{14}	3.4	-	-	-	-	-	-
p_{15}	5.9	3.8	4.5	5.9	5.9	5.9	5.9
p_{31}	2.2	2.2	2.2	0.7	2.2	2.2	2.2
p_{35}	-	-	3.3	2.7	-	-	-
p_{36}	4.6	4.6	4.6	4.6	3.4	4.6	4.6
p_{37}	1.9	3.2	3.2	3.2	3.2	3.2	3.2

Fuente: Elaboración propia

Los demás procesos cuya única opción es el nodo 2 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 42).

Tabla 41. Matriz secundaria (MS). Segunda ronda (E1)

Procesos	Nodos						
	4	5	6	7	8	9	10
p_{17}	0.9	0.9	0.9	0.9	0.9	0.9	0.9
p_{32}	1.5	1.5	1.5	1.5	1.5	1.5	1.5

Fuente: Elaboración propia

Tabla 42. Matriz de procesos no migrables. Segunda ronda (E1)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	2.3	-	-	-	-	-	-	-
p_{13}	4.1	-	-	-	-	-	-	-
p_{34}	2.7	-	-	-	-	-	-	-

Fuente: Elaboración propia

Se continúa con la matriz primaria hasta agotar opciones de procesos mientras los nodos no alcancen el porcentaje de migración fijado.

El segundo proceso de la matriz primaria de la segunda ronda con menor peso de transferencia es el proceso p_{31} , que corresponde al nodo 7.

Lo anterior define la segunda ronda de asignación (Ver Tabla 43), se continua la evaluación mientras existan procesos a evaluar.

Tabla 43. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Segunda ronda (E1)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.6	p_{11}	2
0.7	p_{31}	7

Fuente: Elaboración propia

El siguiente paso es eliminar el proceso p_{31} y el nodo 7 de la matriz primaria porque fueron asignados en la segunda ronda. Aquellos procesos que tenían recursos destino en el nodo 7 (ya eliminado) pero tienen como opciones otros nodos destinos (que no poseen recursos destino de ese proceso) pasan a formar parte de la matriz secundaria (Ver Tabla 44 y Tabla 45). Los demás procesos cuya única opción es el nodo 7 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 46).

Tabla 44. Matriz primaria (MP). Tercera ronda (E1)

Procesos	Nodos					
	4	5	6	8	9	10
p_{14}	3.4	-	-	-	-	-
p_{15}	5.9	3.8	4.5	5.9	5.9	5.9
p_{35}	-	-	3.3	-	-	-
p_{36}	4.6	4.6	4.6	3.4	4.6	4.6
p_{37}	1.9	3.2	3.2	3.2	3.2	3.2

Fuente: Elaboración propia

Tabla 45. Matriz secundaria (MS). Tercera ronda (E1)

Pos	Nodos					
	4	5	6	8	9	10
p_{17}	0.9	0.9	0.9	0.9	0.9	0.9
p_{32}	1.5	1.5	1.5	1.5	1.5	1.5

Fuente: Elaboración propia

Tabla 46. Matriz de procesos no migrables. Tercera ronda (E1)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	2.3	-	-	-	-	-	-	-
p_{13}	4.1	-	-	-	-	-	-	-
p_{34}	2.7	-	-	-	-	-	-	-

Fuente: Elaboración propia

Se continúa con la matriz primaria hasta agotar opciones de procesos mientras que los nodos no alcancen el porcentaje de migración establecido.

El tercer proceso de la matriz primaria de la tercera ronda con menor peso de transferencia es el proceso p_{37} , que corresponde al nodo 4.

Lo anterior define la tercera ronda de asignación (Ver Tabla 47), se continúa la evaluación mientras existan procesos a evaluar.

Tabla 47. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Tercera ronda (E1)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.6	p_{11}	2
0.7	p_{31}	7
1.9	p_{37}	4

Fuente: Elaboración propia

El siguiente paso es eliminar el proceso p_{37} y el nodo 4 de la matriz primaria porque fueron asignados en la tercera ronda. Aquellos procesos que tenían recursos destino en el nodo 4 (ya eliminado) pero tienen como opciones otros nodos destinos (que no poseen recursos destino de ese proceso) pasan a formar parte de la matriz secundaria (Ver Tabla 48 y Tabla 49). Los demás procesos cuya única opción es el nodo 4 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 50).

Tabla 48. Matriz primaria (MP). Cuarta ronda (E1)

Procesos	Nodos				
	5	6	8	9	10
p_{14}	-	-	-	-	-
p_{15}	3.8	4.5	5.9	5.9	5.9
p_{35}	-	3.3	-	-	-
p_{36}	4.6	4.6	3.4	4.6	4.6

Fuente: Elaboración propia

Tabla 49. Matriz secundaria (MS). Cuarta ronda (E1)

Procesos	Nodos				
	5	6	8	9	10
p_{17}	0.9	0.9	0.9	0.9	0.9
p_{32}	1.5	1.5	1.5	1.5	1.5

Fuente: Elaboración propia

Tabla 50. Matriz de procesos no migrables. Cuarta ronda (E1)

Procesos	Nodos						
	2	4	5	6	7	8	9
p_{12}	2.3	-	-	-	-	-	-
p_{13}	4.1	-	-	-	-	-	-
p_{34}	2.7	-	-	-	-	-	-
p_{14}	4.1	3.4	-	-	-	-	-

Fuente: Elaboración propia

Se continúa con la matriz primaria hasta agotar opciones de procesos mientras que los nodos no alcancen el porcentaje de migración establecido. El cuarto proceso de la matriz primaria de la cuarta ronda con menor peso de transferencia es el proceso p_{35} , que corresponde al nodo 6. Lo anterior define la tercera ronda de asignación (Ver Tabla 51); se continua la evaluación mientras existan procesos a evaluar.

Tabla 51. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Cuarta ronda (E1)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.6	p_{11}	2
0.7	p_{31}	7
1.9	p_{37}	4
3.3	p_{35}	6

Fuente: Elaboración propia

El siguiente paso es eliminar el proceso p_{35} y el nodo 6 de la matriz primaria porque fueron asignados en la cuarta ronda. Aquellos procesos que tenían recursos destino en el nodo 6 (ya eliminado) pero tienen como opciones otros nodos destinos (que no poseen recursos destino de ese proceso) pasan a formar parte de la matriz secundaria (Ver Tabla 52 y Tabla 53). Los demás procesos cuya única opción es el nodo 6 se descartan y pasan a formar parte de la matriz de procesos no migrados (Ver Tabla 54).

Teniendo en cuenta que el nodo 3 ya alcanzó el porcentaje máximo de migración de procesos, sólo se evaluarán los procesos del nodo 1. Los procesos del nodo 3 que no fueron asignados en la matriz primaria y secundaria, se almacenan en una matriz de procesos no migrables (Ver Tabla 54).

Tabla 52. Matriz primaria (MP). Quinta ronda (E1)

Procesos	Nodos			
	5	8	9	10
p_{15}	3.8	5.9	5.9	5.9

Fuente: Elaboración propia

Tabla 53. Matriz secundaria (MS). Quinta ronda (E1)

Procesos	Nodos			
	5	8	9	10
p_{17}	0.9	0.9	0.9	0.9

Fuente: Elaboración propia

Tabla 54. Matriz de procesos no migrables. Quinta ronda (E1)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	2.3	-	-	-	-	-	-	-
p_{13}	4.1	-	-	-	-	-	-	-
p_{34}	2.7	-	-	-	-	-	-	-
p_{14}	4.1	3.4	-	-	-	-	-	-
p_{32}	0.9	1.5	1.5	1.5	1.5	1.5	1.5	1.5
p_{36}	3	4.6	4.6	4.6	4.6	3.4	4.6	4.6

Fuente: Elaboración propia

Se continúa con la matriz primaria hasta agotar opciones de procesos mientras que los nodos no alcancen el porcentaje de migración establecido.

El quinto proceso de la matriz primaria de la quinta ronda con menor peso de transferencia es el proceso p_{15} , que corresponde al nodo 5.

Lo anterior define la tercera ronda de asignación (Ver Tabla 55); se continua la evaluación mientras existan procesos a evaluar.

El siguiente paso es eliminar el proceso p_{15} y el nodo 5 de la matriz primaria porque fueron asignados en la quinta ronda. Aquellos procesos que tenían recursos destino en el nodo 5 (ya eliminado) pero tienen como opciones otros nodos destinos (que no poseen recursos destino de ese proceso) pasan a formar parte de la matriz secundaria (Ver Tabla 56). Los demás procesos cuya única opción es el nodo 5 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 57).

Tabla 55. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Quinta ronda (E1)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.6	p_{11}	2
0.7	p_{31}	7
1.9	p_{37}	4
3.3	p_{35}	6
3.8	p_{15}	5

Fuente: Elaboración propia

Tabla 56. Matriz secundaria (MS). Sexta ronda (E1)

Procesos	Nodos		
	8	9	10
p_{17}	0.9	0.9	0.9

Fuente: Elaboración propia

Tabla 57. Matriz de procesos no migrables. Sexta ronda (E1)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	2.3	-	-	-	-	-	-	-
p_{13}	4.1	-	-	-	-	-	-	-
p_{34}	2.7	-	-	-	-	-	-	-
p_{14}	4.1	3.4	-	-	-	-	-	-
p_{32}	0.9	1.5	1.5	1.5	1.5	1.5	1.5	1.5
p_{36}	3	4.6	4.6	4.6	4.6	3.4	4.6	4.6

Fuente: Elaboración propia

Teniendo en cuenta que en la matriz primaria se agotaron las opciones de migración de procesos, sólo se evaluarán las opciones de la matriz secundaria. El proceso de la matriz secundaria de la sexta ronda con menor peso de transferencia es el proceso p_{17} , que corresponde al nodo 8.

Lo anterior define la sexta ronda de asignación (Ver Tabla 58); se continua la evaluación mientras existan procesos a evaluar.

Tabla 58. Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) secundaria. Sexta ronda (E1)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.9	p_{17}	8

Fuente: Elaboración propia

Al finalizar la evaluación de los nodos 3 y 1 y habiéndose alcanzado el porcentaje máximo de migración de procesos establecido, se da por finalizado el presente macrociclo de evaluación.

Seguidamente se procede a la concatenación de la Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) primaria y la Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) secundaria. Se deben ordenar los procesos conforme a sus pesos de transferencia, de menor a mayor, para favorecer en la migración a los procesos que generan menor impacto. Esto da lugar a la Función de Asignación para Migración en Sistemas Distribuidos Concatenada (*FAMSDC*) que se muestra en la Tabla 59.

Tabla 59. Función de Asignación para Migración en Sistemas Distribuidos concatenada (*FAMSDC*) (E1)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.6	p_{11}	2
0.7	p_{31}	7
0.9	p_{17}	8
1.9	p_{37}	4
3.3	p_{35}	6
3.8	p_{15}	5

Fuente: Elaboración propia

2.5.6. Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema (E1)

Al promedio de carga computacional que se obtiene mediante el operador de agregación desarrollado en (La Red Martínez, 2017) se le agrega un criterio adicional que es la sobrecarga, se obtiene de la suma de los promedios de sobrecargas de cada proceso del nodo.

En la Tabla 60 se muestra la evaluación de los promedios de sobrecarga del nodo 1; el promedio A hace referencia al promedio de sobrecarga de los procesos que no se migran, el promedio B hace referencia al promedio de sobrecarga de los procesos migrables a otro nodo, este último promedio se evalúa por cada nodo destino.

Tabla 60. Evaluación de promedios de sobrecarga del nodo 1 (E1)

	Procesos	Nodos destinos	Promedios de sobrecarga
Promedio A			17.60
Promedio B	p_{11}	2	2.97
Promedio B	p_{15}	5	5.53
Promedio B	p_{17}	8	2.07
Promedio A+ Promedio B			28.17

Fuente: Elaboración propia

En la Tabla 61 se muestra la evaluación de los promedios de sobrecarga del nodo 2, el promedio A hace referencia al promedio de sobrecarga de los procesos locales, el promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{11} pasa a ser proceso p_{26} en su nodo receptor.

Tabla 61. Evaluación de promedios de sobrecarga del nodo 2 (E1)

	Procesos	Nodos origen	Promedios de sobrecarga
Promedio A			11.70
Promedio C	$p_{11} \rightarrow p_{26}$	1	2.97
Promedio A+ Promedio C			14.67

Fuente: Elaboración propia

En la Tabla 62 se muestra la evaluación de los promedios de sobrecarga del nodo 3, el promedio A hace referencia al promedio de sobrecarga de los procesos que no se migran, el promedio B hace referencia al promedio de sobrecarga de los procesos migrables a otro nodo, este último promedio se evalúa por cada nodo destino.

Tabla 62. Evaluación de promedios de sobrecarga del nodo 3 (E1)

	Procesos	Nodos destinos	Promedios de sobrecarga
Promedio A			14.50
Promedio B	p_{31}	7	2.10
Promedio B	p_{35}	6	3.73
Promedio B	p_{37}	4	3.30
Promedio A+ Promedio B			23.63

Fuente: Elaboración propia

En la Tabla 63 se muestra la evaluación de los promedios de sobrecarga del nodo 4, el promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{37} pasa a ser proceso p_{41} en su nodo receptor.

Tabla 63. Evaluación de promedios de sobrecarga del nodo 4 (E1)

	Procesos	Nodos origen	Promedios de sobrecarga
Promedio C	$p_{37} \rightarrow p_{41}$	3	3.30

Fuente: Elaboración propia

En la Tabla 64 se muestra la evaluación de los promedios de sobrecarga del nodo 5, el promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{15} pasa a ser proceso p_{51} en su nodo receptor.

Tabla 64. Evaluación de promedios de sobrecarga del nodo 5 (E1)

	Procesos	Nodos origen	Promedios de sobrecarga
Promedio C	$p_{15} \rightarrow p_{51}$	1	5.53

Fuente: Elaboración propia

En la Tabla 65 se muestra la evaluación de los promedios de sobrecarga del nodo 6, el promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{35} pasa a ser proceso p_{61} en su nodo receptor.

Tabla 65. Evaluación de promedios de sobrecarga del nodo 6 (E1)

Procesos	Nodos origen	Promedios de sobrecarga
Promedio C $p_{35} \rightarrow p_{61}$	3	3.73

Fuente: Elaboración propia

En la Tabla 66 se muestra la evaluación de los promedios de sobrecarga del nodo 7, el promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{31} pasa a ser proceso p_{71} en su nodo receptor.

Tabla 66. Evaluación de promedios de sobrecarga del nodo 7 (E1)

Procesos	Nodos origen	Promedios de sobrecarga
Promedio C $p_{31} \rightarrow p_{71}$	3	2.10

Fuente: Elaboración propia

En la Tabla 67 se muestra la evaluación de los promedios de sobrecarga del nodo 8, el promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{17} pasa a ser proceso p_{81} en su nodo receptor.

Tabla 67. Evaluación de promedios de sobrecarga del nodo 8 (E1)

Procesos	Nodos origen	Promedios de sobrecarga
Promedio C $p_{17} \rightarrow p_{81}$	1	2.07

Fuente: Elaboración propia

Los nodos 9 y 10 no se evalúan porque no emigraron ni inmigraron procesos, se quedan con sus promedios de carga iniciales, calculados en la Tabla 17.

Para realizar la comparación de los nuevos valores de cargas computacionales se consideran dos etapas, la primera es aplicar la sobrecarga sin realizar la migración de procesos (ver Tabla 68) y la segunda es aplicar la sobrecarga una vez migrados los procesos (ver Tabla 69).

Tabla 68. Carga computacional sin migración (E1)

Nodo	%CPU	%Mem.	%E/S	Prom.	Prom. de Sobrecar.	Prom. Final
1	80	90	75	81.67	28.13	109.80
2	30	30	50	36.67	11.70	48.37
3	77	49	85	70.33	24.27	94.60
4	6	7	9	7.33	0	7.33
5	5	5	6	5.33	0	5.33
6	7	9	7	7.67	0	7.67
7	7	10	15	10.67	0	10.67
8	9	6	7	7.33	0	7.33
9	10	9	15	11.33	0	11.33
10	8	7	8	7.67	0	7.67

Nota: Los nodos del 4 a 10 no tienen procesos activos, por lo tanto, tienen valor de sobrecarga 0.

Fuente: Elaboración propia

Tabla 69. Carga computacional con migración (E1)

Nodo	%CPU	%Mem.	%E/S	Prom.	Prom. de Sobrecar.	Prom. Final
1	80	90	75	81.67	17.60	99.27
2	30	30	50	36.67	14.67	51.33
3	77	49	85	70.33	14.50	84.83
4	6	7	9	7.33	3.30	10.63
5	5	5	6	5.33	5.53	10.87
6	7	9	7	7.67	3.73	11.40
7	7	10	15	10.67	2.10	12.77
8	9	6	7	7.33	2.07	9.40
9	10	9	15	11.33	0	11.33
10	8	7	8	7.67	0	7.67

Nota: Los nodos 9 y 10 no tienen procesos activos, por lo tanto, su valor de sobrecarga es 0.

Fuente: Elaboración propia

2.5.7. Métrica propuesta para evaluar el balanceo de carga en el sistema distribuido (E1)

Para evaluar el impacto de la migración de procesos y cómo ésta afecta el estado global del sistema, es necesario aplicar algún mecanismo de medición de los estados de los nodos antes y después de la migración. Para ello, se utiliza la desviación estándar, considerando la carga final de los nodos del sistema distribuido.

Se utilizarán los siguientes criterios asociados a cada proceso migrado: sobrecarga de procesador, sobrecarga de memoria y sobrecarga de entrada / salida, que se mostraron en la Tabla 22.

La desviación estándar del nivel de carga de los nodos sin considerar migración aplicando la formula (11) es la siguiente:

Se tiene en cuenta la Tabla 68: 109.80; 48.37; 94.60; 7.33; 5.33; 7.67; 10.67; 7.33; 11.33 y 7.67, cuyo promedio es de 31.01.

$$s = \sqrt{(((109.83-30.95)^2 + (48.37-30.95)^2 + (94.60-30.95)^2 + (7.33-30.95)^2 + (5.33-30.95)^2 + (7.67-30.95)^2 + (10.67-30.95)^2 + (7.33-30.95)^2 + (11.33-30.95)^2 + (7.67-30.95)^2) / (10-1))}$$

$$s = 39.75$$

La desviación estándar del nivel de carga de los nodos con migración aplicando la fórmula (11) es la siguiente.

Se tiene en cuenta la Tabla 69: 99.27; 51.33; 84.83; 10.63; 10.87; 11.40; 12.77; 9.40; 11.33 y 7.67, cuyo promedio es de 30.95.

$$s = \sqrt{(((99.27-30.95)^2 + (51.33-30.95)^2 + (84.83-30.95)^2 + (10.63-30.95)^2 + (10.87-30.95)^2 + (11.40-30.95)^2 + (12.77-30.95)^2 + (9.40-30.95)^2 + (11.33-30.95)^2 + (7.67-30.95)^2) / (10-1))}$$

$$s = 34.81$$

A partir de los resultados obtenidos se observa que la desviación estándar del conjunto de promedios finales sin migración es de 39.75 y la desviación estándar del conjunto de promedios finales con migración es de 34.81. Esto indica que se ha logrado balancear la carga porque los

valores del conjunto de promedios finales con migración están más concentrados alrededor de la media, mientras que los valores de promedios finales sin migración están más dispersos.

2.6. Comentarios y discusiones

El trabajo propone un modelo de migración inteligente de procesos para la gestión de recursos y procesos en sistemas distribuidos, con el objetivo de resolver los inconvenientes relacionados con la sobrecarga que pueden presentarse en los distintos nodos de un sistema distribuido. Se presenta una variante de un enfoque innovador para la gestión de recursos compartidos en sistemas distribuidos que considera el balanceo de carga de trabajo entre los distintos nodos.

La propuesta se basa en la evaluación del estado de las cargas computacionales, lo que permite detectar nodos muy cargados con respecto a nodos inactivos o con baja carga de trabajo y corregir la distribución de carga de trabajo en el sistema.

La solución propuesta se basa en un *Runtime* centralizado que gestiona los procesos y recursos compartidos, y que interactúa con los *Runtimes* de los nodos distribuidos para intercambiar información (todos ellos interactúan con sus respectivos sistemas operativos). Además, se establecen configuraciones iniciales del algoritmo para cada situación en particular, lo que permite adecuar la solución planteada a distintos entornos.

Se recopila información de todos los nodos en un nodo central, se aplica el proceso de agregación y se obtiene la lista de asignaciones de recursos a procesos, y la carga adicional de trabajo que ello significaría sobre los distintos nodos distribuidos.

La evaluación del estado de las cargas computacionales se realiza de manera periódica, lo que permite considerar posibles migraciones para balancear la carga de trabajo de los distintos nodos del sistema.

Como métrica para evaluar la efectividad de la solución propuesta se utiliza la desviación estándar, que es una medida de dispersión muy útil y ampliamente utilizada en el análisis de datos debido a su fácil interpretación, sensibilidad a todos los valores de los datos y propiedades matemáticas útiles.

Capítulo III – Escenario 2. Migración de procesos en condiciones de heterogeneidad

3.1. Trabajos previos

Todos los trabajos previamente mencionados en el Capítulo II del escenario homogéneo (E1) también se emplean como referencia para este Capítulo agregando los siguientes.

En (Beiruti & Ganjali, 2020) y en (Chang et al., 2020) se ha propuesto y discutido nuevas técnicas de migración de procesos, en (Thakkar & Pandya, 2013) se presenta un mecanismo de migración de procesos en un sistema distribuido heterogéneo

En (Junaidi et al., 2020) y en (Kim et al., 2021) proponen un algoritmo de aprendizaje automático que utiliza los recursos del servidor, la CPU y la memoria, para predecir el futuro de las cargas del servidor, en (Banerjee et al., 2019) describe el diseño de un sistema que aborda el reto de programar cargas de trabajo de análisis de datos distribuidos en redes de procesamiento heterogéneos que incluyen CPUs (Central Processing Unit - Unidad Central de Procesamiento), GPUs (Graphics Processing Unit - Unidad de Procesamiento Gráfico), FPGAs (Field Programmable Gate Arrays - Matriz de Puertas Lógicas Programable en Campo) y ASICs (Application Specific Integrated Circuit - Circuito Integrado para Aplicaciones Específicas) en entornos basados en la nube, dinámicos y de múltiples usuarios.

En (Thakkar & Pandya, 2013) se presenta un mecanismo de migración de procesos en un sistema distribuido heterogéneo. Utiliza un mecanismo de representación de datos externos para ayudar a reducir la complejidad del software en el proceso de traducción, considerando el formato del nodo de origen con respecto al formato de datos del nodo de destino.

En (Banerjee et al., 2019) se describe el diseño de un sistema que aborda el reto de programar cargas de trabajo de análisis de datos distribuidos en redes de procesamiento

heterogéneos (que incluyen CPUs, GPUs, FPGAs y ASICs) en entornos basados en la nube, dinámicos y de múltiples usuarios.

3.2. Premisas

El escenario planteado consiste en que el Runtime (software tiempo de ejecución) del nodo central, reciba la información proveniente de los distintos nodos y a través de un operador de agregación determine el orden de asignación de recursos a procesos y de procesos a nodos, según la siguiente premisa:

- Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua sin constituir grupos de procesos, los procesos no requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y sin exigencias estrictas de consenso para lograr el acceso (no se requiere un consenso para asignar de manera consecutiva los recursos solicitados por un proceso o grupo de procesos, es decir, que iniciada la secuencia de asignación de recursos a un proceso, la misma puede ser interrumpida para asignar recursos a otro proceso).
- Bajo las circunstancias mencionadas anteriormente se tendrá en cuenta la aplicación de la migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo heterogeneidad (respecto de compatibilidad de código, SO (Sistema Operativo), arquitectura y set de instrucciones).

El algoritmo de balanceo de carga corresponde a una serie de cálculos que se realizan para lograr la migración de procesos de nodos cargados a nodos más aliviados, partiendo de la estructura de datos utilizados en (La Red Martínez, 2017) y (La Red Martínez et al., 2018).

3.3. Propuesta de solución

Considerando lo desarrollado en (La Red Martínez, 2017) y (La Red Martínez et al., 2018)

se presentará una variante de un enfoque innovador para la gestión de recursos compartidos en sistemas distribuidos, donde se desarrolla un operador de agregación para la asignación de recursos en un sistema distribuido.

Las premisas, estructuras de datos y operadores mencionados en la publicación anterior se utilizan como punto de partida para este trabajo. Para afrontar la problemática planteada, se define un Runtime (software en tiempo de ejecución) central que gestiona los procesos y recursos compartidos e interactúa con sus similares de los nodos distribuidos para intercambiar información. En el nodo central se recopila la información de todos los nodos, se aplica el proceso de agregación y se obtiene la lista de asignaciones de recursos a procesos.

Durante el funcionamiento del sistema distribuido, cada cierto tiempo se reúne y evalúa información de control referida a las cargas nodales para obtener una imagen global del sistema (macro imagen), que refleja el estado de los nodos, recursos y procesos en un determinado momento, lo que permite evaluar posibles migraciones para balancear la carga de trabajo de los distintos nodos del sistema. Las estructuras de datos utilizadas son las mismas mencionadas en el escenario 1 del capítulo anterior.

Para cubrir estas necesidades se plantea un conjunto de propuestas que consideran la migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo heterogeneidad de los nodos (respecto de compatibilidad de código, SO, arquitectura, set de instrucciones, etc.), incluyendo migración de procesos que significarán el acceso a datos distribuidos con alto impacto en E/S y tráfico de red (consumo de ancho de banda), y una posible migración de proceso consecuencia de cambios en el estado global del sistema y de los requerimientos variables de los procesos a lo largo del tiempo, esta verificación se hará cada cierto lapso Δt .

3.4. Descripción del operador de agregación

En cada nodo se define una interfaz entre las aplicaciones y el sistema operativo, que a través de un *Runtime* (software en tiempo de ejecución complementario del sistema operativo) incluido en esa interfaz, gestiona los procesos y recursos compartidos y define el escenario correspondiente. Además, los *Runtime* interactúan entre sí para intercambiar información y en uno de los nodos hay un coordinador global de *Runtime* que evalúa y ejecuta el modelo de decisión y el operador de agregación correspondiente.

Se plantea una propuesta que considera la migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo heterogeneidad de los nodos (respecto de compatibilidad de código, sistema operativo, arquitectura, set de instrucciones, etc.).

El operador de agregación propuesto utiliza información proveniente de (La Red Martínez, 2017), relacionada con la carga computacional actual de los nodos, las categorías de carga computacional y de los vectores de pesos asociados a las mismas, las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo y las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles.

Los cálculos relacionados con la obtención de esta información, considerados preliminares, son el punto de partida para otros que determinarán los procesos que serán migrados.

Al considerar aspectos relacionados con el balanceo de carga, se considera una política efectiva de migración, que tiene en cuenta la prioridad de los procesos, limitaciones en la migración (para determinar el número de procesos que pueden pasar de un nodo a otro), el peso de los recursos cuyo estado puede ser local respecto del proceso que lo requiere, pero podría cambiar a remoto al realizar la migración de procesos y el impacto de ésta en función de las

prestaciones del nodo receptor según características de hardware.

Para un buen diseño en el balanceo de carga se consideran algunas políticas muy utilizadas para esta tarea, entre las cuales se pueden mencionar, políticas de:

- Locación, para determinar el destino del proceso seleccionado.
- Intercambio de información, que permita intercambiar información entre los nodos.
- Asignación, que determine la prioridad de ejecución de los procesos locales y remotos en un determinado nodo.
- Migración, que limite la cantidad de procesos migrables por nodo, y la cantidad de procesos que puede recibir un nodo destino.

Los procesos pueden ser locales o remotos, los segundos se diferencian en que se procesarán en un nodo distinto al original.

Los recursos serán considerados locales si están en el mismo nodo donde se aloja el proceso que los utilizará, y remotos si están en nodos distintos.

Para obtener información de la carga de nodos y estado global del sistema distribuido se deben realizar diferentes observaciones, de las que surgen varios escenarios que se representan mediante un diagrama de flujo que muestra una serie de situaciones en las que se deben tomar decisiones que derivan en dos opciones posibles: "Sí" y "No". Cada opción se define como "Primera alternativa" y "Segunda alternativa" respectivamente.

Dependiendo del resultado de la evaluación, el flujo del diagrama se dirigirá hacia otra situación, donde se tomará otra decisión. Este proceso se repetirá hasta que se alcance una solución final, cuando ya no existan evaluaciones de recursos, procesos y nodos pendientes.

Se considerará la aplicación de las siguientes etapas adicionales para lograr balanceo de carga aplicando migración de procesos:

1. Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos.
2. Selección de los posibles nodos destino mediante la evaluación de su carga y la compatibilidad de los nodos.
3. Evaluación del peso de los recursos y su posible impacto ante la migración de procesos.
4. Establecimiento de las categorías de nodos según las prestaciones de cada uno.
5. Determinación del orden de migración de procesos y de los nodos destino correspondientes.
6. Evaluación de nuevos valores de sobrecarga de los procesos inmigrantes según categoría del nodo receptor.
7. Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema.
8. Evaluación del consumo de energía de los nodos una vez migrado los procesos.

Se presentará una variante innovadora al operador de agregación utilizado en (La Red Martínez, 2017) y (La Red Martínez et al., 2018) , incluyendo la migración inteligente de procesos en sistemas distribuidos heterogéneos.

El modelo se desarrolla en seis etapas bien definidas como se muestra en la Fig. 9.

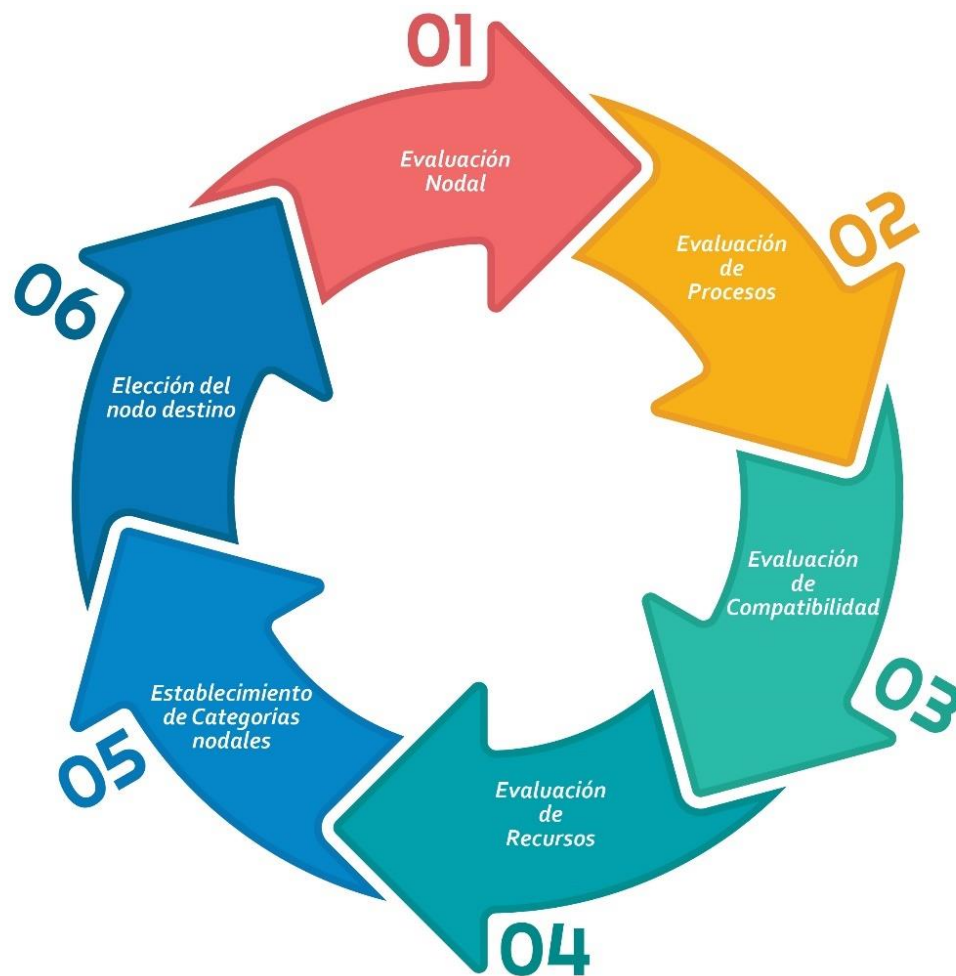


Fig. 9. Etapas del cálculo de migración de procesos (E2).
Fuente: Elaboración propia

Las etapas corresponden a la evaluación de los nodos, clasificándolos según su nivel de carga (alta, media, baja), seguidamente se evalúan los procesos de los nodos con carga alta y se evalúa la compatibilidad de los nodos con carga baja que puedan alojar procesos de otros nodos. Se evalúa los recursos y su posible impacto ante la migración de procesos y luego se categorizan los nodos según las prestaciones de cada uno. Se priorizan los nodos remotos con mejores prestaciones y que contengan mayor peso de recursos para cada proceso. Se obtiene un listado de los procesos candidatos a ser migrados que serán ordenados por su impacto en el ancho

de banda de red, priorizándose los procesos con menor peso de transferencia.

3.4.1. Cálculo de la carga computacional actual de los nodos. Escenario 2 (E2)

En la Fig. 10 se muestra un diagrama que contempla la evaluación a nivel nodal, se realiza la configuración de valores iniciales, en donde se establecen los criterios para indicar los porcentajes de carga, prioridades y otros parámetros que se tendrán en cuenta para la migración.

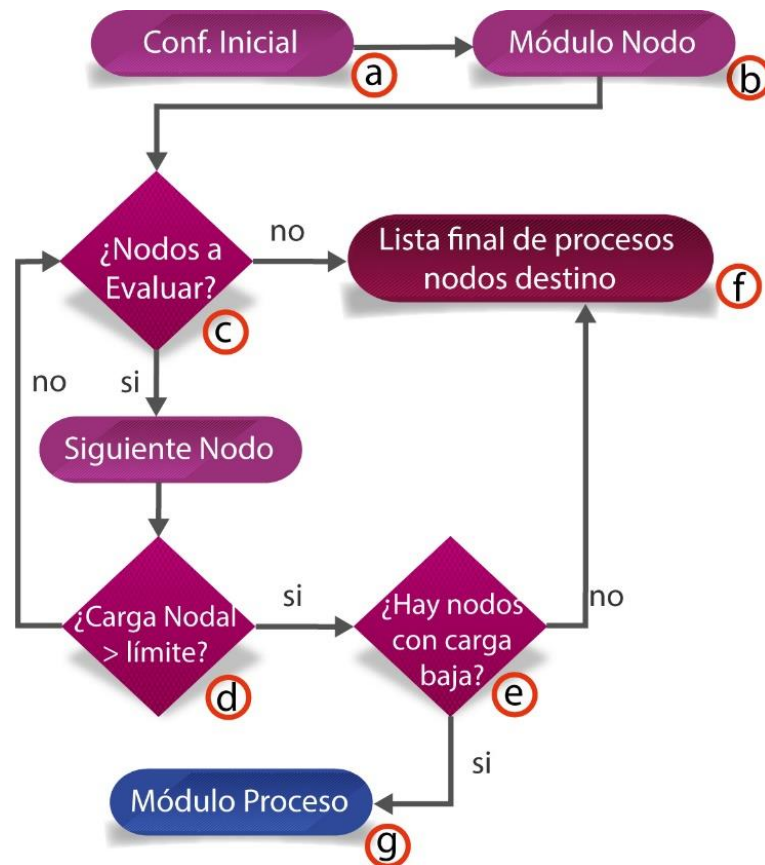


Fig. 10. Módulo de nodo (E2).

Fuente: Elaboración propia

Para establecer las cargas nodales se calcula el promedio de carga del porcentaje de uso de CPU, memoria RAM y entrada/salida, lo que se muestra en la Tabla 1. Se consideran cargados los nodos cuyo promedio de carga sea mayor a 70%, carga media entre 40% y 70%, y como carga baja a los nodos cuyo valor promedio de carga sea menor que 40%.

Esta etapa del operador de agregación del escenario 2 (E2) comparte los mismos procedimientos básicos de cálculos, establecidos en el escenario 1 (E1).

3.4.2. Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos (E2)

La información resultante de la evaluación del estado de los nodos ayuda a identificar aquellos con carga alta, sus procesos y los recursos tanto locales como remotos vinculados con dichos procesos.

En el punto (g) de la Fig. 11, se da inicio al módulo de procesos, y es en donde se analiza el escenario a nivel de procesos, dentro de cada nodo

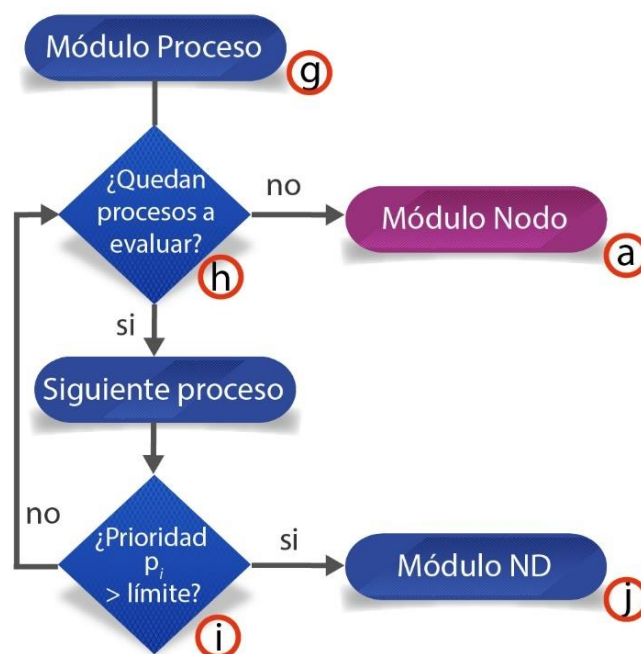


Fig. 11. Módulo proceso (E2).

Fuente: Elaboración propia

Esta etapa del operador de agregación del escenario 2 (E2) comparte los mismos procedimientos básicos de cálculos, establecidos en el escenario 1 (E1).

3.4.3. Selección de los posibles nodos destino mediante la evaluación de compatibilidad y la carga (E2)

En esta etapa, se realiza una evaluación para determinar si hay nodos compatibles con carga baja que puedan alojar procesos de otros nodos. Los procesos que se evaluarán son aquellos que pasaron la etapa anterior. Se identifican los nodos que se encuentran en la categoría de carga baja.

Para evaluar la compatibilidad de los nodos, se considera la Tabla 70 y Tabla 71 que incluye componentes de hardware y software de los nodos. Los componentes de cada nodo se agrupan en conjuntos. La evaluación se basa en determinar si un nodo tiene los mismos componentes o más que el nodo evaluado, realizando la comprobación de si un conjunto O (nodo origen) está contenido en otro D (nodo destino) o si dos conjuntos (O y D) son iguales.

Tabla 70. Componentes de hardware. (E2)

Nodos	h_1	...	h_e	...	h_o
n_1	ph_{11}	...	ph_{1e}	...	ph_{1o}
...	...	...	...	...	...
n_i	ph_{i1}	...	ph_{ie}	...	ph_{io}
...	...	...	...	...	...
n_m	ph_{m1}	...	ph_{me}	...	ph_{mo}

Fuente: Elaboración propia

Con n indicando el nodo, donde $i = 1, \dots, m$ (nº de nodos en el sistema distribuido).

Con h indicado los componentes de hardware, donde $e = 1, \dots, o$ (nº de componente de hardware para cada nodo),

Con ph indicando la presencia o ausencia del componente de hardware, donde $i = 1, \dots, m$ (nº de nodos en el sistema distribuido) y $e = 1, \dots, o$ (nº de componente de hardware para cada nodo).

Tabla 71. Componentes de software (E2)

Nodos	s_1	...	s_u	...	s_c
n_1	ps_{11}	...	ps_{1u}	...	ps_{1c}
...	...	...	...	...	...
n_i	ps_{i1}	...	ps_{iu}	...	ps_{ic}
...	...	...	...	...	...
n_m	ps_{m1}	...	ps_{mu}	...	ps_{mc}

Fuente: Elaboración propia

con n indicando el nodo, donde $i = 1, \dots, m$ (nº de nodos en el sistema distribuido).

con s indicando los componentes de software, donde $u = 1, \dots, c$ (nº de componentes de software para cada nodo).

Con ps indicando la presencia o ausencia del componente de software, donde $i = 1, \dots, m$ (nº de nodos en el sistema distribuido) y $u = 1, \dots, c$ (nº de componentes de software para cada nodo).

Algunos de los componentes de hardware y software pueden o no estar presente en cada nodo.

La fórmula utilizada en este caso es la siguiente:

Compatibilidad (O, D): $O \subseteq D$

Esto significa que todos los elementos de O (nodo origen) también están presentes en D (nodo destino). Si se cumple esta condición, podemos decir que O está contenido en D y, por lo tanto, es compatible.

De la lista de nodos compatibles se denotan los nodos que se encuentran dentro de la categoría de carga baja:

Nodos destino con carga baja compatible = {NDBC} = $\{n_1, \dots, n_n\}$ con n indicando el nº

total de posibles nodos destino según se muestra en la Tabla 72.

Para cada proceso puede haber más de un nodo compatible con carga baja disponible. Teniendo en cuenta el listado de la Tabla 72 se verifican los recursos solicitados por cada proceso p_{ij} para determinar el nodo candidato.

Tabla 72: Nodos compatibles con carga baja (E2)

Nodos compatibles con carga baja
n_1
....
n_i
....
n_n

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

3.4.4. Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E2)

Una vez identificado los procesos del nodo con carga alta y los posibles nodos destinos, se realiza la comparación de los recursos locales en relación con los remotos, con el fin de evaluar el impacto ante la migración de procesos.

En el punto **(j)** de la Fig. 12, se activa el módulo de nodos destino, donde se buscan nodos remotos que puedan albergar recursos solicitados por cada proceso evaluado. Se busca generar el menor peso de transferencia al pasar los recursos locales (que pueda tener en un momento dado un proceso), a remotos (al migrar el proceso). Cualquier nodo remoto compatible con el nodo origen será evaluado, y podrá ser candidato a convertirse en un nodo destino.

(k) Situación: se verifica dentro del sistema distribuido si existen posibles nodos destino **(ND)** que aún no han sido evaluados.

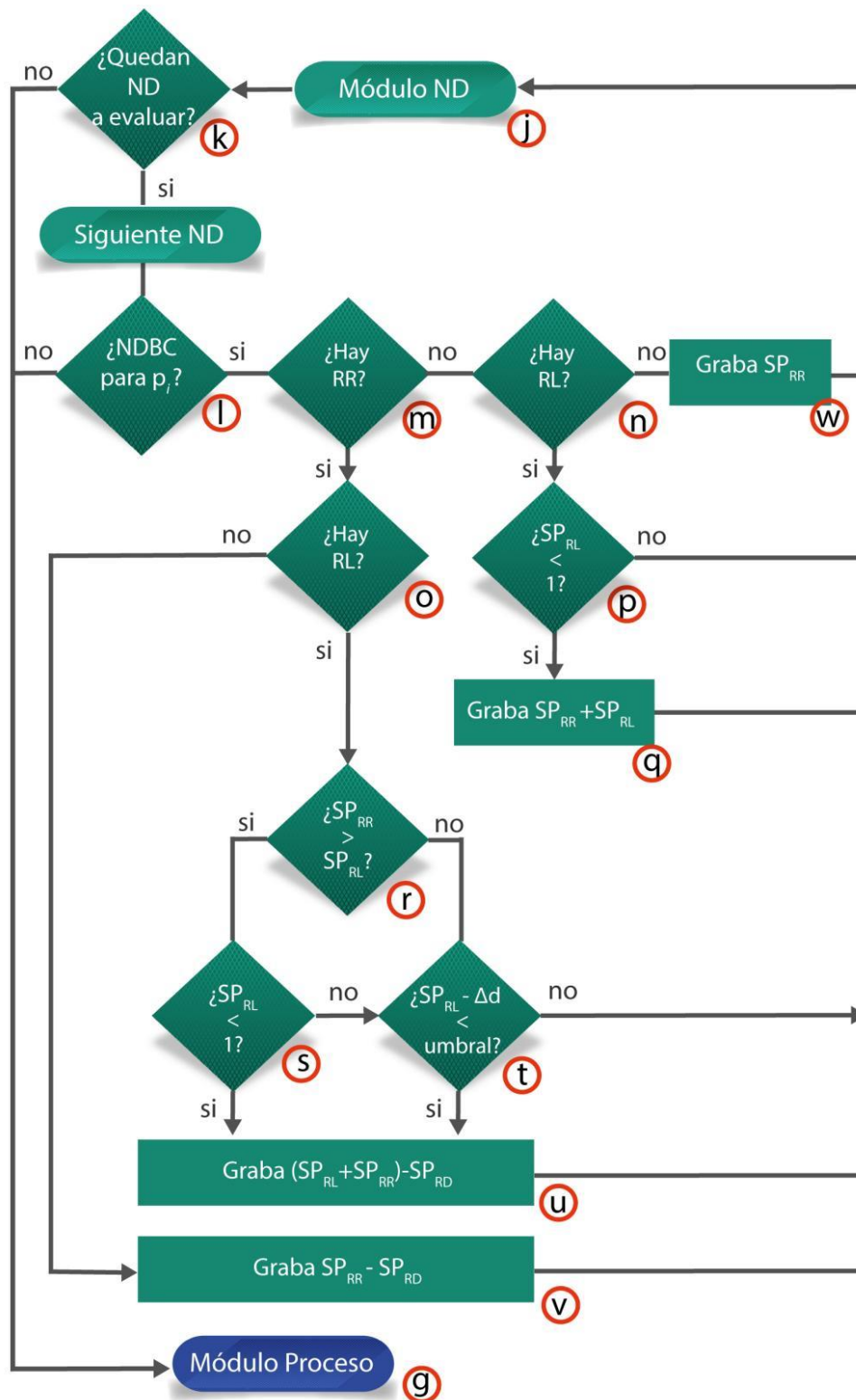


Fig. 12. Módulo nodo destino (E2)

Fuente: Elaboración propia

- *Primera alternativa* (sí): si existen nodos destino que aún no han sido evaluados se selecciona el siguiente nodo destino y se pasa al punto **(l)**.

- *Segunda alternativa* (no): si no existen nodos destino para evaluar entonces se vuelve al módulo de procesos en el punto **(g)**.

(l) Situación: se evalúa si para el proceso en cuestión existen nodos destino compatible y con carga baja (**NDBC**) considerando compatibilidad de códigos, arquitectura, set de instrucciones, etc.

- *Primera alternativa* (sí): si para el proceso existen nodos con carga baja se pasa al punto **(m)**.

- *Segunda alternativa* (no): si no existen nodos con carga baja entonces se vuelve al punto **(g)**.

(m) Situación: se verifica si el nodo tiene alojado recursos remotos (**RR**) asociados al proceso en cuestión.

- *Primera alternativa* (sí): si el nodo tiene recursos remotos asociados al proceso en cuestión se realiza **(o)**.

- *Segunda alternativa* (no): si el nodo no tiene recursos remotos asociados al proceso en cuestión entonces se evalúa **(n)**.

(n) Situación: se verifica si el proceso en cuestión tiene recursos locales.

- *Primera alternativa* (sí): si el proceso tiene recursos de forma local se evalúa **(p)**.

- *Segunda alternativa* (no): si el proceso no tiene recursos de forma local entonces se evalúa **(w)**.

(o) Situación: se verifica si el proceso en cuestión tiene recursos locales.

- *Primera alternativa* (sí): si el proceso tiene recursos de forma local se realiza (r).
- *Segunda alternativa* (no): si el proceso no tiene recursos de forma local entonces se ejecuta el punto (v).

(p) Situación: se verifica si la sumatoria de pesos de los recursos locales (SP_{RL}) es menor a un límite; sin pérdidas de generalidad, en este caso se considera que el valor límite es 1 (uno).

- *Primera alternativa* (sí): si la sumatoria de pesos de los recursos locales es menor a 1 (uno) entonces se ejecuta el punto (q).
- *Segunda alternativa* (no): si la sumatoria de pesos de los recursos locales es no menor a 1 (uno) entonces se vuelve al punto (j).

(q) Situación: en el vector de migración de pesos (Fig. 13) se almacena el valor resultante de la suma de pesos de recursos locales y los pesos de recursos remotos, peso de transferencia que generaría la migración del proceso al nodo destino.

(r) Situación: Se verifica si la suma de los pesos de los recursos remotos (SP_{RR}) es mayor a la suma de los pesos de recursos locales.

- *Primera alternativa* (sí): Si la sumatoria de los pesos de recursos remotos es mayor al peso de los recursos locales se realiza (s).
- *Segunda alternativa* (no): Si la sumatoria de los pesos de recursos remotos es menor al peso de los recursos locales se realiza (t).

(s) Situación: Se verifica si la suma de los pesos del recurso local es menor a un límite (para este caso límite es igual a uno en base a (La Red Martínez, 2017)

- *Primera alternativa* (sí): si la suma de los pesos del recurso local es menor al límite entonces se realiza (**u**).

- *Segunda alternativa* (no): si la suma de los pesos del recurso local es mayor al límite se evalúa (**t**).

(**t**) Situación: se verifica si la suma de los pesos de recursos locales menos Δd (Δd es el peso de los recursos destino más un 20% del total de pesos de recursos del proceso) es menor a un umbral (sin pérdidas de generalidad, en este caso se considera que el umbral es 1 (uno)).

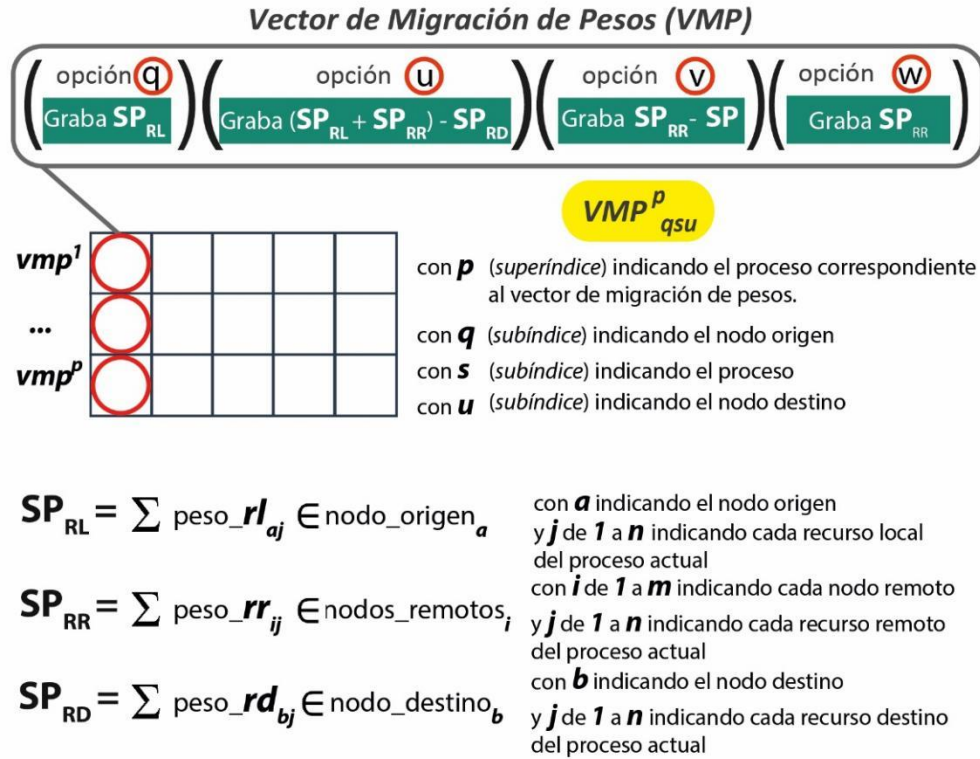
- *Primera alternativa* (sí): si la suma de los pesos de recursos locales menos Δd es menor a un umbral, entonces se realiza el punto (**u**).

- *Segunda alternativa* (no): si la suma de los pesos de recursos locales menos Δd es mayor a un umbral, se vuelve a (**j**).

(**u**) Situación: En el vector de migración de pesos (Fig. 13) se almacena el valor resultante de la suma de pesos de recursos locales más la suma de los pesos de los recursos remotos menos la suma de pesos de recursos destino (SP_{RD}), peso de transferencia que generaría la migración del proceso al nodo destino.

(**v**) Situación: En el vector de migración de pesos (Fig. 13) se almacena el valor resultante de la suma de pesos de recursos locales menos la suma de pesos de recursos destino, peso de transferencia que generaría la migración del proceso al nodo destino.

(**w**) Situación: En el vector de migración de pesos (Fig. 13) se almacena el valor resultante de la suma de pesos de recursos remotos que tiene asociado el proceso, peso de transferencia que generaría la migración del proceso al nodo destino.



*Habr  un vector de migraci n de pesos para cada proceso evaluado y un valor para cada nodo destino calculado.

Fig. 13. Vector de migraci n de procesos (E2).

Fuente: Elaboraci n propia

En la Tabla 73 se observa la matriz general de pesos de migraci n ($MGPM$), resultado de la evaluaci n de las diferentes situaciones que pueden presentarse dentro de un sistema distribuido para la migraci n inteligente de procesos. Para cada proceso, se registra el conjunto de nodos compatibles con carga baja y para cada posible nodo destino evaluado se incluye el valor resultante de la sumatoria del peso de transferencia del vector vmp (obtenido en la Fig. 13).

Tabla 73. Matriz general de pesos de migraci n (E2)

$MGPM$	n_1	...	n_k	...	n_m
$mgpm_1$	vmp^1_{111}	...	vmp^1_{11u}	...	vmp^1_{11m}
...	...	...	...	...	...
$mgpm_i$	vmp^p_{qs1}	...	vmp^p_{qsu}	...	vmp^p_{qsm}
...	...	...	...	...	...
$mgpm_o$	vmp^n_{rt1}	...	vmp^n_{rtu}	...	vmp^n_{rtm}

Elaboraci n propia.

Con i en $mgpm_i$ variando de 1, ..., o , indicando el proceso correspondiente en la matriz general de pesos de migración.

Con p en vmp (*superíndice*) variando de 1, ..., n , indicando el proceso correspondiente al vector de migración de pesos.

Con q,s,u en vmp (*subíndices*) variando q de 1, ..., r , indicando el nodo origen, s de 1, ..., t , indicando el número de proceso y u de 1, ..., m , indicando el nodo destino, del vector de migración de pesos.

Con k en n variando de 1, ..., m , indicando el nodo con carga baja para cada proceso.

3.4.5. Establecimiento de las categorías nodales según sus prestaciones (E2)

Para clasificar los nodos en diferentes categorías, se tendrán en cuenta los componentes de hardware definidos en la Tabla 70 y las características específicas de cada componente (por ejemplo, para el componente procesador: número de núcleos, mflops (millones de instrucciones de punto flotante por segundo), cache, etc.) de la Tabla 74.

Tabla 74. Características específicas del componente de hardware x (E2)

Nodos	ce_1	...	ce_j	...	ce_n
n_1	va_{11}	...	va_{1j}	...	va_{1n}
...	...	...	...	...	...
n_i	va_{i1}	...	va_{ij}	...	va_{in}
...	...	...	...	...	...
n_m	va_{m1}	...	va_{mj}	...	va_{mn}

Elaboración propia.

con n indicando el nodo, donde $i = 1, \dots, m$ (n° de nodos en el sistema distribuido).

Con ce indicando la característica específica del componente de hardware x , donde $j = 1, \dots, n$ (características específicas del componente).

con va indicando el valor de la característica específica del componente, donde $i = 1, \dots, m$ (nº de nodo del sistema distribuido) y donde $j = 1, \dots, n$ (características específicas del componente).

Por cada componente de hardware se define un vector de pesos asociado a sus características específicas, cuyos valores podrán ser diferentes cuando se desee reflejar la importancia relativa de uno sobre otro, ver Tabla 75.

Tabla 75. Vector de peso para características específicas de cada componente (E2)

Nodos	ce_1	...	ce_j	...	ce_n
n_1	pe_{11}	...	pe_{1j}	...	pe_{1n}
...	...	...	...	...	...
n_i	pe_{i1}	...	pe_{ij}	...	pe_{in}
...	...	...	...	...	...
n_m	pe_{m1}	...	pe_{mj}	...	pe_{mn}

Elaboración propia.

Con ce indicando la característica específica del componente de hardware x , donde $j = 1, \dots, n$ (características específicas del componente).

Con pe indicando el valor del peso de la característica específica del componente de hardware x , donde $i = 1, \dots, m$ (nº de nodo del sistema distribuido) y donde $j = 1, \dots, n$ (características específicas del componente). Los valores de las características específicas de cada componente x en la Tabla 74, tomados fila por fila, es decir, respecto a cada nodo, se multiplicarán por el vector de pesos de la Tabla 75. Esto puede verse en la Tabla 76.

Tabla 76. Potencia del componente de hardware X . (E2)

Nodos	h_1	...	h_e	...	h_o
n_1	pot_{11}	...	pot_{1e}	...	pot_{1o}
...	...	...	...	...	...
n_i	pot_{i1}	...	pot_{ie}	...	pot_{io}
...	...	...	...	...	...
n_m	pot_{m1}	...	pot_{me}	...	pot_{mo}

Elaboración propia.

Con h_e indicando el componente de hardware, donde $e = 1, \dots, o$ (nº de componente de hardware).

$$pot_{ie} = \sum (va_{ij} * pe_{ij}). \quad (13)$$

Con pot_{ie} indicando la potencia del componente de hardware, donde $i = 1, \dots, m$ (nº de nodo del sistema distribuido) y $e = 1, \dots, o$ (nº de componente de hardware).

Esto se repite para cada uno de los componentes pendientes.

A continuación, en la Tabla 77 se realiza el proceso de sustituir los valores de la Tabla 70, se reemplaza cada ph por el valor de la potencia calculada para cada componente de hardware en la Tabla 76 (procesador, memoria RAM, disco duro, etc.).

Tabla 77. Potencia de componentes de hardware. (E2)

Nodos	h_1	...	h_e	...	h_o
n_1	pot_{11}	...	pot_{1e}	...	pot_{1o}
...	...	...	...	...	...
n_i	pot_{i1}	...	pot_{ie}	...	pot_{io}
...	...	...	...	...	...
n_m	pot_{m1}	...	pot_{me}	...	pot_{mo}

Elaboración propia

Para establecer un índice ponderado de capacidad para cada nodo, los valores de potencia de hardware presentes en cada nodo (Tabla 77) se multiplica por otro vector de pesos (Tabla 78) que refleja la importancia relativa de cada una, en el nodo.

Tabla 78. Vector de pesos para el índice ponderado de potencia. (E2)

Nodos	h_1	...	h_e	...	h_o
n_1	pe_{11}	...	pe_{1e}	...	pe_{1o}
...	...	...	...	...	...
n_i	pe_{i1}	...	pe_{ie}	...	pe_{io}
...	...	...	...	...	...
n_m	pe_{m1}	...	pe_{me}	...	pe_{mo}

Elaboración propia

Con h indicando el componente de hardware, donde $e = 1, \dots, o$ (nº de componente de hardware para cada nodo).

Con pe indicando el valor del peso del componente de hardware x , donde $i = 1, \dots, m$ (nº de nodo del sistema distribuido) y donde $e = 1, \dots, o$ (nº del componente de hardware).

Los valores de la potencia de componentes de hardware en la Tabla 77, tomados fila por fila, es decir, respecto a cada nodo, se multiplicarán por el vector de pesos de la Tabla 78 y luego se sumarán los resultados para obtener el índice ponderado. Esto puede verse en la Tabla 79

$$indice_ponderado_{if} = \sum (pot_{ie} * pe_{ie}) \quad (14)$$

Tabla 79. Potencia nodal (E2)

Nodos	Índice ponderado calculado
n_1	$indice_ponderado_{11}$
...	...
n_i	$indice_ponderado_{if}$
...	...
n_m	$indice_ponderado_{mg}$

Elaboración propia

Con $indice_ponderado$ indicando la potencia nodal, donde $i = 1, \dots, m$ (nº de nodo del sistema distribuido) y donde $f = 1, \dots, g$ (índice ponderado de la potencia nodal).

Los valores resultantes del índice ponderado se clasifican en diferentes categorías utilizando etiquetas lingüísticas. Se establecen límites inferiores y superiores para cada categoría a la que se asignó una etiqueta lingüística.

3.4.6. Determinación del orden de migración de procesos y de los nodos destino correspondientes (E2)

Para asignar el orden de migración de procesos se debe evaluar la matriz general de pesos de migración (Tabla 73) y los valores resultantes del índice ponderado de la potencia nodal (Tabla

79). En relación a la matriz general de pesos de migración, es importante normalizar sus valores de modo que la suma de los mismos sea igual a 1. Para lograr esto, se emplea la técnica de normalización proporcional. En este método, cada valor en el conjunto de datos de la matriz se divide por la suma total de los valores, garantizando así que la suma de los valores normalizados resulte igual a 1.

Dado que el modelo considera el menor impacto de transferencia, para favorecer como receptores a los nodos con altas prestaciones, se considera la inversa de los valores del índice ponderado de potencia nodal. Se debe normalizar sus valores de modo que la suma de los mismos sea igual a 1.

En la se Tabla 80 observa la matriz general de impacto (*MGI*), resultado de la sumatoria de los pesos de la matriz general de pesos de migración (Tabla 73) y el índice ponderado de potencia invertido y normalizado de cada nodo.

Tabla 80. Matriz general de impacto de migración (E2)

<i>MGI</i>	<i>n</i> ₁	...	<i>n</i> _k	...	<i>n</i> _m
<i>mg</i> <i>i</i> ₁	<i>imp</i> ¹ ₁₁₁	...	<i>imp</i> ¹ _{11u}	...	<i>imp</i> ¹ _{11m}
...	...	...	...	...	...
<i>mg</i> <i>i</i> _i	<i>imp</i> ^p _{qs1}	...	<i>imp</i> ^p _{qsu}	...	<i>imp</i> ^p _{qsm}
...	...	...	...	...	...
<i>mg</i> <i>i</i> _o	<i>imp</i> ⁿ _{rt1}	...	<i>imp</i> ⁿ _{rtu}	...	<i>imp</i> ⁿ _{rtm}

Elaboración propia

Con *k* en *n* variando de 1, ..., *m*, indicando el nodo compatible con carga baja para cada proceso.

Con *i* en *mg**i* variando de 1, ..., *o*, indicando el proceso correspondiente en la matriz general de impacto.

Con *p* en *imp* (*superíndice*) variando de 1, ..., *n*, indicando el proceso correspondiente al vector de migración de impacto.

Con q,s,u en *imp* (subíndices) variando q de 1, ..., r , indicando el nodo origen, s de 1, ..., t , indicando el número de proceso y u de 1, ..., m , indicando el nodo destino, del vector de migración de impacto.

El primer paso que habrá que realizar a partir de los datos de la matriz general de impacto (Tabla 80) es dividirla en dos matrices, una primaria (primera ronda de asignación) y otra secundaria (segunda ronda de asignación). La diferencia radica en que primero se intentará favorecer a los procesos que tienen recursos en los nodos destinos, es decir, cuya migración implicaría una reducción en el impacto de entrada salida (transferencia) con respecto a aquellos que no tengan recursos en nodos destino; para ello se utiliza la matriz primaria. La segunda matriz se analizará luego de haber evaluado la matriz primaria.

Para la elección de los procesos y los nodos destinos de la matriz primaria y secundaria del escenario 2 (E2) se aplica el mismo mecanismo que para el escenario 1 (E1).

3.4.7. Evaluación de nuevos valores de sobrecarga de los procesos inmigrantes según categoría del nodo receptor (E2)

Cada proceso tiene asociados valores de sobrecarga que indican cuánta carga adicional generaría en un posible nodo receptor en términos de memoria, procesador y entrada/salida (Tabla 81). Estos valores son obtenidos mediante el operador de agregación desarrollado en (La Red Martínez, 2017).

Tabla 81. Valores de sobrecarga en nodo origen (E2)

Nodos			Criterios nodo origen		
1	C_{11}	...	C_{1j}	...	C_{1c}
...	...	...	...	...	...
i	C_{i1}	...	C_{ij}	...	C_{ic}
...	...	...	...	...	...
n	C_{n1}	...	C_{nj}	...	C_{nc}

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

$criterios = \{c_{ij}\}$ con $i = 1, \dots, n$ (nº de nodos en el sistema distribuido) y $j = 1, \dots, c$ (nº máximo de criterios para cada nodo).

Cada valor de cada criterio de sobrecarga del nodo origen será multiplicado por un valor de peso que dependerá de la categoría del nodo receptor (Tabla 82).

Tabla 82. Tabla de conversión de categorías nodales (E2)

Categorías	<i>catd₁</i>	...	<i>catd_d</i>	...	<i>catd_z</i>
<i>cato₁</i>	<i>vpc₁₁</i>	...	<i>vpc_{1d}</i>	...	<i>vpc_{1z}</i>
...	...	...	...	...	...
<i>cato_o</i>	<i>vpc_{o1}</i>	...	<i>vpc_{od}</i>	...	<i>vpc_{oz}</i>
...	...	...	...	...	...
<i>cato_z</i>	<i>vpc_{z1}</i>	...	<i>vpc_{zd}</i>	...	<i>vpc_{zz}</i>

Elaboración propia

Con *cato* indicando las categorías de prestaciones del nodo origen, donde $o = 1, \dots, z$, indicando el número de categorías correspondiente.

Con *catd* indicando las categorías de prestaciones del nodo destino, donde $d = 1, \dots, z$, indicando el número de categorías correspondiente.

Con *vpc* indicando el valor del peso de conversión, donde $o = 1, \dots, z$, indicando el número de categorías correspondiente al nodo origen, donde $d = 1, \dots, z$, indicando el número de categorías correspondiente al nodo destino. Aunque los índices sean diferentes podrían representar a las mismas categorías.

La tabla de conversión (Tabla 82) actúa como un factor de ajuste que pondera los valores de sobrecarga según las características relativas de los nodos involucrados. Si el nodo receptor tiene una mayor capacidad y rendimiento en comparación con el nodo origen, la tabla de conversión reducirá los valores de sobrecarga asociados al proceso. En cambio, si el nodo receptor tiene recursos más limitados o un rendimiento inferior, la tabla de conversión mantendrá o incluso

aumentará los valores de sobrecarga.

Cálculo de los nuevos valores de sobrecarga de cada nodo:

$$\text{valor_criterio_destino}(vcd_{ij}) = (\text{valor_criterio_origen}(c_{ij}) * \text{valor_conversión}(vp_{Cod})) \quad (15)$$

Tabla 83. Valores de sobrecarga en nodo destino (E2)

Nodos		Criterios nodo destino			
1	vcd_{11}	...	vcd_{1j}	...	vcd_{1c}
...	...	...	...	...	...
i	vcd_{i1}	...	vcd_{ij}	...	vcd_{ic}
...	...	...	...	...	...
n	vcd_{n1}	...	vcd_{nj}	...	vcd_{nc}

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Los nuevos valores de sobrecarga de la Tabla 83 son los datos utilizados para las siguientes evaluaciones.

3.4.8. Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema (E2)

El objetivo principal es evaluar y comparar el impacto de la migración de procesos en el estado global del sistema. Para lograrlo, se realizarán dos etapas: en la primera etapa, se contemplará la sobrecarga de las asignaciones de recursos a procesos, sin realizar la migración, mientras que en la segunda etapa, realizando la migración.

La evaluación del impacto de migración de procesos en la carga computacional nodal y el rendimiento global del sistema del escenario 2 (E2) se realiza de la misma manera que el escenario 1 (E1).

3.4.9. Evaluación del consumo de energía de los nodos una vez migrados los procesos (E2)

La carga de trabajo que soporta un nodo puede tener un impacto significativo en su consumo de energía.

Para evaluar el consumo de energía de los nodos del sistema distribuido se utilizará el promedio final de sobrecarga multiplicado por un el valor correspondiente del vector de conversión de consumo de energía.

$$\text{consumo_energía} = pf * vc \quad (16)$$

Donde pf indica el promedio final de sobrecarga de cada nodo y vc el valor del vector de conversión.

Mediante la migración de procesos para el balanceo de carga dentro del sistema distribuido, se busca una optimización de la eficiencia energética con la premisa de que un nodo sobrecargado tiende a tener una mayor utilización de sus recursos, incluyendo la CPU, la memoria RAM y otros componentes del hardware. Esta intensa utilización se traduce en un incremento significativo del consumo de energía, ya que los recursos operan en niveles más elevados de rendimiento.

3.4.10. Métrica propuesta para evaluar el balanceo de carga en el sistema distribuido (E2)

La aplicación de una métrica es esencial para la evaluación de la investigación. Al tener una medida objetiva de los resultados, se pueden comparar los resultados de una investigación con otros de investigaciones similares, lo que permite evaluar su relevancia. Para evaluar la eficacia

del modelo propuesto, es necesario medir los resultados para determinar si se comporta según lo deseado.

Para evaluar el impacto de la migración de procesos y cómo esta afecta el estado global del sistema, se utilizará la desviación estándar. Esta medida considerará la carga final de los nodos y el consumo de energía de los nodos del sistema distribuido para evaluar la variabilidad o dispersión de los datos.

La desviación estándar se aplicará a los promedios finales obtenidos de la aplicación de sobrecarga sin migración y de la aplicación de sobrecarga con migración. Si la desviación estándar de los valores de los promedios finales de sobrecarga sin migración es mayor que la desviación estándar de los mismos con migración, esto indica que hay una mayor variabilidad en los resultados. Por lo tanto, los valores de sobrecarga después de la migración estarán más balanceados porque estarán más cerca de la media.

Para el cálculo de la desviación estándar sin considerar migración se procede de la siguiente manera:

$$\text{desvest_sin_migración (carga_final_sin_migración)} = s$$

$$s = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N-1}} \quad (11)$$

Donde $\{x_1, x_2, \dots, x_n\}$ son los valores de carga final de cada uno de los nodos (sin contemplar migración), $\bar{x}$ es el valor medio de todos estos elementos y N es el número total de nodos en el sistema distribuido.

Para el cálculo de la desviación estándar considerando la migración se procede de manera similar considerando los valores de sobrecarga luego de la migración de procesos, aplicando la fórmula (11).

3.5. Ejemplo

En esta sección se explicará en detalle un ejemplo de aplicación del operador de agregación propuesto. El sistema de procesamiento distribuido, las estructuras de datos, los recursos y los procesos que se ejecutan en los diferentes nodos, grupos, cardinalidades, criterios y categorías para evaluar las diferentes cargas y cálculos necesarios, son los mencionados en (La Red Martínez, 2017), (Agostini & La Red Martínez, 2019), (Agostini et al., 2019) y (Agostini et al., 2018).

Para generar la entrada de datos para el ejemplo desarrollado se ha utilizado un simulador (La Red Martínez et al., 2022) haciendo posible la representación de los procesos, recursos y nodos en un modelo dinámico, simulándose diferentes cargas de trabajo para el sistema distribuido.

El sistema de procesamiento distribuido tiene diez nodos:

$$nodos = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Los procesos que se ejecutan en los nodos son los siguientes: siete procesos en el nodo 1, cinco procesos en el nodo 2 y siete procesos en el nodo 3. Los nodos del 4 al 10 son nodos que se agrega al modelo original mencionados en (La Red Martínez, 2017), (Agostini & La Red Martínez, 2019), (Agostini et al., 2019) y (Agostini et al., 2018).

$procesos = \{p_{ij}\}$ con i indicando el nodo y j indicando el proceso, lo que se puede expresar mediante la Tabla 84.

Tabla 84. Procesos en cada nodo (E2)

Nodos		Procesos					
1	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}
2	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}		
3	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Los recursos compartidos disponibles en los nodos son los siguientes: cuatro recursos en el nodo 1, cuatro en el nodo 2, tres en el nodo 3, tres en el nodo 4, tres en el nodo 5, dos en el nodo 6, cuatro en el nodo 7 y tres en el nodo 8.

$recursos = \{p_{ij}\}$ con i indicando el nodo y j indicando el recurso, lo que se puede expresar mediante la Tabla 85.

Tabla 85. Recursos compartidos disponibles en cada nodo (E2)

Nodos		Recursos			
1	r_{11}	r_{12}	r_{13}	r_{14}	
2	r_{21}	r_{22}	r_{23}	r_{24}	
3	r_{31}	r_{32}	r_{33}		
4	r_{41}	r_{42}	r_{43}		
5	r_{51}	r_{52}	r_{53}		
6	r_{61}	r_{62}			
7	r_{71}	r_{72}	r_{73}	r_{74}	
8	r_{81}	r_{82}	r_{83}		

Las solicitudes de recursos por parte de los procesos se muestran en la Tabla 86.

Tabla 86. Recursos solicitados por los procesos (E2)

Recursos	Procesos
r_{11}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{16}, p_{17}, p_{24}, p_{32}, p_{33}, p_{36}, p_{37}$
r_{12}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{16}, p_{17}, p_{21}, p_{23}, p_{24}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$
r_{13}	$p_{13}, p_{14}, p_{16}, p_{17}, p_{21}, p_{31}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}$
r_{14}	p_{16}
r_{21}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{22}, p_{25}, p_{33}, p_{36}, p_{37}$
r_{22}	$p_{11}, p_{12}, p_{13}, p_{14}, p_{16}, p_{21}, p_{22}, p_{33}, p_{34}, p_{35}, p_{36}$
r_{23}	$p_{11}, p_{21}, p_{24}, p_{32}, p_{33}, p_{34}$
r_{24}	$p_{11}, p_{23}, p_{24}, p_{34}, p_{35}, p_{36}$
r_{31}	$p_{12}, p_{13}, p_{16}, p_{21}, p_{22}, p_{23}, p_{31}, p_{34}, p_{35}, p_{36}$
r_{32}	$p_{13}, p_{15}, p_{23}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$

Recursos	Procesos
r_{33}	$p_{12}, p_{13}, p_{15}, p_{16}, p_{21}, p_{22}, p_{23}, p_{31}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$
r_{41}	p_{14}
r_{42}	p_{14}
r_{43}	p_{14}
r_{51}	p_{15}
r_{52}	p_{15}
r_{53}	p_{15}
r_{61}	p_{15}
r_{62}	p_{15}
r_{71}	p_{31}
r_{72}	p_{31}
r_{73}	p_{35}
r_{74}	p_{35}
r_{81}	p_{36}
r_{82}	p_{36}
r_{83}	p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

A continuación, se describirá cada una de las etapas de cálculo.

3.5.1. Cálculo de la carga computacional actual de los nodos (E2)

En esta etapa del operador de agregación del escenario 2 (E2), los cálculos que se realizan para determinar la carga computacional de los nodos del sistema distribuido son iguales que el escenario (E1). Los valores que se obtienen para las categorías de carga en base a los promedios mostrados en la Tabla 87, se indican en la Tabla 88.

Tabla 87. Valores de los criterios de carga computacional (E2)

Nodos	Valores de los Criterios			Promedio
1	80	90	75	81.67
2	30	30	50	36.67
3	77	49	85	70.33
4	6	7	9	07.33
5	5	5	6	05.33
6	7	9	7	07.67
7	7	10	15	10.67
8	9	6	7	07.33
9	10	9	15	11.33
10	8	7	8	07.67

Nota: Valores de los criterios para medir la carga computacional en cada nodo.

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Tabla 88. Valores de las categorías para medir la carga computacional (E2)

Nodos	Valores de las Categorías
1	Alta
2	Baja
3	Alta
4	Baja
5	Baja
6	Baja
7	Baja
8	Baja
9	Baja
10	Baja

Nota: Valores de las categorías para medir la carga computacional en cada nodo.

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo

Los vectores de valoraciones se aplican para cada requerimiento de un recurso hecho por un proceso, según los criterios establecidos para la determinación de la prioridad que en cada caso y momento fija el nodo en el cual se produce el requerimiento; cada vector de valoraciones de cada requerimiento se multiplica escalarmente por el vector de pesos correspondiente a la categoría de carga actual del nodo para obtener la prioridad según cada criterio y la prioridad nodal otorgada a cada requerimiento; esto se muestra en la Tabla 89 que se continúa en la Tabla 90.

Tabla 89. (E2) Valoraciones asignadas a los criterios para calcular la prioridad (N° proceso, %CPU, %Memoria, %MV, Prioridad Proceso) (E2)

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
$p_{11}r_{11}$	0.7	0.5	0.7	0.9	0.8
$p_{11}r_{12}$	0.8	0.7	0.4	0.5	0.3
$p_{11}r_{21}$	0.3	0.4	0.5	0.2	0.9
$p_{11}r_{22}$	0.5	0.5	0.7	0.4	0.8
$p_{11}r_{23}$	0.5	0.6	0.8	0.8	0.9
$p_{11}r_{24}$	0.3	0.5	0.9	0.2	0.6
$p_{12}r_{11}$	0.4	0.7	0.5	0.9	1.0
$p_{12}r_{12}$	0.2	0.7	0.3	0.7	0.8
$p_{12}r_{21}$	0.7	0.4	0.3	0.7	0.8
$p_{12}r_{22}$	0.9	0.6	0.7	0.7	0.8

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
$p_{12}r_{31}$	0.2	0.5	0.7	0.7	0.3
$p_{12}r_{33}$	0.4	0.5	0.7	0.9	0.3
$p_{13}r_{11}$	0.5	0.7	0.7	0.8	0.6
$p_{13}r_{12}$	0.7	0.8	0.7	0.4	0.9
$p_{13}r_{13}$	0.7	0.6	0.7	0.8	0.9
$p_{13}r_{21}$	0.7	0.4	0.9	0.3	0.5
$p_{13}r_{22}$	0.5	0.9	0.8	0.3	0.5
$p_{13}r_{31}$	0.5	0.7	0.3	0.6	0.8
$p_{13}r_{32}$	0.6	0.9	0.3	0.6	0.4
$p_{13}r_{33}$	0.6	0.2	0.4	0.6	0.9
$p_{14}r_{11}$	0.5	0.7	0.6	0.7	0.7
$p_{14}r_{12}$	0.6	0.7	0.6	0.0	0.8
$p_{14}r_{13}$	0.6	0.5	0.6	3.0	0.8
$p_{14}r_{21}$	0.6	0.3	0.8	0.3	0.4
$p_{14}r_{22}$	0.4	0.8	0.7	0.3	0.4
$p_{14}r_{41}$	0.4	0.6	0.4	0.5	0.7
$p_{14}r_{42}$	0.6	0.8	0.2	0.5	0.8
$p_{14}r_{43}$	0.5	0.3	0.3	0.7	0.7
$p_{15}r_{21}$	0.6	0.8	0.8	0.9	0.7
$p_{15}r_{32}$	0.8	0.9	0.8	0.5	0.8
$p_{15}r_{33}$	0.7	0.3	0.5	0.7	0.6
$p_{15}r_{51}$	0.8	0.4	0.8	0.4	0.6
$p_{15}r_{52}$	0.6	0.8	0.9	0.4	0.6
$p_{15}r_{53}$	0.8	0.7	0.8	0.9	0.8
$p_{15}r_{61}$	0.6	0.8	0.4	0.7	0.8
$p_{15}r_{62}$	0.7	0.8	0.4	0.7	0.5
$p_{16}r_{11}$	0.4	0.7	0.5	0.9	1.0
$p_{16}r_{12}$	0.2	0.7	0.3	0.7	0.8
$p_{16}r_{13}$	0.3	0.6	0.3	0.6	0.7
$p_{16}r_{14}$	0.4	0.7	0.2	0.5	0.6
$p_{16}r_{21}$	0.7	0.4	0.3	0.7	0.8
$p_{16}r_{22}$	0.9	0.6	0.7	0.7	0.8
$p_{16}r_{31}$	0.2	0.5	0.7	0.7	0.3
$p_{16}r_{33}$	0.4	0.5	0.7	0.9	0.3
$p_{17}r_{11}$	0.3	0.6	0.5	0.8	0.6
$p_{17}r_{12}$	0.4	0.6	0.4	0.6	0.8
$p_{17}r_{13}$	0.5	0.7	0.6	0.9	0.7
$p_{21}r_{12}$	0.2	0.1	0.3	0.8	0.7
$p_{21}r_{13}$	0.2	0.4	0.8	0.9	0.7
$p_{21}r_{22}$	0.3	0.5	0.8	0.9	0.5
$p_{21}r_{23}$	0.7	0.5	0.6	0.8	0.5
$p_{21}r_{31}$	0.7	0.5	0.8	0.6	0.9
$p_{21}r_{33}$	0.7	0.4	0.9	0.8	0.4

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
$p_{22}r_{21}$	0.5	0.4	0.7	0.8	0.6
$p_{22}r_{22}$	0.5	0.9	0.6	0.8	0.9
$p_{22}r_{31}$	0.5	0.4	0.5	0.2	0.4
$p_{22}r_{33}$	0.8	0.4	0.3	0.2	0.8
$p_{23}r_{12}$	0.5	0.6	0.8	0.3	0.5
$p_{23}r_{24}$	0.6	0.2	0.1	0.7	0.3
$p_{23}r_{31}$	0.3	0.1	0.4	0.8	0.7
$p_{23}r_{32}$	0.4	0.4	0.8	0.2	0.4
$p_{23}r_{33}$	0.3	0.6	0.8	0.2	0.9
$p_{24}r_{11}$	0.4	0.6	0.7	0.9	0.5
$p_{24}r_{12}$	0.3	0.6	0.7	0.8	0.9
$p_{24}r_{23}$	0.4	0.9	0.4	0.8	0.8
$p_{24}r_{24}$	0.5	0.8	0.7	0.9	0.3
$p_{25}r_{21}$	0.2	0.8	0.8	0.9	0.4
$p_{31}r_{13}$	0.6	0.9	0.6	0.9	0.7
$p_{31}r_{71}$	0.8	0.3	0.9	0.5	0.7
$p_{31}r_{72}$	0.4	0.7	0.7	0.5	0.9
$p_{32}r_{11}$	0.6	0.9	0.8	0.5	0.9
$p_{32}r_{12}$	0.7	0.4	0.6	0.9	0.8
$p_{32}r_{13}$	0.8	0.9	1.0	0.7	0.9
$p_{32}r_{21}$	0.8	0.8	1.0	0.9	0.6
$p_{33}r_{11}$	0.2	0.7	0.9	0.8	0.6
$p_{33}r_{12}$	0.9	1.0	0.9	0.7	0.3
$p_{33}r_{13}$	0.9	0.5	0.7	0.9	0.3
$p_{33}r_{21}$	0.4	0.6	0.7	0.8	0.8
$p_{33}r_{22}$	0.9	0.4	0.7	0.8	0.7
$p_{33}r_{23}$	0.4	0.6	0.9	1.0	0.6
$p_{33}r_{32}$	0.6	0.6	0.7	0.8	0.4
$p_{33}r_{33}$	0.6	1.0	0.7	0.4	0.6
$p_{34}r_{12}$	0.8	1.0	0.3	0.5	0.7
$p_{34}r_{13}$	0.2	1.0	0.8	0.5	0.8
$p_{34}r_{22}$	0.9	0.8	0.6	0.8	0.9
$p_{34}r_{23}$	0.4	0.6	0.7	0.8	0.9
$p_{34}r_{24}$	0.9	0.4	0.7	0.4	0.7
$p_{34}r_{31}$	0.5	0.6	0.7	0.9	0.7
$p_{34}r_{32}$	0.8	0.6	0.9	0.5	0.6
$p_{34}r_{33}$	0.4	0.6	0.8	0.5	0.9
$p_{35}r_{31}$	0.9	0.8	0.7	0.4	0.8
$p_{35}r_{32}$	0.9	0.7	0.8	0.6	0.4
$p_{35}r_{33}$	0.5	0.7	0.6	0.9	0.8
$p_{35}r_{61}$	0.2	0.4	0.7	0.4	0.9
$p_{35}r_{62}$	0.8	0.9	0.7	0.3	0.9
$p_{35}r_{72}$	0.3	0.8	0.9	0.4	0.8

Rec.	N° Proceso	%CPU	%Memoria	%MV	Prioridad Proceso
<i>p35r74</i>	0.5	0.8	0.9	0.4	0.6
<i>p36r11</i>	0.8	0.9	0.6	0.5	0.8
<i>p36r12</i>	0.7	0.9	0.4	0.9	0.8
<i>p36r13</i>	0.9	0.9	0.8	0.7	0.8
<i>p36r21</i>	0.8	0.9	0.5	0.3	0.7
<i>p36r22</i>	0.8	0.2	0.1	0.3	0.8
<i>p36r24</i>	0.4	0.7	0.9	0.3	0.8
<i>p36r81</i>	0.5	0.9	0.9	0.7	0.8
<i>p36r82</i>	0.9	0.8	0.6	0.7	0.8
<i>p36r83</i>	0.2	0.7	0.4	0.8	0.8
<i>p37r11</i>	0.9	0.6	0.8	0.6	0.9
<i>p37r12</i>	0.7	0.9	0.8	0.7	0.5
<i>p37r31</i>	0.9	0.7	0.8	0.6	0.6
<i>p37r42</i>	0.8	0.9	0.5	0.3	0.8
<i>p37r43</i>	0.8	0.4	0.6	0.8	0.8

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

Tabla 90. (E2) Valoraciones asignadas a los criterios para calcular la prioridad (Peso del Recurso, Impacto Transferencia, Sobrecarga Memoria, Sobrecarga Procesador, Sobrecarga E/S) (E2)

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
<i>p11r11</i>	0.3	0.5	0.2	0.3	0.4
<i>p11r12</i>	0.3	0.5	0.7	0.2	0.4
<i>p11r21</i>	0.8	0.6	0.2	0.5	0.7
<i>p11r22</i>	0.4	0.3	0.3	0.5	0.6
<i>p11r23</i>	0.7	0.5	0.9	0.7	0.6
<i>p11r24</i>	0.2	0.2	0.6	0.7	0.4
<i>p12r11</i>	0.7	0.6	0.9	0.8	0.8
<i>p12r12</i>	0.3	0.2	0.3	0.8	0.9
<i>p12r21</i>	0.3	0.3	0.9	0.5	0.2
<i>p12r22</i>	0.9	0.7	0.2	0.5	0.4
<i>p12r31</i>	0.8	0.3	0.2	0.7	0.8
<i>p12r33</i>	0.5	0.9	0.4	0.5	0.8
<i>p13r11</i>	0.5	0.3	0.5	0.7	0.8
<i>p13r12</i>	0.8	0.9	0.2	0.9	0.7
<i>p13r13</i>	0.8	0.9	0.4	0.8	0.7
<i>p13r21</i>	0.4	0.9	0.7	0.2	0.3
<i>p13r22</i>	0.6	0.4	0.4	0.9	0.3
<i>p13r31</i>	0.6	0.4	0.9	0.9	0.5
<i>p13r32</i>	0.6	0.6	0.8	0.7	0.8
<i>p13r33</i>	0.8	0.7	0.8	0.5	0.8
<i>p14r11</i>	0.6	0.2	0.4	0.6	0.7
<i>p14r12</i>	0.7	0.8	0.2	0.8	0.6

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
$p_{14}r_{13}$	0.9	0.8	0.3	0.9	0.6
$p_{14}r_{21}$	0.5	0.8	0.8	0.3	0.2
$p_{14}r_{22}$	0.7	0.5	0.5	0.8	0.2
$p_{14}r_{41}$	0.5	0.3	0.8	0.8	0.6
$p_{14}r_{42}$	0.7	0.6	0.9	0.6	0.7
$p_{14}r_{43}$	0.7	0.5	0.9	0.6	0.7
$p_{15}r_{21}$	0.6	0.4	0.6	0.8	0.7
$p_{15}r_{32}$	0.9	0.8	0.3	0.8	0.8
$p_{15}r_{33}$	0.9	0.8	0.9	0.6	0.9
$p_{15}r_{53}$	0.9	0.8	0.5	0.9	0.8
$p_{15}r_{51}$	0.5	0.8	0.8	0.3	0.4
$p_{15}r_{52}$	0.7	0.5	0.5	0.8	0.4
$p_{15}r_{61}$	0.7	0.5	0.8	0.8	0.6
$p_{15}r_{62}$	0.7	0.7	0.9	0.8	0.9
$p_{16}r_{11}$	1.0	0.6	0.8	0.7	0.7
$p_{16}r_{12}$	1.0	0.2	0.4	0.7	0.8
$p_{16}r_{13}$	0.7	0.3	0.3	0.5	0.4
$p_{16}r_{14}$	0.8	0.4	0.5	0.6	0.7
$p_{16}r_{21}$	0.3	0.3	0.9	0.5	0.2
$p_{16}r_{22}$	0.9	0.7	0.2	0.5	0.4
$p_{16}r_{31}$	0.8	0.3	0.2	0.7	0.8
$p_{16}r_{33}$	0.5	0.9	0.4	0.5	0.8
$p_{17}r_{11}$	0.3	0.5	0.8	0.9	0.9
$p_{17}r_{12}$	0.4	0.3	0.4	0.7	0.8
$p_{17}r_{13}$	0.2	0.3	0.3	0.8	0.6
$p_{21}r_{12}$	0.8	0.7	0.7	0.5	0.8
$p_{21}r_{13}$	0.4	0.3	0.4	0.5	0.2
$p_{21}r_{22}$	0.6	0.2	0.6	0.2	0.4
$p_{21}r_{23}$	0.4	0.8	0.2	0.9	0.4
$p_{21}r_{31}$	0.2	0.8	0.4	0.9	0.9
$p_{21}r_{33}$	0.7	0.7	0.8	0.6	0.6
$p_{22}r_{21}$	0.8	0.8	0.4	0.6	0.9
$p_{22}r_{22}$	0.3	0.6	0.4	0.2	0.1
$p_{22}r_{31}$	0.5	0.7	0.8	0.2	0.9
$p_{22}r_{33}$	0.2	0.7	0.9	0.5	0.8
$p_{23}r_{12}$	0.2	0.7	0.7	0.5	0.5
$p_{23}r_{24}$	0.8	0.7	0.8	0.9	0.4
$p_{23}r_{31}$	0.5	0.3	0.9	0.4	0.6
$p_{23}r_{32}$	0.5	0.6	0.6	0.6	0.6
$p_{23}r_{33}$	0.6	0.8	0.6	0.4	0.2
$p_{24}r_{11}$	0.7	0.9	0.7	0.9	0.5

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
<i>p₂₄r₁₂</i>	0.4	0.5	0.7	0.6	0.9
<i>p₂₄r₂₃</i>	0.4	0.8	0.7	0.6	0.3
<i>p₂₄r₂₄</i>	0.6	0.4	0.4	0.8	0.7
<i>p₂₅r₂₁</i>	0.9	0.7	0.5	0.6	0.8
<i>p₃₁r₁₃</i>	0.7	0.8	0.4	0.9	0.8
<i>p₃₁r₇₁</i>	0.6	0.9	0.3	0.9	0.7
<i>p₃₁r₇₂</i>	0.9	0.5	0.9	0.7	0.7
<i>p₃₂r₁₁</i>	0.2	0.6	0.7	0.3	0.7
<i>p₃₂r₁₂</i>	0.4	0.3	1.0	0.9	0.7
<i>p₃₂r₁₃</i>	0.3	0.5	0.3	0.5	0.9
<i>p₃₂r₂₁</i>	0.6	0.6	0.8	0.4	0.9
<i>p₃₃r₁₁</i>	0.4	0.2	0.9	1.0	0.3
<i>p₃₃r₁₂</i>	0.8	0.5	0.5	0.7	0.3
<i>p₃₃r₁₃</i>	0.7	0.5	0.4	0.5	0.8
<i>p₃₃r₂₁</i>	0.4	0.9	0.4	0.8	0.8
<i>p₃₃r₂₂</i>	0.9	0.5	0.4	0.9	0.7
<i>p₃₃r₂₃</i>	0.3	0.8	0.4	0.7	0.8
<i>p₃₃r₃₂</i>	0.3	0.2	0.5	0.8	0.8
<i>p₃₃r₃₃</i>	0.2	0.2	0.8	0.8	0.9
<i>p₃₄r₁₂</i>	0.5	0.6	0.3	0.8	0.7
<i>p₃₄r₁₃</i>	0.7	0.7	0.6	0.9	0.7
<i>p₃₄r₂₂</i>	0.5	0.6	0.8	0.5	0.6
<i>p₃₄r₂₃</i>	0.3	0.3	0.5	0.9	0.6
<i>p₃₄r₂₄</i>	0.4	0.6	0.5	0.9	0.7
<i>p₃₄r₃₁</i>	0.4	0.5	0.9	0.6	0.7
<i>p₃₄r₃₂</i>	0.4	0.2	0.9	0.3	0.9
<i>p₃₄r₃₃</i>	0.7	0.8	0.9	0.4	0.3
<i>p₃₅r₃₁</i>	0.2	0.5	0.3	0.4	0.8
<i>p₃₅r₃₂</i>	0.7	0.7	0.9	0.4	0.7
<i>p₃₅r₃₃</i>	0.6	0.7	0.5	0.9	0.7
<i>p₃₅r₆₁</i>	0.3	0.7	0.5	0.7	0.9
<i>p₃₅r₆₂</i>	0.9	0.9	0.8	0.7	0.4
<i>p₃₅r₇₂</i>	0.9	0.4	0.6	0.3	0.6
<i>p₃₅r₇₄</i>	0.9	0.5	0.9	0.4	0.7
<i>p₃₆r₁₁</i>	0.7	0.6	0.9	0.9	0.6
<i>p₃₆r₁₂</i>	0.8	0.9	0.9	0.4	0.7
<i>p₃₆r₁₃</i>	0.3	0.5	0.7	0.9	0.7
<i>p₃₆r₂₁</i>	0.5	0.4	0.7	0.9	0.9
<i>p₃₆r₂₂</i>	0.5	0.7	0.7	0.6	0.9
<i>p₃₆r₂₄</i>	0.6	0.2	0.5	0.8	0.9
<i>p₃₆r₈₁</i>	0.7	0.7	0.8	0.8	0.7
<i>p₃₆r₈₂</i>	0.3	0.6	0.5	0.6	0.7

Rec.	Peso del Recurso	Impacto Transf.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
<i>p_{36r83}</i>	0.2	0.4	0.9	0.4	0.5
<i>p_{37r11}</i>	0.4	0.9	0.7	0.9	0.8
<i>p_{37r12}</i>	0.7	0.5	0.8	0.8	0.6
<i>p_{37r31}</i>	0.8	0.5	0.9	0.5	0.4
<i>p_{37r42}</i>	0.5	0.9	0.7	0.4	0.6
<i>p_{37r43}</i>	0.8	0.3	0.6	0.9	0.3

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017).

3.5.2. Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos (E2)

En esta etapa del operador de agregación del escenario 2 (E2), el cálculo para determinar el promedio de prioridad de los procesos se realiza de la misma manera que el escenario 1 (E1). Los procesos que clasifican para evaluación para una posible migración a otro nodo se listan en la Tabla 91.

Tabla 91. Lista de procesos con prioridad de procesos superior al límite (E2)

Nodos	Procesos	Promedio de prioridad de proceso
1	<i>p₁₁</i>	0.725
1	<i>p₁₂</i>	0.667
1	<i>p₁₃</i>	0.687
1	<i>p₁₄</i>	0.662
1	<i>p₁₅</i>	0.675
3	<i>p₃₁</i>	0.767
3	<i>p₃₂</i>	0.800
3	<i>p₃₄</i>	0.775
3	<i>p₃₅</i>	0.743
3	<i>p₃₆</i>	0.789
3	<i>p₃₇</i>	0.720

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

3.5.3. Selección de los posibles nodos destino mediante la evaluación de compatibilidad y la carga (E2)

En esta etapa, se realiza una evaluación para determinar si hay nodos compatibles con carga baja que puedan alojar procesos de otros nodos.

De la Tabla 88 que contiene los valores de la carga computacional, se obtienen los nodos con carga alta y con carga baja. Se evalúan la compatibilidad de los nodos con carga alta en relación a los nodos con carga baja.

nodos con carga alta = {1, 3}

nodos con carga baja = {2, 4, 5, 6, 7, 8, 9, 10}

Para evaluar la compatibilidad de los nodos, se considera la Tabla 92 y Tabla 93 que incluye componentes de hardware y componentes de software.

La fórmula utilizada en este caso es la siguiente:

Compatibilidad (O, D): $O \subseteq D$

Tabla 92. Componentes de hardware (E2)

Nodos	h1	h2	h3
1	X	-	X
2	X	X	X
3	-	X	X
4	X	-	X
5	-	X	X
6	X	-	-
7	-	X	X
8	X	-	X
9	X	X	X
10	-	X	X

Fuente: Elaboración propia.

Tabla 93. Componentes de software (E2)

Nodos	s1	s2	s3
1	-	X	-
2	X	X	-
3	X	-	X
4	-	X	-
5	X	-	X
6	-	-	X
7	X	X	X
8	-	X	-
9	X	X	X
10	X	-	X

Fuente: Elaboración propia.

Se evalúa la compatibilidad del nodo 1 en relación a los otros nodos con carga baja.

Nodo 1 (N1) = {h1, h3, s2}

Nodo 2 (N2) = {h1, h2, h3, s1, s2}

Nodo 4 (N4) = {h1, h3, s2}

Nodo 5 (N5) = {h2, h3, s1, s3}

Nodo 6 (N6) = {h1, s3}

Nodo 7 (N7) = {h2, h3, s1, s2, s3}

Nodo 8 (N8) = {h1, h3, s2}

Nodo 9 (N9) = {h1, h2, h3, s1, s2, s3}

Nodo 10 (N10) = {h2, h3, s1, s3}

Para evaluar la compatibilidad del nodo 1 con los demás nodos, se verifica si el conjunto de características de los otros nodos contiene todas las características del nodo 1. Se realiza la siguiente evaluación:

Nodo 2: $N1 \subseteq N2$

Nodo 4: $N1 \subseteq N4$

Nodo 5: No cumple con la condición.

Nodo 6: No cumple con la condición.

Nodo 7: No cumple con la condición.

Nodo 8: $N1 \subseteq N8$

Nodo 9: $N1 \subseteq N9$

Nodo 10: No cumple con la condición.

Según la evaluación, los nodos 2, 4, 8 y 9 son compatibles con el nodo 1, ya que contienen todas las características del nodo 1 o incluso más. Los nodos 5, 6, 7 y 10 no son compatibles debido a la falta de algunas características requeridas.

Se evalúa la compatibilidad del nodo 3 en relación a los otros nodos con carga baja.

Nodo 3 ($N3$) = {h2 h3, s1, s3}

Nodo 2 ($N2$) = {h1, h2, h3, s1, s2}

Nodo 4 ($N4$) = {h1, h3, s2}

Nodo 5 ($N5$) = {h2, h3, s1, s3}

Nodo 6 ($N6$) = {h1, s3}

Nodo 7 ($N7$) = {h2, h3, s1, s2, s3}

Nodo 8 ($N8$) = {h1, h3, s2}

Nodo 9 ($N9$) = {h1, h2, h3, s1, s2, s3}

Nodo 10 ($N10$) = {h2, h3, s1, s3}

Para evaluar la compatibilidad del nodo 3 con los demás nodos, se verifica si el conjunto de características de los otros nodos contiene todas las características del nodo 3. Se realiza la siguiente evaluación:

Nodo 2: No cumple con la condición

Nodo 4: No cumple con la condición

Nodo 5: $N3 \subseteq N5$

Nodo 6: No cumple con la condición

Nodo 7: $N3 \subseteq N7$

Nodo 8: No cumple con la condición

Nodo 9: $N3 \subseteq N9$

Nodo 10: $N3 \subseteq N10$

Según la evaluación, los nodos 5, 7, 9 y 10 son compatibles con el nodo 3, ya que contienen todas las características del nodo 3 o incluso más. Los nodos 2, 4, 6 y 8 no son compatibles debido a la falta de algunas características requeridas.

Para cada proceso puede haber varios nodos compatibles con carga baja disponible. Teniendo en cuenta el conjunto de nodos compatibles con carga baja se verifican los recursos solicitados por los procesos p_{ij} para determinar el nodo candidato para una posible migración.

3.5.4. Evaluación del peso de los recursos y su posible impacto ante la migración de procesos (E2)

Una vez identificados los nodos con carga alta, sus procesos y los posibles nodos destinos, se realiza la comparación de los recursos locales en relación con los remotos, con el fin de elaborar la lista preliminar de los procesos migrables y sus posibles nodos destinos.

En la Fig. 13 se muestra el vector de migración de pesos donde se almacenan valores del peso de migración que generaría el proceso si se migrara a un determinado nodo destino, habrá un

vector de migración de pesos para cada proceso en relación con el nodo destino evaluado.

Para cada nodo con carga alta se realiza la evaluación de sus procesos para las diferentes situaciones que pueden presentarse según la Fig. 12 y la Tabla 90.

En la Tabla 94 se muestra el vector de migración de proceso para el proceso p_{11} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 0.6 y como peso de recursos remotos en el nodo 2 es de 2.1. Analizando la Fig. 12 en relación con el nodo 2 su valor se graba dada la situación (u), y en relación con los nodos 4, 8 y 9 su valor se graba dada la situación (q).

Tabla 94. Vector de migración del proceso p_{11} (E2)

vmp ¹			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	1	2	-	0.6	-	-
1	1	4	2.7	-	-	-
1	1	5	-	-	-	-
1	1	6	-	-	-	-
1	1	7	-	-	-	-
1	1	8	2.7	-	-	-
1	1	9	2.7	-	-	-
1	1	10	-	-	-	-

Fuente: Elaboración propia.

En la Tabla 95 se muestra el vector de migración de proceso para el proceso p_{12} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 1.0, el peso de recursos remotos está distribuido en dos nodos, en el nodo 2 es de 1.2 y en el nodo 3 es de 1.3. Analizando la Fig. 12 en relación con el nodo 2 su valor se graba dada la situación (u). El proceso p_{12} podría migrar únicamente al nodo 2 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 95. Vector de migración del proceso p_{12} (E2)

vmp ²			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	2	2	-	2.3	-	-
1	2	4	-	-	-	-
1	2	5	-	-	-	-
1	2	6	-	-	-	-
1	2	7	-	-	-	-
1	2	8	-	-	-	-
1	2	9	-	-	-	-
1	2	10	-	-	-	-

Fuente: Elaboración propia.

En la Tabla 96 se muestra el vector de migración de proceso para el proceso p_{13} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 2.1, el peso de recursos remotos está distribuido en dos nodos, en el nodo 2 es de 1.0 y en el nodo 3 es de 2.0. Analizando la Fig. 12 en relación con el nodo 2 su valor se graba dada la situación (u). El proceso p_{13} podría migrar únicamente al nodo 2 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 96. Vector de migración del proceso p_{13} (E2)

vmp ³			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	3	2	-	4.1	-	-
1	3	4	-	-	-	-
1	3	5	-	-	-	-
1	3	6	-	-	-	-
1	3	7	-	-	-	-
1	3	8	-	-	-	-
1	3	9	-	-	-	-
1	3	10	-	-	-	-

Fuente: Elaboración propia.

En la Tabla 97 se muestra el vector de migración de proceso para el proceso p_{14} que tiene los siguientes pesos de recursos asociados: el peso de recursos locales es de 2.2, el peso de recursos remotos está distribuido en dos nodos, en el nodo 2 es de 1.2 y en el nodo 4 es de 1.9. Analizando

la Fig. 12 en relación con los nodos 2 y 4 su valor se graba dada la situación (u). El proceso p_{14} podría migrar únicamente al nodo 2 o nodo 4 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 97. Vector de migración del proceso p_{14} (E2)

vmp ⁴			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	4	2	-	4.1	-	-
1	4	4	-	3.4	-	-
1	4	5	-	-	-	-
1	4	6	-	-	-	-
1	4	7	-	-	-	-
1	4	8	-	-	-	-
1	4	9	-	-	-	-
1	4	10	-	-	-	-

Fuente: Elaboración propia.

En la Tabla 98 se muestra el vector de migración de proceso para el proceso p_{15} que tiene los siguientes pesos de recursos asociados: el peso de recursos remotos está distribuido en cuatro nodos, en el nodo 2 es de 0.6, en el nodo 3 es de 1.8, en el nodo 5 es de 2.1 y en el nodo 6 es de 1.4. Analizando la Fig. 12 se observan en los nodos 2, 8, 9, su valor se graba dada la situación (v), y en el nodo 4, su valor se graba dada la situación (q).

Tabla 98. Vector de migración del proceso p_{15} (E2)

vmp ⁵			Opciones			
q	s	u	(q)	(u)	(v)	(w)
1	5	2	-	-	5.3	-
1	5	4	5.9	-	-	-
1	5	5	-	-	-	-
1	5	6	-	-	-	-
1	5	7	-	-	-	-
1	5	8	-	-	5.9	-
1	5	9	-	-	5.9	-
1	5	10	-	-	-	-

Fuente: Elaboración propia.

El proceso p_{16} se descarta de la posibilidad de migración porque no cumple las condiciones para migrarse a otros nodos ya que hacerlo generaría un alto impacto en el ancho de banda.

En la Tabla 99 se muestra el vector de migración de proceso para el proceso p_{17} , el peso de recursos locales de 0.9, no posee recursos remotos y dado que su peso no supera el límite establecido en la situación (p) de la Fig. 12, cualquiera de los nodos compatibles con carga baja del sistema distribuido es candidato para una posible migración y su valor se graba dada la situación (q).

Tabla 99. Vector de migración del proceso p_{17} (E2)

vmp⁷				Opciones		
q	s	u	(q)	(u)	(v)	(w)
1	7	2	0.9	-	-	-
1	7	4	0.9	-	-	-
1	7	5	-	-	-	-
1	7	6	-	-	-	-
1	7	7	-	-	-	-
1	7	8	0.9	-	-	-
1	7	9	0.9	-	-	-
1	7	10	-	-	-	-

Fuente: Elaboración propia.

En la Tabla 100 se muestra el vector de migración de proceso correspondiente al proceso p_{31} del nodo 3 que es el siguiente con carga alta.

Respecto del proceso p_{31} , su peso de recursos remotos está distribuido en dos nodos, en el nodo 1 es de 0.7 y en el nodo 7 es de 1.5. Analizando la Fig. 12 y teniendo en cuenta que el proceso p_{31} no posee recursos locales, en relación con los nodos 5, 9, 10, su valor se graba dada la situación (w) y en relación con el nodo 7 su valor se graba dada la situación (v).

En la Tabla 101 se muestra el vector de migración de proceso para el proceso p_{32} que tiene los siguientes pesos de recursos asociados: el peso de recursos remotos está distribuido en dos nodos, en el nodo 1 es de 0.9 y en el nodo 2 es de 0.6. Analizando la Fig. 12 y teniendo en cuenta que el

proceso p_{32} no posee recursos locales, en relación con los nodos 5, 7, 9 y 10 su valor se graba dada la situación (w).

Tabla 100. Vector de migración del proceso p_{31} (E2)

vmp ¹			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	1	2	-	-	-	-
3	1	4	-	-	-	-
3	1	5	-	-	-	2.2
3	1	6	-	-	-	-
3	1	7	-	-	0.7	-
3	1	8	-	-	-	-
3	1	9	-	-	-	2.2
3	1	10	-	-	-	2.2

Fuente: Elaboración propia.

Tabla 101. Vector de migración del proceso p_{32} (E2)

vmp ²			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	2	2	-	-	-	-
3	2	4	-	-	-	-
3	2	5	-	-	-	1.5
3	2	6	-	-	-	-
3	2	7	-	-	-	1.5
3	2	8	-	-	-	-
3	2	9	-	-	-	1.5
3	2	10	-	-	-	1.5

Fuente: Elaboración propia.

El proceso p_{34} se descarta de la posibilidad de migración porque no cumple las condiciones para migrarse a otros nodos ya que hacerlo generaría un alto impacto en el ancho de banda.

En la Tabla 102 se muestra el vector de migración de proceso para el proceso p_{35} que tiene los siguientes pesos de recursos asociados: el peso de los recursos locales es de 1.5, el peso de recursos remotos está distribuido en dos nodos, en el nodo 6 es de 1.2 y en el nodo 7 es de 1.8. Analizando la Fig. 12 en relación con el nodo 7, su valor se graba dada la situación (u). El proceso

p_{35} podría migrar únicamente al nodo 7 ya que hacerlo a otro nodo generaría un alto impacto en el ancho de banda.

Tabla 102. Vector de migración del proceso p_{35} (E2)

vmp ⁵			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	5	2	-	-	-	-
3	5	4	-	-	-	-
3	5	5	-	-	-	-
3	5	6	-	-	-	-
3	5	7	-	2.7	-	-
3	5	8	-	-	-	-
3	5	9	-	-	-	-
3	5	10	-	-	-	-

Fuente: Elaboración propia.

En la Tabla 103 se muestra el vector de migración de proceso para proceso p_{36} que tiene los siguientes pesos de recursos asociados: el peso de recursos remotos está distribuido en tres nodos, en el nodo 1 es de 1.8, en el nodo 2 es de 1.6 y en el nodo 8 es de 1.8. Analizando la Fig. 12 en relación a los nodos 5, 7, 9 y 10, su valor se graba dada la situación (w).

Tabla 103. Vector de migración del proceso p_{36} (E2)

vmp ⁶			Opciones			
q	s	u	(q)	(u)	(v)	(w)
3	6	2	-	-	-	-
3	6	4	-	-	-	-
3	6	5	-	-	-	4.6
3	6	6	-	-	-	-
3	6	7	-	-	-	4.6
3	6	8	-	-	-	-
3	6	9	-	-	-	4.6
3	6	10	-	-	-	4.6

Fuente: Elaboración propia.

En la Tabla 104 se muestra el vector de migración de proceso para el proceso p_{37} que tiene los siguientes pesos de recursos asociados, el peso de recursos locales es de 0.8, el peso de recursos

remotos está distribuido en dos nodos, en el nodo 1 es de 1.1 y en el nodo 4 es de 1.3. Analizando la Fig. 12 en relación con los nodos 5, 7, 9 y 10, su valor se graba dada la situación (q).

Una vez evaluadas todas las situaciones de cada proceso en relación con cada nodo destino posible, se genera la matriz general de pesos de migración (*MGPM*) que se observa en la Tabla 105. Las celdas verdes indican que el proceso tiene asignado algún recurso en ese nodo, las celdas en rosa que no lo tiene.

Tabla 104. Vector de migración del proceso p_{37} (E2)

q	vmp⁷		Opciones			
	s	u	(q)	(u)	(v)	(w)
3	7	2	-	-	-	-
3	7	4	-	-	-	-
3	7	5	3.2	-	-	-
3	7	6	-	-	-	-
3	7	7	3.2	-	-	-
3	7	8	-	-	-	-
3	7	9	3.2	-	-	-
3	7	10	3.2	-	-	-

Fuente: Elaboración propia.

Tabla 105. Matriz general de pesos de migración (*MGPM*) (E2)

Procesos	2	4	5	Nodos 6	7	8	9	10
p_{11}	0.6	2.7	-	-	-	2.7	2.7	-
p_{12}	2.3	-	-	-	-	-	-	-
p_{13}	4.1	-	-	-	-	-	-	-
p_{14}	4.1	3.4	-	-	-	-	-	-
p_{15}	5.3	5.9	-	-	-	5.9	5.9	-
p_{17}	0.9	0.9	-	-	-	0.9	0.9	-
p_{31}	-	-	2.2	-	0.7	-	2.2	2.2
p_{32}	-	-	1.5	-	1.5	-	1.5	1.5
p_{35}	-	-	-	-	2.7	-	-	-
p_{36}	-	-	4.6	-	4.6	-	4.6	4.6
p_{37}	-	-	3.2	-	3.2	-	3.2	3.2

Fuente: Elaboración propia.

3.5.5. Establecimiento de las categorías nodales según sus prestaciones (E2)

Para clasificar los nodos en diferentes categorías, se tendrán en cuenta los componentes de hardware (Tabla 106) y las características específicas de cada componente (por ejemplo, para el componente procesador: número de núcleos, mflops (millones de instrucciones de punto flotante por segundo), cache, etc.) de la Tabla 106. Para el ejemplo se evaluará los componentes CPU (procesador), memoria RAM, y HDD (disco duro).

Tabla 106. Componentes de hardware (E2)

Nodos	CPU	RAM	HDD
1	X	-	X
2	X	X	X
3	-	X	X
4	X	-	X
5	-	X	X
6	X	-	-
7	-	X	X
8	X	-	X
9	X	X	X
10	-	X	X

Fuente: Elaboración propia.

De cada componente evaluado, se analizarán las características específicas de cada uno.

Tabla 107. Valores de atributos del componente CPU (E2)

Nodos	Nº de núcleos	MFLOPS	Arq.	MB	W
1	8	0.01	32	12	95
2	10	0.012	64	16	110
3	8	0.009	64	10	90
4	6	0.008	32	8	80
5	4	0.006	32	6	70
6	4	0.0045	32	4	60
7	2	0.003	32	2	50
8	4	0.0075	32	6	75
9	6	0.01	64	8	85
10	2	0.0025	32	4	40

Fuente: Elaboración propia.

En el caso del componente CPU, se analizarán los siguientes atributos: N° de núcleos, MFLOPS (millones de instrucciones de punto flotante por segundo), Arquitectura (Arq.), Caché (MBytes), Consumo de energía en watts (W). Los valores de cada atributo del CPU mostrado en la Tabla 107 se multiplicarán por su vector de peso que se detalla a continuación en la Tabla 108.

Tabla 108. Vector de peso CPU (E2)

N° de núcleos	MFLOPS	Arq.	MB	W
0.3	0.2	0.2	0.2	0.1

Fuente: Elaboración propia.

Se realiza la sumatoria de los valores de cada característica del componente para determinar un indicador de la potencia del hardware evaluado (Tabla 109).

Tabla 109. Potencia del componente CPU (E2)

Nodos	N° de núcleos	MFLOPS	Arq.	MB	W	Potencia CPU
1	2.4	0.002	6.4	2.4	9.5	20.702
2	3	0.0024	12.8	3.2	11	30.002
3	2.4	0.0018	12.8	2	9	26.202
4	1.8	0.0016	6.4	1.6	8	17.802
5	1.2	0.0012	6.4	1.2	7	15.801
6	1.2	0.0009	6.4	0.8	6	14.401
7	0.6	0.0006	6.4	0.4	5	12.401
8	1.2	0.0015	6.4	1.2	7.5	16.302
9	1.8	0.002	12.8	1.6	8.5	24.702
10	0.6	0.0005	6.4	0.8	4	11.801

Fuente: Elaboración propia.

En el caso del componente memoria RAM, se analizarán los siguientes atributos: Capacidad (GBytes), Velocidad del reloj (MHz), Ancho de banda (GB/s), Tiempo de acceso (ns), Consumo de energía en watts (W). Los valores de cada atributo de la memoria RAM mostrado en la Tabla 110 se multiplicarán por su vector de peso que se detalla a continuación en la Tabla 111.

Tabla 110. Valores de atributos del componente memoria RAM (E2)

Nodos	GB	MHz	GB/s	ns	W
1	16	3000	40	12	0.8
2	64	3200	50	10	1.2
3	128	3600	60	9	1.5
4	32	3200	45	11	1
5	64	3400	55	10	1.2
6	16	2800	35	13	0.7
7	8	2666	30	14	0.6
8	32	2800	35	11	0.9
9	256	4000	70	8	1.8
10	4	2666	32	15	0.5

Fuente: Elaboración propia.

Tabla 111. Vector de peso memoria RAM (E2)

GB	MHz	GB/s	ns	W
0.3	0.2	0.2	0.2	0.1

Se realiza la sumatoria de los valores de cada característica del componente para determinar un indicador de la potencia del hardware evaluado (Tabla 112).

Tabla 112. Potencia del componente memoria RAM (E2)

Nodos	GB	MHz	GB/s	ns	W	Potencia RAM
1	4.8	600	8	2.4	0.08	615.280
2	19.2	640	10	2	0.12	671.320
3	38.4	720	12	1.8	0.15	772.350
4	9.6	640	9	2.2	0.1	660.900
5	19.2	680	11	2	0.12	712.320
6	4.8	560	7	2.6	0.07	574.470
7	2.4	533.2	6	2.8	0.06	544.460
8	9.6	560	7	2.2	0.09	578.890
9	76.8	800	14	1.6	0.18	892.580
10	1.2	533.2	6.4	3	0.05	543.850

Fuente: Elaboración propia.

En el caso del componente disco duro, se analizarán los siguientes atributos: Capacidad (GB), Velocidad de rotación (RPM), Velocidad de transferencia (MB/s), Caché (MB), Consumo

de energía en watts (W). Los valores de cada atributo del disco duro mostrado en la Tabla 113 se multiplicarán por su vector de peso que se detalla a continuación en la Tabla 114.

Tabla 113. Valores de atributos del componente disco duro (E2)

Nodos	GB	RPM	MB/s	Mb	W
1	2000	5400	150	128	6
2	4000	7200	250	256	7.5
3	8000	7200	300	512	8.2
4	4000	7200	200	256	6.5
5	8000	7200	250	512	7
6	500	5400	100	64	4.5
7	1000	5400	120	128	5
8	4000	5400	200	256	8.3
9	12000	10000	350	1000	9
10	1000	7200	150	128	5.5

Fuente: Elaboración propia.

Tabla 114. Vector de peso Disco Duro (E2)

GB	RPM	MB/s	Mb	W
0.3	0.2	0.2	0.2	0.1

Fuente: Elaboración propia.

Se realiza la sumatoria de los valores de cada característica del componente para determinar un indicador de la potencia del hardware evaluado (Tabla 115).

Tabla 115. Potencia del componente memoria disco duro (E2)

Nodos	GB	RPM	MB/s	Mb	W	Potencia Disco duro
1	600	1080	30	25.6	0.6	1736.200
2	1200	1440	50	51.2	0.75	2741.950
3	2400	1440	60	102.4	0.82	4003.220
4	1200	1440	40	51.2	0.65	2731.850
5	2400	1440	50	102.4	0.7	3993.100
6	150	1080	20	12.8	0.45	1263.250
7	300	1080	24	25.6	0.5	1430.100
8	1200	1080	40	51.2	0.83	2372.030
9	3600	2000	70	200	0.9	5870.900
10	300	1440	30	25.6	0.55	1796.150

Fuente: Elaboración propia.

A continuación, en la Tabla 116 se realiza el proceso de sustituir los valores de la Tabla 106, se reemplaza cada X por el valor de la potencia calculada para cada componente de hardware.

Tabla 116. Potencia de componentes de hardware. (E2)

Nodos	CPU	RAM	HDD
1	20.702	-	615.280
2	30.002	2741.950	671.320
3	-	4003.220	772.350
4	17.802	-	660.900
5	-	3993.100	712.320
6	14.401	-	-
7	-	1430.100	544.460
8	16.302	-	578.890
9	24.702	5870.900	892.580
10	-	1796.150	543.850

Fuente: Elaboración propia.

Los valores de la potencia de componentes de hardware en la Tabla 116, tomados fila por fila, es decir, respecto a cada nodo, se multiplicarán por el vector de pesos de la Tabla 117 y luego se sumarán los resultados para obtener el índice ponderado de la potencia de cada nodo. Esto puede verse en la Tabla 118.

Tabla 117. Vector de pesos para el índice ponderado de potencia (E2)

CPU	RAM	HDD
0.5	0.2	0.3

Fuente: Elaboración propia.

Tabla 118. Potencia nodal (E2)

Nodos	Índice ponderado de cada nodo
1	194.935
2	764.787
3	1032.349
4	207.171
5	1012.316
6	7.200
7	449.358
8	181.818
9	1454.305
10	522.385

Fuente: Elaboración propia.

Los valores resultantes del índice ponderado se clasifican en diferentes categorías utilizando etiquetas lingüísticas. Se establecen límites inferiores y superiores para cada categoría a la que se asignó una etiqueta lingüística.

Para el ejemplo se tomarán tres categorías, sub estándar, estándar y supra estándar clasificados según las potencias de cada nodo del sistema distribuido. Según los valores de la Tabla 118 el valor mínimo es 7.200 y el valor máximo es 1454.305, con estos valores se ha establecido una clasificación en tres categorías siendo los valores límites entre categorías los siguientes: 489.569, 971.937.

La clasificación queda de la siguiente manera (Tabla 119):

Tabla 119. Clasificación de nodos según sus prestaciones (E2)

Nodos	Índice ponderado de cada nodo	Clasificación de Prestaciones
1	194.935	Sub estándar
2	764.787	Estándar
3	1032.349	Supra estándar
4	207.171	Sub estándar
5	1012.316	Supra estándar
6	7.200	Sub estándar
7	449.358	Sub estándar
8	181.818	Sub estándar
9	1454.305	Supra estándar
10	522.385	Estándar

Fuente: Elaboración propia.

3.5.6. Determinación del orden de migración de procesos y de los nodos destino correspondientes (E2)

Para asignar el orden de migración de procesos se debe evaluar la matriz general de pesos de migración (Tabla 105) y los valores resultantes del índice ponderado de la potencia nodal (Tabla 118).

En relación a la matriz general de pesos de migración, es importante normalizar sus valores de modo que la suma de los mismos sea igual a 1 (Tabla 120).

Tabla 120. Matriz general de pesos de migración normalizado (MGPMN) (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{11}	0.006	0.028	-	-	-	0.028	0.028	-
p_{12}	0.024	-	-	-	-	-	-	-
p_{13}	0.043	-	-	-	-	-	-	-
p_{14}	0.043	0.035	-	-	-	-	-	-
p_{15}	0.055	0.061	-	-	-	0.061	0.061	-
p_{17}	0.009	0.009	-	-	-	0.009	0.009	-
p_{31}	-	-	0.023	-	0.007	-	0.023	0.023
p_{32}	-	-	0.016	-	0.016	-	0.016	0.016
p_{35}	-	-	-	-	0.028	-	-	-
p_{36}	-	-	0.048	-	0.048	-	0.048	0.048
p_{37}	-	-	0.033	-	0.033	-	0.033	0.033

Fuente: Elaboración propia.

Dado que el modelo considera el menor impacto de transferencia, para favorecer como receptores a los nodos con altas prestaciones, se considera la inversa de los valores del índice ponderado de potencia nodal (Tabla 118). Se debe normalizar sus valores de modo que la suma de los mismos sea igual a 1 (Tabla 121).

Tabla 121. Potencia nodal invertido y normalizado (E2)

Nodos	Índice ponderado de cada nodo	Inversa del índice ponderado	Índice normalizado
1	194.935	0.005	0.032
2	764.787	0.001	0.008
3	1032.349	0.001	0.006
4	207.171	0.005	0.030
5	1012.316	0.001	0.006
6	7.200	0.139	0.855
7	449.358	0.002	0.014
8	181.818	0.006	0.034
9	1454.305	0.001	0.004
10	522.385	0.002	0.012

Fuente: Elaboración propia.

En la Tabla 122 se observa la matriz general de impacto de migración (*MGI*), resultado de la sumatoria de los pesos de la matriz general de pesos de migración normalizado (Tabla 120) y el índice ponderado de potencia invertido y normalizado de cada nodo (Tabla 121).

Tabla 122. Matriz general de impacto de migración (*MGI*) (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{11}	0.014	0.058	-	-	-	0.062	0.032	-
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-
p_{14}	0.643	0.065	-	-	-	-	-	-
p_{15}	0.655	0.091	-	-	-	0.095	0.065	-
p_{17}	0.609	0.039	-	-	-	0.043	0.014	-
p_{31}	-	-	0.029	-	0.021	-	0.027	0.035
p_{32}	-	-	0.022	-	0.029	-	0.020	0.027
p_{35}	-	-	-	-	0.042	-	-	-
p_{36}	-	-	0.054	-	0.061	-	0.052	0.060
p_{37}	-	-	0.039	-	0.047	-	0.037	0.045

Fuente: Elaboración propia.

En la matriz general de impacto de migración (Tabla 122) se almacenan datos de procesos y el impacto que se generarían al ser migrado cada proceso. Las celdas verdes indican que el proceso tiene asignado algún recurso en ese nodo, las celdas en rosa que no lo tiene.

El primer paso que habrá que realizar a partir de los datos de la matriz general de impacto es dividirla en dos matrices, una primaria (Tabla 123) (primera ronda de asignación) y otra secundaria (Tabla 124) (segunda ronda de asignación). La diferencia radica en que primero se intentará favorecer a los procesos que tienen recursos en los nodos destinos, es decir, cuya migración implicaría una reducción en el impacto de entrada salida (transferencia) con respecto a aquellos que no tengan recursos en nodos destino; para ello se utiliza la matriz primaria. La segunda matriz se analizará luego de haber evaluado la matriz primaria.

Para el modelo propuesto, cada nodo con una carga alta podrá migrar hasta el 45% de sus procesos, con el fin de evitar migrar todos sus procesos y sobrecargar otros nodos.

Tabla 123. Matriz primaria (MP). Primera ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{11}	0.014	0.058	-	-	-	0.062	0.032	-
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-
p_{14}	0.643	0.065	-	-	-	-	-	-
p_{15}	0.655	0.091	-	-	-	0.095	0.065	-
p_{31}	-	-	0.029	-	0.021	-	0.027	0.035
p_{35}	-	-	-	-	0.042	-	-	-

Fuente: Elaboración propia.

Tabla 124. Matriz secundaria (MS). Primera ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{17}	0.609	0.039	-	-	-	0.043	0.014	-
p_{32}	-	-	0.022	-	0.029	-	0.020	0.027
p_{36}	-	-	0.054	-	0.061	-	0.052	0.060
p_{37}	-	-	0.039	-	0.047	-	0.037	0.045

Fuente: Elaboración propia.

El orden de asignación de los procesos a nodos destino de la matriz primaria se indica mediante la Función de Asignación para la Migración en Sistemas Distribuidos (FAMSD) primaria. El primer proceso asignado es aquel que tiene menor peso de transferencia según la Tabla 123. En este caso el proceso p_{11} es el que implica menor impacto de migración, que corresponde migrar al nodo 2. Lo anterior define la primera ronda de asignación (Tabla 125); se continua la evaluación mientras existan procesos a evaluar.

Tabla 125. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Primera ronda (E2)

Impacto de migración	Proceso seleccionado	Nodo destino
0.014	p_{11}	2

Fuente: Elaboración propia.

El siguiente paso es eliminar el proceso p_{11} y el nodo 2 de la matriz primaria porque fueron asignados en la primera ronda. Aquellos procesos que tienen recursos destino en el nodo 2 (ya eliminado como nodo receptor de la Tabla 123) pero tienen como opciones otros nodos destinos para una posible migración (que no poseen recursos destino de esos procesos) pasan a formar parte de la matriz secundaria (Ver Tabla 126 y Tabla 127). Los demás procesos cuya única opción de migración es el nodo 2 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 128).

Tabla 126. Matriz primaria (MP). Segunda ronda (E2)

Procesos	Nodos						
	4	5	6	7	8	9	10
p_{12}	-	-	-	-	-	-	-
p_{13}	-	-	-	-	-	-	-
p_{14}	0.065	-	-	-	-	-	-
p_{31}	-	0.029	-	0.021	-	0.027	0.035
p_{35}	-	-	-	0.042	-	-	-

Fuente: Elaboración propia.

Tabla 127. Matriz secundaria (MS). Segunda ronda (E2)

Procesos	Nodos						
	4	5	6	7	8	9	10
p_{17}	0.039	-	-	-	0.043	0.014	-
p_{15}	0.091	-	-	-	0.095	0.065	-
p_{32}	-	0.022	-	0.029	-	0.020	0.027
p_{36}	-	0.054	-	0.061	-	0.052	0.060
p_{37}	-	0.039	-	0.047	-	0.037	0.045

Fuente: Elaboración propia.

Tabla 128. Matriz de procesos no migrables. Segunda ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-

Fuente: Elaboración propia.

Se continúa con la matriz primaria hasta agotar opciones de procesos mientras los nodos no alcancen el porcentaje de migración fijado.

El segundo proceso de la matriz primaria de la segunda ronda con menor peso de transferencia es el proceso p_{31} , que corresponde migrar al nodo 7.

Lo anterior define la segunda ronda de asignación (Ver Tabla 129), se continua la evaluación mientras existan procesos a evaluar.

Tabla 129. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Segunda ronda (E2)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.014	p_{11}	2
0.021	p_{31}	7

Fuente: Elaboración propia.

El siguiente paso es eliminar el proceso p_{31} y el nodo 7 de la matriz primaria porque fueron asignados en la segunda ronda. Aquellos procesos que tenían recursos destino en el nodo 7 (ya eliminado como nodo receptor de la Tabla 123) pero tienen como opciones otros nodos destinos para una posible migración (que no poseen recursos destino de esos procesos) pasan a formar parte de la matriz secundaria (Ver Tabla 130 y Tabla 131).

Los demás procesos cuya única opción de migración es el nodo 7 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 132).

Tabla 130. Matriz primaria (MP). Tercera ronda (E2)

Procesos	Nodos					
	4	5	6	8	9	10
p_{14}	0.065	-	-	-	-	-

Fuente: Elaboración propia.

Se continúa con la matriz primaria hasta agotar opciones de procesos mientras que los nodos no alcancen el porcentaje de migración establecido.

El tercer proceso de la matriz primaria de la tercera ronda con menor peso de transferencia es el proceso p_{14} , que corresponde migrar al nodo 4.

Lo anterior define la tercera ronda de asignación (Ver Tabla 133), se continúa la evaluación mientras existan procesos a evaluar.

Tabla 131. Matriz secundaria (MS). Tercera ronda (E2)

Procesos	Nodos					
	4	5	6	8	9	10
p_{17}	0.039	-	-	0.043	0.014	-
p_{15}	0.091	-	-	0.095	0.065	-
p_{32}	-	0.022	-	-	0.020	0.027
p_{36}	-	0.054	-	-	0.052	0.060
p_{37}	-	0.039	-	-	0.037	0.045

Fuente: Elaboración propia.

Tabla 132. Matriz de procesos no migrables. Tercera ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-
p_{35}	-	-	-	-	0.042	-	-	-

Fuente: Elaboración propia.

Tabla 133. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) primaria. Tercera ronda (E2)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.014	p_{11}	2
0.021	p_{31}	7
0.065	p_{14}	4

Fuente: Elaboración propia.

El siguiente paso es eliminar el proceso p_{14} y el nodo 4 de la matriz primaria porque fueron asignados en la tercera ronda. Aquellos procesos que tenían recursos destino en el nodo 4 (ya eliminado como nodo receptor de la Tabla 130) pero tienen como opciones otros nodos destinos para una posible migración (que no poseen recursos destino de esos procesos) pasan a formar parte de la matriz secundaria (Tabla 134). Los demás procesos cuya única opción de migración es el nodo 4 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 135).

Tabla 134. Matriz secundaria (MS). Cuarta ronda (E2)

Procesos	Nodos				
	5	6	8	9	10
p_{17}	-	-	0.043	0.014	-
p_{15}	-	-	0.095	0.065	-
p_{32}	0.022	-	-	0.020	0.027
p_{36}	0.054	-	-	0.052	0.060
p_{37}	0.039	-	-	0.037	0.045

Fuente: Elaboración propia.

Tabla 135. Matriz de procesos no migrables. Cuarta ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-
p_{35}	-	-	-	-	0.042	-	-	-

Fuente: Elaboración propia.

Teniendo en cuenta que en la matriz primaria se agotaron las opciones de migración de procesos, sólo se evaluarán las opciones de la matriz secundaria.

El cuarto proceso de la matriz secundaria de la cuarta ronda con menor peso de transferencia es el proceso p_{17} , que corresponde migrar al nodo 9.

Lo anterior define la cuarta ronda de asignación (Ver Tabla 136); se continua la evaluación mientras existan procesos a evaluar.

Tabla 136. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) secundaria. Cuarta ronda (E2)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.014	p_{17}	9

Fuente: Elaboración propia.

El siguiente paso es eliminar el proceso p_{17} y el nodo 9 de la matriz secundaria porque fueron asignados en la cuarta ronda. Aquellos procesos que tenían recursos destino en el nodo 6 (ya eliminado como nodo receptor de la Tabla 130) pero tienen como opciones otros nodos destinos para una posible migración (que no poseen recursos destino de esos procesos) pasan a

formar parte de la matriz secundaria (Ver Tabla 137). Los demás procesos cuya única opción de migración es el nodo 9 se descartan y pasan a formar parte de la matriz de procesos no migrados (Ver Tabla 138).

Tabla 137. Matriz secundaria (MS). Quinta ronda (E2)

Procesos	Nodos			
	5	6	8	10
p_{15}	-	-	0.095	-
p_{32}	0.022	-	-	0.027
p_{36}	0.054	-	-	0.060
p_{37}	0.039	-	-	0.045

Fuente: Elaboración propia.

Tabla 138. Matriz de procesos no migrables. Quinta ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-
p_{35}	-	-	-	-	0.042	-	-	-

Fuente: Elaboración propia.

Se continúa con la matriz secundaria hasta agotar opciones de procesos mientras que los nodos no alcancen el porcentaje de migración establecido.

El quinto proceso de la matriz secundaria de la quinta ronda con menor peso de transferencia es el proceso p_{32} , que corresponde migrar al nodo 5.

Lo anterior define la tercera ronda de asignación (Ver Tabla 139); se continua la evaluación mientras existan procesos a evaluar.

Tabla 139. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) secundaria. Quinta ronda (E2)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.014	p_{17}	9
0.022	p_{32}	5

Fuente: Elaboración propia.

El siguiente paso es eliminar el proceso p_{32} y el nodo 5 de la matriz secundaria porque fueron asignados en la quinta ronda. Aquellos procesos que tenían recursos destino en el nodo 5

(ya eliminado como nodo receptor de la Tabla 137) pero tienen como opciones otros nodos destinos para una posible migración (que no poseen recursos destino de esos procesos) pasan a formar parte de la matriz secundaria (Ver Tabla 140). Los demás procesos cuya única opción de migración es el nodo 5 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 141).

Tabla 140. Matriz secundaria (MS). Sexta ronda (E2)

Procesos	Nodos	
	8	10
p_{15}	0.095	-
p_{36}	-	0.060
p_{37}	-	0.045

Fuente: Elaboración propia.

Tabla 141. Matriz de procesos no migrables. Sexta ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-
p_{35}	-	-	-	-	0.042	-	-	-

Fuente: Elaboración propia.

Se continúa con la matriz secundaria hasta agotar opciones de procesos mientras que los nodos no alcancen el porcentaje de migración establecido. El sexto proceso de la matriz secundaria de la sexta ronda con menor peso de transferencia es el proceso p_{37} , que corresponde migrar al nodo 10.

Lo anterior define la sexta ronda de asignación (Ver Tabla 142); se continua la evaluación mientras existan procesos a evaluar.

Tabla 142. Función de Asignación para Migración en Sistemas Distribuidos (FAMSD) secundaria. Sexta ronda (E2)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.014	p_{17}	9
0.022	p_{32}	5
0.045	p_{37}	10

Fuente: Elaboración propia.

El siguiente paso es eliminar el proceso p_{37} y el nodo 10 de la matriz secundaria porque fueron asignados en la sexta ronda. Aquellos procesos que tenían recursos destino en el nodo 10 (ya eliminado como nodo receptor de la Tabla 140) pero tienen como opciones otros nodos destinos para una posible migración (que no poseen recursos destino de esos procesos) pasan a formar parte de la matriz secundaria (Ver Tabla 143). Los demás procesos cuya única opción de migración es el nodo 10 se descartan y pasan a formar parte de la matriz de procesos no migrables (Ver Tabla 144).

Tabla 143. Matriz secundaria (MS). Séptima ronda (E2)

Procesos	Nodo 8
p_{15}	0.095

Fuente: Elaboración propia.

Tabla 144. Matriz de procesos no migrables. Séptima ronda (E2)

Procesos	Nodos							
	2	4	5	6	7	8	9	10
p_{12}	0.624	-	-	-	-	-	-	-
p_{13}	0.643	-	-	-	-	-	-	-
p_{35}	-	-	-	-	0.042	-	-	-
p_{36}	-	-	0.054	-	0.061	-	0.052	0.060

Fuente: Elaboración propia.

Se continúa con la matriz secundaria hasta agotar opciones de procesos mientras que los nodos no alcancen el porcentaje de migración establecido.

El séptimo proceso de la matriz secundaria de la séptima ronda con menor peso de transferencia es el proceso p_{15} , que corresponde migrar al nodo 8.

Lo anterior define la séptima ronda de asignación (Ver Tabla 145); se continua la evaluación mientras existan procesos a evaluar.

Tabla 145. Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) secundaria. Séptima ronda (E2)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.014	p_{17}	9
0.022	p_{32}	5
0.045	p_{37}	10
0.095	p_{15}	8

Fuente: Elaboración propia.

Al finalizar la evaluación de los nodos 3 y 1 y habiéndose agotado opciones de migración o alcanzado el porcentaje máximo de migración de procesos establecido, se da por finalizado el presente macrociclo de evaluación.

Seguidamente se procede a la concatenación de la Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) primaria y la Función de Asignación para Migración en Sistemas Distribuidos (*FAMSD*) secundaria. Esto da lugar a la Función de Asignación para Migración en Sistemas Distribuidos Concatenada (*FAMSDC*) que se muestra en la Tabla 146.

Tabla 146. Función de Asignación para Migración en Sistemas Distribuidos concatenada (*FAMSDC*) (E2)

Peso de transferencia del proceso	Proceso seleccionado	Nodo destino
0.014	p_{11}	2
0.021	p_{31}	7
0.065	p_{14}	4
0.014	p_{17}	9
0.022	p_{32}	5
0.045	p_{37}	10
0.095	p_{15}	8

Fuente: Elaboración propia.

3.5.7. Evaluación de nuevos valores de sobrecarga de los procesos inmigrantes según categoría del nodo receptor (E2)

Cada proceso tiene asociados valores de sobrecarga que indican cuánta carga adicional generaría en un posible nodo receptor en términos de memoria, procesador y entrada/salida (Tabla 147). Estos valores son obtenidos mediante el operador de agregación desarrollado en (La Red

Martínez, 2017).

Tabla 147. Valores de sobrecarga en nodo origen (E2)

Rec.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
$p_{11}r_{11}$	0.2	0.3	0.4
$p_{11}r_{12}$	0.7	0.2	0.4
$p_{11}r_{21}$	0.2	0.5	0.7
$p_{11}r_{22}$	0.3	0.5	0.6
$p_{11}r_{23}$	0.9	0.7	0.6
$p_{11}r_{24}$	0.6	0.7	0.4
$p_{12}r_{11}$	0.9	0.8	0.8
$p_{12}r_{12}$	0.3	0.8	0.9
$p_{12}r_{21}$	0.9	0.5	0.2
$p_{12}r_{22}$	0.2	0.5	0.4
$p_{12}r_{31}$	0.2	0.7	0.8
$p_{12}r_{33}$	0.4	0.5	0.8
$p_{13}r_{11}$	0.5	0.7	0.8
$p_{13}r_{12}$	0.2	0.9	0.7
$p_{13}r_{13}$	0.4	0.8	0.7
$p_{13}r_{21}$	0.7	0.2	0.3
$p_{13}r_{22}$	0.4	0.9	0.3
$p_{13}r_{31}$	0.9	0.9	0.5
$p_{13}r_{32}$	0.8	0.7	0.8
$p_{13}r_{33}$	0.8	0.5	0.8
$p_{14}r_{11}$	0.4	0.6	0.7
$p_{14}r_{12}$	0.2	0.8	0.6
$p_{14}r_{13}$	0.3	0.9	0.6
$p_{14}r_{21}$	0.8	0.3	0.2
$p_{14}r_{22}$	0.5	0.8	0.2
$p_{14}r_{41}$	0.8	0.8	0.6
$p_{14}r_{42}$	0.9	0.6	0.7
$p_{14}r_{43}$	0.9	0.6	0.7
$p_{15}r_{21}$	0.6	0.8	0.7
$p_{15}r_{32}$	0.3	0.8	0.8
$p_{15}r_{33}$	0.9	0.6	0.9
$p_{15}r_{53}$	0.5	0.9	0.8
$p_{15}r_{51}$	0.8	0.3	0.4
$p_{15}r_{52}$	0.5	0.8	0.4
$p_{15}r_{61}$	0.8	0.8	0.6
$p_{15}r_{62}$	0.9	0.8	0.9
$p_{16}r_{11}$	0.8	0.7	0.7
$p_{16}r_{12}$	0.4	0.7	0.8

Rec.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
$p_{16}r_{13}$	0.3	0.5	0.4
$p_{16}r_{14}$	0.5	0.6	0.7
$p_{16}r_{21}$	0.9	0.5	0.2
$p_{16}r_{22}$	0.2	0.5	0.4
$p_{16}r_{31}$	0.2	0.7	0.8
$p_{16}r_{33}$	0.4	0.5	0.8
$p_{17}r_{11}$	0.8	0.9	0.9
$p_{17}r_{12}$	0.4	0.7	0.8
$p_{17}r_{13}$	0.3	0.8	0.6
$p_{21}r_{12}$	0.7	0.5	0.8
$p_{21}r_{13}$	0.4	0.5	0.2
$p_{21}r_{22}$	0.6	0.2	0.4
$p_{21}r_{23}$	0.2	0.9	0.4
$p_{21}r_{31}$	0.4	0.9	0.9
$p_{21}r_{33}$	0.8	0.6	0.6
$p_{22}r_{21}$	0.4	0.6	0.9
$p_{22}r_{22}$	0.4	0.2	0.1
$p_{22}r_{31}$	0.8	0.2	0.9
$p_{22}r_{33}$	0.9	0.5	0.8
$p_{23}r_{12}$	0.7	0.5	0.5
$p_{23}r_{24}$	0.8	0.9	0.4
$p_{23}r_{31}$	0.9	0.4	0.6
$p_{23}r_{32}$	0.6	0.6	0.6
$p_{23}r_{33}$	0.6	0.4	0.2
$p_{24}r_{11}$	0.7	0.9	0.5
$p_{24}r_{12}$	0.7	0.6	0.9
$p_{24}r_{23}$	0.7	0.6	0.3
$p_{24}r_{24}$	0.4	0.8	0.7
$p_{25}r_{21}$	0.5	0.6	0.8
$p_{31}r_{13}$	0.4	0.9	0.8
$p_{31}r_{71}$	0.3	0.9	0.7
$p_{31}r_{72}$	0.9	0.7	0.7
$p_{32}r_{11}$	0.7	0.3	0.7
$p_{32}r_{12}$	1.0	0.9	0.7
$p_{32}r_{13}$	0.3	0.5	0.9
$p_{32}r_{21}$	0.8	0.4	0.9
$p_{33}r_{11}$	0.9	1.0	0.3
$p_{33}r_{12}$	0.5	0.7	0.3
$p_{33}r_{13}$	0.4	0.5	0.8
$p_{33}r_{21}$	0.4	0.8	0.8
$p_{33}r_{22}$	0.4	0.9	0.7
$p_{33}r_{23}$	0.4	0.7	0.8
$p_{33}r_{32}$	0.5	0.8	0.8

Rec.	Sobrecarga Memoria	Sobrecarga Procesador	Sobrecarga E/S
$p_{33}r_{33}$	0.8	0.8	0.9
$p_{34}r_{12}$	0.3	0.8	0.7
$p_{34}r_{13}$	0.6	0.9	0.7
$p_{34}r_{22}$	0.8	0.5	0.6
$p_{34}r_{23}$	0.5	0.9	0.6
$p_{34}r_{24}$	0.5	0.9	0.7
$p_{34}r_{31}$	0.9	0.6	0.7
$p_{34}r_{32}$	0.9	0.3	0.9
$p_{34}r_{33}$	0.9	0.4	0.3
$p_{35}r_{31}$	0.3	0.4	0.8
$p_{35}r_{32}$	0.9	0.4	0.7
$p_{35}r_{33}$	0.5	0.9	0.7
$p_{35}r_{61}$	0.5	0.7	0.9
$p_{35}r_{62}$	0.8	0.7	0.4
$p_{35}r_{72}$	0.6	0.3	0.6
$p_{35}r_{74}$	0.9	0.4	0.7
$p_{36}r_{11}$	0.9	0.9	0.6
$p_{36}r_{12}$	0.9	0.4	0.7
$p_{36}r_{13}$	0.7	0.9	0.7
$p_{36}r_{21}$	0.7	0.9	0.9
$p_{36}r_{22}$	0.7	0.6	0.9
$p_{36}r_{24}$	0.5	0.8	0.9
$p_{36}r_{81}$	0.8	0.8	0.7
$p_{36}r_{82}$	0.5	0.6	0.7
$p_{36}r_{83}$	0.9	0.4	0.5
$p_{37}r_{11}$	0.7	0.9	0.8
$p_{37}r_{12}$	0.8	0.8	0.6
$p_{37}r_{31}$	0.9	0.5	0.4
$p_{37}r_{42}$	0.7	0.4	0.6
$p_{37}r_{43}$	0.6	0.9	0.3

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Al promedio de carga computacional que se obtiene mediante el operador de agregación desarrollado en (La Red Martínez, 2017) se le agrega un criterio adicional que es la sobrecarga, que se obtiene de la suma de los promedios de sobrecargas de cada proceso del nodo.

Al migrar los procesos, se considerará la tabla de clasificación de nodos según sus prestaciones (Tabla 119). Esto permitirá determinar la categoría a la que pertenecen tanto el nodo

origen como el nodo destino. De esta manera, se podrán calcular los nuevos valores de sobrecarga utilizando la tabla de conversión de categorías nodales (Tabla 148).

Tabla 148. Tabla de conversión de categorías nodales (E2)

Categorías Origen	Destino		
	1	2	3
1	1.00	0.33	0.25
2	3.00	1.00	0.75
3	4.00	1.33	1.00

Nota: La categoría 1 corresponde a sub estándar, el 2 corresponde a estándar y 3 corresponde a supra estándar.

En la Tabla 149 se muestra la evaluación de los promedios de sobrecarga del nodo 1 (nodo origen), que migra procesos a los nodos destino 2, 4, 8 y 9. El promedio A hace referencia al promedio de sobrecarga de los procesos que no se migran, el promedio B hace referencia al promedio de sobrecarga de los procesos que serán migrados a otros nodos, este último promedio se evalúa por cada nodo destino.

Tabla 149. Evaluación de promedios de sobrecarga del nodo 1 (E2)

	Procesos	Nodos destinos	Promedios de sobrecarga
Promedio A			13.00
Promedio B	p_{11}	2	2.97
Promedio B	p_{14}	4	4.57
Promedio B	p_{15}	8	5.53
Promedio B	p_{17}	9	2.07
Promedio A+ Promedio B			28.17

Fuente: Elaboración propia

En la Tabla 150 se muestra la evaluación de los promedios de sobrecarga del nodo 2 (nodo destino), que recibe el proceso p_{11} del nodo 1. Según las prestaciones de los nodos involucrados el proceso se migra de un nodo origen sub estándar a un nodo receptor con categoría estándar. El valor de sobrecarga del proceso en el nodo origen es multiplicado por el valor del peso de la tabla

de conversión de categorías nodales (Tabla 148) para obtener el nuevo valor que generará en el nodo receptor.

El promedio A hace referencia al promedio de sobrecarga de los procesos locales, el promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{11} pasa a ser proceso p_{26} en su nodo receptor.

Tabla 150. Evaluación de promedios de sobrecarga del nodo 2 (E2)

Procesos	Promedio de sobrec. en nodo origen	Promedios de sobrec. en nodo receptor
Promedio A	$p_{11} \rightarrow p_{26}$	11.70
Promedio C		0.98
Promedio A+ Promedio C		12.68

Fuente: Elaboración propia

En la Tabla 151 se muestra la evaluación de los promedios de sobrecarga del nodo 3 (nodo origen), que migra procesos a los nodos destino 7, 5, y 10. El promedio A hace referencia al promedio de sobrecarga de los procesos que no se migran, el promedio B hace referencia al promedio de sobrecarga de los procesos migrados a otros nodos, este último promedio se evalúa por cada nodo destino.

Tabla 151. Evaluación de promedios de sobrecarga del nodo 3 (E2)

Procesos	Nodos destinos	Promedios de sobrec. en nodo origen
Promedio A		16.17
Promedio B	p_{31}	2.10
Promedio B	p_{32}	3.73
Promedio B	p_{37}	3.30
Promedio A+ Promedio B		23.63

Fuente: Elaboración propia

En la Tabla 152 se muestra la evaluación de los promedios de sobrecarga del nodo 4 (nodo destino), que recibe el proceso p_{14} del nodo 1. Según las prestaciones de los nodos involucrados el proceso se migra de un nodo origen estándar a un nodo receptor con categoría estándar. El valor

de sobrecarga del proceso en el nodo origen es multiplicado por el valor del peso de la tabla de conversión de categorías nodales (Tabla 148) para obtener el nuevo valor que generará en el nodo receptor.

El promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{14} pasa a ser proceso p_{41} en su nodo receptor.

Tabla 152. Evaluación de promedios de sobrecarga del nodo 4 (E2)

Procesos	Promedio de sobrec. en nodo origen	Promedios de sobrec. en nodo receptor
Promedio C $p_{14} \rightarrow p_{41}$	4.57	4.57

Fuente: Elaboración propia

En la Tabla 153 se muestra la evaluación de los promedios de sobrecarga del nodo 5 (nodo destino), que recibe el proceso p_{32} del nodo 3. Según las prestaciones de los nodos involucrados el proceso se migra de un nodo origen supra estándar a un nodo receptor con categoría supra estándar. El valor de sobrecarga del proceso en el nodo origen es multiplicado por el valor del peso de la tabla de conversión de categorías nodales (Tabla 148) para obtener el nuevo valor que generará en el nodo receptor.

El promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{15} pasa a ser proceso p_{51} en su nodo receptor.

Tabla 153. Evaluación de promedios de sobrecarga del nodo 5 (E2)

Procesos	Promedio de sobrec. en nodo origen	Promedios de sobrec. en nodo receptor
Promedio C $p_{32} \rightarrow p_{51}$	2.70	2.70

Fuente: Elaboración propia

En la Tabla 154 se muestra la evaluación de los promedios de sobrecarga del nodo 7 (nodo destino), que recibe el proceso p_{31} del nodo 3. Según las prestaciones de los nodos involucrados el proceso se migra de un nodo origen supra estándar a un nodo receptor con categoría sub estándar.

El valor de sobrecarga del proceso en el nodo origen es multiplicado por el valor del peso de la tabla de conversión de categorías nodales (Tabla 148) para obtener el nuevo valor que generará en el nodo receptor.

El promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{31} pasa a ser proceso p_{71} en su nodo receptor.

Tabla 154. Evaluación de promedios de sobrecarga del nodo 7 (E2)

Procesos	Promedio de sobrec. en nodo origen	Promedios de sobrec. en nodo receptor
Promedio C $p_{31} \rightarrow p_{71}$	2.10	8.40

Fuente: Elaboración propia

En la Tabla 155 se muestra la evaluación de los promedios de sobrecarga del nodo 8 (nodo destino), que recibe el proceso p_{15} del nodo 1. Según las prestaciones de los nodos involucrados el proceso se migra de un nodo origen estándar a un nodo receptor con categoría estándar. El valor de sobrecarga del proceso en el nodo origen es multiplicado por el valor del peso de la tabla de conversión de categorías nodales (Tabla 148) para obtener el nuevo valor que generará en el nodo receptor.

El promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{15} pasa a ser proceso p_{81} en su nodo receptor.

Tabla 155. Evaluación de promedios de sobrecarga del nodo 8 (E2)

Procesos	Promedio de sobrec. en nodo origen	Promedios de sobrec. en nodo receptor
Promedio C $p_{15} \rightarrow p_{81}$	5.53	5.53

Fuente: Elaboración propia

En la Tabla 156 se muestra la evaluación de los promedios de sobrecarga del nodo 9 (nodo destino), que recibe el proceso p_{17} del nodo 1. Según las prestaciones de los nodos involucrados el proceso se migra de un nodo origen sub estándar a un nodo receptor con categoría supra estándar.

El valor de sobrecarga del proceso en el nodo origen es multiplicado por el valor del peso de la tabla de conversión de categorías nodales (Tabla 148) para obtener el nuevo valor que generará en el nodo receptor.

El promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{17} pasa a ser proceso p_{91} en su nodo receptor.

Tabla 156. Evaluación de promedios de sobrecarga del nodo 9 (E2)

Procesos	Promedio de sobrec. en nodo origen	Promedios de sobrec. en nodo receptor
Promedio C $p_{17} \rightarrow p_{91}$	2.07	0.52

Fuente: Elaboración propia

En la Tabla 157 se muestra la evaluación de los promedios de sobrecarga del nodo 10 (nodo destino), que recibe el proceso p_{37} del nodo 3. Según las prestaciones de los nodos involucrados el proceso se migra de un nodo origen supra estándar a un nodo receptor con categoría estándar. El valor de sobrecarga del proceso en el nodo origen es multiplicado por el valor del peso de la tabla de conversión de categorías nodales (Tabla 148) para obtener el nuevo valor que generará en el nodo receptor.

El promedio C hace referencia al promedio de sobrecarga de los procesos inmigrantes. El proceso p_{37} pasa a ser proceso p_{101} en su nodo receptor.

Tabla 157. Evaluación de promedios de sobrecarga del nodo 10 (E2)

Procesos	Promedio de sobrec. en nodo origen	Promedios de sobrec. en nodo receptor
Promedio C $p_{37} \rightarrow p_{101}$	3.30	4.40

Fuente: Elaboración propia

El nodo 6 no se evalúa porque no emigraron ni inmigraron procesos, se quedan con sus promedios de carga iniciales, calculados en la Tabla 87 que contine los valores de los criterios de carga computacional (E2) y el promedio de carga de los nodos.

3.5.8. Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema (E2)

El objetivo principal es evaluar y comparar el impacto de la migración de procesos en el estado global del sistema. Para lograrlo, se realizarán dos etapas: en la primera etapa, se contemplará la sobrecarga de las asignaciones de recursos a procesos, sin realizar la migración, mientras que en la segunda etapa, realizando la migración con los nuevos valores de sobrecarga según nodo receptor.

Al comparar ambos escenarios, se podrá determinar el impacto de la migración en el rendimiento y funcionamiento global del sistema.

La carga computacional de cada nodo se calculará de dos maneras.

a. Cálculo de sobrecarga sin migración

Se aplica el promedio de sobrecarga (Prom. de Sobrecar.) de las asignaciones de recursos a procesos al promedio carga computacional (Prom.) obtenida mediante el operador de agregación desarrollado en (La Red Martínez, 2017). Esto se observa en la Tabla 158.

Tabla 158. Carga computacional sin migración (E2)

Nodo	%CPU	%Mem.	%E/S	Prom.	Prom. de Sobrecar.	Prom. Final
1	80	90	75	81.67	28.13	109.80
2	30	30	50	36.67	11.70	48.37
3	77	49	85	70.33	24.27	94.60
4	6	7	9	7.33	0	7.33
5	5	5	6	5.33	0	5.33
6	7	9	7	7.67	0	7.67
7	7	10	15	10.67	0	10.67
8	9	6	7	7.33	0	7.33
9	10	9	15	11.33	0	11.33
10	8	7	8	7.67	0	7.67

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Nota: Los nodos del 4 a 10 no tienen procesos activos, por lo tanto, tienen valor de sobrecarga 0.

b. Cálculo de sobrecarga con migración

Se aplica el promedio de sobrecarga (Prom. de Sobrecar.) de las asignaciones de recursos a procesos al promedio carga computacional (Prom.) obtenida mediante el operador de agregación desarrollado en (La Red Martínez, 2017). Los valores de promedio de sobrecarga aplicado son los valores calculados para cada nodo destino desde la Tabla 149 hasta la Tabla 157. Esto se observa en la Tabla 159.

Tabla 159. Carga computacional con migración (E2)

Nodo	%CPU	%Mem.	%E/S	Prom.	Prom. de Sobrecar.	Prom. Final
1	80	90	75	81.67	13.00	94.67
2	30	30	50	36.67	12.68	49.35
3	77	49	85	70.33	16.17	86.50
4	6	7	9	7.33	4.57	11.90
5	5	5	6	5.33	2.70	8.03
6	7	9	7	7.67	0	7.67
7	7	10	15	10.67	8.40	19.07
8	9	6	7	7.33	5.53	12.87
9	10	9	15	11.33	0.52	11.85
10	8	7	8	7.67	4.40	12.07

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

Nota: El nodo 6 no tiene procesos activos, por lo tanto, su valor de sobrecarga es 0.

3.5.9. Evaluación del consumo de energía de los nodos una vez migrado los procesos (E2)

Para evaluar el consumo de energía de los nodos del sistema distribuido se utilizará el promedio final de sobrecarga multiplicado por un el valor correspondiente del vector de conversión de consumo de energía (Tabla 160).

Tabla 160. Vector de conversión consumo de energía. (E2)

	Categorías		
	Subestándar	Estándar	Supra estándar
Consumo	1	2	3

Fuente: Elaboración propia

Se realiza el siguiente cálculo:

$$\text{consumo_energía} = pf * vc \quad (16)$$

En donde pf hace referencia al promedio final de los nodos aplicando la sobrecarga con y sin migración y vc hace referencia al vector de conversión de la Tabla 160.

El consumo de energía se calculará de dos maneras y serán mostrados en el punto a (Tabla 161) y en el punto b (Tabla 162).

a. Cálculo de consumo de energía sin migración

Tabla 161. Consumo de energía sin migración (E2)

Nodo	%CPU	%Mem.	%E/S	Prom.	Prom. de Sobrecar.	Prom. Final	Consumo de energía
1	80	90	75	81.67	28.13	109.80	109.80
2	30	30	50	36.67	11.70	48.37	34.71
3	77	49	85	70.33	24.27	94.60	283.80
4	6	7	9	7.33	0	7.33	7.33
5	5	5	6	5.33	0	5.33	16.00
6	7	9	7	7.67	0	7.67	7.67
7	7	10	15	10.67	0	10.67	10.67
8	9	6	7	7.33	0	7.33	7.33
9	10	9	15	11.33	0	11.33	34.00
10	8	7	8	7.67	0	7.67	15.33

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

b. Cálculo de consumo de energía con migración

Tabla 162. Consumo de energía con migración (E2)

Nodo	%CPU	%Mem.	%E/S	Prom.	Prom. de Sobrecar.	Prom. Final	Consumo de energía
1	80	90	75	81.67	13.00	94.67	63.27
2	30	30	50	36.67	12.68	49.35	17.95
3	77	49	85	70.33	16.17	86.50	55.10
4	6	7	9	7.33	4.57	11.90	19.50
5	5	5	6	5.33	2.70	8.03	23.36
6	7	9	7	7.67	0	7.67	23.73

Nodo	%CPU	%Mem.	%E/S	Prom.	Prom. de Sobrecar.	Prom. Final	Consumo de energía
7	7	10	15	10.67	8.40	19.07	12.33
8	9	6	7	7.33	5.53	12.87	18.53
9	10	9	15	11.33	0.52	11.85	19.55
10	8	7	8	7.67	4.40	12.07	19.33

Fuente: Elaboración propia. Basado en (La Red Martínez, 2017)

3.5.10. Métrica propuesta para evaluar el balanceo de carga y consumo de energía en el sistema distribuido (E2)

Para evaluar el impacto de la migración de procesos y cómo ésta afecta el estado global del sistema, es necesario aplicar algún mecanismo de medición de los estados de los nodos antes y después de la migración. Para ello, se utiliza la desviación estándar, considerando promedio final de carga de la Tabla 158 y Tabla 159 y el consumo de energía de los nodos del sistema distribuido mostrados en la Tabla 161 y Tabla 162.

Para la evaluación de la carga se utilizarán los siguientes criterios asociados a cada proceso migrado: sobrecarga de procesador, sobrecarga de memoria y sobrecarga de entrada / salida, que se mostraron en la Tabla 90.

La desviación estándar del nivel de carga de los nodos sin considerar migración aplicando la formula (11) es la siguiente:

Se tiene en cuenta la Tabla 158: 109.80; 48.37; 94.60; 7.33; 5.33; 7.67; 10.67; 7.33; 11.33 y 7.67, cuyo promedio es de 31.01.

$$s = \sqrt{(((109.83-30.95)^2 + (48.37-30.95)^2 + (94.60-30.95)^2 + (7.33-30.95)^2 + (5.33-30.95)^2 + (7.67-30.95)^2 + (10.67-30.95)^2 + (7.33-30.95)^2 + (11.33-30.95)^2 + (7.67-30.95)^2) / (10-1))}$$

$$s = 39.75$$

La desviación estándar del nivel de carga de los nodos con migración aplicando la fórmula (11) es la siguiente.

Se tiene en cuenta la Tabla 159: 94.67; 49.35; 86.50; 11.90; 8.03; 7.67; 19.07; 12.87; 11.85 y 12.07, cuyo promedio es de 31.40.

$$s = \sqrt{(((94.67-30.95)^2 + (49.35-30.95)^2 + (86.50-30.95)^2 + (11.90-30.95)^2 + (8.03-30.95)^2 + (7.67-30.95)^2 + (19.07-30.95)^2 + (12.87-30.95)^2 + (11.85-30.95)^2 + (12.07-30.95)^2) / (10-1))}$$

$$s = 33.50$$

La desviación estándar del consumo de energía de los nodos sin considerar migración aplicando la formula (11) es la siguiente:

Se tiene en cuenta la Tabla 161: 109.80; 34.71; 283.80; 7.33; 16.00; 7.67; 10.67; 7.33; 34.00; 15.33; cuyo promedio es 52.66.

$$s = \sqrt{(((109.80-52.66)^2 + (34.71-52.66)^2 + (283.80-52.66)^2 + (7.33-52.66)^2 + (16.00-52.66)^2 + (7.67-52.66)^2 + (10.67-52.66)^2 + (7.33-52.66)^2 + (34.00-52.66)^2 + (15.33-52.66)^2) / (10-1))}$$

$$s = 86.92.$$

La desviación estándar del consumo de energía de los nodos con migración aplicando la fórmula (11) es la siguiente.

Se tiene en cuenta la Tabla 162: 94.67; 98.69; 259.50; 4.57; 3.00; 7.67; 19.07; 12.87; 35.55; 24.13, cuyo promedio es 55.97.

$$s = \sqrt{(((94.67-55.97)^2 + (98.69-55.97)^2 + (259.50-55.97)^2 + (4.57-55.97)^2 + (3.00-55.97)^2 + (7.67-55.97)^2 + (19.07-55.97)^2 + (12.87-55.97)^2 + (35.55-55.97)^2 + (24.13-55.97)^2) / (10-1))}$$

s= 79.71.

A partir de los resultados obtenidos referente al nivel de carga de los nodos se observa que la desviación estándar del conjunto de promedios finales sin migración es de **39.75** y la desviación estándar del conjunto de promedios finales con migración es de **33.50**. Referente a los resultados del consumo de energía se observa que la desviación estándar del conjunto de datos sobre consumo de energía sin migración es de **86.92** y la desviación estándar del conjunto de datos sobre consumo de energía con migración es de **79.71**.

Esto indica que se ha logrado balancear la carga y el consumo de energía dentro del sistema distribuido porque los valores del conjunto de promedios finales y el conjunto de datos sobre consumo de energía con migración están más concentrados alrededor de la media, mientras que los valores de promedios finales y conjunto de datos sobre consumo de energía sin migración están más dispersos.

3.6. Comentarios y discusiones

El trabajo propone un modelo de migración inteligente de procesos para la gestión de recursos y procesos en sistemas distribuidos, con el objetivo de resolver los inconvenientes relacionados con la sobrecarga que pueden presentarse en los distintos nodos de un sistema distribuido. Se presenta una variante de un enfoque innovador para la gestión de recursos compartidos en sistemas distribuidos que considera el balanceo de carga de trabajo entre los distintos nodos.

La propuesta se basa en la evaluación del estado de las cargas computacionales, lo que permite detectar nodos muy cargados con respecto a nodos inactivos o con baja carga de trabajo y corregir la distribución de carga de trabajo en el sistema.

La solución propuesta se basa en un *Runtime* centralizado que gestiona los procesos y recursos compartidos, y que interactúa con los *Runtimes* de los nodos distribuidos para intercambiar información (todos ellos interactúan con sus respectivos sistemas operativos). Además, se establecen configuraciones iniciales del algoritmo para cada situación en particular, lo que permite adecuar la solución planteada a distintos entornos.

Se recopila información de todos los nodos en un nodo central, se aplica el proceso de agregación y se obtiene la lista de asignaciones de recursos a procesos, y la carga adicional de trabajo que ello significaría sobre los distintos nodos distribuidos.

La evaluación del estado de las cargas computacionales se realiza de manera periódica, lo que permite considerar posibles migraciones para balancear la carga de trabajo de los distintos nodos del sistema.

Como métrica para evaluar la efectividad de la solución propuesta se utiliza la desviación estándar, que es una medida de dispersión muy útil y ampliamente utilizada en el análisis de datos debido a su fácil interpretación, sensibilidad a todos los valores de los datos y propiedades matemáticas útiles.

Capítulo IV. Comparación de resultados

Durante el funcionamiento del sistema distribuido homogéneo y heterogéneo, cada cierto tiempo se reúne y se evalúa información de control referida a las cargas nodales para obtener una imagen global del sistema (macro imagen), que refleja el estado de los nodos, recursos y procesos en un determinado momento, lo que permite evaluar posibles migraciones para balancear la carga de trabajo de los distintos nodos del sistema.

El algoritmo de balanceo de carga corresponde a una serie de cálculos que se realizan para lograr la migración de procesos de nodos cargados a nodos más aliviados, partiendo de la estructura de datos utilizados en (La Red Martínez, 2017) y (La Red Martínez et al., 2018). A continuación, se realiza la comparativa de resultados de los escenarios desarrollados.

En la Tabla 163 se muestran los valores de promedios finales de la carga computacional sin migración y los valores de promedios finales de la carga computacional aplicando la migración de procesos en cada escenario (homogéneo y heterogéneo).

Tabla 163. Comparativa de carga computacional con migración de procesos

Nodo	Promedio final sin migración	Promedio final con migración homogéneo (E1)	Promedio final con migración heterogéneo (E2)
1	109.80	99.27	94.67
2	48.37	51.33	49.35
3	94.60	84.83	86.50
4	7.33	10.63	11.90
5	5.33	10.87	8.03
6	7.67	11.40	7.67
7	10.67	12.77	19.07
8	7.33	9.40	12.87
9	11.33	11.33	11.85
10	7.67	7.67	12.07
Desviación estándar	39.75	34.81	33.50

Fuente: Elaboración propia.

Nota: Los valores del promedio final sin migración son los mismos para ambos escenarios

Tabla 164. Comparativa del consumo de energía en entornos heterogéneo

Nodo	Consumo de energía sin migración heterogéneo (E2)	Consumo de energía con migración heterogéneo (E2)
1	99.27	94.67
2	51.33	49.35
3	84.83	86.50
4	10.63	11.90
5	10.87	8.03
6	11.40	7.67
7	12.77	19.07
8	9.40	12.87
9	11.33	11.85
10	7.67	12.07
Desviación estándar	86.92	79.71

Fuente: Elaboración propia.

Nota: Los valores del promedio final sin migración son los mismos para ambos escenarios

En cada uno de los escenarios, se aplica el promedio de sobrecarga de las asignaciones de recursos a procesos al promedio carga computacional, obtenida mediante el operador de agregación desarrollado en (La Red Martínez, 2017). Al comparar estos valores entre los escenarios, se evidencian diferencias significativas en los resultados obtenidos. A continuación, se procede a analizar estos resultados utilizando la métrica propuesta.

Para evaluar la efectividad de la solución planteada en cada escenario, se utiliza la desviación estándar como métrica de referencia sobre el promedio final de la carga computacional. La desviación estándar del conjunto de valores de promedios finales para ambos escenarios es de **39.75**. En el contexto de la migración de procesos en un sistema distribuido de entorno homogéneo, el valor resultante de la métrica aplicado al conjunto de valores de promedios finales es de **34.81**, mientras que, en un sistema distribuido de entorno heterogéneo, el valor resultante de la métrica aplicado al conjunto de valores de promedios finales es de **33.50**.

Referente a los resultados del consumo de energía de los nodos que se analizó en la migración de procesos de entornos heterogéneos (E2), se observa que la desviación estándar del conjunto de datos sobre consumo de energía sin migración es de **86.92** y la desviación estándar del conjunto de datos sobre consumo de energía con migración es de **79.71**.

Estos resultados indican que se ha logrado efectivamente el balanceo de carga en ambos escenarios y el balanceo del consumo de energía en el escenario 2 (E2), aunque es importante destacar que en el entorno heterogéneo se han desarrollado cálculos adicionales que no se tuvieron en cuenta en el escenario homogéneo. En el entorno heterogéneo, se consideraron las prestaciones individuales de cada nodo como un factor determinante en el proceso de migración, realizando una evaluación minuciosa de las características específicas de cada componente hardware en cada nodo.

Se priorizó la migración de procesos hacia nodos con mayores prestaciones como primera opción. En este sentido, se utilizaron vectores de peso para asignar una importancia relativa a las diferentes características y componentes de hardware, lo que permitió determinar con mayor precisión cuáles características o componentes de hardware debían considerarse de manera prioritaria en cada situación específica. Este enfoque detallado y personalizado en el entorno heterogéneo contribuyó a optimizar la distribución de la carga computacional y el consumo de energía, teniendo en cuenta las capacidades individuales de cada nodo y las necesidades del sistema en tiempo real.

Comparativa del operador de agregación para cada escenario

Las diferencias más significativas con respecto al operador de agregación para cada escenario se pueden observar en la Tabla 165.

Tabla 165. Etapas del operado de agregación para cada escenario

Etapas	SD homogéneo	SD heterogéneo
Cálculo de la carga computacional actual de los nodos	✓	✓
Establecimiento de las categorías nodales según la carga computacional	✓	✓
Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo	✓	✓
Determinación de procesos migrables mediante el cálculo de prioridad promedio de los procesos	✓	✓
Selección de los posibles nodos destino mediante la evaluación de la carga	✓	✓
Selección de los posibles nodos destino mediante la evaluación de compatibilidad	✗	✓
Evaluación del peso de los recursos y su posible impacto ante la migración de procesos	✓	✓
Establecimiento de las categorías de nodos según las prestaciones de cada uno	✗	✓
Determinación del orden de migración de procesos y de los nodos destino correspondientes	✓	✓
Evaluación de nuevos valores de sobrecarga de los procesos inmigrantes según categoría del nodo receptor	✗	✓
Evaluación del impacto de la migración de procesos en la carga computacional nodal y el rendimiento global del sistema	✓	✓
Evaluación del consumo de energía de los nodos una vez migrado los procesos	✗	✓

Fuente: Elaboración propia.

Capítulo V - Conclusiones y futuras líneas de investigación

Conclusiones

En el capítulo I se ha hecho la descripción de antecedentes, el marco teórico, se ha planteado la hipótesis y los objetivos de la investigación, se ha considerado agregar a un modelo de decisión la capa de migración inteligente de procesos dentro de un sistema distribuido considerando homogeneidad (escenario 1) y heterogeneidad (escenario 2), que permita la redistribución de la carga de trabajo entre los nodos del sistema distribuido para mejorar la utilización de los recursos, el tiempo de respuesta de los procesos, la mejora del rendimiento y verificar el impacto en el consumo de energía.

También se describieron las estructuras de datos, los elementos que intervienen dentro del sistema, los nodos, recursos y procesos, con prioridades y cargas de trabajo y que en diferentes situaciones permiten que el sistema se autorregule de manera reiterada en función del estado local de los n nodos y del estado global del sistema.

En el capítulo II se desarrolla un operador de agregación para la asignación de recursos en un sistema distribuido considerando la migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo homogeneidad de los nodos (respecto de compatibilidad de código, SO, arquitectura, set de instrucciones, etc.), incluyendo migración de procesos que significarán el acceso a datos distribuidos con alto impacto en E/S y tráfico de red (consumo de ancho de banda), y una posible migración de proceso consecuencia de cambios en el estado global del sistema y de los requerimientos variables de los procesos a lo largo del tiempo, esta verificación se hará cada cierto lapso Δt .

La propuesta presentada es una solución innovadora y efectiva para la gestión de recursos compartidos en sistemas distribuidos homogéneos, que permite resolver problemas relacionados con la sobrecarga y el balanceo de carga de trabajo. La evaluación del estado de las cargas computacionales y la comparación final utilizando la desviación estándar, que son herramientas útiles para detectar y corregir desequilibrios en el sistema.

En el capítulo III se desarrolla un operador de agregación para la asignación de recursos en un sistema distribuido considerando la migración de procesos para lograr un mejor balanceo de carga de trabajo suponiendo heterogeneidad de los nodos (respecto de compatibilidad de código, SO, arquitectura, set de instrucciones, etc.).

Al igual que el capítulo anterior la propuesta presentada es una solución innovadora y efectiva para la gestión de recursos compartidos en sistemas distribuidos, pero en entornos heterogéneos, que permite resolver problemas relacionados con la sobrecarga y el balanceo de carga de trabajo. La evaluación del estado de las cargas computacionales y la comparación final utilizando la desviación estándar, son herramientas útiles para detectar y corregir desequilibrios en el sistema.

En el capítulo IV se realiza la comparativa de resultados de ambos escenarios propuestos demostrando que, en cada escenario se logró el balanceo de carga y que gracias a la implementación de estrategias más avanzadas en el entorno heterogéneo, se logró un balance de carga aún más efectivo en el capítulo III, aprovechando las ventajas de las prestaciones diferenciales de los nodos y teniendo en cuenta las características específicas de su hardware, lo que ha contribuido significativamente a mejorar el rendimiento y la eficiencia del sistema distribuido en su conjunto.

Futuras líneas de investigación

Como futuras líneas de investigación relacionadas al desarrollo de este trabajo podemos mencionar las siguientes:

- Adaptar el simulador para la evaluación de algoritmos clásicos y nuevos para la gestión de recursos compartidos en sistemas distribuidos contemplando exclusión mutua, agregando la capa de migración desarrollada.
- Para mejorar los algoritmos de migración, se pretende investigar y desarrollar algoritmos más avanzados y eficientes para la migración de procesos, que puedan adaptarse dinámicamente a cambios en la carga de trabajo y condiciones del sistema distribuido. Esto podría incluir enfoques basados en aprendizaje automático para una toma de decisiones más precisa.
- También se prevé la integración de políticas de seguridad. Investigar cómo integrar políticas de seguridad sólidas en el proceso de migración de procesos, garantizando que la migración no comprometa la integridad y la confidencialidad de los datos y procesos.
- Desarrollar un operador de agregación global teniendo en cuenta los modelos de decisión y operadores de agregación particulares, para la administración de procesos en sistemas distribuidos según los distintos escenarios y características (Etiquetas lingüísticas, imputación de datos, migración controlada de procesos).
- Explorar la aplicación de este modelo en entornos de nube, donde la gestión de recursos y la migración de procesos son desafíos clave, así como evaluar su integración con sistemas de contenedores para mejorar la eficiencia en la gestión de recurso.

Bibliografía

- Agostini, F. (2019). *Nueva propuesta para la administración de recursos y procesos en sistemas distribuidos*. Maestría en sistemas y redes de telecomunicaciones. Universidad Nacional del Nordeste.
- Agostini, F., & La Red Martínez, D. L. (2019). Allocation of shared resources. *14th Iberian Conference on Information Systems and Technologies - CISTI 2019*, 1–6.
- Agostini, F., La Red Martínez, D. L., & Acosta, J. C. (2018). Nueva propuesta para la administración de recursos y procesos en sistemas distribuidos. *Workshop de Investigadores En Ciencias de La Computación*.
- Agostini, F., la Red Martínez, D. L., & Acosta, J. C. (2019). Assignment of resources in distributed systems with strict consensus requirements. *IMCIC 2019 - 10th International Multi-Conference on Complexity, Informatics and Cybernetics, Proceedings, 1*.
- Agrawal, D., & El Abbadi, A. (1991). An Efficient and Fault-Tolerant Solution for Distributed Mutual Exclusion. *ACM Transactions on Computer Systems (TOCS)*, 9(1).
<https://doi.org/10.1145/103727.103728>
- Alakeel, A. M. (2010). A Guide to Dynamic Load Balancing in Distributed Computer Systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(6).
- Andrade, H., Crnkovic, I., & Bosch, J. (2020). Refactoring Software in the Automotive Domain for Execution on Heterogeneous Platforms. *Proceedings - 2020 IEEE 44th Annual Computers, Software, and Applications Conference, COMPSAC 2020*.
<https://doi.org/10.1109/COMPSAC48688.2020.00-37>

- Asadi, A. N., Azgomi, M. A., & Entezari-Maleki, R. (2020). Analytical evaluation of resource allocation algorithms and process migration methods in virtualized systems. *Sustainable Computing: Informatics and Systems*, 25, 100370. <https://doi.org/10.1016/j.suscom.2019.100370>
- Banerjee, S. S., El-Hadedy, M., Lim, J. Bin, Kalbarczyk, Z. T., Chen, D., Lumetta, S. S., & Iyer, R. K. (2019). ASAP: Accelerated Short-Read Alignment on Programmable Hardware. *IEEE Transactions on Computers*, 68(3), 331–346. <https://doi.org/10.1109/TC.2018.2875733>
- Beirut, M. A., & Ganjali, Y. (2020). Load Migration in Distributed SDN Controllers. *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2020: Management in the Age of Softwarization and Artificial Intelligence, NOMS 2020*. <https://doi.org/10.1109/NOMS47738.2020.9110292>
- Cao, G., & Singhal, M. (2001). A delay-optimal quorum-based mutual exclusion algorithm for distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, 12(12). <https://doi.org/10.1109/71.970560>
- Chang, C., Hadachi, A., & Srirama, S. N. (2020). Adaptive edge process migration for IoT in heterogeneous fog and edge computing environments. *International Journal of Mobile Computing and Multimedia Communications*, 11(3). <https://doi.org/10.4018/IJMCMC.2020070101>
- Chen, J., Banerjee, S. S., Kalbarczyk, Z. T., & Iyer, R. K. (2020). Machine learning for load balancing in the Linux kernel. *APSys 2020 - Proceedings of the 2020 ACM SIGOPS Asia-Pacific Workshop on Systems*. <https://doi.org/10.1145/3409963.3410492>

- Chiclana, F., Herrera, F., & Herrera-Viedma, E. (2001). Integrating multiplicative preference relations in a multipurpose decision-making model based on fuzzy preference relations. *Fuzzy Sets and Systems*, 122(2). [https://doi.org/10.1016/S0165-0114\(00\)00004-X](https://doi.org/10.1016/S0165-0114(00)00004-X)
- Deshmukh, S. C., & Deshmukh, S. S. (2015). Improved load balancing for distributed file system using self acting and adaptive loading data migration process. *2015 4th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2015*. <https://doi.org/10.1109/ICRITO.2015.7359283>
- Dong, Y., Zhang, H., & Herrera-Viedma, E. (2016). Consensus reaching model in the complex and dynamic MAGDM problem. *Knowledge-Based Systems*, 106. <https://doi.org/10.1016/j.knosys.2016.05.046>
- Duolikun, D., Aikebaier, A., Enokido, T., & Takizawa, M. (2014). Power consumption models for migrating processes in a server cluster. *Proceedings - 2014 International Conference on Network-Based Information Systems, NBiS 2014*. <https://doi.org/10.1109/NBiS.2014.41>
- Elleuch, A., & Muntean, T. (1994). Process migration protocols for massively parallel systems. *MPCS 1994 - 1st International Conference on Massively Parallel Computing Systems: The Challenges of General-Purpose and Special-Purpose Computing*. <https://doi.org/10.1109/MPCS.1994.367089>
- Eskicioglu, M. R. (1995). Design Issues of Process Migration Facilities in Distributed Systems. *IEEE Computer Society Press*, 414–424.
- Fornerón Martínez, J. T., Agostini, F., & La Red Martínez, D. L. (2020). *International Journal of Information Systems and Software Engineering for Big Companies (IJISEBC)*. www.ijisebc.com

- Fullér, R. (1996). OWA operators in decision making. *Exploring the Limits of Support Systems, TUCS General Publications*, 3, 85–104.
- Hac, A., & Johnson, T. (1986). A Study of Dynamic Load Balancing in a Distributed System. In *Proceedings of the ACM SIGCOMM Conference on Communications Architectures & Protocols*, 348–356.
- Junaidi, J., Wibowo, P., Yuniasri, D., Damayanti, P., Shiddiqi, A. M., & Pratomo, B. A. (2020). Applied Machine Learning In Load Balancing. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 18(2), 76. <https://doi.org/10.12962/j24068535.v18i2.a940>
- Kansal, N. J., & Chana, I. (2012). Cloud Load Balancing Techniques: A Step Towards Green Computing. *International Journal of Computer Science Issues*, 9(1), 238–246.
- Kim, Y., Lee, J., Kim, A., & Hong, J. (2021). Load Balancing Scheme for Machine Learning Distributed Environment. *Korean Institute of Smart Media*, 10(1), 25–31. <https://doi.org/10.30693/smj.2021.10.1.25>
- La Red Martínez, D. L. (2004). *Sistemas Operativos*. EUDENE.
- La Red Martínez, D. L. (2017). Aggregation Operator for Assignment of Resources in Distributed Systems. *International Journal of Advanced Computer Science and Applications*, 8(10). <https://doi.org/10.14569/ijacsa.2017.081053>
- La Red Martínez, D. L., & Acosta, J. C. (2014). Revisión de Operadores de Agregación. *Campus Virtuales*, 3(2).
- La Red Martinez, D. L., & Acosta, J. C. (2015). Review of Modeling Preferences for Decision Models. *European Scientific Journal*, 11(36).

- La Red Martínez, D. L., Acosta, J. C., & Agostini, F. (2018). Assignment of resources in distributed systems. *IMCIC 2018 - 9th International Multi-Conference on Complexity, Informatics and Cybernetics, Proceedings, 2*.
- La Red Martínez, D. L., Agostini, F., Acosta, J. C., Gerzel, S., & Latyn, L. (2022). Simulador para la evaluación de algoritmos para la gestión de recursos compartidos en sistemas distribuidos. *Revista de Investigación En Tecnologías de La Información, 10*(20), 62–79. <https://doi.org/10.36825/riti.10.20.006>
- La Red Martínez, D. L., Agostini, F., & Primorac, C. (2017). Modelo de Asignación de Recursos para la Enseñanza de los Procesos Distribuidos. *Primer Congreso Latinoamericano de Ingeniería - CLADI*.
- Leyva-vázquez, M., Fernández Marín, M. A., & Pérez Teruel, K. (2012). Operador TW-OWA su aplicación en la fusión de la información. *Taller de Cibernética Aplicada*, 6–8.
- Liu, H., Jin, H., Xu, C. Z., & Liao, X. (2013). Performance and energy modeling for live migration of virtual machines. *Cluster Computing, 16*(2), 249–264. <https://doi.org/10.1007/s10586-011-0194-3>
- Lodha, S., & Kshemkalyani, A. (2000). A fair distributed mutual exclusion algorithm. *IEEE Transactions on Parallel and Distributed Systems, 11*(6). <https://doi.org/10.1109/71.862205>
- Mallikarjuna, B., Arunkumar Reddy, D., & Venkata Vinod Kumar, N. (2018). Green Computing: Efficient Energy Load Balancing Technique in Cloud Computing. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.3169031>

- Marecos Brizuela, T. D., Agostini, F., & La Red Martínez, D. L. (2022). Migración controlada de procesos en sistemas distribuidos. *Memorias Del Encuentro Argentino de Ingenieria*, 173–180.
- Maurya, S. K., & Ahmad, K. (2013). Load balancing in distributed system through task migration. *International Journal of Engineering and Technology*, 5(2).
- Milošević, D. S., Douglass, F., Paindaveine, Y., Wheeler, R., & Zhou, S. (2000). Process migration. *ACM Computing Surveys*, 32(3). <https://doi.org/10.1145/367701.367728>
- Naghash Asadi, A., Abdollahi Azgomi, M., & Entezari-Maleki, R. (2019a). Evaluation of the impacts of failures and resource heterogeneity on the power consumption and performance of IaaS clouds. *Journal of Supercomputing*, 75(5). <https://doi.org/10.1007/s11227-018-2699-5>
- Naghash Asadi, A., Abdollahi Azgomi, M., & Entezari-Maleki, R. (2019b). Unified power and performance analysis of cloud computing infrastructure using stochastic reward nets. *Computer Communications*, 138. <https://doi.org/10.1016/j.comcom.2019.03.004>
- Patel, D. (2015). Process migration and load balancing. *International Journal of Research in Advance Engineering*, 1(1), 25. <https://doi.org/10.26472/ijrae.v1i1.10>
- Peláez, J., Doña, J. M., & Gómez-Ruiz, J. A. (2007). Analysis of OWA operators in decision making for modelling the majority concept. *Applied Mathematics and Computation*, 186(2). <https://doi.org/10.1016/j.amc.2006.07.161>
- Peláez, J., Dona, J., & Red, D. La. (2004). Majority opinion in group decision making using the qma-owa operator. 449–454.

- Rathore, N., & Chana, I. (2014). Load Balancing and Job Migration Techniques in Grid: A Survey of Recent Trends. *Wireless Personal Communications*, 79(3). <https://doi.org/10.1007/s11277-014-1975-9>
- Ricart, G., & Agrawala, A. K. (1981). An optimal algorithm for mutual exclusion in computer networks. *Communications of the ACM*, 24(1). <https://doi.org/10.1145/358527.358537>
- Sánchez Dams, R.-D. (2014). Load balancing for distributed databases access using a random heuristic algorithm. *Prospectiva*, 12(1). <https://doi.org/10.15665/rp.v12i1.150>
- Shah, V., & Kapadia, V. (2012). Load Balancing by Process Migration in Distributed Operating System. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(1).
- Sharma, V., Tech Scholar, M., & Sharma, V. (2012). A Survey on Process Migration. *International Journal of Advanced Research in Computer Science*, 3(6), 29–34.
- Smith, J. M. (1988). Survey Of Process Migration Mechanisms. *Operating Systems Review (ACM)*, 22(3). <https://doi.org/10.1145/47671.47673>
- Sohrabi, Z., & Mousavi Khaneghah, E. (2020). Challenges of Using Live Process Migration in Distributed Exascale Systems. *Azerbaijan Journal of High Performance Computing*, 3(2), 151–163. <https://doi.org/10.32010/26166127.2020.3.2.151.163>
- Stallings, W. (2005). *Sistemas operativos Aspectos internos y principios de diseño: Vol. Quinta Edi.*
- Tanenbaum, A., Guerrero, G., & Velasco, Ó. (1996). *Sistemas operativos distribuidos*. Prentice-Hall Hispanoamericana S.A.
- Tanenbaum, A. S. (2009). *Sistemas Operativos Modernos Tercera Edición*. In *Pearson education*.

- Tanenbaum, A. S., & Van Steen, M. (2008). *Sistemas Distribuidos - Principios y Paradigmas* (2da.). Pearson Educación S. A.
- Thakkar, N., & Pandya, A. (2013). *Process Migration in Heterogeneous Systems*. 1(7), 1425–1427.
- Thapliyal, N., & Dimri, P. (2020). Green Cloud Computing Strategies for Load Balancing. *Journal of Computational and Theoretical Nanoscience*, 17(6), 2695–2698.
<https://doi.org/10.1166/jctn.2020.8969>
- Upadhyay, A., & Lakkadwala, P. (2015). Migration of over loaded process and schedule for resource utilization in Cloud Computing. *2015 4th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2015*.
<https://doi.org/10.1109/ICRITO.2015.7359325>
- Wang, C., Mueller, F., Engelmann, C., & Scott, S. L. (2012). Proactive process-level live migration and back migration in HPC environments. *Journal of Parallel and Distributed Computing*, 72(2). <https://doi.org/10.1016/j.jpdc.2011.10.009>

Apéndice de publicaciones

1. Marecos, T., Agostini, F., La Red Martínez, D. (2022). Migración controlada de procesos en sistemas distribuidos. *6° Congreso Argentino de Ingeniería. 12° Congreso Argentino de Enseñanza de Ingeniería CADI-CAEDI*. Presentación oral. (PUBLICADO).
2. Marecos, T., Agostini, F., La Red Martínez, D. (2023). Migración controlada de procesos en sistemas distribuidos. *Workshop de Investigadores en Ciencias de la Computación (WICC)*. Póster. (PUBLICADO).
3. Marecos, T., Agostini, F., La Red Martínez, D. (2023). Migración Inteligente de procesos para balancear la carga de trabajo en sistemas distribuidos. *Libro de la Academia Nacional de Ciencias de Buenos Aires (ANCBA)*. ISBN 978-987-537-180-4. Pág. 67-111 (PUBLICADO).
4. Marecos, T., Agostini, F., La Red Martínez, D., Acosta, J. (2023). Migración inteligente de procesos en sistemas distribuidos heterogéneos. *Congreso Nacional de Ingeniería Informática / Sistemas de Información (CoNaII SI)*. Presentación oral. (PUBLICADO).
5. Marecos, T., Agostini, F., La Red Martínez, D. (2023). Migración inteligente de procesos en sistemas distribuidos heterogéneos. *Sociedad Científica del Paraguay – VIII Encuentro de Investigadores*. Póster. (PRESENTADO).