



ISSN: 1646-9895

Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

F e v e r e i r o 2 1 • F e b r u a r y 2 1



©RISTI 2021 <http://www.risti.xyz>

Nº E42

Edição / Edition

N.º E42, 02/2021

ISSN: 1646-9895

Indexação / Indexing

Academic Journals Database, CiteFactor, Dialnet, DOAJ, DOI, EBSCO, GALE, Index-Copernicus, Index of Information Systems Journals, Latindex, ProQuest, QUALIS, SCImago, SCOPUS, SIS, Ulrich's.

Publicação / Publication

RISTI – Revista Ibérica de Sistemas e Tecnologias de Informação
Rua Quinta do Roseiral 76, 4435-209 Rio Tinto, Portugal

Web: <http://www.risti.xyz>

Director

Álvaro Rocha, Universidade de Lisboa, PT

Coordenadores da Edição / Issue Coordinators

Teresa Guarda, Universidad Estatal Península de Santa Elena, EC

Datzania Villao, Universidad Estatal Península de Santa Elena, EC

Marcelo Leon, Universidad Tecnológica Empresarial de Guayaquil, EC

Conselho Editorial / Editorial Board

Carlos Ferrás Sexto, Universidad de Santiago de Compostela, ES

Gonçalo Paiva Dias, Universidade de Aveiro, PT

Jose Antonio Calvo-Manzano Villalón, Universidad Politécnica de Madrid, ES

Manuel Pérez Cota, Universidad de Vigo, ES

Ramiro Gonçalves, Universidade de Trás-os-Montes e Alto Douro, PT

Conselho Científico / Scientific Board

A. Augusto Sousa, FEUP, Universidade do Porto, PT

Abel Méndez Porras, Instituto Tecnológico de Costa Rica, CR

Abel Suing, Universidad Técnica Particular de Loja, EC

Adolfo Lozano-Tello, Universidad de Extremadura, ES

Adrián Hiebra Pardo, Universidad de Santiago de Compostela, ES

Alberto Fernández, Universidad Rey Juan Carlos, ES

Alberto Freitas, FMUP, Universidade do Porto, PT

Alcinia Zita Sampaio, IST, Universidade de Lisboa, PT

Alejandro Peña, Escuela de Ingeniería de Antioquia, CO

Alexandre L'Erario, Universidade Tecnológica Federal do Paraná, BR

Alicia García-Holgado, Universidad de Salamanca, ES

Alma Gomez-Rodríguez, Universidade de Vigo, ES

Ana Amélia Carvalho, Universidade de Coimbra, PT

Ana Beatriz Blanco-Ariza, Universidad Simón Bolívar, CO

Ana Isabel Veloso, Universidade de Aveiro, PT

Ana Maria Correia, ISEGI, Universidade Nova de Lisboa, PT

Ana Paula Afonso, Instituto Politécnico do Porto, PT

Anabela Mesquita, Instituto Politécnico do Porto, PT

Índice / Index

EDITORIAL

Nuevas tendencias en la evolución y transformación digital: Volumen II.....ix
Teresa Guarda, Datzania Villao, Marcelo Leon

ARTIGOS / ARTICLES

Identificación de relaciones entre medicamentos en
base a códigos extraídos de un sistema informático 1
Lisbeth Leiva, Ruth Reátegui

La Computación en la Nube en el proceso formativo en Programación Web.....10
Pablo Alejandro Quezada-Sarmiento, Cristóbal Suárez-Guerrero

Estimación de cloruros por medio de sensores remotos con
Sentinel 2 usando redes neuronales en el Lago de Tota, Colombia 20
Luis M. Guerrero, Luz Angela Rocha, Javier Medina

Uso de la metodología STEAM para motivar a
niños el uso de Inteligencia Artificial.....31
Juan José Játiva, Jefferson Beltrán Morales

Modelo de apropiación de Tecnologías de la
Información Geográfica como apoyo a la consolidación
de Ciudades Inteligentes, caso de estudio Bogotá (Colombia)..... 46
Yeni-Andrea Nieto-Vera, Luz-Ángela Rocha-Salamanca

Desarrollo e implementación de sistema de gestión integral de calidad59
*Manuel Andrés López Martínez, Nora Osuna-Millán,
Jesús Manuel Niebla Zatarain Ricardo Rosales Esperanza Manrique Rojas*

Máquina Social: um paradigma emergente ainda desconhecido..... 70
Brunno Wagner Lemos de Souza, Silvio Romero de Lemos Meira

Propuesta de implementación de un modelo
conceptual de infraestructura de datos espaciales
para el departamento del Meta, Colombia..... 90
Helber Leonardo Gutierrez Guerrero, Luz Angela Rocha Salamanca

Implementación de un Dashboard para el apoyo
en la toma de decisiones en el sector privado en Distribución
de Maquinaria y Materia Prima para productos de Panificación.....103
Rodrigo Cuevas Hernández, Hilda Beatriz Ramírez Moreno

KARE: Metodología para el desarrollo de prototipos de Interfaces Tangibles de Usuario basada en la definición de la Arquitectura de Información.....	245
<i>Javier M Reyes V, Paola-J Rodríguez-C, Ivette Kafure Muñoz</i>	
Sistema IoT para la monitorización de niveles de ruido en zonas aledañas al aeropuerto de Cartagena de Indias	257
<i>Harold Adrián Peña R, Gabriel Elías Chanchí G, Wilmar Yesid Campo M</i>	
Sistema IoT para el seguimiento y análisis del nivel de saturación y ritmo cardíaco en el diagnóstico temprano de Covid-19	272
<i>Gabriel Elías Chanchí G, María Clara Gómez A, Claudia Estella Hernández-Londoño</i>	
Perspetivas sobre o e-Government em Portugal: Um Estudo Qualitativo.....	286
<i>Adriana Zoqui de Freitas Cayres Nishimura, Manuel Au-Yong-Oliveira</i>	
Técnicas de ajuste de hiperparámetros de algoritmos de aprendizaje automático para la estimación de esfuerzo: un mapeo de literatura.....	305
<i>Leonardo Villalobos-Arias, Christian Quesada-López, Alexandra Martínez, Marcelo Jenkins</i>	
UDLOntology: An ontology for education in the diversity	319
<i>María Viola Deambrosis, Regina Motz and María Amelia Eliseo</i>	
IoT nas Smart Cities: Revisão da literatura	330
<i>Pedro Oliveira, Isabel Pedrosa, Jorge Bernardino</i>	
Uma Proposta de Chatbot Para Apoio a Gestantes no Contexto do Sistema de Saúde Brasileiro	344
<i>João Benedito dos Santos Junior, Julia Gomes, Julia da Silva Dias, Larissa Neris Oliveira de Souza, Ana Carolina Nogueira Zanotti, Raeli Pereira Dias, Ângela Borges de Carvalho</i>	
Modelo de Decisión para Gestión de Procesos y Recursos en Sistemas Distribuidos con Balanceo de Carga de Trabajo.....	353
<i>Jorge Tomás Fornerón Martínez, Federico Agostini, David Luis La Red Martínez</i>	
Uma Proposta de ChatBot para Tele Orientação sobre Aleitamento Materno	367
<i>João Benedito dos Santos Junior, Lara Dias, Livia Justino Figueiredo, Luis Fernando Coutinho de Brito, Tainara Maria de Souza Abrão, Thatia Regina Bonfim</i>	
Impactos da Lei de Proteção de Dados (LGPD) brasileira no uso da Computação em Nuvem	374
<i>Márcio Aurélio de Souza Fernandes, Fernando Gonçalves de Oliveira, Felipe Silva Ferraz, Daniel Alves da Silva, Edna Dias Canedo, Rafael Timóteo de Sousa Jr</i>	

Modelo de Decisión para Gestión de Procesos y Recursos en Sistemas Distribuidos con Balanceo de Carga de Trabajo

Jorge Tomás Fornerón Martínez¹, Federico Agostini², David Luis La Red Martínez²

jforneron@aplicadas.edu.py; fagostini@exa.unne.edu.ar; lrm david@exa.unne.edu.ar

¹ Facultad de Ciencias Aplicadas. Universidad Nacional de Pilar. Pilar, Paraguay.

² Facultad de Cs. Ex. y Naturales y Agrimensura. Universidad Nacional del Nordeste. Corrientes, Argentina.

Pages: 353-366

Resumen: La aplicación general de modelos de decisión en los sistemas distribuidos funciona con algoritmos de intercambio de permisos y acuerdos sobre procesos que intervienen para realizar acciones. La creación de un operador de agregación que controle esta asignación de recursos a grupos de procesos y el desarrollo de un modelo de decisión que permita adaptarse a diferentes requerimientos, permitiría asegurar la exclusión mutua, la sincronización y lograría la autorregulación y el balanceo dinámico del sistema. En cada nodo se define un *Runtime* (software en tiempo de ejecución complementario al sistema operativo). Existe un coordinador global de *Runtime* en uno de los nodos que evalúa y ejecuta el modelo de decisión y el operador de agregación correspondiente, resultando en el balanceo dinámico y a la autorregulación del sistema. El escenario propuesto es la gestión de recursos y procesos de sistemas distribuidos, aplicados al entorno de las comunicaciones de datos.

Palabras-clave: Modelos de decisión; operadores de agregación; exclusión mutua; sistemas operativos distribuidos; sistemas autorregulados.

Decision Model for Process and Resource Management in Distributed Systems with Workload Balancing

Abstract: The generic application of decision models in distributed systems works with algorithms for exchanging permissions and agreements on processes involved in carrying out actions. The creation of an aggregation operator that controls this allocation of resources to groups of processes and the development of a decision model that can adapt to different requirements, would ensure mutual exclusion, synchronization and achieve self-regulation and dynamic balancing of the system. In each node a Runtime is defined (software in execution time complementary to the operating system). There is a global Runtime coordinator in one of the nodes that evaluates and executes the decision model and the corresponding aggregation operator, resulting in the dynamic balancing and self-regulation of the system. The proposed scenario is the management of resources and processes of distributed systems, applied to the data communications environment.

Keywords: Decision models; aggregation operators; mutual exclusion; distributed operating systems; self-regulating systems.

1. Introducción

En los sistemas de telecomunicaciones, existen un conjunto de nodos, separados físicamente, cada uno con sus propias características y recursos heterogéneos, y conforman lo que se denomina Sistema Distribuido. La implementación de un sistema distribuido se puede realizar sobre cualquier red de comunicaciones, permite la comunicación entre múltiples nodos, y en donde todos los dispositivos forman un solo equipo de cómputo. Para que esto suceda y para que los nodos que poseen múltiples procesos que cooperan entre sí logren una determinada función, se requiere modelos de decisión que permitan a grupos de procesos utilizar recursos compartidos a los que sólo se puede acceder en la modalidad de exclusión mutua (Tanenbaum, 1996), (Naimi & Thiare, 2013), (Fitzek & Katz, 2014), (Cicirelli & Nigro, 2016), (Kamei & Kakugawa, 2017) y (Sakamoto & Ogata, 2018).

Los métodos considerados tradicionales para este problema se encuentran en (Tanenbaum, 1996 y 2009), donde se describen los principales algoritmos de sincronización en sistemas distribuidos. En (Lodha y Kshemkalyani, 2000), (Cao y Singhal, 2001) y (Ricart y Agrawala, 1981) presentan algoritmos para gestionar la exclusión mutua en redes informáticas.

Temas relacionados con la exclusión mutua distribuida y estados globales distribuidos, se abordan en (Stallings, 2005). Algunas soluciones consideradas tradicionales para diferentes tipos de sistemas distribuidos en (Attiya y Welch, 2004), (Guerraoui y Rodrigues, 2006), (Andrews, 2000), (Tel, 2000) y (Lynch, 1996).

La formulación matemática de diferentes conceptos sociales, como la teoría de la elección social y la justicia (Gaertner, 2009) como base para una equitativa distribución de recursos entre los miembros de sistemas distribuidos como se observa en (Pitt et al., 2015), han tomado fuerza a la hora de solucionar problemas de ingeniería teniendo como punto de partida el éxito de las comunidades.

Algunos métodos utilizados para comprender los procesos biológicos se han aplicado a problemas de la ingeniería (Maldonado & Gomez, 2011). El éxito de estos paradigmas es un claro ejemplo de las redes neuronales, lógica difusa, o en inteligencia artificial.

En el mecanismo de elección utilizado en redes de telecomunicaciones, es necesario tener en cuenta la perspectiva desde donde se está tomando la decisión, desde el punto de vista del nodo y desde el punto de vista de la red, como una articulación de las opiniones de los miembros, en donde las acciones públicas están determinadas por los miembros de la sociedad; estas opiniones son agregadas a una preferencia social que refleja la opinión general de la sociedad (Gaertner, 2009).

La elección de una opción entre varias es un proceso en la toma de decisiones que se realiza, mediante un método que pueda resaltar algunas de las opciones de otras, de manera que se puedan organizar jerárquicamente (Saaty, 1990). Para ello, se debe elegir uno o más criterios para poder resaltar esa alternativa de las demás. Entre mayor

número de criterios haya, el problema de elección se vuelve un poco más complejo y se pueden resolver con métodos denominados de optimización multiobjetivo. Si lo posible viene definido por un conjunto discreto de alternativas existen métodos multicriterio discretos para resolver el problema, como el método multicriterio discreto Scoring (Maher et al., 2016) y (Bouyssou et al. 2000).

En este trabajo se presentará una alternativa similar al sistema discreto de Scoring, en el cual, un modelo de decisión, utilizará un operador de agregación específico para resolver el problema mencionado anteriormente. Esto cae dentro de la categoría de operadores de OWA (Ordered Weight Average - Promedio Ponderado Ordenado). El uso de operadores de agregación en modelos de decisión ha sido ampliamente estudiado. En (Yager, 1988 y 1993) y (La Red Martínez y Pinto, 2015), se presenta una revisión de los operadores de agregación, especialmente los de la familia OWA.

En este caso, se desarrolla un nuevo modelo de decisión, que utiliza etiquetas lingüísticas con el objetivo de favorecer el acceso de los grupos de procesos a los recursos solicitados. Algunas de las premisas, estructuras de datos y operadores mencionados en (La Red Martínez, 2017) y (Agostini et al., 2019) se utilizan como punto de partida para crear un nuevo operador y establecer el modelo de decisión para el escenario mencionado.

2. Materiales y Métodos

2.1. Estructura de datos

El objetivo principal de las redes de computadoras es compartir recursos, de tal manera que todos los programas, equipos y datos se encuentren disponibles para quien lo solicite sin importar su ubicación. Para ello se tendrá en cuenta el acceso a recursos compartidos en la modalidad de exclusión mutua distribuida y mediante requisitos de sincronización. Se debe decidir, en función a la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar cada uno. Para satisfacer esto, el permiso de acceso a los recursos compartidos de un nodo no sólo dependerá de si los nodos los utilizan o no, sino también del valor de agregación de las preferencias (prioridades) de los distintos nodos en cuanto a la concesión de acceso a los recursos compartidos (alternativas). Los nodos deberán expresar sus prioridades para la asignación de los diferentes recursos compartidos de acuerdo con las necesidades de recursos de cada proceso.

Las premisas, estructuras de datos y el operador mencionado en (La Red Martínez, 2017) y (Agostini et al., 2019) son aplicadas para resolver el escenario que se describe a continuación. Se operará con lógica difusa, etiquetas lingüísticas y estructuras del tipo 2-tupla, basado en (Dutta et al., 2019).

2.2. Escenario propuesto

En la Figura 1 se muestra una macroimagen del sistema, en la cual, también se consideran transiciones de estados de acuerdo a la actualización iterativa de los requerimientos del sistema, en donde se encuentran los distintos nodos, cuyo estado puede ser activo, cuando interactúa con los demás nodos aunque no envíe información ni participe del uso de recursos o la ejecución de procesos con recursos externos, activo compartido,

cuando interactúa con los demás nodos, enviando y recibiendo información, creando procesos que utilizan recursos de otros nodos y compartiendo recursos propios con nodos externos, inactivo, cuando no se tiene información del estado real del nodo (puede estar sobrecargado y no puede realizar el intercambio de información, o bien, pudo haberse caído el enlace que lo conecta con los demás nodos), por último, existe un nodo central, que a través de un Runtime gestiona la información de todos los nodos que participen en las asignaciones de recursos.

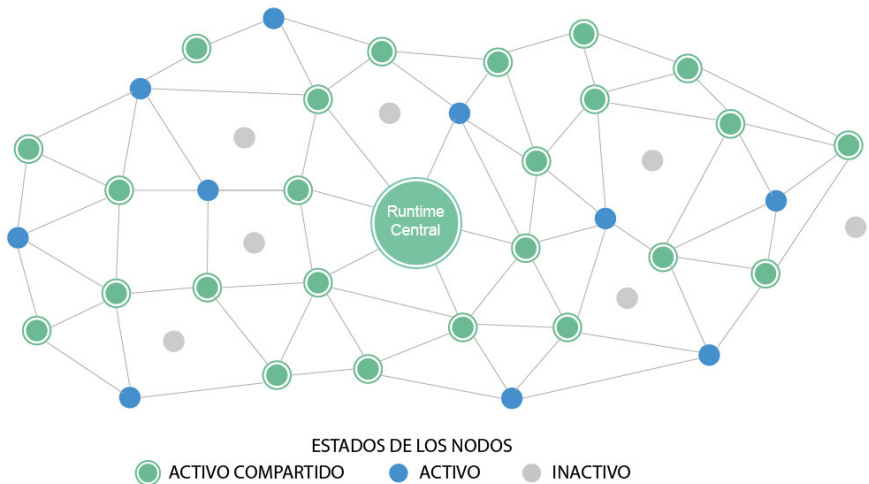


Figura 1 – Macroimagen del estado de los nodos en un Sistema Distribuido

El nodo central, es el encargado de recopilar la información de todos los nodos, aplicar el proceso de agregación y obtener la lista de asignaciones. Un macrociclo contempla la iteración de este proceso, en el cual el modelo de decisión es una etapa que se repite n veces. En este macrociclo hay un paso previo de recopilación de información del estado de red, nodos activos y conectados, cuantos están entregando información, un sondeo previo. Luego de la recepción se continúa con el proceso de evaluación y la emisión de los resultados. Se vuelve a aplicar el loop y el universo va cambiando en cada iteración.

2.3. Operador de agregación utilizado por el MD

En (La Red Martínez, 2017), las características de los operadores de agregación descritos permiten considerar que el método propuesto pertenece a la familia de operadores de agregación Neat-OWA. La propuesta desarrollada en este trabajo consiste en generar un modelo de decisión y su correspondiente operador de agregación para la gestión de grupos de procesos que comparten recursos, realizando modificaciones de los operadores de la familia OWA mencionados.

Este método es una manera rápida y sencilla para realizar una elección en un problema de decisión; el proceso para realizar una decisión Multicriterio con este operador es el siguiente:

- Identificar el propósito del problema.
- Identificar las alternativas.
- Listar los criterios que van a ser tenidos en cuenta a elegir la mejor alternativa.
- Asignar una ponderación para cada uno de los criterios.
- Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo
- Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles y determinación del orden en que se asignarán los recursos y a qué proceso será asignado cada recurso.

Identificar el problema

Este paso consiste en identificar qué es lo que se desea alcanzar. Para este trabajo lo que se intenta resolver son las solicitudes de recursos por parte de los procesos que se encuentran en nodos distribuidos. Para ello se utilizará la expresión de los valores calculados en términos de 2-tupla utilizando un conjunto de etiquetas lingüísticas.

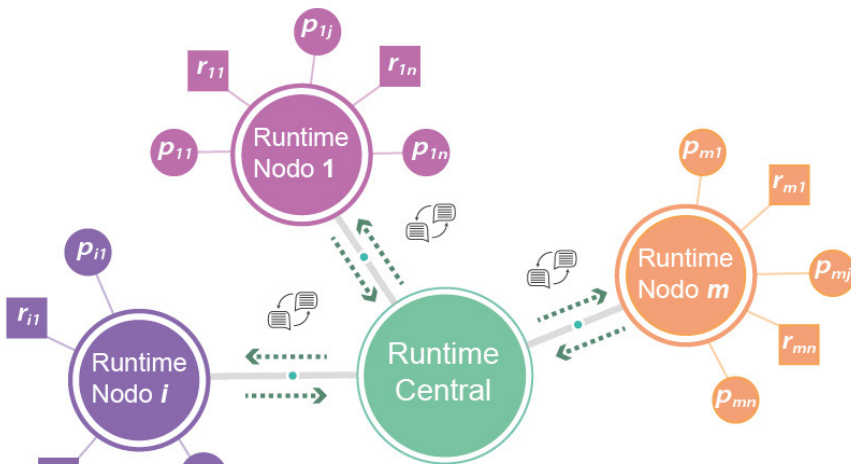


Figura 2 – Traslación Simbólica Ascendente-Descendente

El conjunto de etiquetas utilizado puede variar en cada nodo, es decir, que el conjunto es independiente del nodo central, y es este el encargado de realizar la translación simbólica, proceso que consiste en convertir los conjuntos de etiquetas recibidos en el nodo central, en el conjunto que este último trabaja, el cual debe ser el de mayor número de etiquetas.

De igual manera, esa traslación simbólica se realiza cuando el nodo central devuelve la información final de prioridades a cada nodo. Esto se puede observar en la Figura 2.

Las valoraciones expresadas en un formato lingüístico utilizando las etiquetas lingüísticas y semánticas mencionadas se pueden ver en la Figura 3.

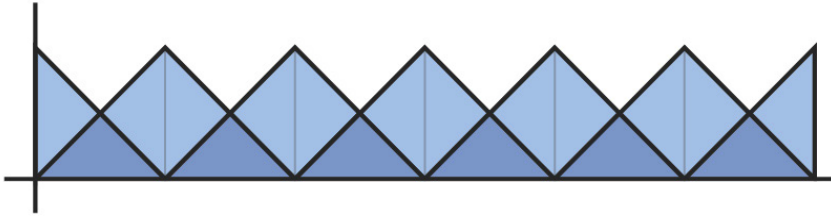


Figura 3 – Representación del conjunto de etiquetas

Identificar las alternativas

Son las opciones que podrían ayudar a la consecución del propósito que se trazó en el primer paso. 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 al 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.

Listar los criterios que van a ser tenidos en cuenta a elegir la mejor alternativa

Se contemplará el cálculo de la carga actual de los nodos. Para obtener un indicador de la carga computacional actual de cada nodo, se pueden adoptar diferentes criterios; en esta propuesta los criterios serán el porcentaje de uso de la CPU, el porcentaje de uso de la memoria y el porcentaje de uso de la operación de entrada/salida.

La carga computacional de cada nodo, el número de criterios para determinar la carga de los nodos, los criterios que se aplican y el cálculo de la carga computacional de cada nodo, son los mencionados en (La Red Martínez, 2017).

Asignar una ponderación para cada uno de los criterios

Es decir, indicar qué tan importante es para el decisor el criterio en cuestión. Una representación lingüística podría representarse mediante la siguiente escala de 1 a 5.

Las categorías de carga computacional actuales de cada nodo, el número de categorías para determinar la carga de los nodos, las categorías que se aplican, los vectores de pesos asociados a las categorías de carga computacional actuales de cada nodo pueden ser representados con la Tabla 1.

Escala	Significado
1	Muy poco satisfactorio
2	Poco satisfactorio
3	Satisfacción media
4	Algo satisfactorio
5	Muy satisfactorio

Tabla 1 – Ponderación de criterios

Para cada alternativa podría darse con el siguiente cálculo:

$$s_j = \sum_i w_i r_{ij}$$

Donde:

r_{ij} = Valor de la alternativa j para el criterio i .

w_i = Ponderación de cada criterio.

S_j = Valoración para cada alternativa j .

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

Estas prioridades se calculan en cada nodo para cada solicitud de recursos originada en cada proceso; el cálculo considera el vector de peso correspondiente según la carga actual del nodo y el vector de los valores concedidos por el nodo según los criterios de evaluación de la solicitud.

Los vectores de valoración que se aplicarán para cada solicitud de un recurso por un proceso, de acuerdo con los criterios establecidos para la determinación de la prioridad que en cada caso y momento fijará el nodo en el que se produce la solicitud.

Los valores son representados en el formato de 2-tupla, considerando las etiquetas lingüísticas propuestas anteriormente. Por lo tanto, el valor de cada criterio deberá compararse con el valor medio que tendrá cada etiqueta, la diferencia mínima de esa comparación será la etiqueta apropiada. Algunos de estos conceptos pueden verse en (Dutta et al., 2019).

El primer elemento de la 2-tupla será el valor lingüístico de esa etiqueta. El segundo elemento será la diferencia entre el valor de los criterios buscados y el valor medio de la etiqueta seleccionada.

dm = la diferencia mínima entre las diferencias de cpm (criterios) y el valor más representativo de cada etiqueta lingüística.

En resumen, la prioridad nodal (que debe calcularse en el nodo en el que se produce la solicitud) de un proceso de acceso a un recurso determinado (que puede ser en

cualquier nodo) se calcula mediante el producto escalar de los vectores mencionados: prioridad nodal ($r_{ij} p_{kl}$) = $\sum w_{o_m} * T(\text{etiqueta}_m; dm_m) = T(\text{etiqueta}_n; dm_n) = TPN_{ijkl}$ (Tupla de Prioridad Nodal) con la o indicando el vector de pesos según la carga del nodo, manteniendo todos los demás subíndices los significados explicados anteriormente. Con m y n indicando la etiqueta lingüística correspondiente dentro del conjunto adoptado definido anteriormente.

Las valoraciones de los criterios establecidos para la representación de las etiquetas lingüísticas se pueden representar de la siguiente manera: 1) Extremadamente bajo; 2) Muy bajo; 3) Bajo; 4) Relativamente bajo; 5) Medio; 6) Relativamente alto; 7) Alto; 8) Muy alto; 9) Extremadamente alto. La satisfacción de criterios, indica, para cada alternativa; en cuanto se satisface cada uno de los criterios en una escala de 1 a 9.

Esta prioridad nodal debe transformarse en el formato de 2-tuplas, teniendo en cuenta las etiquetas lingüísticas ya mencionadas. Por lo tanto, será necesario comparar cada valor de prioridad nodal con el valor medio de cada etiqueta, la diferencia mínima de estas comparaciones indicará la etiqueta correspondiente.

Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles y determinación del orden en que se asignarán los recursos y a qué proceso será asignado cada recurso

Para calcular las prioridades finales, se colocan las prioridades o preferencias nodales calculadas en la etapa anterior; en esta tabla cada fila contiene la información de las prioridades nodales de los diferentes procesos para acceder a un determinado recurso.

A continuación, es necesario calcular el vector de pesos finales que se utilizará en el proceso de agregación para determinar el orden o la prioridad de acceso a los recursos.

El siguiente paso es normalizar los pesos recién obtenidos dividiendo cada uno de ellos por la suma de todos. Así se obtiene un vector de peso normalizado (en el rango de 0 a 1 inclusive) y con la restricción de que la suma de los elementos del vector debe dar 1.

Las prioridades nodales tomadas fila por fila para cada recurso serán multiplicadas escalarmente por el vector de peso final normalizado. De esta manera es posible obtener las prioridades globales finales de acceso de cada proceso a cada recurso. El orden o la prioridad con que se asignarán los recursos y a qué proceso se asignará cada uno, estará determinado por el cálculo de la Prioridad global final ($r_{ij} p_{kl}$) = $TPN_{ijkl} = TPGF_{ijkl}$ (Tupla de prioridad global final) con el r_{ij} indicando el recurso j del nodo i , TPN_{ijkl} es el formato de 2-tuplas, ij indicando el recurso j del nodo i , kl el proceso l del nodo k y el producto de la prioridad global final del proceso para acceder a dicho recurso.

El mayor de estos productos realizados para los diferentes procesos en relación con el mismo recurso indicará cuál de los procesos tendrá acceso al recurso.

La suma de todos estos productos en relación con el mismo recurso indicará la prioridad que se le asignará a ese recurso, en relación con los demás recursos que también deberán asignarse. Esto es lo que se denominará Función de Asignación de Sistemas Distribuidos Lingüística (FASDL):

$$FASDL(r_{ij}) = \sum TPGFN_{ijkl} = \text{prioridad de asignación de recursos } r_{ij}.$$

Calculando el *FASDL* para todos los recursos se obtendrá un vector 2-tupla, y ordenando sus elementos de mayor a menor, se obtendrá el orden de prioridad de asignación de recursos.

Además, como ya se ha indicado, el mayor de los productos $TPGFN_{ijkl}$ para cada recurso indicará el proceso al que se asignará el recurso.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas; cabe señalar que los recursos asignados estarán disponibles una vez que los procesos los liberen y, por lo tanto, podrán asignarse a otros procesos.

Dado que el sistema se autorregula liberando los recursos ya asignados a los procesos en el paso anterior, y dado que hay solicitudes de recursos de los procesos que aún no han sido satisfechas.

3. Modelo de decisión

El objetivo principal del modelo propuesto es considerar el entorno de ejecución distribuida de procesos, que podrán estar agrupados, con distintas exigencias de consenso respecto del acceso a recursos compartidos, para seleccionar el método de agregación más adecuado a cada escenario posible, para generar la secuencia de asignación de los recursos a los procesos que los solicitan, respetando la exclusión mutua en el acceso a dichos recursos.

El modelo considera la información proveniente de la caracterización de los procesos (si los procesos pertenecen a grupos o son independientes, y si la aplicación a la que pertenecen requiere algún tipo de consenso respecto del acceso a los recursos), las prioridades nodales, estado de los nodos, recursos, procesos y del sistema para producir una solución global satisfactoria. Este método evalúa esta información, para determinar cuál es el escenario según la carga de trabajo proporcionada, y en base a ello elige el operador de agregación correspondiente. En el modelo propuesto, en cada nodo se define una interfaz entre las aplicaciones y el sistema operativo, ver Figura 4.

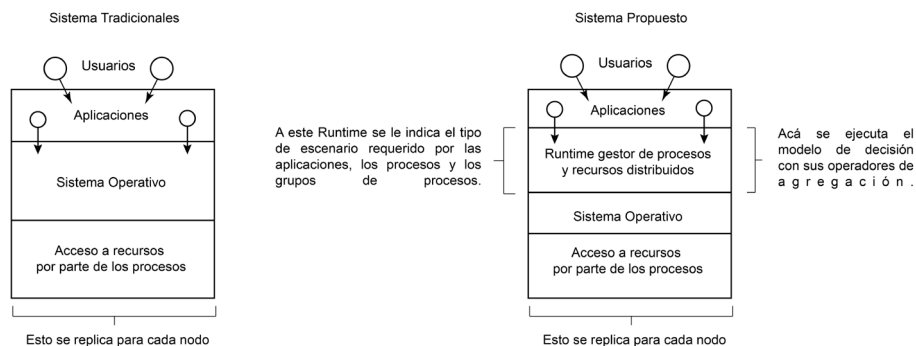


Figura 4 – Comparación del modelo propuesto contra los modelos tradicionales

Comparación de las diferencias más significativas del método propuesto con respecto a los métodos tradicionales: se consideran el algoritmo centralizado, el Algoritmo Distribuido de Lamport, Ricart y Agrawala y el Algoritmo de *Token Ring* (Colouris et al., 1988) y (Tanenbaum & Van Steen, 2001). El algoritmo propuesto establece el orden inicial de asignación por prioridades, mientras que los demás lo hacen teniendo en cuenta la *timestamp*. Los requisitos de acceso a recursos en el algoritmo propuesto se establecen calculando prioridades, mientras que en el algoritmo centralizado es por mensajes al coordinador, el Algoritmo Distribuido de Lamport, Ricart y Agrawala a través de mensajes a procesos $n-1$ y el Algoritmo *Token Ring* a través de mensajes a sus vecinos.

El método de asignación en el algoritmo propuesto contempla la mayor prioridad, mientras que en los demás se tiene en cuenta la *timestamp* más antigua. Sólo el método propuesto tiene en cuenta los grupos de procesos, mientras que los demás sólo tienen en cuenta procesos independientes. La predisposición (prioridad nodal), carga de nodos, estado nodal (nodos, procesos, grupos, recursos) y estado global del sistema, sólo se tiene en cuenta en el algoritmo propuesto. Algunas de las diferencias más significativas con respecto a los métodos considerados tradicionales se mencionan a continuación. Con respecto al orden inicial con el que se establecen las asignaciones, el método propuesto lo hace por prioridades, mientras que los demás lo hacen por marca de tiempo. Las solicitudes de recursos del método propuesto se realizan mediante el cálculo de prioridades, mientras los tradicionales lo hacen por mensajes al coordinador, mensajes a $n-1$ procesos o mensajes al vecino. En los métodos de asignación, el propuesto lo hace por mayor prioridad calculada, mientras que los otros consideran la menor marca de tiempo o la prioridad inicial de los procesos. El método propuesto considera los grupos de procesos mientras que los tradicionales solo tienen en cuenta procesos independientes. El método mencionado considera la predisposición (prioridad nodal), carga del nodo, el estado nodal (nodos, procesos, grupos, recursos) y el estado global del sistema, características que no son tenidas en cuenta en los métodos tradicionales.

3.1. Prioridad nodal colaborativa

En los sistemas distribuidos, se utilizan distintos protocolos de enrutamiento, que permiten la incorporación de nodos a la red (por ejemplo, en una red Ad Hoc). No existe un registro en el sistema sobre la incorporación de un nodo ni de los recursos que este puede ofrecerle a la red, cada nodo puede salir perjudicado en cuanto a cantidad de recursos si ingresa a una red que no llene sus expectativas y que por el contrario sólo le solicite recursos. La red puede estar aceptando un nodo que es egoísta y que solo va a explotar recursos de la red y en realidad no va a aportar nada (Toh et al., 2010) y (Loo et al., 2012).

En este ámbito, se podría tener un escenario muy acorde al incluir la capacidad de toma de decisiones en los nodos, de manera que autónomamente indiquen la prioridad colaborativa que tiene cada uno, dependiendo de la cantidad y el orden en el que ofrezcan sus recursos.

Partiendo de la idea aplicada a los sistemas de grid, pero adecuándola a un concepto totalmente colaborativo, esta solución es superadora del problema mencionado anteriormente, dado que el algoritmo proporciona el orden de asignaciones, pero no hay restricciones en el acceso a los recursos, es un esquema colaborativo.

Ningún nodo restringe el acceso a sus recursos, solamente se ordena el acceso en base a las pautas de prioridad, consumo de recursos, etc. El modelo opera en base a distintos ciclos, recoge información y resuelve, posterior a cada finalización de ciclo, se calcula una prioridad colaborativa, donde se agrega un ítem más de prioridad, que va a entrar en el cálculo de las prioridades nodales, donde en función de la cantidad de recursos y la prioridad de otorgar recursos de un nodo, obtenga más prioridad para acceder a sus propios requerimientos. En este ciclo de realimentación, se genera una nueva variable que marca la prioridad colaborativa, y en donde se da más prioridad al nodo que más compartió recursos.

Se implementa un indicador de prioridad por nodo, que depende de la cantidad de recursos y del orden en cual los ofreció. Para el cálculo se utiliza un vector de pesos asociado a orden y cantidad, y es un vector variable, dado que se podría evaluar que se va a ponderar más, si la cantidad de recursos ofrecidos por el nodo, o el orden en el cual los ofrece. Por ejemplo, dos nodos que compartieron la misma cantidad de recursos cada uno, pero uno los compartió de entrada y el otro lo hizo al final, se debería poder balancear ambos criterios, mediante un vector de ponderación en función de la cantidad de recursos y el orden, para obtener un valor de prioridad. Una ponderación sería priorizar de igual manera los dos criterios, el coeficiente sería igual para ambos, también se podría priorizar el orden, por sobre el de la cantidad, o viceversa, por lo cual los valores de coeficientes serían distintos.

Como resultado del ciclo de realimentación, se tendría una variable más, actualizada y ponderada, que indicara que tan colaborativo fue el nodo con los demás, el que fue poco colaborativo va a tener un valor de prioridad colaborativa muy baja, porque habrá tenido muy pocos recursos que ofreció, o porque los ofreció con un orden muy bajo, y otro nodo muy colaborativo va a tener mayor prioridad. Esto **contribuye al balanceo dinámico y a la autorregulación del sistema.**

4. Conclusión

4.1. Evaluación

Se proporciona una solución global, donde se resalta el dinamismo, la magnitud de los nodos y de los procesos y de la cantidad de requerimientos, que pueden ser aplicables a grandes sistemas, pero también para sistemas de menor cantidad de nodos y requerimientos. Esa variación se explica porque la carga de tráfico, de procesos y de requerimientos varía, sistemas que estaban operando dejan de operar porque terminan y aparecen otros, de pronto se activan nodos con recursos y procesos que hacen falta, los nodos pueden estar activos, pero son incorporados al algoritmo cuando algún proceso solicita algún recurso, o algún recurso es solicitado por otro proceso de otro nodo. Un nodo puede estar activo, pero no formar parte de las evaluaciones de asignaciones.

El modelo propuesto logra establecer un consenso que permite a los grupos de procesos acceder a todos sus recursos de manera secuencial y que éstos no puedan ser removidos hasta que el mismo grupo de procesos que los mantiene, los libera. El orden de asignación será determinado por la prioridad promedio general de todas las asignaciones de cada grupo. El sistema distribuido regula y actualiza constantemente el estado local de

cada nodo, las decisiones de acceso a los recursos modifican estos estados por lo que debe ser reajustado repetidamente, garantizando la exclusión mutua y reordenando nuevas prioridades. El método debe repetirse siempre que haya grupos de procesos que requieran recursos compartidos.

4.2. Futuras líneas de investigación

Como trabajo futuro, se prevé desarrollar variantes del método propuesto considerando otros operadores de agregación (especialmente la familia OWA), utilizando etiquetas lingüísticas para establecer los criterios de evaluación de los estados nodales, la consideración de la migración controlada de procesos, con el objetivo de disminuir el tráfico de la red en sistemas distribuidos y la clasificación de grupos de procesadores para la agrupación de nodos según rangos que cumplan ciertas características (velocidad, tiempo de cálculo, procesamiento, etc.).

Se considera desarrollar modelos de decisión desde la óptica cognitiva para la toma de decisiones en grupos de procesos, contemplando los principios de la cibernética de segundo orden, en el contexto de sistemas complejos de auto-regulación, que trasciendan el enfoque tradicional de las ciencias de la computación considerando la posibilidad de imputación de datos faltantes.

5. Agradecimientos

Este trabajo ha sido apoyado por el Proyecto: “Modelos de decisión y operadores de agregación para la gestión de procesos en sistemas distribuidos”, código 16FO01 de la Universidad Nacional del Nordeste (Argentina).

Referencias

- Agostini, F., Fornerón Martínez J. T. & La Red Martínez D. L. (2019). Nuevo operador de agregación para grupos de procesos. *16ª Conferencia Ibero Americana*. Lisboa, Portugal. (pp. 223-230), ISBN: 978-989-8533-96-8.
- Andrews, G. (2000). *Foundation of Multithreaded, Parallel, and Distributed Programming*. Reading, MA: Addison-Wesley, USA.
- Attiya, H. & Welch, J. (2004). *Distributed Computing Fundamentals, Simulations, and Advanced Topics*, John Wiley, 2nd ed., New York, USA.
- Bouyssou, D., Marchant, T., Piriot, M., Tsoukias, A. & Vincke, P. (2000). Evaluation and decision models: a critical perspective. *Kluwer Academic Publishers International series in Operations Research & Management Sciences*. Massachusetts. United States of America. (Vol. 32, pp. 262). ISBN 978-0-7923-7250-9.
- Cao, G. & Singhal, M. (2001). A Delay-Optimal Quorum-Based Mutual Exclusion Algorithm for Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*. USA. (Vol. 12, No. 12, pp. 1256-1268).

- Cicirelli, F., & Nigro, L. (2016). *Modelling and Verification of Mutual Exclusion Algorithms*. IEEE/ACM 20th International Symposium on Distributed Simulation and Real Time Applications, pp. 136-144.
- Colouris, G. et al. (1988). *Distributed Systems. Concepts and Design*. Pearson.
- Dutta, B., Labella, A., Rodríguez, R. & Martinez, L. (2019). Aggregating Interrelated Attributes in Multi-Attribute Decision-Making With ELICIT Information Based on Bonferroni Mean and Its Variants. *International Journal of Computational Intelligence Systems*. (Vol. 12, pp. 1179-1196). DOI: 10.2991/ijcis.d.190930.002.
- Fitzek, F. & Katz, M. (2014). *MOBILE CLOUDS: Exploiting Distributed Resources in Wireless, Mobile and Social Networks* L. John Wiley & Sons, ed., New Delhi, India. ISBN: 978-0-470-97389-9
- Gaertner, W. (2009). *A Primer in Social Choice Theory Revised*. Oxford, *Oxford University Press*. ISBN: 9780199565306
- Guerraoui R. & Rodrigues L. (2006). *Introduction to Reliable Distributed Programming*, Berlin, *Springer-Verlag*.
- Kamei, S. & Kakugawa, H. (2017). *An asynchronous message-passing distributed algorithm for the global critical section problem*. Fifth International Symposium on Computing and Networking, pp. 414-419.
- Maher A. and Maysam F. (2016). Classifiers consensus system approach for credit scoring. *Knowledge Based Systems*. (Vol. 104, pp. 89-105). ISSN: 0950- 7051 DOI:10.1016/j.knosys.2016.04.013.
- Maldonado, C.E. & Gomez, N.A. (2011). *Sistemas Bioinspirados: Un Marco Teórico para la Ingeniería de Sistemas Complejos*, Bogotá. ISSN: 0124-8219
- Naimi, M. & Thiare, O. (2013). *A Distributed Deadlock-Free Quorum-Based Algorithm for Mutual Exclusion*. (IJCSIS) International Journal of Computer Science and Information Security, vol. 11, n° 8, pp. 7-13.
- La Red Martínez D. L. (2017). Aggregation Operator for Assignment of Resources in Distributed Systems. *International Journal of Advanced Computer Science and Applications (IJACSA)*. The Science and Information (SAI) Organization, England. U.K. (Vol. 8, No. 10, pp. 406-419). ISSN N° 2156-5570.
- La Red Martínez D. L. & Pinto N. (2015). Brief Review of Aggregation Operators. *Wulfenia Journal*. Austria, (Vol. 22, No. 4, pp 114-137).
- Lodha S. & Kshemkalyani A. (2000). A Fair Distributed Mutual Exclusion Algorithm. *IEEE Trans. Parallel and Distributed Systems*. USA. (Vol. 11, No. 6, pp. 537-549).
- Loo, J., Lloret, J. & Hamilton, J. (2012). *Mobile Ad Hoc Networks: Current Status and Future Trends*, Taylor & Francis Group, LLC. ISBN 9781439856505.
- Lynch N. (1996). *Distributed Algorithms*. Morgan Kauffman, San Mateo, CA, USA.

- Ricart G., Agrawala A. K. (1981). An Optimal Algorithm for Mutual Exclusion in Computer Networks. *Commun. of the ACM*, (Vol. 24, pp 9-17).
- Pitt, J., Busquets, D., & Riveret, R. (2015). The pursuit of computational justice in open systems. *AI & Society*, 30(3), (pp. 359-378). doi:10.1007/s00146-013-0531-6
- Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European journal of operational research*, (48(1), pp. 9-26). [https://doi.org/10.1016/0377-2217\(90\)90057-1](https://doi.org/10.1016/0377-2217(90)90057-1)
- Sakamoto, S. & Ogata, K. (2018). Model Checking of the Suzuki-Kasami Distributed Mutual Exclusion Algorithm with SPIN. 5th International Conference on Dependable Systems and Their Applications (DSA), pp. 136-141.
- Stallings W. (2005). *Sistemas Operativos*, 5ta. Edición, Pearson Educación S.A. España.
- Tanenbaum A. S. (1996). *Sistemas Operativos Distribuidos*. Prentice - Hall Hispanoamericana S.A., México.
- Tanenbaum A. S. (2009). *Sistemas Operativos Modernos*. 3ra. Edición. Pearson Educación S. A., México.
- Tanenbaum A. S. & Van Steen M. (2001). *Distributed Systems: Principles and Paradigms*. Pearson.
- Tel G. (2000). *Introduction to Distributed Algorithms*, Cambridge University Press, 2nd ed. Cambridge, UK.
- Toh, C. K., Kim, D., Oh, S., & Yoo, H. (2010, February). The controversy of selfish nodes in ad hoc networks. In 2010 *The 12th International Conference on Advanced Communication Technology (ICACT)* (Vol. 2, pp. 1087-1092). IEEE.
- Yager R. (1988). On Ordered Weighted Averaging Aggregation Operators in Multi-Criteria Decision Making. *IEEE Trans. On Systems, Man and Cybernetics*. (Vol. 18, No. 1, pp 183-190).
- Yager R. (1993). Families of OWA Operators. *Fuzzy Sets and Systems*. (Vol. 59, No. 2, pp 125-148).

Critérios Editoriais

A RISTI (Revista Ibérica de Sistemas e Tecnologias de Informação) é um periódico científico, que foca a investigação e a aplicação prática inovadora no domínio dos sistemas e tecnologias de informação.

O Conselho Editorial da RISTI incentiva potenciais autores a submeterem artigos originais e inovadores para avaliação pelo Conselho Científico.

A submissão de artigos para publicação na RISTI deve realizar-se de acordo com as chamadas de artigos e as instruções e normas disponibilizadas no sítio Web da revista (<http://www.risti.xyz>).

Todos os artigos submetidos são avaliados por um conjunto de membros do Conselho Científico, não inferior a três elementos.

Em cada número da revista são publicados entre cinco a oito dos melhores artigos submetidos.

Criterios Editoriales

La RISTI (Revista Ibérica de Sistemas y Tecnologías de la Información) es un periódico científico, centrado en la investigación y en la aplicación práctica innovadora en el dominio de los sistemas y tecnologías de la información.

El Consejo Editorial de la RISTI incentiva autores potenciales a enviar sus artículos originales e innovadores para evaluación por el Consejo Científico.

Lo envío de artículos para publicación en la RISTI debe hacerse de conformidad con las llamadas de los artículos y las instrucciones y normas establecidas en el sitio Web de la revista (<http://www.risti.xyz>).

Todos los trabajos enviados son evaluados por un número de miembros del Consejo Científico de no menos de tres elementos.

En cada número de la revista se publican cinco a ocho de los mejores artículos enviados.



Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

©RISTI 2021 <http://www.risti.xyz>

