

Modelado Matemático de un Operador de Agregación Para Asignación de Recursos en Sistemas Distribuidos

Mathematical Modeling of an Aggregation Operator for Resources Allocation in Distributed Systems

David L. La Red Martínez

Dpto. de Informática, Universidad Nacional del Nordeste
9 de julio 1449 (3400), Corrientes, Argentina
lrmdavid@exa.unne.edu.ar

Julio C. Acosta

Dpto. de Matemática, Universidad Nacional del Nordeste
9 de julio 1449 (3400), Corrientes, Argentina
julioa@exa.unne.edu.ar

Stella M. Gerzel

Dpto. de Informática, Universidad Nacional del Nordeste
9 de julio 1449 (3400), Corrientes, Argentina
licgerzelstella@gmail.com

Alice R. Rambo

Dpto. de Informática, Universidad Nacional de Misiones
Félix de Azara 1552 (3300),
Posadas, Argentina
alirambo@gmail.com

Resumen—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 dichos recursos serán asignados a los procesos que los requieren; en este trabajo 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.

PalabrasClave—sistemas operativos; exclusión mutua; comunicación entre grupos de procesos; operadores de agregación.

Abstract—In distributed processing systems it is often necessary to coordinate the allocation of shared resources that should be assigned to the processes in the modality of mutual exclusion; in such cases, the order in which the shared resources will be assigned to the processes that require them must be decided; in this paper we propose an aggregation operator (which could be used by a shared resources manager module) that will decide the order of allocation of the resources to the processes considering the requirements of the processes (shared resources) and the state of the distributed nodes where the processes operate (their computational load).

Keywords—operating systems; mutual exclusion; communication between groups of processes; aggregation operators.

I. INTRODUCCIÓN

La proliferación de sistemas informáticos, muchos de ellos distribuidos en diferentes nodos con múltiples procesos que cooperan para el logro de una determinada función, hace necesario disponer de modelos de decisión que permitan a los grupos de procesos acceder a los recursos compartidos que sólo se pueden acceder en la modalidad de exclusión mutua.

Las soluciones tradicionales para esta problemática se encuentran en [1] y [2], donde se describen los principales algoritmos de sincronización en sistemas distribuidos; en [3], donde se presenta una solución eficiente y tolerante a fallas para el problema de la exclusión mutua distribuida; en [4], [5] y en [6], donde se presentan unos algoritmos para gestionar la exclusión mutua en redes de computadoras; en [7], donde se detallan los principales algoritmos para la gestión distribuida de procesos, los estados globales distribuidos y la exclusión mutua distribuida y en [8] se presentan los principales algoritmos de coordinación distribuida y gestión de la exclusión mutua.

Las soluciones tradicionales para la asignación de recursos compartidos distribuidos en la modalidad de exclusión mutua se concentran en garantizar esta última, sin considerar la carga computacional de los nodos donde operan los procesos y el impacto que sobre dicha carga producirá el acceso a los recursos compartidos solicitados por los procesos. Sin embargo, esta asignación de recursos a los procesos debería hacerse teniendo en cuenta las prioridades de los procesos y también el

estado en cuanto a carga de trabajo de los nodos computacionales en los cuales los procesos se ejecutan.

Los nuevos modelos de decisión para la asignación de recursos compartidos podrían ejecutarse en el contexto de un administrador de recursos compartidos para el sistema distribuido, el cual recibiría los requerimientos de recursos compartidos de los procesos que ejecutan en los distintos nodos distribuidos, como así también el estado de carga computacional de los nodos y, considerando esa información, debería decidir el orden (prioridad) de asignación de los recursos solicitados a los procesos solicitantes de los mismos, para lo cual es necesario disponer de operadores de agregación específicamente diseñados.

En este trabajo se presentará un nuevo operador de agregación específico para la problemática mencionada, que encuadra en la categoría de operadores OWA, más concretamente Neat OWA.

La utilización de operadores de agregación en los modelos de decisión en grupo ha sido estudiada ampliamente, por ejemplo, en [9] y [10] se presentan y analizan los operadores OWA (Ordered Weighted Averaging: Promedio de Pesos Ordenados) aplicados a la toma de decisiones multicriterio; en [11] se presenta un modelo de decisión en grupo con la utilización de operadores de agregación de la familia OWA; en [12] se presenta la utilización de operadores de agregación de la familia OWA para la toma de decisiones y en [13] se presentan metodologías para resolver problemas en presencia de múltiples atributos y criterios.

Este artículo, en el que se presentará un método innovador para la gestión de recursos compartidos en sistemas distribuidos, se ha estructurado de la siguiente manera: en la Sección II explicarán las estructuras de datos, en la Sección III se describirá el operador de agregación, luego se presentarán las Conclusiones y Líneas Futuras de Trabajo, finalizándose con los Agradecimientos y las Referencias.

II. ESTRUCTURAS DE DATOS A UTILIZAR

Se utilizarán 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, cuáles serán las prioridades para asignar los recursos (sólo intervendrán 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 su disponibilidad, 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 (que podrá ser parte de un grupo de procesos).

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

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

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 I.

TABLE I. PROCESOS EN CADA NODO

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

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

grupos = $\{p_{ij}\}$ con i indicando el nodo y j 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:

$$\text{card} = \{\text{card}(g_i)\}$$
 con $i = 1, \dots, g$ 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:

$$\text{prg} = \{\text{prg}_i = \text{card}(g_i)\}$$
 con $i = 1, \dots, g$ 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 II.

TABLE II. RECURSOS COMPARTIDOS DISPONIBLES EN CADA NODO

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

Estos recursos compartidos disponibles alojados en distintos nodos del sistema distribuido podrán ser requeridos por los procesos en ejecución; estas solicitudes de recursos por parte de los procesos se muestran en la Tabla III.

Estados posibles de cada uno de los p procesos:

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

TABLE III. RECURSOS SOLICITADOS POR LOS PROCESOS

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

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 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.

III. DESCRIPCIÓN DEL OPERADOR DE AGREGACIÓN

El operador propuesto consta de las siguientes etapas:

- Cálculo de la carga computacional actual de los nodos.

- Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas.
- Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo (se las calcula en cada nodo para cada proceso).
- Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles (se las calcula en el administrador centralizado de recursos compartidos) y determinación del orden en que se asignarán los recursos y a qué proceso será asignado cada recurso.

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

Cálculo de la carga computacional actual de los nodos

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.

La carga computacional de cada nodo se calculará de la siguiente manera:

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

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

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

$\text{criterios} = \{c_{ij}\}$ 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 IV.

TABLE IV. CRITERIOS PARA MEDIR LA CARGA COMPUTACIONAL EN CADA NODO

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

Eventualmente todos los nodos podrán utilizar el mismo conjunto de criterios.

Cálculo de la carga computacional de cada nodo:

$$\text{carga}_i = (\text{valor}(c_{i1}) + \dots + \text{valor}(c_{ic})) / c \text{ con } i = 1, \dots, n$$

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

Para establecer las categorías de carga computacional actual de cada nodo se pueden adoptar distintos criterios; en esta propuesta las categorías serán: 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%).

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

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

Establecimiento de las categorías que se aplicarán (podrán diferir de un nodo a otro):

$\text{categorías} = \{\text{cat}_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, a$ (n° máximo de categorías para cada nodo), lo que se puede expresar mediante la Tabla V.

TABLE V. CATEGORÍAS PARA MEDIR LA CARGA COMPUTACIONAL EN CADA NODO

| Nodos | Categorías | | | |
|-------|-------------------|-------------------|-------------------|-------------------|
| | 1 | cat_{11} | cat_{12} | |
| | | | ... | |
| i | cat_{i1} | cat_{i2} | | cat_{ia} |
| | | | ... | |
| n | cat_{n1} | cat_{n2} | | cat_{na} |

Eventualmente todos los nodos podrían utilizar el mismo conjunto de categorías.

Para establecer los vectores de pesos asociados a las categorías de carga computacional actual de cada nodo se pueden adoptar distintos criterios; en esta propuesta los criterios serán: 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), 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) y sobrecarga de entrada / salida (entrada / salida adicional que requerirá disponer el recurso solicitado, si el dato está disponible).

Establecimiento del n° de criterios para determinar la prioridad o preferencia que se otorgará en cada nodo según su carga a cada pedido de un recurso compartido hecho por cada proceso:

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

Establecimiento de los criterios que se aplicarán (iguales para todos los nodos):

$\text{criterios para preferencias} = \{\text{cp}_{ij}\}$ con $i = 1, \dots, a$ (n° de categorías de carga computacional) y $j = 1, \dots, e$ (n° máximo de criterios), lo que se puede expresar mediante la Tabla VI.

TABLE VI. CRITERIOS PARA CALCULAR LA PRIORIDAD QUE CADA NODO OTORGARÁ A CADA REQUERIMIENTO DE CADA PROCESO SEGÚN LA CARGA DEL NODO

| Categorías | Criterios | | | |
|------------|------------------|------------------|------------------|------------------|
| | 1 | cp_{11} | cp_{12} | |
| | | | ... | |
| i | cp_{i1} | cp_{i2} | | cp_{ie} |
| | | | ... | |
| a | cp_{a1} | cp_{a2} | | cp_{ae} |

Eventualmente todos los nodos podrían utilizar distintos conjuntos de criterios aplicables a las distintas categorías de carga computacional.

Una vez determinadas las categorías para indicar la carga de los nodos y los criterios que se aplicarán para evaluar la prioridad a otorgar a cada requerimiento de recursos de cada proceso, se podrán establecer los valores correspondientes a los criterios constituyendo así los vectores de pesos para las distintas categorías de carga.

Establecimiento de los vectores de pesos que se aplicarán (iguales para todos los nodos):

$\text{pesos} = \{w_{ij}\}$ con $i = 1, \dots, a$ (n° de categorías de carga computacional) y $j = 1, \dots, e$ (n° máximo de criterios), lo que se puede expresar mediante la Tabla VII.

TABLE VII. 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

| Categorías | Pesos | | | |
|------------|----------|----------|----------|----------|
| | 1 | w_{11} | w_{12} | |
| | | | ... | |
| i | w_{i1} | w_{i2} | | w_{ie} |
| | | | ... | |
| a | w_{a1} | w_{a2} | | w_{ae} |

La asignación de pesos a los distintos criterios será función de estudios estadísticos previamente realizados acerca del sistema distribuido; habrá entonces una función de asignación de pesos a los criterios para constituir los vectores de pesos de cada categoría de carga:

$w_{ij} = \text{norm}(\text{función}(\text{cp}_{ij}))$ con $i = 1, \dots, a$ (n° de categorías) y $j = 1, \dots, e$ (n° de criterios); *norm* indica que los valores deben estar normalizados (en el intervalo de 0 a 1 inclusive) y con la restricción de que la sumatoria de los elementos de un vector de pesos debe dar 1:

$$\sum \{w_{ij}\} = 1 \text{ conj } j = 1, \dots, e \text{ para cada } i \text{ constante. (1)}$$

Esto significa que la sumatoria de los pesos asignados a los distintos criterios será 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 (se las calcula en cada nodo para cada proceso y podría llamárselas prioridades nodales)

Estas prioridades son calculadas en cada nodo para cada petición de recursos originada en cada proceso; el cálculo considera el vector de pesos correspondiente según la carga actual del nodo y el vector de los valores otorgados por el nodo según los criterios de evaluación de la petición. El rango de valores es entre 0 y 1, donde un valor cercano a 0 significa que el criterio relacionado aportará poco al cálculo de la prioridad de la petición en tanto que un valor cercano a 1 significa lo contrario. Esto permite que un nodo ante una petición de un recurso por parte de un proceso, podrá influir en la misma según su estado y el impacto o carga adicional que significaría asignar el recurso solicitado al proceso solicitante, por ejemplo, si acceder a la petición significa incrementar el uso de memoria y el nodo tiene poca memoria disponible, entonces podría asignar a dicho criterio un valor cercano a 0, a su vez, si el consumo adicional de procesador se considera bajo y el uso de

CPU del nodo es reducido, entonces a dicho criterio se asignaría un valor cercano a 1.

Los vectores de valoraciones que se aplicarán para cada requerimiento de un recurso por parte de un proceso, según los criterios establecidos para la determinación de la prioridad que en cada caso y momento fijará el nodo en el cual se produce el requerimiento, son los siguientes:

valoraciones $(r_{ij} p_{kl}) = \{cp_m\}$ con $i = 1, \dots, n$ (nodo donde reside el recurso), $j = 1, \dots, r$ (recurso en el nodo i), $k = 1, \dots, n$ (nodo donde reside el proceso), $l = 1, \dots, p$ (proceso en el nodo k) y $m = 1, \dots, e$ (criterios de valoración de la prioridad del requerimiento), los que se pueden expresar mediante la Tabla VIII.

TABLE VIII. VALORACIONES ASIGNADAS A LOS CRITERIOS PARA CALCULAR LA PRIORIDAD O PREFERENCIA QUE CADA NODO OTORGARÁ A CADA REQUERIMIENTO DE CADA PROCESO SEGÚN LA CARGA DEL NODO

| Recursos - Procesos | Criterios | | | | |
|---------------------|-----------|------|--------|------|--------|
| | CP_1 | | CP_m | | CP_e |
| $r_{11} p_{11}$ | CP_1 | | CP_m | | CP_e |
| | | | | | |
| $r_{ij} p_{kl}$ | CP_1 | | CP_m | | CP_e |
| | | | | | |
| $r_{nr} p_{np}$ | CP_1 | | CP_m | | CP_e |

Resumiendo, la prioridad nodal (por ser calculada en el nodo en el que se produce la petición) de un proceso para acceder a un recurso determinado (que puede estar en cualquier nodo) se calcula mediante el producto escalar de los vectores mencionados anteriormente:

prioridad nodal $(r_{ij} p_{kl}) = \sum w_{om} * cp_m$ con o indicando el vector de pesos según la carga del nodo, manteniendo los demás subíndices los significados explicados anteriormente. (2)

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

En esta etapa se consideran las prioridades nodales calculadas en la etapa anterior para cada requerimiento de acceso a los recursos por parte de los procesos. A partir de estas prioridades nodales se deben calcular las prioridades globales o finales, es decir, con qué prioridad, o sea en qué orden, los recursos solicitados serán otorgados y a qué procesos se hará dicho otorgamiento. Los requerimientos que no puedan ser atendidos por resultar con bajas prioridades, serán nuevamente considerados en la siguiente iteración del método.

Para el cálculo de las prioridades finales se utiliza la Tabla IX, en la cual 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 distintos procesos para acceder a un determinado recurso.

Seguidamente corresponde calcular el vector de pesos finales que se utilizará en el proceso final de agregación para determinar el orden o prioridad de acceso a los recursos.

pesos finales = $\{wf_{kl}\}$ con $k = 1, \dots, n$ (n° de nodos) y $l = 1, \dots, p$ (n° máximo de procesos por nodo), lo que se puede expresar mediante la Tabla X, donde np es el número de procesos en el sistema y prg_i es la prioridad del grupo de procesos al que pertenece el proceso (explicada en la Sección anterior).

TABLE IX. PRIORIDADES NODALES DE LOS PROCESOS PARA ACCEDER A CADA RECURSO

| Recursos | Prioridades Nodales de los Procesos | | | | |
|----------|-------------------------------------|------|----------|------|----------|
| | p_{11} | | p_{kl} | | p_{np} |
| r_{11} | p_{11} | | p_{kl} | | p_{np} |
| | | | | | |
| r_{ij} | p_{11} | | p_{kl} | | p_{np} |
| | | | | | |
| r_{nr} | p_{11} | | p_{kl} | | p_{np} |

TABLE X. PESOS ASIGNADOS A LOS PROCESOS PARA CALCULAR LA PRIORIDAD O PREFERENCIA FINAL DE ACCESO A LOS RECURSOS

| Procesos | Pesos Finales | |
|----------|---------------------------------|---------------------|
| | Si integra un grupo de procesos | Si es independiente |
| p_{11} | $wf_{11}=(prg_i)/np$ | $wf_{11}=1/np$ |
| | | |
| p_{kl} | $wf_{kl}=(prg_i)/np$ | $wf_{kl}=1/np$ |
| | | |
| p_{np} | $wf_{np}=(prg_i)/np$ | $wf_{np}=1/np$ |

El siguiente paso es normalizar los pesos recientemente obtenidos dividiendo cada uno por la sumatoria de todos ellos, lo cual se indica en la Tabla XI.

TABLE XI. PESOS FINALES NORMALIZADOS ASIGNADOS A LOS PROCESOS PARA CALCULAR LA PRIORIDAD O PREFERENCIA FINAL DE ACCESO A LOS RECURSOS

| Procesos | Pesos Finales Normalizados |
|----------|-------------------------------------|
| p_{11} | $nwf_{11} = wf_{11} / \sum wf_{kl}$ |
| | |
| p_{kl} | $nwf_{kl} = wf_{kl} / \sum wf_{kl}$ |
| | |
| p_{np} | $nwf_{np} = wf_{np} / \sum wf_{kl}$ |

Es así como se obtiene un vector de pesos normalizados (en el intervalo de 0 a 1 inclusive) y con la restricción de que la sumatoria de los elementos del vector debe dar 1:

$\sum \{nwf_{kl}\} = 1$ con $k = 1, \dots, n$ (n° de nodos) y $l = 1, \dots, p$ (n° máximo de procesos por nodo). (3)

Las prioridades nodales indicadas en la Tabla IX tomadas fila por fila, es decir, respecto de cada recurso, se multiplicarán escalarmente por el vector de pesos finales normalizados indicado en la Tabla XI para obtener las prioridades globales finales de acceso de cada proceso a cada recurso y de allí, el orden o prioridad con que se asignarán los recursos y a qué proceso de asignará cada uno de ellos; esto se indica a continuación.

prioridad final global $(r_{ij} p_{kl}) = nwf_{kl} * p_{kl}$ con r_{ij} indicando el recurso j del nodo i , p_{kl} el proceso l del nodo k y el producto

la prioridad final global de dicho proceso para acceder al mencionado recurso. El mayor de estos productos hechos para los distintos procesos en relación al mismo recurso indicará cuál de los procesos tendrá acceso al recurso. (4)

La sumatoria de todos estos productos en relación al mismo recurso indicará la prioridad que tendrá dicho recurso para ser asignado, en relación a los demás recursos que también tendrán que ser asignados. Esto constituye lo que se denominará Función de Asignación para Sistemas Distribuidos (FASD):

$$FASD(r_{ij}) = \sum nwf_{kl} * p_{kl} = \text{prioridad de asignación del recurso } r_{ij}. \quad (5)$$

Calculando la FASD para todos los recursos se obtendrá un vector y , ordenando sus elementos de mayor a menor se obtendrá el orden prioritario de asignación de los recursos. Además, como ya se ha indicado, el mayor de los productos $nwf_{kl} * p_{kl}$ respecto de cada recurso indicará el proceso al cual será asignado el recurso. Esto se indica en la Tabla XII.

TABLE XII. ORDEN O PRIORIDAD FINAL DE ASIGNACIÓN DE LOS RECURSOS Y PROCESO AL CUAL SE ASIGNA CADA RECURSO

| Orden de asignación de los recursos | Proceso al que se asignará el recurso |
|--|--|
| 1°: r_{ij} del Máximo($FASD(r_{ij})$) | p_{kl} del Máximo($nwf_{kl} * p_{kl}$) para el r_{ij} seleccionado |
| 2°: r_{ij} del Máximo($FASD(r_{ij})$) para los r_{ij} no asignados | p_{kl} del Máximo($nwf_{kl} * p_{kl}$) para el r_{ij} seleccionado |
| **** | **** |
| último: r_{ij} no asignado | p_{kl} del Máximo($nwf_{kl} * p_{kl}$) para el r_{ij} seleccionado |

Por razones de espacio no se agrega un ejemplo del uso del operador y mayores consideraciones respecto de la tipificación del operador.

IV. CONCLUSIONES

El modelo propuesto logra que el sistema distribuido se auto regule reiteradamente en función del estado local de los n nodos, produciéndose una actualización de los estados locales de los mismos como consecuencia de la evolución de sus respectivos procesos y de las decisiones de acceso a los recursos: 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, garantizándose además la exclusión mutua en el acceso a los recursos compartidos, indicándose la prioridad de otorgamiento de acceso a cada recurso y el proceso al cual se lo asigna; este

proceso se repite mientras haya procesos que soliciten acceso a recursos compartidos.

El modelo propuesto incluye como caso particular uno de los métodos más utilizados, consistente en considerar sólo la prioridad de los procesos, en vez de un grupo de variables de estado de cada nodo.

Otra característica destacable de la propuesta es su facilidad de implementación en el entorno de un administrador centralizado de recursos compartidos de un sistema distribuido.

Se tiene previsto desarrollar variantes del método propuesto considerando otros operadores de agregación (especialmente de la familia OWA) y la posibilidad de que sea utilizado por un administrador de recursos compartidos distribuido (en vez de centralizado como en el método propuesto).

AGRADECIMIENTOS

Este trabajo ha sido realizado en el contexto del proyecto de investigación: "Modelos de decisión para sincronización de procesos en sistemas distribuidos", código 12F003 de la Universidad Nacional del Nordeste, Argentina (Resolución N° 960/12 C.S.).

REFERENCIAS BIBLIOGRÁFICA

- [1] A. S. Tanenbaum. *Sistemas Operativos Distribuidos*. Prentice - Hall Hispanoamericana S.A. México, 1996.
- [2] A. S. Tanenbaum. *Sistemas Operativos Modernos*. 3ra. Edición. Pearson Educación S. A. México, 2009.
- [3] D. Agrawal and A. El Abbadi. *An Efficient and Fault-Tolerant Solution of Distributed Mutual Exclusion*. ACM Trans. on Computer Systems. Vol. 9. Pp. 1-20. USA, 1991.
- [4] G. Ricart, and A. K. Agrawala. *An Optimal Algorithm for Mutual Exclusion in Computer Networks*. Commun. of the ACM. Vol. 24. Pp. 9-17, 1981.
- [5] G. Cao and M. Singhal. *A Delay-Optimal Quorum-Based Mutual Exclusion Algorithm for Distributed Systems*. IEEE Transactions on Parallel and Distributed Systems. Vol. 12, no. 12. Pp. 1256-1268. USA, 2001.
- [6] S. Lodha and A. Kshemkalyani. *A Fair Distributed Mutual Exclusion Algorithm*. IEEE Trans. Parallel and Distributed Systems. Vol. 11. N° 6. Pp. 537-549. USA, 2000.
- [7] W. Stallings. *Sistemas Operativos*. 5ta. Edición. Pearson Educación S.A. España, 2005.
- [8] A. Silberschatz, P. B. Galvin and G. Gagne. *Fundamentos de Sistemas Operativos*. 7ma. Edición. McGraw-Hill / Interamericana de España. S.A.U. España, 2006.
- [9] R. Yager. *On Ordered Weighted Averaging Aggregation Operators In Multi-Criteria Decision Making*. IEEE Trans. On Systems, Man and Cybernetics 18: 183-190, 1988.
- [10] R. Yager. *Families Of OWA Operators*. Fuzzy Sets and Systems. 59: 125-148, 1993.
- [11] J. M. Doña, A. M. Gil, J. I. Peláez and D. L. La Red Martínez. *A System Based on the Concept of Linguistic Majority for the Companies Valuation*. Revista EconoQuantum. V. 8 N° 2. Pp. 121-142. México, 2011.
- [12] R. Fullér. *OWA Operators in Decision Making*. En Carlsson, C. ed. Exploring the Limits of Support Systems, TUCS General Publications, No. 3, Turku Centre for Computer Science. 85-104, 1996.

- [13] S. Greco, B. Matarazzo and R. Slowinski. *Rough sets methodology for sorting problems in presence of multiple attributes and criteria*. European Journal of Operational Research. V.138. Pp. 247-259, 2002.