Revista Cubana de Ingenieria Vol. VIII, No. 3, septiembre - diciembre , 2017, pp. 34 - 41

Artículo Original

Algoritmo para el descubrimiento del modelo organizacional utilizando el patrón reducción

Algorithm for the discovery of the organizational model using reduction parallel pattern

Ronny Álvarez Pérez[1], Alex Rivero Botta[2] , Lester Guerra Denis[3], Humberto Díaz Pando[4]
[1-4] Universidad Tecnológica de La Habana José Antonio Echeverría, Cujae, La Habana, Cuba
Correo electrónico: lguerra@ceis.cujae.edu.cu

Recibido: 27 de mayo del 2017 Aprobado: 15 de julio del 2017

Resumen

Los avances alcanzados con las tecnologías de la información y las comunicaciones han resultado en un crecimiento de todos los datos almacenados y/o intercambiados electrónicamente. Las técnicas de minería de procesos son capaces de extraer conocimiento de los registros de eventos comúnmente disponibles en los sistemas de información actuales. Asimismo, el procesamiento paralelo es un tipo de procesamiento de la información, que permite que se ejecuten varios procesos concurrentemente, logrando impresionantes poderes de cálculo. El trabajo que a continuación se presenta es un diseño del algoritmo de minería de procesos para el descubrimiento del modelo organizacional utilizando el patrón paralelo reducción, aprovechando las operaciones que en su diseño secuencial se ejecutan independientemente. Los experimentos realizados y los resultados obtenidos con las pruebas de hipótesis no paramétricas de Mann-Whitney arrojan que la solución propuesta disminuye los tiempos de ejecución en relación con su variante secuencial.

Palabras claves: modelo organizacional, patrón paralelo reducción

Abstract

The progress made with the information technology and communications have resulted in a growth of all data stored and / or exchanged electronically. The process mining techniques are able to extract knowledge from the common-mind records events available in the current information systems. Likewise, parallel processing is a type of information processing, which allows multiple processes to run concurrently, achieving impressive powers of calculation. The work presented below is a design of mining process algorithm for the discovery of organizational model using the reduction parallel pattern, according to its design operations in sequential run independently. The experiments conducted and results obtained with non-parametric tests Mann-Whitney hypothesis that the proposed solution yield decreases execution times relative to its sequential variant.

Key words: organizational model, reduction parallel pattern

INTRODUCCIÓN

Los avances alcanzados con las tecnologías de la información y las comunicaciones han permitido recopilar y almacenar gran cantidad de información de todo tipo de procesos: industriales, empresariales, bancarios y publicitarios, entre otros.

El registro de los eventos de un proceso, en los actuales sistemas de información, hace posible aplicar técnicas de minería de procesos [1], capaces de extraer conocimiento de ellos.

Con el transcurso del tiempo, son cada vez mayores los datos almacenados en estos sistemas, propiciándose un escenario favorable para el descubrimiento de modelos en una organización.

Desafortunadamente, un crecimiento de los datos implica un mayor tiempo de procesamiento, careciendo, en muchos casos, de la operatividad necesaria para la extracción de conocimientos y la toma de decisiones que este implica.

El procesamiento paralelo es un tipo de procesamiento de la información, que permite que se ejecuten varios procesos concurrentemente, logrando impresionantes poderes de cálculo [2]. En este marco, existe un conjunto de patrones que combinan la distribución de tareas y el acceso a los datos para resolver un problema en específico en el diseño de algoritmos paralelos [3, 4].

El trabajo que a continuación se presenta propone un diseño del algoritmo para el descubrimiento del modelo organizacional basado en el patrón de computación paralela reducción. Este patrón es uno de los más usados en el diseño eficiente de algoritmos.

MATERIALES Y MÉTODOS

Minería de procesos

Es una técnica de administración de procesos que permite analizar los procesos de negocios de acuerdo con un registro de eventos. A través de esta, es posible extraer conocimientos desde los registros de eventos de los procesos almacenados por los sistemas. Existen varios tipos de minería de proceso:

• El descubrimiento (automático) de procesos. En este tipo no existe un modelo a priori, es decir, sobre la base de un log de eventos se construye un modelo de proceso que puede ser descubierto basado en eventos de bajo nivel (ejemplo: extraer modelos de procesos a partir de un registro de eventos) [1].

• La verificación de conformidad. En la verificación de conformidad hay un modelo a priori. Este modelo se compara con el registro de eventos y se analizan las discrepancias entre el registro y el modelo (ejemplo: monitorear desviaciones al comparar el modelo y el registro de eventos) [1].

• Extensión. En los algoritmos de tipo extensión también existe un modelo a priori. Este modelo se amplía con un nuevo aspecto o perspectiva, es decir, el objetivo no es para comprobar la conformidad, sino para enriquecer el modelo. Un ejemplo es la extensión de un modelo de proceso con los datos de rendimiento [1].

Algoritmo para el descubrimiento del modelo organizacional

El algoritmo de minería de procesos para el descubrimiento del modelo organizacional tiene como propósito mostrar, mediante un modelo, los originadores y las tareas que estos ejecutan en una organización.

Los originadores representan actores, ya sean personas, sistemas, etc., que ejecutan actividades. Las tareas son las actividades u operaciones ejecutadas por uno o varios originadores. Este algoritmo posee gran importancia a la hora de descubrir quién ejecuta qué tareas en una entidad determinada. Mediante el modelo se puede observar las relaciones existentes entre los originadores y las tareas.

La esencia de este algoritmo es iterar sobre el registro de eventos y para cada uno preguntar si el originador de ese evento ya se encuentra en el modelo, si no es así, agregarlo con la tarea correspondiente, y de encontrarse en el modelo, se verifica si la tarea a la que está asociado en dicho evento está registrada en la lista de tareas del originador en el modelo. Al finalizar el algoritmo se obtiene una estructura conformada por los originadores y cada uno tiene las tareas que ejecuta. La figura 1 muestra esta idea utilizando un diagrama de actividades en UML.

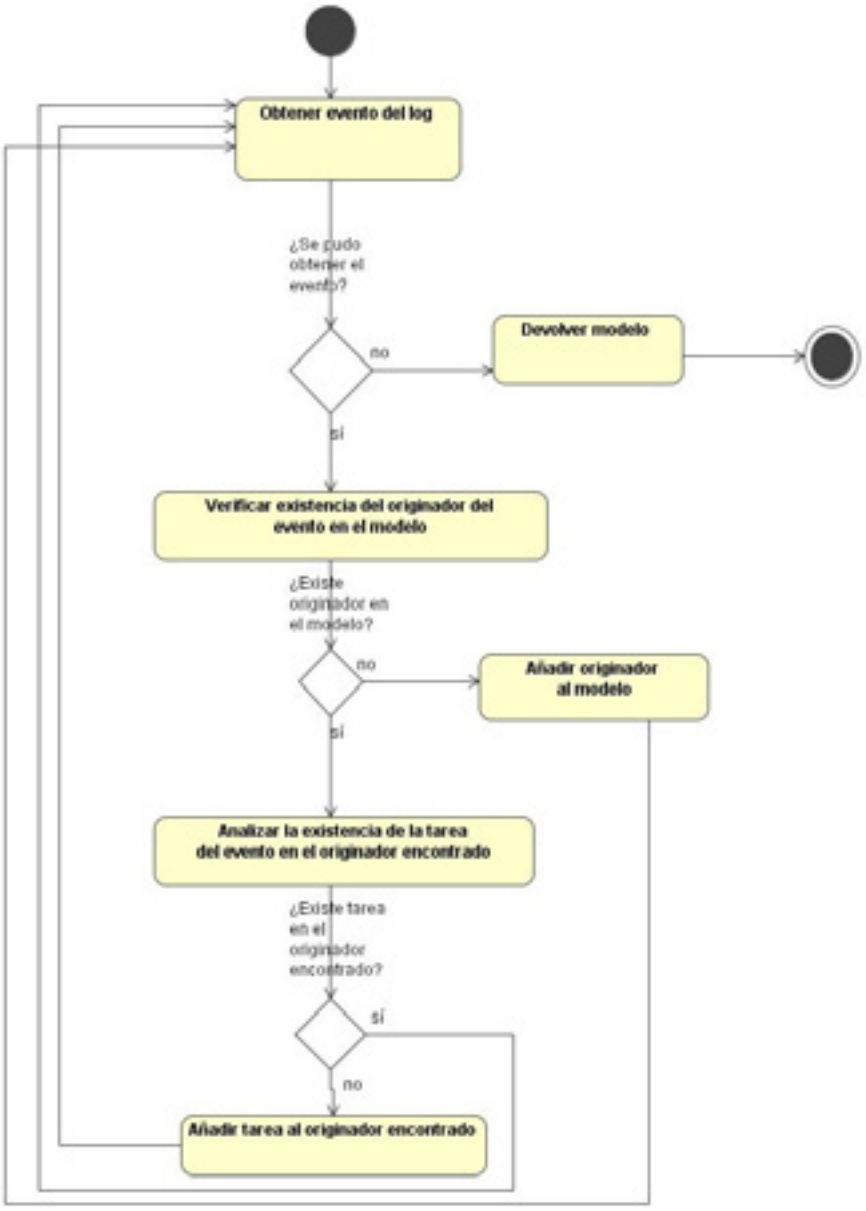


Fig. 1. Diagrama de actividades del algoritmo para el descubrimiento del modelo organizacional de forma secuencial

A continuación se muestra el pseudocódigo del algoritmo con el objetivo de describir la secuencia de pasos que ocurren en el mismo:

```
1 DescubrimientoMO(Registro de Eventos)
2       Iterar sobre el Registro de Eventos
3             Mientras existan Eventos
4                   Si evento.originador no está en el modelo

5                         Añadir evento.originador al modelo y su tarea
6                   Sino
7                         Si evento.tarea no está en la lista de tareas del originador (modelo)
8                               Añadir tarea a la lista de tareas del originador
9       Fin iterar
10      Devolver modelo
11 Fin DescubrimientoMO
```

El tiempo de ejecución de este algoritmo estará determinado fundamentalmente por las búsquedas que se realizan (entre las líneas 4 y 8), que en el caso de que la lista sea muy grande esto requerirá un gran costo computacional. Para ello en el siguiente acápite se explicará un patrón de computación paralela, el cual será explorado como solución a este problema.

Patrones de computación paralela

Los patrones de computación paralela, se refieren a una combinación de distribución de tareas y accesos a datos que resuelven un problema específico sobre el diseño de algoritmos paralelos. Dentro de estos patrones se encuentra el patrón Reducción.

Patrón Reducción

El patrón Reducción combina los elementos de una colección, lo cual trae como resultado la obtención de un solo elemento, utilizando una función de combinación asociativa [5, 6]. Debido a la propiedad asociativa de la función de combinación, varias formas de combinar son posibles. Un ejemplo de cómo implementar reducción de manera secuencial o paralela se muestra en la figura 1. Los estereotipos verdes indican elementos de datos y los azules elementos de tareas.

Fig. 2. Patrón reducción: a) diseño secuencial; b) diseño paralelo [5]

Para el diseño del algoritmo paralelo, se decidió utilizar una variante que se puede generalizar en dos fases. En la primera se crean grupos de eventos que se procesan de manera paralela y dentro de los cuales se utiliza reducción secuencial, para obtener un modelo por cada grupo. En la segunda fase se utiliza reducción entre los elementos resultantes de la primera (modelos), para obtener el modelo resultante. Este método se muestra en la figura 3 y en la figura 4 se observa un diagrama de actividades de este algoritmo para el diseño discutido anteriormente.

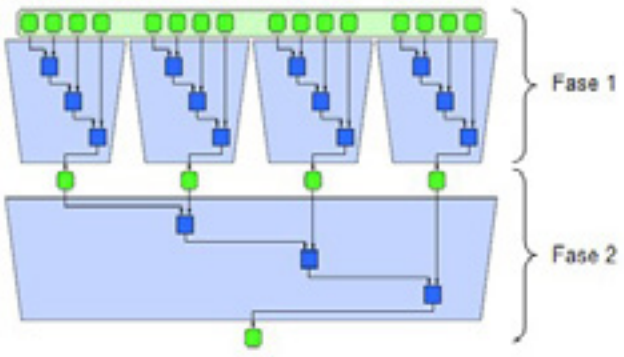


Fig. 3. Variante de diseño de reducción que combina la implementación paralela y secuencial del mismo [5]

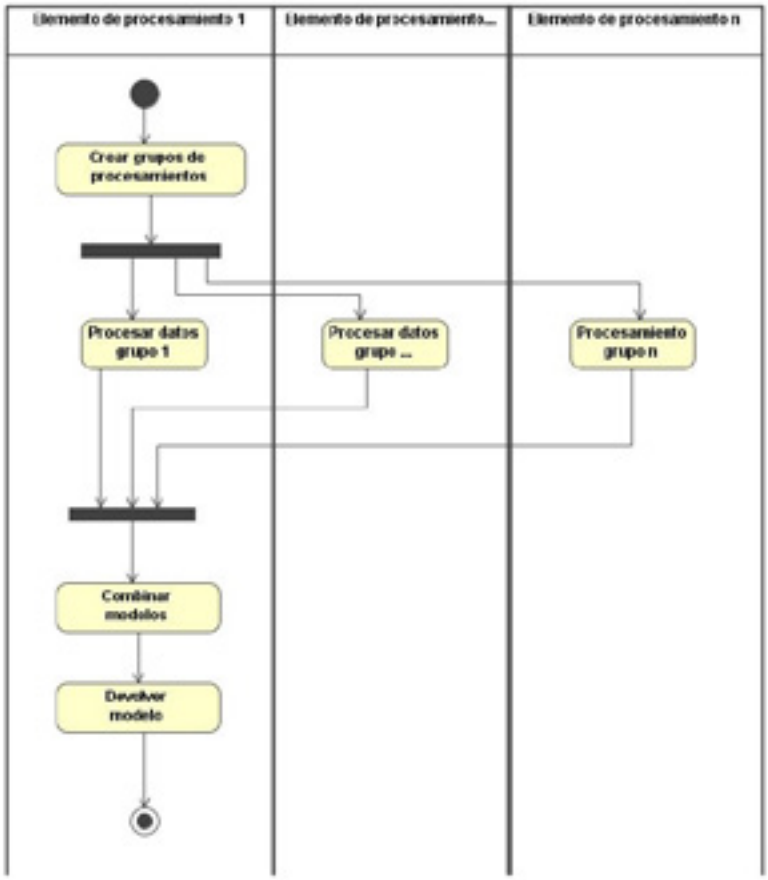


Fig. 4. Diagrama de actividades del algoritmo diseñado de forma paralela

Pseudocódigo de la solución paralela

A continuación se muestra el pseudocódigo del algoritmo con el diseño paralelo propuesto anteriormente.

Fase 1

```
DescubrimientoMO1(Registro de Eventos)
        Mientras no hayan tantos modelos como numGrupos
                Crear Modelo
        Fin Mientras
        Parallel start
        Repetir hasta llegar a numGrupos
            Añadir a cada modelo el primer originador y su tarea
        Fin Repetir
        Parallel end
        Parallel start
        Repetir hasta llegar a numGrupos
                Mientras existan eventos a analizar en el grupo
                        Si evento.originador no está en el modelo
                                Añadir evento.originador al modelo y su tarea
                        Sino
                                Si evento.tarea no está en la lista de tareas del originador (modelo)
                                        Añadir tarea a la lista de tareas del originador
                        Fin Si
                Fin Mientras
        Fin repetir
        Parallel end
    Devolver ListaModelosResultantes
Fin DescubrimientoMO1
```

Fase 2

```
DescubrimientoMO2(ListaModelosResultantes)
    Obtener primer modelo
    Mientras existan modelos a analizar a partir del segundo modelo
        Obtener modelo a analizar
        Mientras existan originadores dentro del modelo a analizar
            Si el originador actual no está en el primer modelo
                Añadir originador actual al primer modelo con su lista de tareas
            Sino
                Para cada tarea en originador actual
                    Si tarea no existe en originador encontrado (primer modelo)
                        Añadir tarea a originador encontrado (primer modelo)
                Fin Para
            Fin Si
        Fin Mientras
    Fin Mientras
    Devolver primer modelo (modelo actualizado)
Fin DescubrimientoMO2
```

RESULTADOS

Para determinar si el diseño propuesto mejora el tiempo de ejecución del algoritmo secuencial, se realizó la prueba estadística no paramétrica de Mann-Whitney. Se seleccionó esta porque no se cuenta con evidencias para afirmar que los datos sigan una distribución normal. El valor de α (máximo nivel de riesgo aceptable para rechazar una hipótesis nula verdadera) utilizado fue 0,05.

Como en la paralelización de este algoritmo utilizando el patrón reducción, la carga de la base de datos es la misma para las versiones secuencial y paralela, el tiempo de ejecución del algoritmo no contempla esta carga.

La arquitectura utilizada para realizar las pruebas fue: microprocesador Intel Core 2 Duo a 2.66 MHz con 2 GB de memoria RAM y una base de datos con 1 408 539 eventos. La tabla 2 muestra el mínimo, el máximo y la media de los tiempos del algoritmo secuencial y paralelo, después de realizar 10 ejecuciones.

Tabla. 1. Tabla de tiempos (en segundos) de ejecución del algoritmo en las versiones secuencial y paralelo sin tener en cuenta la carga de la base de datos

	Algoritmo secuencial (s)	Algoritmo paralelo (s)
Mínimo	1,066 03	0,747 654
Máximo	1,078 96	0,776 694
Media	1,075 514	0,754 807 9

La prueba de hipótesis se formuló de la siguiente manera:

H_0: $T_s = T_p$

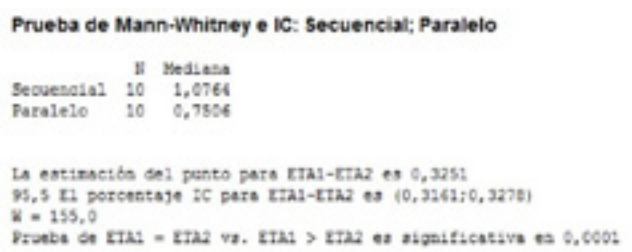
H_1: $T_s > T_p$

donde:

T_s: Tiempo de ejecución del algoritmo secuencial.

T_p: Tiempo de ejecución del algoritmo paralelo.

La prueba se ejecutó en el software Minitab [7]. La figura 5 muestra una vista de los resultados obtenidos.



Prueba de Mann-Whitney e IC: Secuencial; Paralelo

	N	Mediana
Secuencial	10	1,0766
Paralelo	10	0,7506

La estimación del punto para ETA1-ETA2 es 0,3251
95,5 El porcentaje IC para ETA1-ETA2 es (0,3161;0,3278)
W = 155,0
Prueba de ETA1 = ETA2 vs. ETA1 > ETA2 es significativa en 0,0001

Fig. 5. Resultados de la prueba Mann-Whitney

Las métricas de rendimiento utilizadas para evaluar el diseño paralelo implementado fueron:

Aceleración (**Speed Up**): es la relación entre el tiempo de ejecución utilizando solo un procesador y el tiempo de ejecución utilizando varios procesadores. Se calcula de la siguiente forma:

$$S=\frac{T_s}{T_p} \quad (1)$$

donde:
S: Aceleración.
T_s: Tiempo de ejecución del algoritmo secuencial.
T_p: Tiempo de ejecución del algoritmo paralelo.

Eficiencia: La eficiencia puede ser definida como la parte del tiempo que los procesadores dedican al trabajo útil, o sea a las tareas. La eficiencia muestra cuán bien se han utilizado los procesadores. La fórmula que define a la eficiencia es la siguiente:

$$E = \frac{S_p}{P} = \frac{Ts}{P \cdot Tp} \quad (2)$$

donde:
E: Eficiencia.
P: Número de procesadores.
S_p: Aceleración asociada a la ejecución del programa paralelo usando P procesadores.

Para realizar el cálculo de las medidas de rendimiento, fue utilizado el valor promedio de tiempo propuesto en la tabla 2:

Aceleración: $s = \frac{1{,}075\ 514\ s}{0{,}754\ 807\ 9\ s} = 1{,}424\ 884$

Eficiencia: $E = \frac{1{,}424\ 884}{2} = 0{,}712\ 44$

DISCUSIÓN

Como puede apreciarse en la figura 4, p = 0,000 1 y este valor es menor que α, entonces no existen evidencias que permitan aceptar la hipótesis nula, por lo que esta se rechaza y se concluye que el algoritmo paralelo presenta menores tiempos que el algoritmo secuencial en el 95 % de sus ejecuciones.

La aceleración obtenida significa que la variante paralela es 1,42 veces más rápida que la variante secuencial, obteniéndose una mejora de aproximadamente un 70 %. La eficiencia obtenida de 0,71 implica que se aprovechan en un 71 % los recursos del procesador.

CONCLUSIONES

En la presente investigación se abordó el estudio de un método que permite reducir el tiempo de ejecución de la implementación del algoritmo para el descubrimiento del modelo organizacional. Los resultados indican que:

Es posible paralelizar el procesamiento de los eventos para el descubrimiento del modelo organizacional.

Aunque el tiempo del algoritmo secuencial no es muy elevado, se propone un diseño que disminuye su tiempo de ejecución.

Se utiliza el patrón de computación paralela Reducción, evidenciándose la pertinencia de su utilización.

El algoritmo paralelo presenta menores tiempos que el algoritmo secuencial en el 95 % de sus ejecuciones, según la prueba de hipótesis no paramétrica de Mann-Whitney realizada.

El cálculo de las medidas de rendimiento arroja que la variante paralela es 1,42 veces más rápida que la variante secuencial y que los recursos del procesador se aprovechan en un 71 %.

Por todo lo planteado se concluye que el objetivo de investigación ha sido cumplido satisfactoriamente. La investigación realizada permite obtener un algoritmo que puede ser extrapolado a cualquier escenario que necesite descubrir irregularidades en los procesos de negocios.

Se recomienda implementar el algoritmo siguiendo el diseño propuesto con otros modelos de programación, específicamente MPI, y comparar los resultados obtenidos con los expuestos para OpenMP en este trabajo.

REFERENCIAS

1. Aalst van der W, et al. Manifiesto sobre Minería de Procesos. Berlin, Alemania: Springer-Verlag; 2011.
2. Barney B. Introduction to Parallel Computing, Available. EE.UU.2016. [Citado Disponible en: https://computing.llnl.gov/tutorials/parallel_comp/
3. Mattson TG, Sanders BA, Massingill B. Patterns for Parallel Programing. Massachusetts, EE.UU.: Addison-Wesley Professional; 2004.
4. Massingill BL, Mattson TG, Sanders BA. Parallel programming with a pattern language. International Journal on Software Tools for Technology Transfer. 2001;3( ):217-34.
5. McCool M, Robinson AD, Reinders J. Structured Parallel Programming Patterns for Efficient Computation Ámsterdan, Holanda: Morgan Kaufman; 2012.
6. McCool M. Ámsterdan, Holanda; 2009. [Citado Disponible en: http://www.drdobbs.com/architecture-and-design/parallel-pattern-7-reduce/222000718]
7. Minitab. Soporte de Minitab; 2016. [Citado Disponible en: http://support.minitab.com.