Herramientas de Modelado y Simulación para Sistemas de Gran Escala

Jair Lobos[1], Veronica Gil-Costa[2,3], Andrea Giubergia[2] and Marcela Printista[1,3]

[(1)]Departamento de Computación
Facultad de Ciencias Físico Matemáticas y Naturales
Universidad Nacional de San Luis
Ejército de los Andes 950, 1º piso.
(02652-420823)

[(2)]Departamento de Minería
Facultad de Ciencias Físico Matemáticas y Naturales
Universidad Nacional de San Luis
Chacabuco y Pedernera.
(02664-4436531)

[(3)] CONICET San Luis
Almirante Brown 907
(02664 421654)

CONTEXTO

La línea de investigación presentada en este trabajo recurre a un proyecto que vincula estrechamente dos temas que han cobrado gran interés en los últimos años debido al avance de la tecnología y a los costos excesivos que requieren las pruebas y ejecuciones sobre plataformas reales. Nos referimos a las líneas de Modelado y Simulación.

En particular, nos enfocamos en el modelado de aplicaciones de gran escala para plataformas paralelas que no pueden ser probadas en sistemas y hardware reales debido al costo de los mismos. Para ello, es posible utilizar diferentes herramientas como las Petri Nets[Petri62], Devs[Zeig76], Análisis Operacional [Den78] y UML. Otra ventaja de las técnicas de modelado y simulación utilizadas en este proyecto, es que permite obtener estimaciones de las métricas utilizadas en las aplicaciones para determinar el costo-beneficio de implementar y desplegar la aplicación en un hardware real.

RESUMEN

El crecimiento constante de la tecnología, diseño e implementación de nuevas arquitecturas de computadoras, placas aceleradoras como las GPU, redes de alta velocidad, etc., son el resultado de la gran demanda introducida por los usuarios de sistemas de gran escala que requieren no sólo procesar sus requerimientos rápidamente (fracciones de segundo) sino de la gran cantidad de datos utilizados en estos sistemas.

Algunos de estos datos (los más cercanos al usuario de redes) se encuentran en las páginas o documentos Web que son procesadas e indexadas por grandes motores de búsqueda Web como Google, Yahoo! o Bing. Pero existen datos más complejos como las imágenes satelitales que son utilizados por especialistas en otras disciplinas como la geología y la minería. Estos datos (ya sean imágenes, información referente a perforaciones realizadas en la superficie de la tierra, etc.) deben ser eficiente y eficazmente procesadas por sistemas de información en un tiempo razonable.

En ambos casos, al utilizar documentos Web como datos de sistemas industrializados de propósito específico, existe la complejidad de que dichos datos tienden a ser modificados a lo largo del tiempo. Esto último introduce una dimensión adicional que debe ser analizada al momento de diseñar estos sistemas complejos.

En este trabajo, se presentan los objetivos y los desafíos que se pretenden abordar desde el grupo interdisciplinario de investigación de la Universidad Nacional de San Luis, para abordar los desafíos que involucra el modelado y diseño de sistemas de gran escala que deben ser capaces de procesar grandes volúmenes de datos e información.

Palabras clave: *Sistemas de Gran Escala, Herramientas de modelado: Petri Nets, Devs, UML.*

1.INTRODUCCION

La gran demanda de procesamiento y análisis eficiente de grandes volúmenes de datos e información (conocida recientemente como “Big Data” y “stream processing”) ha introducido el desafío del desarrollo de sistemas que sean capaces de resolver los requerimientos de

procesamiento realizados por diferentes usuarios sobre estos datos. En este trabajo se describen los lineamientos de investigación que se llevan a cabo sobre dos tipos de sistemas que involucra la administración de datos masivos. El objetivo principal es poder modelar eficientemente (a través de herramientas existentes) estos sistemas para luego mediante técnicas de simulación poder verificar el comportamiento de los mismos sobre diferentes escenarios de carga de trabajo.

El primer sistema consiste en motores de búsqueda Web de gran escala, que deben ser capaces de procesar miles de consultas por segundo. Estos motores de búsqueda utilizan servicios enfocados a realizar diferentes tareas en simultáneo. Algunas de estas tareas involucran (a) la administración de memorias caches que almacenan las consultas mas referenciadas, (b) el ranking de los top-k documentos más relevantes para las consultas de los usuarios, (c) selección de publicidades que mejor se ajustan a las consultas y a los usuarios, (d) gestión de historial de usuarios, etc..

El segundo sistema comprende el procesamiento de información y datos (generalmente imágenes) que son requeridos para evaluar proyectos de gran escala como lo son los proyectos de extracción de minerales. En particular, en este caso, se utiliza el lenguaje de modelado UML y se lo extiende mediante el uso de perfiles para soportar características propias de los sistemas bajo estudio.

1.1 Petri Nets

Las Redes de Petri corresponde a un concepto introducido por Carl Adam Petri en 1962 [Petri62]. Las Redes de Petri son una representación matemática y gráfica para describir y estudiar sistemas a eventos discretos en el cual se puede describir la topología de un sistema concurrente, paralelo o distribuido [Murata89, Molloy81]. Algunos trabajos introductorios se pueden encontrar en [Peterson77, Peterson81] y algunos libros en [Aalst11, Silva85]

Una Red de Petri es un tipo particular de grafo dirigido que posee un estado inicial, denominado M_o. El grafo generado N de una Red de Petri es un grafo dirigido, con peso y bipartito que posee 3 tipos de elementos: (1) nodos, también llamados lugares (places); (2) transiciones (transitions), que reflejan las acciones o eventos y (3) arcos (arcs), que conectan un lugar/nodo con una transición o conectan una transición a un lugar/nodo.

Gráficamente los lugares se representan con un círculo, las transiciones con barras o cajas. Los arcos son etiquetados con sus pesos (enteros positivos). Las etiquetas para pesos unitarios usualmente son omitidas. Un arco que esté etiquetado con k puede ser interpretado como k arcos paralelos. Una marca (estado) se asigna a cada lugar como un entero no negativo. Si se marca un lugar p con un valor k no negativo, se dice que p se marca con k tokens, lo cual representa a k recursos los cuales se mueven entre los lugares de la red. Los tokens o fichas se representan, en una Red de Petri, por medio de un punto negro e indican los recursos que posee un lugar.

A diferencia de los simuladores de eventos discretos, un modelo basado en Redes de Petri [Petri62] es más simple y eficiente, y es más fácil de extender de forma tal de incluir nuevas características o cambiar el comportamiento de algunos componentes del modelo utilizando un lenguaje gráfico y un test de verificación del modelo.

Sin embargo, una limitación de las Redes de Petri tradicionales es que al modelar un sistema real, se debe generar una gran cantidad de lugares (places) y transiciones (transitions), por lo cual su análisis se hace complejo. Además, en los sistemas reales, a menudo, se presentan procesos similares que se producen en paralelo o de forma simultánea, y se diferencian unos de los otros por sus entradas y sus salidas. Al utilizar Redes de Petri coloreadas, la cantidad de lugares, transiciones y arcos, en general se reducen. Lo cual hace que el modelo tenga una mayor facilidad de entendimiento.

1.2 DEVS

La "Especificación de Sistemas de Eventos Discretos (DEVS)" es un formalismo para modelar sistemas y realizar y simulaciones. Se basa en conceptos de teoría de sistemas y fue desarrollada por Zeigler en [Zeig76].

DEVS fue creado para modelar y simular sistemas dinámicos de eventos discretos, de manera que permite especificar sistemas cuyo estado se altera ya sea por la recepción de un evento de entrada o por el vencimiento de una demora de tiempo. Como una manera de rebajar la complejidad del sistema a estudiar es que el modelo se organiza jerárquicamente, en que los componentes de alto nivel del sistema se descomponen en elementos más simples. La separación entre modelo y simulador, y su naturaleza de módulos jerárquicos ha permitido llevar a cabo pruebas formales en las diferentes entidades bajo estudio [Wai09].

Por medio de DEVS, un sistema real se representa mediante la composición de componentes atómicos y acoplados (un componente acoplado corresponde a la composición de dos o más componentes atómicos). Un componente atómico está definido por el conjunto de eventos y puertos de entrada, el conjunto de puertos y eventos de salida, el conjunto de estados, una función de transición de estados externa, una función de transición de estados interna, la función de salida y la función de avance de tiempo. Un estado para el cual la función de avance del tiempo es cero se denomina "estado transiente" y dispara una transición interna de forma inmediata. Por el contrario, si el tiempo es infinito, se denomina "estado pasivo". El sistema continuará en dicho estado hasta que reciba un evento externo [Zeig76, Zeig00].

Un modelo DEVS acoplado se compone de uno o más modelos atómicos y/o submodelos acoplados. Está definido formalmente por el conjunto de puertos y eventos de entrada, el conjunto de puertos y eventos de salida, el conjunto de nombres de los componentes, el conjunto de componentes que lo forman, el conjunto de acoples externos de entrada, el conjunto de acoples externos de salida, el conjunto interno de acoples y, finalmente, por la función de selección. Esta función de selección permite resolver la ambigüedad que surge cuando hay más de un evento que debe realizar una transición interna al mismo tiempo [Zeig76].

1.3 Análisis Operacional

Por medio del análisis operacional se puede realizar predicciones respecto del rendimiento de sistemas representados mediante modelos de redes de colas. En análisis Operacional [Den78], todas las ecuaciones derivan al menos los siguientes tres principios:

- Todas las cantidades deben ser definidas de manera que sean precisamente medibles, y las suposiciones de forma que sean directamente comprobables.
- El sistema debe ser balanceado en su flujo.
- Los dispositivos deben ser homogéneos.

En base a estos tres principios del análisis operacional se pueden derivar ecuaciones que permiten caracterizar el comportamiento y rendimiento de sistemas tales como los computacionales. Desde el punto de vista del análisis operacional, siempre deben existir los elementos “sistema" y “periodo de tiempo" en un problema. El sistema puede ser uno real o supuesto y el periodo de tiempo puede ser pasado, presente o uno futuro.

Las variables del análisis operacional poseen el mismo valor durante dicho periodo de tiempo. Estas pueden ser obtenidas directamente mediante mediciones o derivadas de estas. Entre las variables básicas - obtenidas durante el periodo de observación - se tienen:

- T: el periodo de observación.
- A: el número de arribos durante T.
- B: el tiempo total en que el sistema estuvo ocupado durante T.
- C: El numero de “trabajo" completos durante T.

A partir de las cuales se derivan las siguientes métricas de rendimiento:

- $\lambda = A/T$, es la tasa de arribos medida en trabajos por segundo.
- $X = C/T$, es la tasa de salida también medida en trabajos por segundo (throughput).
- $U = B/T$, es la utilización, o fracción del tiempo T en que el sistema estuvo ocupado.
- $S = B/C$, es el tiempo de servicio promedio por trabajo completado.

Todas estas variables pueden cambiar de valor de un periodo de observación a otro.

1.4 UML y Simulación

El Lenguaje de Modelado Unificado, UML (Unified Modelling Language) [Booch98], fue construido para diseñar sistemas de información y para facilitar el desarrollo y mantenimiento de sus procesos. Con el pasar de los años, el uso de UML se ha extendido y actualmente es el estándar más utilizado para especificar y documentar sistemas.

UML es una notación de propósito general, pero no siempre satisface las necesidades que requiere una aplicación en particular. Es por ello que UML permite extender su sintaxis y su semántica a través de mecanismos propios, para que lo conviertan en un lenguaje con características más específicas orientadas a ciertos dominios.

Estos mecanismos de extensión, inherente al mismo UML, que permiten extender y adaptar las metaclases de un metamodelo cualquiera a las necesidades concretas de un dominio de aplicación, se denominan Perfiles UML (UML Profiles) [Deb06].

Por otro lado, la técnica de Simulación [Gold07], [Sad07] implica replicar artificialmente las características de un sistema a través de un modelo e imitar su operación a medida que transcurre el tiempo. En base al análisis de comportamiento del modelo, luego será posible inferir las características operacionales del sistema real. El modelado y la simulación brindan la posibilidad de estudiar nuevas estrategias y de predecir el efecto de la aplicación de nuevas políticas, que de otra manera, serían excesivamente costosas o incluso imposibles de reproducir y de estudiar.

Ambos enfoques, modelado UML y simulación, son ampliamente usados en ingeniería de sistemas. Si bien estas dos teorías han evolucionado por separado, existen pocos trabajos [Teil08] que integran las herramientas de modelado con las herramientas de software de los lenguajes de simulación para desarrollar parcial o totalmente un sistema.

2. LINEAS DE INVESTIGACION y DESARROLLO

La línea de investigación descripta en la sección anterior involucra una serie de desarrollos individuales que en su conjunto logran obtener el objetivo planteado. Para ello es necesario estudiar formas.

3. RESULTADOS OBTENIDOS ESPERADOS

Modelado y Simulación mediante Análisis Operacional

Los resultados obtenidos hasta el momento son:

- Modelado de un motor de búsqueda Web basado en servicios [Gil13].
- Diseño e implementación de un simulador basado en procesos.
- Estudio de formulas que permitan determinar la cantidad de recursos requeridos por el motor de búsqueda.

Los resultados esperados son:

- Desarrollo de una metodología que permita evaluar la capacidad computacional de un motor de búsqueda.
- Validación de las fórmulas y metodología desarrollada.

Modelado a través de UML

Los resultados obtenidos hasta el momento son:

- Mostrar que UML proporciona una alta flexibilidad a la hora de modelar sistemas [Giuber12].
- Crear un metamodelo que incluya el perfil específico, que luego pueda ser trasladado a un ambiente grafico de simulación

Los resultados esperados son:

- Mostrar que UML puede emplearse como una etapa previa o de transición hacia la implementación de la simulación.

Modelado y simulación a través de DEVS

Los resultados obtenidos hasta el momento son:

- Estudio de métodos de formalización como DEVS
- Modelo y diseño de un motor de búsqueda Web (sin jerarquías de cache) utilizando DEVS

- Los resultados esperados son:
  - Extender el primer modelo para incluir características más relevantes como jerarquías de cache y costos de consultas individuales.
  - Estudiar técnicas de integración de módulos de software.

Modelado y simulación con Petri Nets

Los resultados obtenidos hasta el momento son:

- Modelado de un motor de búsqueda Web mediante petri Nets[Gil12]
- Análisis y validación del modelo.

Los resultados esperados son:

- Diseño de un simulador de petri nets paralelo que se ajuste a las características de un motor de búsqueda.

4. FORMACION DE RECURSOS HUMANOS

Actualmente, se cuenta con dos doctores en ciencias de la computación realizando la investigación teórica y dirección de los algoritmos propuestos. También se cuenta con un alumno de doctorado que se encuentra iniciando su carrera doctoral, un segundo alumno de doctorado que se encuentra realizando una estadía en Ottawa,Canada con un grupo de investigación especializado en DEVS; y una alumna de maestría próxima a finalizar su tesis.

Mediante este trabajo de investigación se podrán formar profesionales que puedan modelar, diseñar e implementar algoritmos eficientes que se ejecuten en sistemas de gran escala y requieren el procesamiento de datos masivos.

5. BIBLIOGRAFIA

[Aalst11] W.M.P. van der Aalst and C. Stahl. Modeling Business Processes - A Petri Net-Oriented Approach. The MIT Press, 2011.

[Booch98] Booch G.; Rumbaugh J.; Jacobson I., (1998), The Unified Modeling Language User Guide (Addison-Wesley Object Technology Series).

[Buzen76] J. P. Buzen. Fundamental operational laws of computer system performance. Acta Inf., 7:167-182, 1976

[Deb06] Debnath N.C., Garis A., Riesco D., Montejano G., (2006), Defining Patterns Using UML Profiles.

[Den78] P. J. Denning and J. P. Buzen. The operational analysis of queueing network models. ACM Computing Surveys, 10:225-261, 1978.

[Gil12] “Capacity Planning for Vertical Search Engines: An approach based on Coloured Petri Nets”. Gil-Costa Veronica, Lobos Jair, Inostrosa-Psijas Alonso and Mauricio Marin. Petri Nets 2012.

[Gil13] “Service Deployment Algorithms for Vertical Search Engines”. Alonso Inostrosa-Psijas, Veronica Gil-Costa, Mauricio Marin and Esteban Feuestein. PDP 2013.

[Giub12] "Estereotipos UML para Aplicar en un Ambiente de Simulación de Procesos Mineros”. Andrea Giubergia, Daniel Riesco, Marcela Printista y Veronica Gil Costa. CACIC 2012, Argentina.

[Gold07] Goldsman D., (2007), Introduction to simulation. Proceedings of the 2007 Winter Simulation Conference. IEEE.

[Molloy81] M.K. Molloy. On the integration of delay and throughput measures in distributed processing models. Technical report, Phd Thesis, UCLA, 1981.

[Murata89] Murata, T. 1989. Petri nets: Properties, analysis and applications. Proceedings of the IEEE 77, no. 4: 541-580.

[Peterson77] J.L. Peterson. Petri nets. Computing Surveys, 9:223–252, 1977.

[Peterson81] J.L. Peterson. Petri net theory and the modeling of systems. Prentice Hall, Englewood Cliffs, 1981.

[Petri62] C.A. Petri. “Communication with Automata” New York: Griffiss Air Force Base. Tech. Rep. RADC-TR-65-377, vol.1, Suppl. 1, 1962.

[Sad07] Sadowski D.A., (2007), Tips for successful practice of simulation. Proceedings of the 2007 Winter Simulation Conference.

[Silva85] M. Silva. Las Redes de Petri en la Automatica y la Informatica. AC, Madrid, 1985.

[Teil08] Teilans A.; Kleins A.; Merkuryev Y.; Grinbergs A., (2007), Design of UML models and their simulation using ARENA. WSEAS TRANSACTIONS ON COMPUTER RESEARCH. Issue 1, Volume 3, January 2008.

[Wai09]Wainer, G. 2009. Discrete-Event Modeling and Simulation: A Practitioner's Apporach.

[Zeig76] Zeigler, B. P. 1976. Theory of Modeling and Simulation. Wiley-Interscience.

[Zeig00] Zeigler, B. P, H. Praehofer, and T.G. Kim. 2000. Theory of Modeling and Simulation, 2nd. ed. New York: Academic Press.