

WICC 2014 XVI Workshop de Investigadores en Ciencias de la Computación

Modelado y Diseño de Sistemas Complejos mediante Técnicas de Simulación

Andrea Giubergia², Verónica Gil-Costa^{2,3}, Jair Lobos¹, Cesar Ochoa¹, A. Marcela Printista^{1,3} and Cristian Tisera¹

⁽¹⁾Departamento de Informática
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 que se presenta en este trabajo está vinculada a un tema que ha cobrado un gran interés durante el último tiempo. Dado el avance de tecnología y al costo que requieren realizar pruebas y ejecuciones sobre plataformas reales, es posible obtener estimaciones sobre estos puntos por medio de técnicas de simulaciones.

Utilizando técnicas de simulaciones eficientes es posible diseñar y desarrollar sistemas de gran escala como lo son los sistemas de stream processing. Con ello se pueden probar alternativas en el diseño y predecir su impacto sobre un sistema por medio de métricas de rendimiento, lo cual permite determinar el costo-beneficio e implementar la aplicación en un hardware real.

Para la simulación se han utilizando herramientas como Simulación Aproximada [Marin13], Redes de Petri [Petri62] [Peterson77], Análisis Operacional [Den78], simuladores orientados a procesos y XML. En particular, este proyecto se enfoca en considerar el modelado de aplicaciones complejas y de gran escala donde intervienen diferentes variables de incertidumbre para plataformas paralelas y aplicadas a sistemas de stream processing, que no pueden ser probadas en sistemas y hardware reales sin acarrear un costo económico que puede ser de gran impacto en el caso de que las nuevas aplicaciones o algoritmos no cumplan con sus objetivos.

RESUMEN

Actualmente, existe una gran demanda de los usuarios frente a los sistemas de gran escala, la cual requiere no solo procesar sus requerimientos de manera rápida sino que

manejan una gran cantidad de datos. Este nuevo desafío ha llevado a que la tecnología tenga un crecimiento tanto en el diseño e implementación de nuevas arquitecturas de computadores, redes de alta velocidad, placas aceleradoras de GPU, etc.

Los datos de gran volumen son generalmente representados mediante documentos Web que son procesados e indexados por los grandes motores de búsqueda como Google, Yahoo! o Bing. A su vez existen otros tipos de datos complejos como lo pueden ser las imágenes satelitales, imágenes médicas, etc. Adicionalmente, es necesario considerar el efecto que tiene el comportamiento de los usuarios sobre un sistema de gran escala, comportamiento frente a la publicidad, acciones sobre juegos, respecto a eventos mundiales como los terremotos, etc. Todos estos desafíos promueven el desarrollo de tecnologías y algoritmos que deben ser eficaces (en términos de calidad de recuperación de la información) y eficientes (en términos de tiempo de procesamiento).

En este trabajo, se presentan los objetivos, trabajo realizado y desafíos que aborda el grupo de investigación interdisciplinario de la Universidad Nacional de San Luis, para abordar los temas que involucra el diseño de sistemas complejos y de gran escala para procesar grandes volúmenes de datos e información.

Palabras clave: Sistemas Complejos y de Gran Escala, Simulación, Stream Processing.

INTRODUCCION

Un nuevo desafío en el desarrollo de sistemas ha causado gran demanda de procesamiento y análisis eficiente de grandes volúmenes de datos e información. Estos desafíos

corresponden a “Big Data” y “Stream Processing”, los cuales deben ser procesados por sistemas capaces de entregar los resultados esperados en un tiempo de procesamiento razonable. En este trabajo se describen los lineamientos de investigación realizados en este último tiempo y se proponen futuros temas de investigación para el problema de sistemas que involucra el diseño y desarrollo de algoritmos capaces de administrar datos masivos sobre plataformas de hardware de gran escala, como lo son los motores de búsqueda Web y sistemas de procesamiento de streaming.

El objetivo principal diseñar, analizar, desarrollar y evaluar diferentes técnicas y algoritmos basados en técnicas de modelado y simulación que permita verificar el comportamiento de los sistemas sobre distintas configuraciones y diferentes escenarios de carga de trabajo. Un punto crucial es probar algoritmos y estrategias que permitan balancear la carga de trabajo de los componentes a evaluar. Para ello, se ha realizado investigación y se han desarrollado herramientas que permiten el modelado de estos sistemas. Estas herramientas hacen uso de Redes de Petri, UML, análisis operacional, simulación aproximada paralela y XML.

Un escenario alternativo donde se ha incursionado recientemente son los proyectos generados para la extracción de minerales. Estos proyectos generan una gran cantidad de datos referidos al tipo y calidad de rocas o minerales que pueden ser extraídos y procesados. En este punto, las tecnologías informáticas (especialmente las tecnologías basadas en simulaciones) pueden ser utilizadas para estimar la productividad del proyecto a desarrollar y analizar diferentes situaciones de incertidumbre generadas por las fallas en los sistemas mecánicos.

A continuación se describen algunas de las técnicas, herramientas y plataformas utilizadas en el proyecto de investigación.

1.1 Modelado mediante UML

UML (Unified Modelling Language) [Booch98] es una notación de propósito general que permite extender su sintaxis y su semántica a través de mecanismos propios (denominado Perfil), que lo convierten en un lenguaje con características más específicas orientadas a ciertos dominios. Esto permite utilizar una herramienta como UML en un

ámbito de simulación de eventos discretos, para aplicar en el campo de la minería [Giu14]. Puede emplearse como una etapa previa hacia la implementación de la simulación, donde la combinación de ambas herramientas conduce a facilitar el desarrollo de un sistema y potenciar la implementación efectiva del mismo.

1.2 Simulación Paralela Aproximada

Una simulación de eventos discretos consiste de un conjunto de eventos que tienen asociados un tiempo o timestamp. Si la simulación se ejecuta sobre una sola máquina o procesador, los eventos de la simulación pueden ser colocados en una cola central de forma tal que el algoritmo en ejecución pueda ordenarlos cronológicamente. Sin embargo, la simulación de sistemas de gran escala requiere largos tiempos de ejecución. Inclusive una simulación de red con miles de nodos puede tomar varias semanas en finalizar. Para reducir el tiempo de ejecución, es posible correr las simulaciones en ambientes paralelos.

Cuando las simulaciones son ejecutadas en paralelo, la lista de eventos se encuentra distribuida entre los procesadores (cada procesador posee sólo una porción de la lista de eventos). Otra complicación adicional es que los eventos pueden llegar en forma des-sincronizada desde otros procesadores.

Los modelos de simulación proveen una mejora en términos de tiempo de ejecución. Bajo este paradigma el modelo de simulación es particionado en diferentes procesadores físicos que se comunican entre sí intercambiando mensajes (eventos). El objetivo principal de las simulaciones paralelas tradicionales (PDES) es proveer los mismos resultados obtenidos por la simulación secuencial.

Recientemente, en [Marin13] se propone priorizar los tiempos de ejecución y perder hasta un 5% de precisión en los resultados finales de la simulación. Este trabajo se basa en la utilización del modelo de computación paralelo BSP [Val90], donde el cómputo se organiza en una secuencia de supersteps. Durante un superstep, cada procesador realiza operaciones sobre sus datos locales y/o envía mensajes a otros procesadores. Al final de cada superstep se ejecuta una barrera de sincronización que permite que los mensajes enviados en el superstep actual, se encuentren disponibles en

las colas de entrada de los procesadores destino en el próximo superstep.

Adicionalmente, la implementación de la herramienta de simulación paralela, hace uso de multi-threading. Con el fin de implementar correctamente la barrera de sincronización, en cada procesador existe un thread master que se sincroniza con los P-1 threads master restantes. Los T-1 threads restantes se sincronizan con el thread master para evitar avanzar en la ejecución del próximo superstep.

1.3 Análisis Operacional

Utilizando el análisis operacional [Den78] es posible realizar predicciones respecto al rendimiento de sistemas representados mediante modelos de redes de cola. Al utilizar análisis operacional todas las cantidades deben ser definidas de manera que sean medibles. Adicionalmente, exige que exista homogeneidad entre los dispositivos modelados y el sistema debe ser balanceado.

Las ecuaciones obtenidas a partir del modelado del sistema permiten caracterizar el comportamiento y rendimiento del mismo. En el análisis operacional, en un problema, debe existir el elemento “sistema” y el elemento “período de tiempo”. El sistema puede ser real o supuesto y el período de tiempo puede ser pasado, presente o futuro.

1.4 XML

Extensible Markup Language (XML) [XML] ha sido un estándar para representar datos y meta-datos para categorizar datos e información por medio de etiquetas o tags. Por medio de estas etiquetas se pueden definir un conjunto de reglas en un formato que puede ser leído e interpretado por un usuario como por un sistema. XML permite la creación de documentos en que los datos agrupados pueden generar un elemento mucho más robusto. Por tal razón, dada estas características se puede generar elementos que representen estructuras, como lo pueden ser sentencias de código fuente: clases, atributos, métodos, etc. Existen trabajos que por medio de definiciones caracterizan estructuras de código fuente, para luego interpretar y generar un programa. En la literatura se pueden encontrar trabajos como los presentados en [Chiu04] en el cual usa XML para generar código en C++ utilizando un autómata generalizado. A su vez el trabajo presentado en [Kostoulas06] muestra un generador que

traduce esquemas XML a código C. Ese trabajo es extendido en [Matsa07] y principalmente se centra en un parser utilizando técnicas de optimización. Por su parte el trabajo presentado en [Dai10] pone énfasis en performance del parsing.

La idea de nuestra investigación es utilizar el modelado dirigido por especificaciones realizadas en XML para generar la descripción de componentes de hardware y generar el código ejecutable a ser simulado. El rendimiento de la lectura de las estructuras XML en la generación del código fuente están fuera de nuestro campo de investigación por el momento.

1.5 Stream Processing

Stream processing es un paradigma que soporta la recolección, el proceso y el análisis de grandes volúmenes de información heterogénea que se encuentra en continuo flujo, para extraer ideas y resultados en tiempo real. La diferencia que se puede encontrar entre stream processing y el análisis tradicional radica básicamente en la vía en que la información es leída. En una arquitectura convencional se presume que la información se encuentra almacenada en un repositorio. Por el contrario, al utilizar stream processing la información se divide en segmentos y cada segmento se procesa antes de que otro segmento llegue. Ejemplos donde se puede aplicar este paradigma es la Web 2.0, Sensores de redes, análisis de mercado – transacciones comerciales, etc. Dentro del mercado existen herramientas disponibles para realizar stream processing sobre clusters de computadoras como lo son: SPC (IBM) [Amini06], S4 (Yahoo!) [Neumeyer10][S4], Storm (Twitter) [Storm], Kinesis (Amazon Web Services) [Kinesis], entre otros.

Dentro del proyecto de investigación descrito en este trabajo, se considera que debido a que actualmente no existen herramientas que sean capaces de considerar adecuadamente las complejidades respecto de la predicción y evaluación del rendimiento computación para estas plataformas, se estudia una herramienta para aplicar modelos de rendimiento en la planificación para estos sistemas. La herramienta a considerar para el estudio es la plataforma de S4.

1.6 Simulación de Evacuaciones

En los últimos años se ha incrementado el interés en desarrollo de modelos de procesos de evacuaciones de emergencias. En nuestro proyecto nos enfocamos en modelos microscópicos que usan el concepto de autómatas celulares (AC) para modelar las características del ambiente. La conexión entre geometría del espacio simulado, la población y la propagación del fuego y del humo se realiza a través de reglas y/o ecuaciones.

Recientemente se han desarrollado modelos de simulación híbrida [TPL12] que consiste en dos sub-modelos denominados ambiental (EsM) y peatón [PsM]. Este modelo junto con la metodología computacional permite construir ambientes artificiales poblados con agentes autónomos que son capaces de interactuar entre sí.

2. LINEAS DE INVESTIGACION y DESARROLLO

La línea de investigación descrita en la sección anterior involucra una serie de desarrollos individuales que en su conjunto logran obtener el objetivo planteado. Este objetivo contempla el modelado y diseño de sistemas complejos mediante diferentes herramientas, algunas de las cuales han sido desarrolladas por los integrantes del proyecto.

Mediante estas herramientas se pretende facilitar el entendimiento del sistema, realizar profiling a la ejecución de los algoritmos y mejorar el análisis de resultados.

3. RESULTADOS OBTENIDOS ESPERADOS

Modelado y simulación

Los resultados obtenidos hasta el momento son:

- Se ha modelado mediante UML un sistema que debe procesar grandes volúmenes de datos y posee una gran cantidad de situaciones de incertidumbre debido a las fallas que pueden ocurrir. Este trabajo se realizó sobre un escenario de explotación minera [Giu14].
- Se ha mostrado que es posible modelar por medio de XML componentes correspondientes a un sistema de gran escala y luego traducirlo a un modelo de Redes de Petri. [Gil12]
- Diseño de un simulador de petri nets paralelo que se ajuste a las caracterizaras de un motor de búsqueda [Gil14]. El sistema incluye el

modelado de operaciones de lecturas y escrituras concurrentes.

Los resultados esperados son:

- Extender el modelado del sistema mediante UML presentado en [Giu14] para desarrollar sistemas de simulación que permitan estimar métricas y factores relevantes.
- Mostrar que XML puede emplearse como una etapa previa o de transición hacia la implementación de la simulación.
- Producir una herramienta gráfica que facilite la creación del código XML.
- Analizar el consumo de energía tanto en los centros de datos donde se ejecutan las aplicaciones como en las redes de intercomunicación.

Técnicas de 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 sistema de stream processing.
- Obtención de parámetros que permiten estimar el comportamiento de la plataforma de S4.

Los resultados esperados son:

- Desarrollo de una metodología que permita evaluar la capacidad computacional de un sistema de stream processing.
- Validación de las fórmulas y metodología desarrollada.
- Diseño e implementación de un simulador para plataformas de stream processing.

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. Se cuenta con dos estudiantes de doctorando. Uno en etapa final y otro se encuentra iniciando su trabajo en el análisis de sistemas de stream processing como lo es S4, para luego realizar un simulador para dicha plataforma. También se cuenta con 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 (previamente evaluados y analizados con un mínimo porcentaje de error mediante las técnicas de modelado y simulación) que se ejecuten en sistemas complejos donde intervienen una gran cantidad de variables y requieren el procesamiento de datos masivos.

5. BIBLIOGRAFIA

- [Amini06] "SPC: a distributed, scalable platform for data mining". Lisa Amini, Henrique Andrade, Ranjita Bhagwan, Frank Eskesen, Richard King, Philippe Selo, Yoonho Park, and Chitra Venkatramani. S. In Proceedings of the 4th international workshop on Data mining standards, services and platforms (DMSSP '06). 2006.
- [Booch98] Booch G.; Rumbaugh J.; Jacobson I., (1998), The Unified Modeling Language User Guide (Addison-Wesley Object Technology Series).
- [Chiu04] "A Compiler-Based Approach to Schema-Specific XML Parsing". Kenneth Chiu and Wei Lu. 2004.
- [Dai10] "A 1 cycle-per-byte XML parsing accelerator". Dai, Zefu and Ni, Nick and Zhu, Jianwen. Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays. 2010.
- [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.
- [Gil14] "Modelling Search Engines Performance using Coloured Petri Nets". Veronica Gil-Costa, Mauricio Marin, Alonso Inostrosa-Psijas, Jair Lobos, Carolina Bonacic. *Fundamenta Informaticae*, 130 (2014) 1–28. DOI 10.3233/FI-2014-900 – 2014.
- [Giu14] "UML Profile for Mining Process: Supporting Modeling and Simulation based on Metamodels of Activity Diagram", Andrea Giubergia, Daniel Riesco, Veronica Gil-Costa and Marcela Printista. *Journal Modelling and Simulation in Engineering*. Hindawi. Vol. 2014. Article ID 974850, 10 pages. <http://dx.doi.org/10.1155/2014/974850> - 2014.
- [Kinesis] Amazon Kinesis. <https://aws.amazon.com/es/kinesis/>. 2014.
- [Kostoulas06] "XML screamer: an integrated approach to high performance XML parsing, validation and deserialization". Margaret Gaitatzes Kostoulas, Morris Matsa, Noah Mendelsohn, Eric Perkins, Abraham Heifets, Martha Mercaldi. *WWW*. 2006.
- [Marin13] Approximate Parallel Simulation of Web Search Engines", M. Marin, V. Gil-Costa, C. Bonacic and R. Solar. In *ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS 2013)*, May 2013
- [Matsa07] "A high-performance interpretive approach to schema-directed parsing". Matsa Morris, Perkins Eric, Heifets Abraham, Kostoulas Margaret Gaitatzes, Silva Daniel, Mendelsohn Noah Leger Michelle. *Proceedings of the 16th international conference on World Wide Web*. *WWW'07*. 2007.
- [Neumeyer10] Leonardo Neumeyer, Bruce Robbins, Anish Nair, Anand Kesari: S4: Distributed Stream Computing Platform. *ICDM Workshops 2010*: 170-177
- [Peterson77] J.L. Peterson. *Petri nets*. *Computing Surveys*, 9:223–252, 1977.
- [Petri62] C.A. Petri. "Communication with Automata" New York: Griffiss Air Force Base. Tech. Rep. RADC-TR-65-377, vol.1, Suppl. 1, 1962.
- [S4] S4: Distributed stream computing platform. <http://incubator.apache.org/s4/>, 2014.
- [Storm] Storm. <http://github.com/nathanmarz/storm/wiki>. 2014.
- [TPL12] Pablo Cristian Tissera, A. Marcela Printista, and Emilio Luque. *ICCS, volume 9 of Procedia Computer Science, page 266-275. Elsevier, (2012)*
- [Val90] L. G. Valiant. A bridging model for parallel computation. *Comm. ACM*, 33(8) 1990.
- [XML] W3C. Extensible markup language (xml)