

Aspectos de Implementación de Servidores Web con el modelo BSP de computación paralela^{*}

Esteban Gesto, Daniel Laguía, Osiris Sofia

Universidad Nacional de la Patagonia Austral

Río Gallegos, Argentina

{egesto;dlaguia;osofia}@unpa.edu.ar

and

Mauricio Marín, José Canumán

Universidad de Magallanes

Punta Arenas, Chile

{mauricio.marin;jose.canuman}@umag.cl

Resumen

En el marco del Proyecto de Investigación *Paralelización Eficiente de Servidores Web*, de la Universidad Nacional de la Patagonia Austral se ha abierto una línea de investigación que da continuidad al desarrollo de servidores web soportados en clusters de PC a través del modelo BSP de computación paralela y que tiene como objetivo estudiar estrategias de implementación de servidores paralelos en entornos reales.

El propósito de este trabajo es presentar los resultados alcanzados en esta línea de investigación, los desarrollos en progreso y los trabajos futuros.

Palabras claves: Bases de Datos, Procesamiento Paralelo de Consultas SQL, Computación Paralela y Distribuida, BSP

1. Introducción

La web se ha convertido en un recurso ubicuo para la computación distribuida, haciendo relevante la investigación de nuevos caminos para proveer acceso eficiente a los servicios disponibles en los sitios dedicados. El crecimiento exponencial que ha experimentado desde sus comienzos en cuanto al volumen de información y al número de usuarios que la utilizan hace que la búsqueda, organización, acceso y mantenimiento de sus contenidos sea cada vez más difícil.

En respuesta a esta expansión de las fuentes potenciales de información, los motores de búsqueda

han hecho énfasis en ampliar su velocidad y cobertura, brindando poca importancia a la eficiencia.

Debido a esto, diversos estudios se han abocado al desarrollo de nuevas estrategias que permitan satisfacer estas demandas a través del procesamiento paralelo [18], el cual ha demostrado ser un paradigma que permite mejorar los tiempos de ejecución de los algoritmos.

Particularmente nuestro grupo de investigación, integrado por docentes investigadores de la Universidad Nacional de la Patagonia Austral (Argentina) y de la Universidad de Magallanes (Chile), se ha abocado al estudio y desarrollo de herramientas de programación basadas en el modelo BSP [17, 2] de computación paralela, el cual utiliza una configuración de base de datos distribuida para acelerar las consultas.

1.1. Modelo de computación paralela BSP

En BSP un computador paralelo es visto como un conjunto de procesadores con memoria local e interconectados a través de una red de comunicaciones de topología transparente al usuario. En este modelo, la computación es organizada como una secuencia de *supersteps*. Tal como lo indica la Fig. 1, un superstep está formado por una fase en la que cada procesador puede realizar operaciones sobre datos locales únicamente y depositar mensajes a ser enviados a otros procesadores. Al final del superstep, todos los mensajes son enviados a sus destinos y los procesadores son sincronizados en forma de barrera para iniciar el siguiente superstep. Es decir, los mensajes están disponibles en sus destinos al instante en que se inicia el siguiente superstep.

^{*}Este trabajo fue financiado por la Universidad Nacional de la Patagonia Austral, Santa Cruz, Argentina, Proyecto 29/A164

El modelo práctico de programación paralela en BSP es el conocido SPMD (Simple Program Multiple Data), el cual es realizado mediante P copias del mismo programa corriendo en un cluster de P procesadores, cada una actuando sobre un subconjunto de los datos, donde la comunicación y sincronización de las copias es realizada mediante librerías tales como BSPlib o BSP-PUB [1].

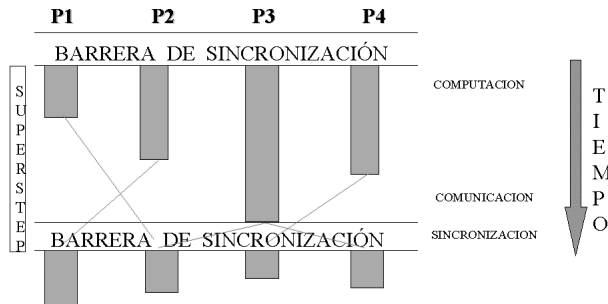


Figura 1: Modelo BSP y supersteps.

En el marco del Proyecto de Investigación *Paralelización Eficiente de Servidores Web*, de la Universidad Nacional de la Patagonia Austral se ha abierto una línea de investigación que da continuidad al desarrollo de servidores web soportados en clusters de PC a través del modelo BSP de computación paralela y que tiene como objetivo estudiar estrategias de implementación de servidores paralelos en entornos reales.

El propósito de este trabajo es presentar los resultados alcanzados en esta línea de investigación, los desarrollos en progreso y los trabajos futuros.

2. Resultados de Implementaciones con BSP

Para la implementación de una aplicación que permitiera una aproximación a la solución para el problema planteado del acceso a grandes volúmenes de información, se realizaron varios estudios, tendientes inicialmente a la implementación de cluster y la distribución de los registros de la base de datos [9, 7, 8]. La solución adoptada implementa una base de datos distribuida del tipo relacional, sobre una plataforma computacional formada por un grupo de PC con sistema operativo Linux, los cuales están conectados en red mediante un switch de alto desempeño y en estas máquinas se ejecutan los programas con supersteps de BSP. En cada PC se encuentra instalado un administrador de bases de datos relacional llamado MySQL [15] y la librería ApiMySQL que

proporciona clases que permiten enviar string con sentencias SQL desde un programa C++ al servidor MySQL.

Esta implementación es de bajo costo, ya que el software involucrado es de dominio público y el equipamiento corresponde a computadoras de escritorio de bajo costo, en contraposición a los servidores secuenciales de alto desempeño y costo.

La base de datos se distribuye uniformemente para permitir el balance de carga en los procesadores involucrados. Cada superstep envía la consulta a todos los procesadores, y los servidores de bases de datos locales realizan la consulta sobre su porción de la base de datos de manera secuencial. El resultado de la consulta SQL corresponde a la unión de todas las respuestas parciales de los procesadores del cluster.

La estructura del modelo BSP facilita la predicción del desempeño de programas y algoritmos. El costo de un programa está dado por la suma del costo de todos sus supersteps, donde el costo temporal de cada uno de ellos está dado por la suma del tiempo de computación sobre datos locales, el tiempo de comunicación entre procesadores y el tiempo de sincronización.

Aprovechando esta facilidad del modelo BSP se ha desarrollado una herramienta gráfica [11, 10, 14, 12, 13] que a través de metáforas visuales permite observar y administrar la cola de consultas y los supersteps ejecutados con distintos niveles de granularidad, evaluando el desempeño de cada uno de los parámetros que los integran.

En los objetivos del proyecto se encuentra el desarrollo de una aplicación funcional, para lo cual se ha elegido la implementación de un *Digesto Digital Institucional* [5, 6] que realiza consultas complejas del tipo *Join* [13] y la misma posee características de Base de Datos Textual.

Esta herramienta de software se compone de los siguientes elementos: una aplicación web desarrollada fundamentalmente con PHP, instalada en un servidor Web con infraestructura *LAMP* (Linux, Apache, MySQL, PHP), un servidor de base de datos distribuida implementado con MySQL, un programa ejecutable desarrollado bajo el modelo de computación paralela BSP y un *broker* realizado en base a la modificación del *daemon* del BSP-PUB, *pubd*.

Estos componentes se interrelacionan de la siguiente manera: en primer lugar los clientes acceden a la aplicación a través del servidor web, donde elaboran la consulta a realizar. El servidor web envía la consulta al *broker* a través de sockets, ejecutando el programa BSP en cada procesador que forma parte del *cluster*. En cada

procesador se ejecuta la consulta en el servidor local de base de datos MySQL y, cuando se produce la etapa de sincronización del *superstep* BSP, los resultados son enviados a la máquina *broker*, quien los agrupa y envía al servidor Web a través de los sockets definidos para que de esta manera el cliente pueda obtener los resultados requeridos.

3. Conclusiones

En este trabajo se ha presentado una de las líneas de investigación del Proyecto de Investigación *Paralelización Eficiente de Servidores Web*, de la Universidad Nacional de la Patagonia Austral que da continuidad al desarrollo de servidores web soportados en clusters de PC a través del modelo BSP de computación paralela y que tiene como objetivo estudiar estrategias de implementación de servidores paralelos en entornos y aplicaciones reales.

Los trabajos realizados presentan una solución concreta al problema de acceso a grandes volúmenes de datos a través de la Web, mediante el desarrollo de varios componentes que conforman un motor de búsquedas paralelo, con acceso a una base de datos distribuida, implementando el modelo de computación paralela BSP, en particular a través de la librería BSP-PUB. Este modelo soporta una metodología estructurada de diseño de software que es simple de utilizar y permite el uso de tecnología existente y gratuita para obtener sistemas de bajo costo y alta eficiencia.

La distribución de los registros influye significativamente en la velocidad de respuesta de las consultas, obteniendo hasta el momento los mejores resultados con una distribución uniforme.

Se han desarrollado herramientas gráficas para visualizar y administrar el desempeño del servidor de base de datos, mediante distintas metáforas visuales, que permiten observar con distintos niveles de granularidad las consultas realizadas y los *supersteps* generados a los largo de la ejecución de la aplicación.

A través de la experimentación de laboratorio se han obtenido resultados satisfactorios en la etapa de simulación, con valores iguales o mayores al óptimo en relación con el caso secuencial, para los casos de tratamiento de grandes volúmenes de datos donde se justifica la utilización de modelos paralelos. Similares resultados se obtuvieron en la implementación de una aplicación real y funcional del Digesto Digital Institucional. Para este último caso fue necesario la modificación de la librería BSP-PUB para permitir su funcionamiento continuo en el servidor a través de su definición como *daemon*.

Actualmente nos encontramos estudiando distintas alternativas de implementación de la cola de consultas [16, 3, 4] para mejorar el rendimiento

ajustando parámetros y características de la cola de consultas SQL, de manera de tomar en cuenta el tiempo de espera de los clientes que han generado las consultas para maximizar su satisfacción al realizar un acceso a la aplicación.

4. Trabajos Futuros

Entre nuestros trabajos futuros se encuentran el estudio de la recuperación de la base de datos distribuida a raíz de la caída de una máquina del cluster, mediante la replicación de registros u otra estrategia. También es posible su aplicación a redes de Datos del tipo *Grid*, en el supuesto de la necesidad de contar con una base de datos distribuida geográficamente para respetar la autonomía de las organismos intervinientes para una aplicación dada, tal como el Digesto Digital Institucional.

Referencias

- [1] O. Bonorden, B. Juurlink, I. von Otte, and I. Rieping. The paderborn university bsp (pub) library - design, implementation and performance. In *13th International Parallel Processing Symposium*, San Juan, Puerto Rico, Apr. 1999.
- [2] BSP and Worldwide Standard. <http://www.bsp-worldwide.org/>.
- [3] Ricardo Cao. *Introducción a La Simulación y a La Teoría De Colas*. Pearson Publications, 2004.
- [4] A. V. Gerbessiotis, C. J. Siniolakis, and A. Tiskin. Parallel priority queue and list contraction: The bsp approach. *Computing and Informatics*, 21:59–90, 2002.
- [5] Esteban Gesto, Daniel Laguia, Natalia Trejo, Osiris Sofia, and Jose Canumán. Implementación de un digesto digital paralelo. In *Anales del VIII Workshop de Investigadores en Ciencias de la Computación*, Universidad de Morón - Argentina, Junio 2006. WICC 2006.
- [6] Esteban Gesto, Daniel Laguia, Natalia Trejo, Osiris Sofia, and Jose Canumán. Implementación de un motor de búsquedas paralelo con bsp. In *32a Conferencia Latinoamericana de informática*, Santiago de Chile - Chile, Agosto 2006. CLEI 2006.
- [7] M. Marín, J. Canuman, M. Becerra, D. Laguia, and O. Sofia. Procesamiento paralelo de consultas sql generadas desde la web.

- In *Jornadas Chilenas de Computación 2001*, Punta Arenas-Chile, Nov. 2001.
- [8] M. Marín, J. Canuman, M. Becerra, D. Laguna, and O. Sofia. Servidor paralelo sql-bsp para aplicaciones web. In *VII Congreso Argentino de Ciencia de la Computación*, El Calafate - Argentina, Oct. 2001. CACIC 2001.
- [9] M. Marin, J. Canumán, and D. Laguna. Un modelo de predicción de desempeño para bases de datos relacionales paralelas sobre bsp. In *VI Congreso Argentino de Ciencia de la Computación*, Ushuaia - Argentina, Oct 2000. CACIC 2000.
- [10] Paula Millado, Daniel Laguna, Albert Sofia, and Mauricio Marín. Representación visual para la administración del procesamiento paralelo de consultas sql. In *IX Congreso Argentino de Ciencia de la Computación*, La Plata - Argentina, Oct. 2003. CACIC 2003.
- [11] Paula Millado, Daniel Laguna, Albert Sofia, and Mauricio Marín. Visualización gráfica de consultas sql en paralelo. In RITOS2, editor, *Ingeniería de Software en la década del 2000*, Cartagena-Colombia, Aug. 2003. IX Jornadas Iberoamericanas de Informática.
- [12] Paula Millado, Daniel Laguna, Albert Sofia, Mauricio Marín, and Claudio Delrieux. Administrador visual de entornos bsp. In *VI Workshop de Investigadores en Ciencias de la Computación*, Neuquén - Argentina, May. 2004. WICC 2004.
- [13] Paula Millado, Daniel Laguna, Albert Sofia, Mauricio Marín, and Claudio Delrieux. Simulación y visualización de la performance de un administrador bsp. In *30TH Conferencia Latinoamericana de Informática*, Arequipa - Perú, Sept. 2004. CLEI 2004.
- [14] Paula Millado, Daniel Laguna, Albert Sofia, Mauricio Marín, and Claudio Delrieux. Visualización y control de congestión en un administrador bsp. In *Simposio Argentino de Tecnología 2004*, Córdoba - Argentina, Jun. 2004. AST 2004.
- [15] MySQL Web Page. <http://www.mysql.com/>.
- [16] J. Pasos. *Teoría de Colas y Simulación de Eventos Discretos*. Pearson Publications, 2004.
- [17] D.B. Skillicorn, J.M.D. Hill, and W.F. McColl. Questions and answers about BSP. Technical Report PRG-TR-15-96, Computing Laboratory, Oxford University, 1996. Also in *Journal of Scientific Programming*, V.6 N.3, 1997.
- [18] L.T. Yang and M. Guo, editors. *High Performance Computing: Paradigm and Infrastructure*. John Wiley and Sons, 2005.