

Paralelización del D-Index para Búsquedas en Espacios Métricos Aplicado a los Frameworks BSP y Westershed

Norma Beatriz Perez, Mario Berón

Facultad de Ciencias Físico Matemáticas y Naturales - Universidad Nacional de San Luis
Ejército de los Andes, 950 CP: 5700 - San Luis - Argentina
email: {nbperez, mberom}@unsl.edu.ar

Fernando Magno Quintão Pereira

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais
Av. Antônio Carlos, 6627 - Prédio do ICEX - Pampulha - CEP: 31270-010
Belo Horizonte Minas Gerais - Brasil
email: proneto@ufmg.br

Resumen

La gran cantidad de datos que se producen hoy en día hace que la recuperación eficiente de la información sea un problema fundamental en la ciencia de la computación. La paralelización de los algoritmos de búsqueda, diseñados originalmente para ambientes de ejecución secuencial es una manera natural de lograr la velocidad que se necesita. Un algoritmo de este tipo es el D-Index: emplea una técnica de búsqueda por similitud basado en la agrupación de datos similares, siendo introducido en el 2003.

El objetivo de este proyecto, que reúne a investigadores de la Universidad de San Luis (UNSL), Argentina, y la Universidade Federal de Minas Gerais (UFMG), Brasil, es la implementación de la primera versión paralela del D-Index. Actualmente estamos experimentando con varios enfoques diferentes para conseguir paralelismo. Ca-

da enfoque difiere de los otros en la manera de dividir los datos entre los elementos de procesamiento distribuido, y cómo enviar consultas a estos elementos. Implementamos estos enfoques sobre dos modelos diferentes de ejecución en paralelo: *Bulk Synchronous Parallel (BSP)* y *filter-stream*.

Los experimentos preliminares muestran que cada esquema de paralelización ofrece speedup casi lineal sobre el número de procesadores.

Palabras clave: D-Index, Modelo BSP, filter-stream.

1. Contexto

La línea de investigación descrita en este trabajo se encuentra enmarcada en el contexto del proyecto: *“Ingeniería del Software: Conceptos, Métodos, Técnicas y Herramientas en un Con-*

texto de Ingeniería de Software en Evolución” de la Universidad Nacional de San Luis. Dicho proyecto, es reconocido por el programa de incentivos y es la continuación de diferentes proyectos de investigación de gran éxito a nivel nacional e internacional. Además se forma parte del proyecto de co-tutela entre la Universidad Nacional de San Luis - Argentina y la Universidade Federal de Minas Gerais, Belo Horizonte - Brasil. Este proyecto denominado *CAFP-BA 004/08* fue aprobado por la Secretaría de Políticas Universitarias (SPU) dependiente del Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación (MinCyT) [1]. Ambos entes soportan económicamente la realización de diferentes misiones de investigación desde Argentina a Brasil y viceversa.

2. Introducción

Hoy en día las colecciones de datos han incluido la capacidad de almacenar nuevos tipos de datos, tales como imágenes, videos, documentos entre otros objetos multimedia. Sin embargo, es muy complicado, y hasta imposible estructurar un video, una huella dactilar, etc. en campos de un registro haciendo que los métodos tradicionales de búsqueda exacta ya no sean suficientes. Es necesario contar con nuevas formas de búsqueda sobre estos tipos de datos no estructurados.

En los últimos años se ha producido un importante avance en la investigación de métodos que permitan resolver *búsquedas por similitud*, las que consisten en buscar aquellos objetos en una colección de datos que sean *similares* a un objeto de consulta dado, satisfaciendo a la necesidad de una respuesta rápida y adecuada, logrando la aparición de nuevos índices que incluyen estos aspectos.

El modelo de los *Espacios Métricos* [2] provee un marco formal dentro del cual se pueden estudiar dichos métodos. Sea D un universo de objetos, y d una función de distancia total definida como $d : D \times D \mapsto \mathbb{R}^+$. Si la función d cumple con las propiedades:

1. $(\forall x, y \in D), d(x, x) = 0 \wedge x \neq y \Leftrightarrow d(x, y) > 0$
2. $(\forall x, y \in D), d(x, y) = d(y, x)$
3. $(\forall x, y, z \in D), d(x, z) \leq d(x, y) + d(y, z)$

el par $\mathcal{M} = (D, d)$ define un *espacio métrico*. Partiendo de la noción de distancia, definimos las consultas típicas como sigue. Sea $S \subseteq D$:

- **Búsqueda por Rango:** recupera todos los objetos S a distancia r de un objeto q dado.
- **Búsqueda de K vecinos más cercanos:** dado q , recupera los k objetos más cercanos a q de S .

Los índices para los espacios métricos se pueden clasificar en técnicas basadas en pivotes [3] o basadas en clustering [3]. Donde existen métodos efectivos para buscar sobre espacios D -dimensionales, tales como R-Trees [4], Kd-Trees [5], X-Trees [6], etc. Sin embargo, para 20 dimensiones o más estos índices dejan de funcionar correctamente.

Definido por Vlastislav Dohnal [7], el D-Index [8] es un índice de búsqueda por similitud que es capaz de reducir tanto el costo de I/O como el de CPU [9]. Combina una innovadora técnica de clustering con estimación de distancias basadas en pivotes para acelerar la ejecución de las consultas por similitud, para grandes colecciones de datos con objetos almacenados en memoria secundaria. Este índice crea una estructura de almacenamiento de múltiples niveles, utilizando en cada nivel una función denominada

ρ -split [10] para separar los objetos en diferentes particiones. Cada nivel se divide en buckets y uno de estos buckets, denominado bucket de exclusión almacena los objetos que no pertenecen a ninguno de los buckets del nivel que esta siendo analizado.

La resolución de consultas sobre este tipo de índices tiende a ser muy costosa debido a la dificultad que implica la ejecución de la función de similitud, la cual depende del tipo de dato utilizado. Además, existen un gran número de aplicaciones que reciben grandes lotes de consultas a resolver por unidad de tiempo donde construir índices, en muchos casos, no es suficiente.

Por otro lado, la diversidad de modelos de computación paralela que han surgido en los años 80 y las numerosas investigaciones han dejado en evidencia que estos modelos no son los suficientemente generales como para ser modelos de referencia. Cada modelo se ajusta con demasiada precisión a una arquitectura específica considerada. Hoy en día, el número de modelos y lenguajes de computación paralela excede la cantidad de arquitecturas diferentes, pero la mayoría son inadecuadas porque es difícil obtener portabilidad, performance o ambas. Estas dificultades trazan nuestro objetivo de investigación.

Nos centramos en estudiar el modelo BSP, propuesto en los años 90 por Leslie Valiant [11]. BSP tiene como principal objetivo permitir el desarrollo de software escalable e independiente de la arquitectura y proveer un entorno de trabajo simple y práctico para computaciones paralelas de propósito general. Las características claves de BSP son el tratamiento del medio de comunicación como una red abstracta totalmente conectada y un modelo de costo de la comunicación y sincronización independiente y explícito.

En adición al modelo BSP, en este proyecto también estamos experimentando con el modelo

filter-stream. Este modelo de paralelización tiene sus orígenes en trabajos como procesadores de flujo de datos de Duane Adam [12], y lenguajes de programación orientado al flujo, tal como Lucid [13]. Desde esas ideas pioneras, el modelo de programación *filter-stream* ha evolucionado sustancialmente.

Hoy en día, el flujo de procesamiento del hardware y los lenguajes son parte de la caja de herramientas de las compañías como IBM [14] y SPI [15]. La característica principal del modelo de *filter-stream* es la separación entre el procesamiento de datos y la comunicación de datos en una forma modular. Los datos son agrupados en flujos, y luego procesados en un ambiente de ejecución de los filtros que ellos procesaran. En este proyecto se utiliza el marco de trabajo *Westershed* [16], desarrollado en la UFMG, con el objetivo de facilitar la implementación del *filter-stream* para aplicaciones orientadas.

3. Líneas de investigación y desarrollo

Este proyecto tiene como objetivo principal reducir los costos de búsqueda en colecciones organizadas sobre los espacios métricos. Con tal propósito, estamos trabajando en la paralelización del D-Index en un ambiente de computación distribuida en donde se aplican dos modelos de computación paralela.

Nuestro estudio, para el diseño y construcción de nuestros algoritmos se planificaron en base a:

- Un balance de carga adecuado con la finalidad de que ningún procesador quede ocioso.
- Una distribución uniforme de la colección de datos, esto asegura un mejor balance de carga.

- Reducir el número de iteraciones (o superpasos) esto permite reducir el número de sincronizaciones por barrera.

La implementación paralela del D-Index se realizó con tres enfoques diferentes.

- Enfoque Trivial: utilizada como punto de referencia para las comparaciones, consiste en la simple replica de la colección completa en cada nodo del clúster.
- Enfoque Global: divide al D-Index por nivel entre los nodos.
- Enfoque Local: no sólo divide los datos, sino también sus índices entre todos los nodos.

Para el análisis experimental se utilizan diferentes colecciones de datos que varían en tipos de objetos, cantidad de objetos, dimensión, función de distancia, etc.

4. Resultados

Durante este proyecto, esperamos (a) obtener una mejora sustancial en la performance del sistema basados en tres enfoques. (b) realizar un estudio comparativo de los modelos de computación paralela propuestos.

5. Formación de Recursos Humanos

Los trabajos elaborados en la presente línea de investigación forman parte del desarrollo de tesis para optar a los grados de Magister en Ingeniería del Software (UNSL) y Mestre em Ciência da Computação (UFMG). Es importante mencionar que tanto el equipo de la Argentina como

el de Brasil se encuentran dedicados a la captura de alumnos de grado y posgrado para la realización de estudios de investigación relacionados con las temáticas presentadas en este trabajo. Dichos estudios pretenden fortalecer la relación entre UNSL y la UFMG.

Referencias

- [1] "Ministerio de ciencia, tecnología e innovación productiva," 2007. [Online]. Available: <http://www.mincyt.gov.ar/>
- [2] G. Navarro, "Searching in metric spaces by spatial approximation," in *SPIRE*. IEEE, 1999, pp. 141–148.
- [3] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín, "Searching in metric spaces," *ACM Comput. Surv.*, vol. 33, pp. 273–321, 2001.
- [4] A. Guttman, "R-trees: a dynamic index structure for spatial searching," in *SIGMOD*. ACM, 1984, pp. 47–57.
- [5] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, pp. 509–517, 1975.
- [6] S. Berchtold, D. A. Keim, and H. Kriegel, "The X-tree: An index structure for high-dimensional data," in *VLDB*. ACM, 1996, pp. 28–39.
- [7] V. Dohnal, C. Gennaro, P. Savino, and P. Zezula, "D-Index: Distance searching index for metric data sets," *Multimedia Tools and Applications*, vol. 21, pp. 9–33, 2003.
- [8] V. Dohnal, "Indexing structure for searching in metric spaces," Ph.D. dissertation, Masaryk University, 2004.

- [9] V. Dohnal, C. Gennaro, P. Savino, and P. Zezula, "D-index: Distance searching index for metric data sets," *Multimedia Tools Appl.*, vol. 21, no. 1, pp. 9–33, 2003.
- [10] P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search: The Metric Space Approach*, ser. ADS. Springer, 2006, vol. 32.
- [11] L. G. Valiant, "A bridging model for parallel computation," *Commun. ACM*, vol. 33, pp. 103–111, 1990.
- [12] D. A. Adams, "A computation model with data flow sequencing," Ph.D. dissertation, Stanford University, 1969.
- [13] E. A. Aschcroft and W. W. Wadge, "Lucid, a nonprocedural language with iteration," *Communications of the ACM*, vol. 20, pp. 519–526, 1977.
- [14] B. Gedik, H. Andrade, K.-L. Wu, P. S. Yu, and M. Doo, "SPADE: The System S declarative stream processing engine," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2008, pp. 1123–1134.
- [15] *A Programmable 512 GOPS Stream Processor for Signal, Image, and Video Processing*, 2007.
- [16] R. A. Ferreira, W. Meira Jr., D. Guedes, L. M. A. Drummond, B. Coutinho, G. Teodoro, T. Tavares, R. Araujo, and G. T. Ferreira, "Anthill: A scalable runtime environment for data mining applications," in *Proceedings of the 17th International Symposium on Computer Architecture on High Performance Computing (SBAC-PAD)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 159–167.