Paralelización de Estructuras Métricas para Búsquedas por Similaridad en Servidores Web.*

Osiris Sofia
Universidad Nacional de la Patagonia Austral
Río Gallegos, Argentina
osofia@unpa.edu.ar

and

Roberto Uribe Paredes
Departamento de Ingeniería en Computación
Universidad de Magallanes
Punta Arenas, Chile
ruribe@ona.fi.umag.cl

Resumen

La *búsqueda por similaridad* consiste en recuperar todos aquellos objetos dentro de una base de datos que sean parecidos o relevantes a una determinada consulta. Este concepto tiene una amplia gama de aplicaciones en áreas como bases de datos multimediales, reconocimiento de patrones, minería de datos, recuperación de información, etc.

La posibilidad de fusionar dos líneas de investigación independiente, como es, el desarrollo de estructuras de datos para búsquedas por similitud y la necesidad de procesar grandes volúmenes de datos usando computación paralela, permitirá la utilización de estas nuevas estructuras en aplicaciones reales.

El presente artículo describe la línea de investigación conjunta de un grupo de investigadores de la Universidad de Magallanes y de la Universidad Nacional de la Patagonia Austral a través del programa de investigación "Paralelización de Estructuras de Datos y Algoritmos para la Recuperación de Información", el cual permitirá el diseño, implementación y evaluación de estructuras métricas paralelas.

Palabras claves: bases de datos, estructuras de datos, algoritmos, espacios métricos, consultas por similaridad, paralelismo, modelo BSP.

*Este trabajo fue financiado por la Universidad Nacional de la Patagonia Austral, proyecto "Paralelización de Estructuras de Datos y Algoritmos para la Recuperación de Información" y la Universidad de Magallanes

1. Introducción

1.1. Antecedentes

Uno de los problemas de gran interés en ciencias de la computación es el de "búsqueda por similitud", es decir, encontrar los elementos de un conjunto más similares a una muestra. Esta búsqueda es necesaria en múltiples aplicaciones, como ser en reconocimiento de voz e imagen, compresión de video, genética, minería de datos, recuperación de información, etc. En casi todas las aplicaciones la evaluación de la similaridad entre dos elementos es cara, por lo que usualmente se trata como medida del costo de la búsqueda la cantidad de similaridades que se evalúan.

Interesa el caso donde la similaridad describe un espacio métrico, es decir, está modelada por una función de distancia que respeta la desigualdad triangular. En este caso, el problema más común y difícil es en aquellos espacios de "alta dimensión" donde el histograma de distancias es concentrado, es decir, todos los objetos están más o menos a la misma distancia unos de otros.

El aumento de tamaño de las bases de datos y la aparición de nuevos tipos de datos sobre los cuales no interesa realizar búsquedas exactas, crean la necesidad de plantear nuevas estructuras para búsqueda por similaridad o búsqueda aproximada. Asimismo, se necesita que dichas estructuras sean dinámicas, es decir, que permitan agregar o eliminar elementos sin necesidad de crearlas nuevamente, así como también que sean óptimas en la administración de memoria secundaria. La necesidad de procesar grandes volúmenes de datos obligan a aumentar la capacidad de procesamiento y con ello la paralelización de los algoritmos y la distribución de las bases de datos.

1.2. Marco teórico

La similaridad se modeliza en muchos casos interesantes a través de un espacio métrico, y la búsqueda de objetos más similares a través de una búsqueda por rango o de vecinos más cercanos.

Definición 1 (*Espacios Métricos*): Un espacio métrico es un conjunto X con una función de distancia $d : X^2 \rightarrow R$, tal que $\forall x, y, z \in X$,

1. $d(x, y) \geq 0$ *and* $d(x, y) = 0$ *ssi* $x = y$. *(positividad)*
2. $d(x, y) = d(y, x)$. *(Simetría)*
3. $d(x, y) + d(y, z) \geq (d(x, z)$. *(Desigualdad Triangular)*

Definición 2 (*Consulta por Rango*): Sea un espacio métrico *(X,d)*, un conjunto de datos finito $Y \subseteq X$, una consulta $x \in X$, y un rango $r \in R$. La consulta de rango alrededor de x con rango r es el conjunto de puntos $y \in Y$, tal que $d(x, y) \leq r$.

Definición 3 (*Los k Vecinos más Cercanos*): Sea un espacio métrico *(X,d)*, un conjunto de datos finito $Y \subseteq X$, una consulta $x \in X$ *y un entero k*. Los k vecinos más cercanos a x son un subconjunto A de objetos de Y, donde la $|A| = k$ y no existe un objeto $y \in A$ tal que *d(y,x)* sea menor a la distancia de algún objeto de A a x.

El objetivo de los algoritmos de búsqueda es minimizar la cantidad de evaluaciones de distancia realizadas para resolver la consulta. Los métodos para buscar en espacios métricos se basan principalmente en dividir el espacio empleando la distancia a uno o más objetos seleccionados. El no trabajar con las características particulares de cada aplicación tiene la ventaja de ser más general, pues los algoritmos funcionan con cualquier tipo de objeto [6].

Existen distintas estructuras para buscar en espacios métricos, las cuales pueden ocupar funciones discretas o continuas de distancia. Algunos son BKTree [4], MetricTree [18], GNAT [2], VpTree [22], FQTree [1], MTree [7], SAT [14], SlimTree [17], EGNAT [20].

Algunas de las estructuras anteriores basan la búsqueda en pivotes y otras en clustering. En el primer caso se seleccionan pivotes del conjunto de datos y se precalculan las distancias entre los elementos y los pivotes. Cuando se realiza una consulta, se calcula la distancia de la consulta a los pivotes y se usa la desigualdad triangular para descartar candidatos.

Los algoritmos basados en clustering dividen el espacio en áreas, donde cada área tiene un *centro*. Se almacena alguna información sobre el área que permita descartar toda el área mediante sólo comparar la consulta con su centro. Los algoritmos de clustering son los mejores para espacios de alta dimensión, que es el problema más difícil en la práctica.

Existen dos criterios para delimitar las áreas en las estructuras basadas en clustering, *hiperplanos* y *radio cobertor* (*covering radius*). El primero divide el espacio en particiones de *Voronoi* y determina el hiperplano al cual pertenece la consulta según a qué centro corresponde. El criterio de radio cobertor divide el espacio en esferas que pueden intersectarse y una consulta puede pertenecer a más de una esfera.

Definición 4 (*Diagrama de Voronoi*): Considérese un conjunto de puntos $\{c_1, c_2, \ldots, c_n\}$(centros). Se define el diagrama de Voronoi como la subdivisión del plano en n áreas, una por cada c_i, tal que $q \in$ al área c_i sí y sólo sí la distancia euclidiana $d(q, c_i) < d(q, c_j)$ para cada c_j, con $j \neq i$.

El ***EGNAT*** es una estructura basada principalmente en el diagrama de Voronoi, aunque igualmente usa radio cobertor. Está basada en el *GNAT* [2] que es una generalización del *Generalized Hyperplane Tree* (*GHT*) [18].

1.3. Modelo de computación paralela BSP

El modelo BSP de computación paralela fue propuesto en 1990 con el objetivo de permitir que el desarrollo de software sea portable y tenga desempeño eficiente y escalable [21, 16]. BSP propone alcanzar este objetivo mediante la estructuración de la computación en una secuencia de pasos llamados *supersteps* y el empleo de técnicas aleatorias para el ruteo de mensajes entre procesadores. El computador paralelo, independiente de su arquitectura, es visto como un conjunto de pares procesadores-memoria, los cuales son conectados mediante una red de comunicación cuya topología es transparente al programador. Los supersteps son delimitados mediante la sincronización de procesadores. Los procesadores proceden al siguiente superstep una vez que todos ellos han alcanzado el final del superstep, los cuales son agrupados en bloques para optimizar la eficiencia de la comunicación. Durante un superstep, los procesadores trabajan asincrónicamente con datos almacenados en sus memorias locales. Cualquier mensaje enviado por un procesador está disponible para procesamiento en el procesador destino sólo al comienzo del siguiente

superstep. Dada la estructura particular del modelo de computación, el costo de los programas BSP puede ser obtenido utilizando técnicas similares a las empleadas en el análisis de algoritmos secuenciales. En BSP, el costo de cada superstep esta dado por la suma del costo en computación (el máximo entre los procesadores), el costo de sincronización entre procesadores, y el costo de comunicación entre procesadores (el máximo enviado/recibido entre procesadores).

En el marco del Proyecto de Investigación *Paralelización de Estructuras de Datos y Algoritmos para la Recuperación de Información*, 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 estructuras métricas para búsquedas por similitud tanto en la paralelización de los algoritmos como en la distribución de las estructuras de datos.

2. Resultados Preliminares

La unificación de experiencias de los equipos de las distintas Universidades está formalizada en las distintas publicaciones en las áreas de Paralelismo ([10, 9, 13, 11, 8]), como en la de implementación y evaluación de estructuras métricas ([20, 3]). También se pueden mencionar resultados preliminares en la paralelización de estructuras métricas por parte de uno de los equipos ([5, 19, 15, 12]).

Inicialmente el trabajo de los equipos estará orientado a dos problemas, la *paralelización de los algoritmos* y a las estrategias utilizadas para la *distribución de la base de datos* sobre el cluster de PCs.

El contexto común para el estudio de las distintas estrategias de distribución de las bases de datos y paralelización de los algoritmos, es que existe una máquina broker que reparte las consultas de forma circular entre todas las máquinas.

En cada superstep cada máquina toma Q consultas (enviadas desde la máquina broker) y hace el proceso de búsqueda con dichas consultas, luego recoge todas las consultas provenientes de las demás máquinas y realiza el proceso de búsqueda con ellas. Entonces se procede a repartir las Q consultas (ya procesadas anteriormente) a las demás máquinas, y también se envían los resultados de las consultas a las máquinas que corresponda.

Entre las distintas medidas de costo a considerar en los estudios están, los cálculos de distancia y los accesos a disco durante la construccion y búsqueda de objetos. A su vez, es relevante mantener en forma adecuada el balance de carga sobre los procesadores como también un balance en la distribución de los datos entre los distintos componentes del cluster. En el análisis secuencial de las estruturas es de suma importancia mantener metodos eficientes de almacenamiento de éstas, de tal manera de evitar altos costos, tanto de accesos como de espacio en memoria secundaria.

3. Conclusiones

En este trabajo se ha presentado una de las líneas de investigación de un grupo conformado por investigadores de la Universidad Nacional de la Patagonia Austral, Argentina y de la Universidad de Magallanes, Chile. Esto da continuidad tanto al desarrollo de servidores web soportados en clusters de PC a través del modelo BSP de computación paralela como al diseño e implementación de estructuras métricas que permitan búsquedas aproximadas mas eficientes.

Estudios preliminares realizados por los equipos sobre paralelización de estructuras métricas, han generado resultados exitosos y prometedores en términos de obtener resultados que permitan soluciones adecuadas al problema presentado.

Se espera contar, al finalizar el proyecto de investigación conjunto, con el desarrollo de parte de una máquina de búsqueda por similitud, soportada sobre un cluster de PCs, que pueda ser utilizada como prototipo en aplicaciones de tipo real.

Referencias

[1] R. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. Proximity matching using fixedqueries trees. In *5th Combinatorial Pattern Matching (CPM'94)*, LNCS 807, pages 198–212, 1994.

[2] Sergei Brin. Near neighbor search in large metric spaces. In *the 21st VLDB Conference*, pages 574–584. Morgan Kaufmann Publishers, 1995.

[3] Nieves R. Brisaboa, Oscar Pedreira, Diego Seco, Roberto Solar, and Roberto Uribe. Clustering-based similarity search in metric spaces with sparse spatial centers. In *SOFSEM 2008: 34rd Conference on Current Trends in Theory and Practice of Computer Science*, volume 4910 of *Lecture Notes in Computer Science*, pages 186–197, Novy Smokovec, High Tatras, Slovakia, January, 19-25 2008. Springer.

[4] W. Burkhard and R. Keller. Some approaches to best-match file searching. *Communication of ACM*, 16(4):230–236, 1973.

[5] Roberto Uribe-Paredes Carlos Subiabre, Enrique Árias. Paralelización de los procesos de búsqueda y optimización en memoria secundaria para la etructura spaghettis. In *XIII Congreso Argentino de Ciencias de la Computación (Cacic2007)*, Corrientes, Argentina, Oct. 2007.

[6] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José L. Marroquín. Searching in metric spaces. In *ACM Computing Surveys*, pages 33(3):273–321, September 2001.

[7] P. Ciaccia, M. Patella, and P. Zezula. M-tree : An efficient access method for similarity search in metric spaces. In *the 23st International Conference on VLDB*, pages 426–435, 1997.

[8] 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.

[9] 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.

[10] M. Marin, J. Canumán, and D. Laguia. 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.

[11] M. Marín and S. Casas. Procesamiento paralelo de consultas a bases de datos textuales distribuidas. In *III Workshop de Investigadores en Ciencias de la Computación*. WICC 2001, May. 2002.

[12] Mauricio Marín, Roberto Uribe, and Ricardo J. Barrientos. Searching and updating metric space databases using the parallel egnat. In *Proc. of International Conference on Computational Science 2007 (ICCS 2007)*, volume 4487 (1) of *Lecture Notes in Computer Science*, pages 229–236, Beijing, China, May 2007. Springer.

[13] Paula Millado, Daniel Laguia, 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.

[14] Gonzalo Navarro. Searching in metric spaces by spatial approximation. *The Very Large Databases Journal (VLDBJ)*, 11(1):28–46, 2002.

[15] Eduardo Peña-Jaramillo. Estructuras métricas paralelas en la recuperación de imágenes. Master's thesis, Escuela de Ingeniería, Departamento de Ciencias de la Computación, Pontificia Católica de Chile, Santiago, Chile, Nov. 2006.

[16] 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.

[17] Caetano Traina, Agma Traina, Bernhard Seeger, and Christos Faloutsos. Slim-trees: High performance metric trees minimizing overlap between nodes. In *VII International Conference on Extending Database Technology*, pages 51–61, 2000.

[18] J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. In *Information Processing Letters*, pages 40:175–179, 1991.

[19] Roberto Uribe, Gonzalo Navarro, Ricardo J. Barrientos, and Mauricio Marín. An index data structure for searching in metric space databases. In *Proc. of International Conference on Computational Science 2006 (ICCS 2006)*, volume 3991 of *Lecture Notes in Computer Science*, pages 611–617. Springer, 2006.

[20] Roberto Uribe-Paredes. Manipulación de estructuras métricas en memoria secundaria. Master's thesis, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Santiago, Chile, Abril 2005.

[21] L.G. Valiant. A bridging model for parallel computation. *Comm. ACM*, 33:103–111, Aug. 1990.

[22] P. Yianilos. Data structures and algoritms for nearest neighbor search in general metric spaces. In *4th ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 311–321, 1993.