Búsquedas en Grandes Volúmenes de Datos

Luis Britos, María E. Di Gennaro, Veronica Gil-Costa, Fernando Kasián, Jair Lobos, Verónica Ludueña, Romina Molina, Marcela Printista, Nora Reyes, Patricia Roggero, Guillermo Trabes
LIDIC, Dpto. de Informática, Fac. de Cs. Físico Matemáticas y Naturales, Universidad Nacional de San Luis
{*lebritos, mdigena, gvcosta, fkasian, vlud, mprinti, nreyes,proggero*}@unsl.edu.ar
jairlobos@gmail.com, mromy00@gmail.com, guillermotrabes@hotmail.com
Edgar Chávez
Centro de Investigación Científica y de Educación Superior de Ensenada, México
elchavez@cicese.mx
Claudia Deco
Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario
deco@fceia.unr.edu.ar

Resumen

En la actualidad los sistemas de información demandan no sólo poder realizar búsquedas eficientes sobre distintos tipos de datos, tales como texto libre, audio, video, secuencias de ADN, etc., sino también poder manejar grandes volúmenes de estos datos. Dada una consulta, el objetivo de un sistema de recuperación de información es obtener lo que podría ser útil o relevante para el usuario, usando una estructura de almacenamiento especialmente diseñada para responderla eficientemente.

Nuestra línea de investigación tiene como principal objetivo desarrollar herramientas para sistemas de información sobre bases de datos masivas, conteniendo datos multimedia, que sean eficientes. Con este fin, se investigan nuevas técnicas que soporten la interacción con el usuario, nuevas estructuras de datos (índices) capaces de manipular eficientemente datos multimedia y que permitan manejar bases de datos masivas de este tipo de datos y se desarrollan nuevas aplicaciones que soporten la recolección y el procesamiento de grandes volúmenes de datos no estructurados.

Palabras Claves: *recuperación de información, computación de alto desempeño, grandes bases de datos.*

1. Contexto

Esta línea de investigación se encuentra enmarcada dentro del Proyecto Consolidado 3-30114 de la Universidad Nacional de San Luis (UNSL) y en el Programa de Incentivos (código 22/F434): "Tecnologías Avanzadas Aplicadas al Procesamiento de Datos Masivos", dentro de la línea "Recuperación de Datos e Información", desarrollada en el Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la UNSL.

En este contexto, se pretende aportar a la incorporación de información no estructurada en los procesos de toma de decisiones y resolución de problemas, no considerados en los enfoques clásicos. Por lo tanto, nuestra línea se dedica, principalmente, al diseño de índices eficientes que sirvan de apoyo a sistemas de recuperación de información orientados a conjuntos masivos de datos multimedia. Se espera así contribuir a estos sistemas obteniendo índices más eficientes para memorias jerárquicas, dinámicos, con E/S eficiente, escalables (capaces de manejar grandes volúmenes de datos), considerando técnicas de computación de alto desempeño (HPC).

2. Introducción y Motivación

Con el uso masivo de internet, estamos en presencia de un fenómeno donde la aceleración tanto del crecimiento del volumen de datos capturados y almacenados, como la creciente variación en los tipos de datos requeridos, hace que las técnicas tradicionales para el procesamiento, análisis y obtención de información útil deban ser redefinidas para formular nuevas metodologías de abordaje.

Los sistemas de computación tradicionales hacen uso intensivo de información estructurada; es decir, datos generados con un formato específico. En estos casos, la estructura o formato de esta información puede ser fácilmente interpretada y directamente utilizada por un programa de computadora. Pero el hecho de restringirse al uso de este tipo de información conduce, muchas veces, a representar una visión parcial del problema y dejar fuera información que podría ser relevante para la resolución efectiva del mismo. En este contexto gran parte de la infor-

mación que se requiere para la toma de decisiones y la resolución de problemas de índole más general proviene de información no estructurada.

Habitualmente, se utilizan diferentes métodos de acceso o índices [5] para responder eficientemente a consultas para recuperación de información sobre bases de datos multimedia, principalmente por la gran cantidad de datos con los que se trabaja. Los índices pueden tener distintas características que los hacen indicados para aplicaciones reales: eficientes, dinámicos, escalables, resistentes a la *maldición de la dimensión*, entre otras. Un enfoque prometedor para sistemas de recuperación usando búsqueda por similitud es "la búsqueda basada en contenidos", la cual usa el dato multimedia mismo. Para calcular la similitud entre dos objetos multimedia, se debe definir una función de distancia. Dicha función mide la disimilitud entre dos objetos.

El concepto de espacios métricos da un marco formal, independiente del dominio de la aplicación, para definir el concepto de búsqueda por similitud. Un espacio métrico está compuesto por un *universo* $\mathcal{U}$ de objetos y una función de distancia $d : \mathcal{U} \times \mathcal{U} \longrightarrow \mathbb{R}^+$, que satisface las propiedades que la hacen una métrica. Las consultas por similitud, sobre una *base de datos* $\mathcal{S} \subseteq \mathcal{U}$, son básicamente de dos tipos: *Búsqueda por rango* y *Búsqueda de los* k *vecinos más cercanos*. La función de similitud (distancia) mide el mínimo esfuerzo (costo) necesario para transformar un objeto en otro. Dependiendo de los tipos de datos multimedia reales el cálculo de dicha función puede ser muy costoso. En particular, para ahorrar cálculos de distancia es importante que dicha distancia satisfaga la desigualdad triangular.

Si la base de datos $\mathcal{S}$ posee n objetos, las consultas se pueden responder llevando a cabo n evaluaciones de distancia. Sin embargo, en la mayoría de las aplicaciones, las distancias son costosas de computar (por ej.: comparación de huellas digitales). En conjuntos masivos de datos la búsqueda secuencial es impráctica y, en general, los repositorios de datos multimedia son grandes volúmenes de datos. Para responder a las consultas con la menor cantidad de cálculos de distancia se debe preprocesar la base de datos para construir un índice. En algunos casos, es probable que la base de datos, el índice, o ambos, no puedan almacenarse en memoria principal. Por lo tanto, para lograr eficiencia, se debe minimizar el número de operaciones de E/S, considerar la jerarquía de memorias y utilizar técnicas paralelas.

Esta propuesta se enfoca en obtener herramientas de recuperación de información, desarrollando nuevas técnicas y aplicaciones que soporten la interacción con el usuario, diseñando estructuras de datos (índices), capaces de manipular eficientemente grandes volúmenes de datos no estructurados y facilitando la realización de diferentes consultas, de modo de acercar las bases de datos multimedia al nivel de desarrollo de las bases de datos tradicionales.

3. Líneas de Investigación

Se pretende investigar sobre distintos aspectos de los sistemas de recuperación de información multimedia: el diseño de nuevos índices, representaciones que reflejen características de interés de los objetos, distintas consultas sobre estos tipos de bases de datos y eficiencia al considerar grandes volúmenes de datos.

Diseño de Índices

Existen muchos índices para espacios métricos [5]. En su gran mayoría usan la desigualdad triangular para evitar el análisis secuencial de la base de datos. Ésta es la propiedad que permite estimar la distancia entre la consulta q y los objetos de la base de datos, si se han calculado de antemano algunas distancias a objetos distinguidos, y evita calcular las distancias reales desde q a algunos objetos durante una búsqueda. Las distintas técnicas difieren en si esos objetos distinguidos son *pivotes* o *centros*. Si son pivotes se almacenan las distancias de todos los objetos de la base de datos a ellos. Si son centros se particiona el espacio en zonas denominadas *particiones compactas*, por cercanía a los centros y se almacena un radio de cobertura para determinar la zona de cada centro.

Si los objetos de la base de datos se conocen de antemano y luego de construir el índice se realizarán las consultas, los índices se denominan *estáticos*. Por el contrario, si los objetos no se conocen de antemano y el índice se irá creando, preferentemente de manera incremental, a medida que arriben los elementos, permitiendo consultas en cualquier momento, entonces los índices se denominan *dinámicos*. Las estructuras estáticas se benefician al conocer la base de datos completa, ya que pueden seleccionar los mejores puntos de referencia para una estructura de datos determinada. Por el contrario, en las estructuras de datos dinámicas esto no es posible, porque tanto los objetos como las consultas arriban al azar.

Otro aspecto importante para buscar una solución es saber si se puede trabajar en memoria principal

o, por el contrario, si por ser conjuntos de datos masivos se deberá trabajar en otros niveles de la jerarquía de memorias. Además, en este último caso, una manera de lograr eficiencia en las operaciones sobre los índices es aplicando técnicas de computación de alto desempeño y en otros casos mediante la adaptación o diseño de las estructuras que sean concientes de la jerarquía de memorias, minimizando no sólo la cantidad de cálculos de distancia, sino también el número de operaciones de E/S. Otra manera de acelerar la respuesta a una consulta es admitir una respuesta aproximada, permitiendo que la misma sea de menor calidad o menos exacta, pero muy rápida.

Por un lado, nos interesa mejorar el desempeño de índices dinámicos jerárquicos (árboles) para espacios métricos, los cuales se construyen incrementalmente vía inserciones. La raíz del árbol es el primer objeto que llega, y esto se repite recursivamente en cada nivel del árbol. El *Árbol de Aproximación Espacial Distante DiSAT* [4] es un índice muy eficiente en cuanto al número de cálculos de distancias realizados tanto en construcción como en búsquedas. La desventaja del *DiSAT* es que no admite inserciones ni eliminaciones y no es posible construirlo incrementalmente. Sin embargo, una opción para transformarlo en una estructura dinámica es mediante la aplicación de la técnica de Bentley y Saxe [1] que permite lograr dinamismo a partir de una estructura estática, cuando la búsqueda sobre ella cumple con ser un problema que se puede descomponer en partes independientes. Por lo tanto, se está desarrollando un *DiSAT* dinámico usando esta técnica, con un algoritmo eficiente de búsqueda de k-vecinos más cercanos, a pesar de no poder descomponer dicha consulta en búsquedas independientes.

Por otra parte, considerando aplicaciones sobre conjuntos de datos masivos, donde los volúmenes de información con los que se debe trabajar (millones de imágenes en la Web), se hace necesario que los índices sean almacenados en memoria secundaria. En este caso, para hacerlos eficientes, no sólo se debe considerar que en las búsquedas se realice el menor número de cálculos de distancia sino también, dado el costo de las operaciones sobre disco, se efectúe la menor cantidad posible de operaciones de E/S. Hemos diseñado e implementado las siguientes estructuras *DSACL*-tree* y el *DSACL+-tree* [2], optimizadas para memoria secundaria, que demostraron ser competitivas frente a otras de las estructuras conocidas tales como el *M-tree* y *DSA*-tree* y *DSA+-tree* [11]. También, existe una nueva propuesta en evaluación (una nueva versión del *DSATCL-tree*) que promete ser aún más eficiente. Además, es posible mejorar su desempeño mediante la aplicación de técnicas paralelas. Por lo tanto, se buscará aplicar y comparar distintas estrategias de paralelización con el fin de determinar la más adecuada.

Por otro lado, tomando como base al índice para búsquedas aproximadas *Lista de Permutaciones Agrupadas* (LPA), que combina un algoritmo basado en *Permutaciones* con una *Lista de Clusters* [6], se ha propuesto una nueva versión de la *LPA* que permite realizar búsquedas por similitud aproximadas sobre conjuntos de datos masivos [13]. Esta nueva versión de la *LPA* es conciente que trabaja en memoria secundaria y no sólo considera minimizar los costos en cantidad de distancias calculadas, sino también en cantidad y tipo de operaciones de E/S.

Consultas sobre Bases de Datos Multimedia

Las operaciones tradicionales sobre bases de datos multimedia son las búsquedas por rango o de k-vecinos más cercanos. Sin embargo, existen otras operaciones de interés como las distintas variantes del *join* por similitud. Para estas operaciones se consideran dos bases de datos A y B, ambas subconjuntos del mismo universo del espacio métrico $\mathcal{U}$. El resultado de cualquier operación de join por similitud entre A y B obtiene el conjunto de pares formados por un objeto de A y otro de B, tales que entre ellos se satisface el predicado de similitud Φ considerado. Las variantes más conocidas son: el join por rango, el join de k-vecinos más cercanos y el join de k pares de vecinos más cercanos; entre otras.

Formalmente, dadas $A, B \subseteq \mathcal{U}$, se define el *join por similitud* entre A y B ($A \underset{\Phi}{\bowtie} B$) como el conjunto de todos los pares (x, y), donde $x \in A$ e $y \in B$; es decir, $(x, y) \in A \times B$, tal que $\Phi(x, y)$ es verdadero (se satisface el criterio de similitud Φ entre x e y). Al resolver el join por similitud es posible que ambas, una o ninguna de la bases de datos posean un índice; o que ambas bases de datos se indexen conjuntamente con un índice diseñado para el join. Calcular cualquiera de las variantes del join por similitud de manera exacta es muy costoso [12], así vale la pena analizar posibilidades de obtener una respuesta aproximada al join, más rápidamente, aunque siempre buscando buena calidad en la respuesta.

PostgreSQL es el primer sistema de base de datos que permite realizar consultas por similitud sobre algunos atributos, particularmente indexa para búsquedas de k-vecinos más cercanos (índices *KNN-GiST*). Estos índices pueden ser usados sobre tex-

to, comparación de ubicación geoespacial, etc. Sin embargo, los índices *K-NN GiST* proveen plantillas sólo para índices con estructura de *árbol balanceado* (*B-tree*, *R-tree*), pero el "balance" no siempre es bueno para los índices que se utilizan en búsquedas por similitud [3]. Además, no se dispone de este tipo de consultas para todo tipo de datos métricos. Así, es importante proveer un DBMS para bases de datos métricas que maneje todos los posibles datos métricos y las operaciones de interés sobre ellos [7].

Simulaciones Paralelas Aproximadas para Sistemas de Gran Escala

Los *motores de búsqueda Web* (WSE) son sistemas complejos y sumamente optimizados que operan sobre clusters de procesadores. Los WSE gestionan cargas de trabajo altamente dinámicas. La evaluación del desempeño de estos WSE es de suma importancia para la implementación y funcionamiento de los centros de datos y para labores de investigación.

La simulación secuencial de eventos discretos y la simulación orientada a procesos han demostrado ser útiles en estudios de evaluación de desempeño en modelos de pequeña escala. Sin embargo, para el estudio de modelos de gran escala se utiliza simulación paralela por eventos discretos, ya que permite satisfacer requerimientos críticos, como tiempos de espera de resultados de la simulación o de consumo de memoria. Existen dos enfoques principales para efectuar simulaciones paralelas: el enfoque optimista y el conservativo. Ambos tienen como objetivo garantizar el cumplimiento de la causalidad de eventos. Los protocolos conservativos resultan inviables en simulaciones de grandes WSE, que se caracterizan porque los objetos simulados presentan patrones de comunicación aleatorios en que todos los elementos se comunican con todos, o que pueden incluir implementaciones reales como cachés de resultados, y en que los incrementos de tiempo de los eventos pueden presentar grandes diferencias entre ellos.

Dentro del proyecto de investigación descripto en este trabajo se propone desarrollar y evaluar un esquema optimista de simulación paralela semiasíncrona y aproximado [10, 14], que reduzca los tiempos de ejecución y el consumo de memoria de simulaciones optimistas. El objetivo es que los resultados estimados por simulaciones aproximadas, que violan la restricción de causalidad, presenten alta precisión respecto a resultados de simulaciones secuenciales. El nivel de optimismo se podría gestionar eficientemente a lo largo de la simulación, y se ajustaría automáticamente a características propias de la simulación. El esquema a abordar utiliza un algoritmo para balancear la carga de los procesadores que ejecutan la simulación paralela.

Por lo tanto, se busca: (a) proponer y evaluar un nuevo enfoque de simulación paralela aproximado para sistemas de búsqueda Web de gran escala y (b) utilizar el framework PCD++ [8] basado en DEVS para diseñar y desarrollar el nuevo enfoque.

Sistema CBIR con Plataforma de Base FPGA

Con el crecimiento constante de los datos que se desarrollan en diferentes aplicaciones, pero con mayor dimensión en la Web, estas nuevas aplicaciones exigen el uso de datos más complejos que sólo texto sin formato. En estos casos, los espacios métricos han demostrado ser útiles y prácticos para realizar una búsqueda de similitud en grandes colecciones de objetos complejos. En este caso, las consultas son objetos del mismo tipo de los almacenados en la base de datos cuando, por ejemplo, uno está interesado en la recuperación de los k objetos más similares a una consulta dada. La similitud entre dos objetos se calcula mediante una función de distancia dependiente de la aplicación, que generalmente suele ser costosa para calcular en términos computacionales.

Las aplicaciones de recuperación de imágenes basada en contenido visual también se conocen como sistemas de recuperación de imágenes basada contenido (CBIR). Los sistemas CBIR se pueden modelar con espacios métricos e implican básicamente dos operaciones: (a) cálculos de descriptores visuales que caracterizan y describen el contenido de una imagen (los vectores característicos o descriptor); y (b) la medida de similitud para seleccionar imágenes candidatas de la base de datos similares a una consulta (para el proceso de búsqueda).

Dentro de este proyecto se propone implementar un sistema CBIR sobre una plataforma SoC basado en FPGA para acelerar la búsqueda de similitud mediante el aprovechamiento de los beneficios de las herramientas de síntesis de alto nivel. El objetivo es reducir el tiempo de ejecución requerido para construir el vector de descriptor de una imagen, cuyo proceso representa uno de los cuellos de botella en los sistemas de CBIR [15]. Así, se tiene por objetivo: (1) proponer, diseñar y evaluar un sistema CBIR sobre una plataforma SoC basada en FPGAs y (2) tomar ventaja de las propiedades de área, consumo de energía y velocidad de procesamiento provista por las nuevas plataformas SoCs.

4. Resultados

Se implementaron el *DSACL*-tree* y *DSACL+-tree*, que trabajan con grandes volúmenes de datos, diseñadas para memoria secundaria y que mostraron ser competitivas contra otras estructuras diseñadas para tal fin [2]. Se espera lograr una implementación paralela eficiente de estos índices. Se ha obtenido una versión para memoria secundaria de la *LPA* [13]. Se ha completado el análisis del *DiSAT* [4] y se espera proponer una versión dinámica del mismo, que mantenga su eficiencia. Se está trabajando en la extensión de *PostgreSQL* que incluya más consultas por similitud y sobre distintos tipos de datos.

Habiendo previamente desarrollado un conjunto de aplicaciones de comparación para determinar las operaciones más relevantes y costosas de la plataforma S4 [9], se ha propuesto y evaluado un simulador paralelo asíncrono, diseñado para procesamiento de "streams" distribuidos sobre dicha plataforma [14].

5. Formación de Recursos

En esta línea se están realizando los siguientes trabajos de formación en Ciencias de la Computación:
Tesis de Doctorado: (1) "Planeación de Capacidad en Centros de Datos para Sistemas Escalables para la Web" (con beca de posgrado de CONICET), (2) "Planeación de Capacidad para Motores de Búsqueda Web" y (3) "Optimización Dinámica de Funcionamiento de Motor de Búsqueda con Máquinas de Aprendizaje y Transformada Discreta de Fourier".
Tesis de Maestría: (1) "Estructuras Eficientes sobre Datos Masivos para Búsquedas en Espacios Métricos", (2) "Simulación Paralela Aproximada sobre S4 para Motores de Búsqueda en la Web", (3) "Recuperación de Imágenes sobre Plataformas de Sistemas de Cómputo de Alta Productividad" y (4) un "Sistema Administrador para Bases de Datos Métricas".
Trabajo Final de Licenciatura: "Aplicación de Multi-BSP para la Estimación de Costo de Algoritmos sobre Plataformas Multi-core".

Referencias

[1] J. Bentley and J. Saxe. Decomposable searching problems: Static-to-dynamic transformation. *J. Algorithms*, 1(4):301–358, 1980.

[2] L. Britos, A. M. Printista, and N. Reyes. Dynamic spatial approximation trees with clusters for secondary memory. In *XVI CACIC Selected Papers*, Computer Science & Technology Series, 205–215. Editorial UNLP, 2011.

[3] E. Chávez, V. Ludueña, and N. Reyes. Revisiting the VP-forest: Unbalance to improve the performance. In *Procs. of JCC*, page 26, 2008.

[4] E. Chávez, V. Ludueña, N. Reyes, and P. Roggero. Faster proximity searching with the distal {SAT}. *Information Systems*, pages–, 2016. In Press, Available online.

[5] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM*, 33(3):273–321, Sept. 2001.

[6] K. Figueroa and R. Paredes. List of clustered permutations for proximity searching. In *Procs. of the 6th SISAP.*, vol. 8199, *LNCS*, 50–58. Springer, 2013.

[7] F. Kasián and N. Reyes. Búsquedas por similitud en PostgreSQL. In *Actas del XVIII CACiC*, 1098–1107, 2012.

[8] Q. Liu and G. Wainer. Lightweight time warp - a novel protocol for parallel optimistic simulation of large-scale devs and cell-devs models. In *12th IEEE/ACM Int. Symp. on DS-RT.* , 131–138, Oct. 2008.

[9] J. Lobos, V. Gil-Costa, and M. Marin. Benchmark applications for stream processing profiling. In *Procs. of JCC*, Talca, Chile, 2014.

[10] M. Marin, V. Gil-Costa, C. Bonacic, and R. Solar. Approximate parallel simulation of web search engines. In *Procs. of 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 189–200, 2013. ACM.

[11] G. Navarro and N. Reyes. Dynamic spatial approximation trees for massive data. *Procs. of 2nd SISAP*, 81–88. IEEE Comp. Soc., 2009.

[12] R. Paredes and N. Reyes. Solving similarity joins and range queries in metric spaces with the list of twin clusters. *JDA*, 7:18–35, 2009.

[13] P. Roggero, N. Reyes, K. Figueroa, and R. Paredes. List of clustered permutations in secondary memory for proximity searching. *JCS&T*, 15(2):107 – 113, Nov. 2015.

[14] E. Tapia, V. Gil-Costa, and M. Marin. Evaluation of a parallel simulation algorithm for the s4 stream processing platform. In *Procs. of JCC*, Santiago, Chile, 2015.

[15] Y. Zhang. *Advances in Face Image Analysis: Techniques and Technologies*. IGI Global, USA, 1st edition, 2010.