

Procesamiento y Recuperación en Bases de Datos Masivas

Luis Britos, Veronica Gil-Costa, Fernando Kasián, 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, gvcosta, fkasian, vlud, mprinti, nreyes, proggero}@unsl.edu.ar
romy00@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 es cada vez más evidente la necesidad de procesar conjuntos de datos, de manera tal de poder obtener información útil a partir de ellos. 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.

En esta línea de investigación el principal objetivo es desarrollar herramientas eficientes para sistemas de información sobre bases de datos masivas, conteniendo datos multimedia. Por lo tanto, se analizan nuevas técnicas que permitan una buena interacción con el usuario, nuevas estructuras de datos (índices) capaces de manipular eficientemente datos multimedia y que puedan utilizarse para administrar bases de datos multimedia masivas. Además, se busca desarrollar herramientas que permitan soportar tanto la recolección como el procesamiento de grandes volúmenes de datos multimedia.

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

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 contribuir 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, el objetivo principal es diseñar e implementar índices eficientes para conjuntos masivos de datos multimedia, que puedan servir de apoyo a sistemas de recuperación de información sobre distintos tipos de datos no estructurados. Para ello, se debe considerar que son necesarios índices más eficientes para memorias jerárquicas, dinámicos, con E/S eficiente, escalables (capaces de manejar grandes volúmenes de datos), considerando además 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 vi-

sión parcial del problema y dejar fuera información que podría ser relevante para la resolución efectiva del mismo. En este contexto gran parte de la información 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 [2] 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 U de objetos y una función de distancia $d: U \times U \rightarrow \mathbb{R}^+$, que satisface las propiedades que la hacen una métrica. Las consultas por similitud sobre una base de datos S 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 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 minimi-

zar el número de operaciones de E/S, considerar la jerarquía de memorias, en algunos casos admitir respuestas no exactas 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 sobre grandes volúmenes de datos: 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.

Procesamiento y Modelado de Grandes Volúmenes de Datos

Con el continuo crecimiento de la Web y las redes sociales, es cada vez mayor la cantidad de datos que se generan. Además de estos datos de aspecto social, también están los datos científicos generados por diferentes disciplinas como la biología, la física, la astronomía y la medicina. Estos datos provenientes de diferentes fuentes generalmente deben ser procesadas y analizadas en el menor tiempo posible.

Para abordar esta problemática, desde la UNSL, se han realizado investigaciones que analizan la posibilidad de utilizar arquitecturas heterogéneas basadas en dispositivos programables (FPGAs - Field Programmable Gate Array) y en sistemas sobre chip (SoCs -System-on-a-chip) para representar datos multimedia y procesar búsquedas de elementos complejos [5]. Estas plataformas tienen la característica de ser re-configurables y se pueden adaptar a medida para resolver un problema específico. De esta manera, logran reducir los tiempos de procesamiento. Adicionalmente, reportan un menor consumo de energía que las unidades de procesamiento gráfico (GPUs).

Por otro lado, el procesamiento masivo de datos debe ser acompañado de un modelo de computación que permita organizar las comunicaciones y los cómputos. En la literatura existen varios mode-

los de computación que intentan generar un puente entre la arquitectura y las aplicaciones. Sin embargo, muchos de estos modelos fallan debido a que su complejidad crece con el número de parámetros que utilizan para caracterizar las arquitecturas. En este sentido, el modelo BSP (Bulk Synchronous Parallel) [12] ha sido ampliamente aceptado y utilizado por gran parte de la comunidad científica, debido a que es un modelo simple que permite abstraer los costos de las arquitecturas. Este modelo BSP, originalmente fue propuesto para clusters de computadoras conectadas mediante una red de comunicación. En el año 2011, se presentó el modelo Multi-BSP [13] para arquitecturas multi-core.

Recientemente hemos utilizado estos modelos (BSP y Multi-BSP) para modelar algoritmos MapReduce. En particular, el objetivo es identificar los componentes que más costo tienen y el efecto de las sincronizaciones impuestas por el modelo [11]. Actualmente, se está estudiando el efecto que tienen las barreras de sincronización en arquitecturas multi-core, para evaluar la jerarquía más adecuada en que las sincronizaciones de componentes atómicos reducen los costos adicionales impuestos por el modelo de computación.

Esta línea de investigación involucra a dos tesis de doctorado, una beca de CONICET y dos tesis de maestría (una finalizada).

Índices para Bases de Datos Multimedia

Los índices que resultan apropiados, para luego realizar búsquedas sobre bases de datos multimedia, son los índices métricos [2]. Todos ellos aprovechan que la distancia satisface la propiedad de desigualdad triangular para ahorrar algunos cálculos de distancia. Esta propiedad permite estimar la distancia entre cualquier objeto de consulta q y los objetos de la base de datos, si se mantienen algunas distancias de los elementos de la base de datos a objetos distinguidos. Los distintos enfoques se diferencian 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.

En nuestro caso, nos interesa poder diseñar buenos índices que consideren:

Dinamismo: Los índices pueden construirse de manera estática, si los objetos de la base de datos

se conocen de antemano. En estos índices denominados *estáticos* las búsquedas se realizan luego de construido el índice. Por el contrario, si no se pueden tener los objetos de antemano y la única manera de construir el índice es a medida que se incorporan los elementos a la base de datos; es decir, de manera incremental, se considera que las búsquedas pueden realizarse en cualquier momento. Esta clase de índices se denominan *dinámicos*. Los índices estáticos, por conocer a toda la base de datos, pueden seleccionar los mejores objetos distinguidos para una estructura de datos determinada. En cambio, en los índices dinámicos esto no es posible.

Jerarquía de Memorias: 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. En caso que el índice deba alojarse en memoria secundaria, se deben minimizar la cantidad de cálculos de distancia y también el número de operaciones de E/S.

Computación de Alto Desempeño: En algunos casos, si no se logra la eficiencia deseada mediante la optimización del índice en sí mismo, se pueden aplicar técnicas de computación de alto desempeño con el fin de acelerar los tiempos de respuesta a las consultas.

Exactitud de la Respuesta: Otra manera de acelerar la respuesta a una consulta por similitud es admitir una respuesta aproximada, permitiendo que la misma sea de menor calidad o menos exacta, pero muy rápida.

Dimensionalidad Intrínseca: los índices para búsquedas por similitud, al trabajar sobre el modelo de espacios métricos, pueden también sufrir de la llamada *maldición de la dimensión*; es decir, los índices se degradan a medida que la dimensión de los espacios aumenta. Existen índices que se comportan mejor en espacios difíciles (dimensión intrínseca mediana a alta) y otros que son adecuados para espacios fáciles (dimensión intrínseca baja).

Considerando que nos interesa trabajar sobre conjuntos de datos masivos, los volúmenes de información con los que se debe trabajar (millones de imágenes en la Web) hace necesario que los índices sean almacenados en memoria secundaria. En este caso, para lograr eficiencia, 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. Por ello, en esta línea nos hemos dedicado a diseñar índices especialmente adaptados para trabajar en memoria secundaria, que tengan buen desempeño principalmente en las búsquedas. Así, se ha diseñado e implementado una versión paralela del *Conjunto Dinámico de Clusters* (DSC) [8]. Este índice, basado en la *Lista de Clusters* (LC), está especialmente diseñado para

memoria secundaria y es completamente dinámico, admite inserciones y eliminaciones y tiene un buen desempeño en las búsquedas, principalmente en la cantidad de operaciones de E/S. DSC ha demostrado ser muy competitivo frente a otras de las buenas estructuras del estado del arte. 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 LC [4], se ha propuesto una nueva versión dinámica de la LPA que permite realizar búsquedas por similitud aproximadas sobre conjuntos de datos masivos [3]. Esta nueva versión de la LPA es consciente 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.

Existen en la actualidad pocas medidas que permitan reflejar adecuadamente la dimensionalidad intrínseca de los espacios métricos [2]. Sin embargo, si se pudiera calcular la dimensionalidad intrínseca de un espacio métrico con cierta confiabilidad, se podría elegir el índice que tuviera mejor desempeño en las búsquedas para esa dimensión en particular. Por lo tanto, se han propuesto nuevas medidas de evaluación de la dimensionalidad intrínseca y se las ha evaluado experimentalmente junto a otras medidas ya conocidas, para ver cuál de ellas puede reflejar de manera más confiable la dimensionalidad intrínseca de un espacio métrico [7].

En esta línea de investigación se están desarrollando dos tesis de maestría.

Sistema Administrador para Bases de Datos Multimedia

A pesar de que las operaciones más comunes sobre bases de datos multimedia son las búsquedas por rango o de k -vecinos más cercanos, existen otras operaciones de interés tales como las distintas variantes del *join* por similitud. La operación de *join* por similitud se considera una de las operaciones

que debería brindar típicamente un sistema administrador para bases de datos multimedia.

Existen diferentes variantes para el *join* por similitud, dependiendo del criterio de similitud Φ utilizado, pero ellas tienen en común que se aplican entre dos bases de datos A y B , ambos subconjuntos del mismo universo del espacio métrico U que modela a la base de datos multimedia. El resultado de cualquiera de las variantes del *join* por similitud entre A y B obtendrá el conjunto de pares formados por un objeto de A y otro de B , tales que entre ellos se satisface el criterio 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 $A, B \subseteq U$, se define el *join por similitud* entre A y B ($A \star_{\Phi} B$) 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 las 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 [9], así como 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 texto, 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 [1]. 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 [6].

Más aún, dado que las respuestas a consultas de *join* suelen ser conjuntos muy grandes de pares de objetos y muchos de esos pares son muy similares entre sí, se planea introducir sobre las operaciones de *join* la posibilidad de diversificar las respuestas [10]; es decir, un operador de *join* por similitud que asegure un conjunto más pequeño, más diversifica-

do de respuestas útiles y, de ser posible, más rápido de obtener. Estos desarrollos, entre otros, permitirán tener un DBMS con mayores posibilidades de aplicación en sistemas de información reales.

Esta línea corresponde a una tesis de maestría.

4. Resultados

Se ha propuesto, diseñado y evaluado un sistema CBIR sobre una plataforma SoC basada en FPGAs [5]. Se ha implementado la versión paralela del índice DSC, que trabaja con grandes volúmenes de datos, diseñado especialmente para memoria secundaria, que admite inserciones y eliminaciones de elementos y que permitirá responder eficientemente a lotes de consultas por similitud. Se ha obtenido una versión para memoria secundaria y dinámica de la LPA [3]. Se continúa trabajando en la extensión de PostgreSQL para que brinde facilidades de soporte a más tipos de consultas por similitud, sobre distintos tipos de datos y que considere opciones de respuesta aproximada, como así también la posibilidad de obtener una respuesta diversificada en el caso de los joins por similitud.

5. Formación de Recursos

En esta línea se están realizando las siguientes tesis de posgrado en Ciencias de la Computación:

Tesis de Doctorado: (1) “Simulación Semi-asíncrona DEVs de Motores de Búsqueda Web” y (2) “Predicción de Tiempo de Ejecución de Consultas para Motores de Búsqueda Web”.

Tesis de Maestría: (1) “Estructuras Eficientes sobre Datos Masivos para Búsquedas en Espacios Métricos”, (2) “Cómputo Aproximado del Grafo de Todos los k -Vecinos”, (3) “Simulación Paralela Aproximada sobre S4 para Motores de Búsqueda en la Web”, (4) “Recuperación de Imágenes sobre Plataformas de Sistemas de Cómputo de Alta Productividad” y (5) “Sistema Administrador para Bases de Datos Métricas”.

Referencias

- [1] E. Chávez, V. Ludueña, and N. Reyes. Revisiting the VP-forest: Unbalance to improve the performance. In *Proc. de las JCC08*, page 26, 2008.
- [2] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM*, 33(3):273–321, Sept. 2001.
- [3] K. Figueroa, C. Martínez, R. Paredes, N. Reyes, and P. Roggero. Dynamic list of clustered permutations on disk. *Computer Science and Technology*, 201–211, 2016.
- [4] K. Figueroa and R. Paredes. List of clustered permutations for proximity searching. In *Similarity Search and Applications*, volume 8199 of LNCS, 50–58. Springer Berlin Heidelberg, 2013.
- [5] V. Gil-Costa, R. Molina, R. Petrino, C. Soza Paez, A. Printista, and J. Dondo Gazzano. *Hardware Acceleration of CBIR System with FPGA-Based Platform*, 138–170. Advances in Computer and Electrical Engineering. IGI Global, 2016.
- [6] F. Kasián and N. Reyes. Búsquedas por similitud en PostgreSQL. In *Actas del XVIII CACIC*, 1098–1107, Bahía Blanca, Argentina, Oct. 2012.
- [7] G. Navarro, R. Paredes, N. Reyes, and C. Bustos. An empirical evaluation of intrinsic dimension estimators. *Information Systems*, 64:206–218, 2017.
- [8] G. Navarro and N. Reyes. New dynamic metric indices for secondary memory. *Information Systems*, 59:48–78, 2016.
- [9] R. Paredes and N. Reyes. Solving similarity joins and range queries in metric spaces with the list of twin clusters. *JDA*, 7:18–35, March 2009. doi:10.1016/j.jda.2008.09.012.
- [10] L. F. D. Santos, L. Olmes Carvalho, W. D. Oliveira, A. J.M. Traina, and C. Jr. Traina. Diversity in similarity joins. In *Similarity Search and Applications*, volume 9371 of LNCS, 42–53. Springer International Pub., 2015.
- [11] H. Senger, V. Gil-Costa, L. Arantes, C. A. C. Marcondes, M. Marín, L. M. Sato, and F. A.B. da Silva. Bsp cost and scalability analysis for mapreduce operations. *Concurrency and Computation: Practice and Experience*, 28(8):2503–2527, 2016. cpe.3628.
- [12] Leslie G. Valiant. A bridging model for parallel computation. *Commun. ACM*, 33(8):103–111, Aug. 1990.
- [13] Leslie G. Valiant. A bridging model for multi-core computing. *J. Comput. Syst. Sci.*, 77(1):154–166, January 2011.