Recuperación de Datos e Información en Bases de Datos Masivas

Luis Britos, Fernando Kasián, Verónica Ludueña, Franco Merenda,
Marcela Printista, Nora Reyes, Patricia Roggero
LIDIC, Dpto. de Informática, Fac. de Cs. Físico Matemáticas y Naturales, Universidad Nacional de San Luis
{*lebritos, fkasian, vlud, mprinti, nreyes,proggero*}@unsl.edu.ar, merenda.franco83@gmail.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 grandes conjuntos de datos, de manera tal de poder obtener información útil a partir de ellos. Sin embargo, la evolución de las tecnologías de información y comunicación, en conjunto con la gran cantidad y variedad de información disponible digitalmente, han llevado en las últimas décadas al surgimiento de nuevos depósitos no estructurados de información, en los cuales los datos que no se adaptan fácilmente al modelo relacional. A tipos de datos tales como texto libre, imágenes, audio, video, secuencias biológicas de ADN o proteínas, entre otros; no se los puede estructurar más en claves y registros, o tal estructuración es muy dificultosa (tanto manual como computacionalmente), y restringe de antemano los tipos de consultas que luego se pueden realizar. Como muchas aplicaciones computacionales necesitan recuperar datos e información desde estas grandes bases de datos conteniendo datos no estructurados, es necesario lograr eficiencia en formas más sofisticadas de búsqueda que la habitual sobre datos estucturados. Así, 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.

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

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. Actualmente se encuentra en proceso de evaluación la nueva presentación.

En esta línea de investigación se busca desarrollar herramientas eficientes para sistemas de información sobre bases de datos masivas, conteniendo datos no estructurados. 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 no estructurados y que puedan utilizarse para administrar bases de datos masivas que contienen datos no estructurados. Así, 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 de esta línea es el diseño y desarrollo de índices que sirvan de apoyo a sistemas de recuperación dedicados a conjuntos de datos no estructurados masivos tales como: datos multimedia, texto, secuencias de ADN, etc. , permitiendo que estos sistemas cuenten con estructuras de datos eficientes y escalables, para memorias jerárquicas, que hagan uso, de ser necesario, de técnicas de computación de alto desempeño (HPC).

2. Introducción y Motivación

En la actualidad, gracias al uso masivo de internet, se ha producido una significativa aceleración tanto en el crecimiento del volumen de datos capturados y almacenados, como en la creciente variación en los tipos de datos que aparecen. En este contexto, se hace necesario que las técnicas tradicionales para el

procesamiento, análisis y obtención de información útil deban ser redefinidas para formular nuevas metodologías de abordaje.

En general, los sistemas tradicionales de computación utilizan principalmente información estructurada, la cual puede organizarse en claves y registros, sobre los cuales tiene sentido aplicar búsquedas tradicionales y donde su estructura puede interpretarse y utilizarse en programas casi directamente. Pero, por el volumen y variedad de los datos disponibles actualmente, dos de las características de los datos en el ámbito de problemas de "big data", no es posible restringirse a búsquedas sobre datos estructurados, porque obligaría a representar una visión parcial del problema, dejando fuera información que podría ser relevante para la resolución efectiva del mismo. Por lo tanto, en la era de "big data" es necesario administrar eficientemente información no estructurada y considerar tipos de búsqueda más generales que puedan servir de apoyo, por ejemplo, en la toma de decisiones. Uno de estos tipos de búsqueda más generales son las búsquedas por similitud, las cuales se suelen sustentar sobre métodos de acceso o índices métricos [3], que permiten responderlas más eficientemente.

Así, dada la gran cantidad de datos con los que se trabaja, ante consultas de recuperación de información sobre bases de datos conteniendo datos no estructurados, se pueden utilizar estos índices para lograr eficiencia en la respuesta. Dichos 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 útil para sistemas de recuperación usando búsqueda por similitud es "la búsqueda basada en contenidos", la cual usa el dato no estructurado en sí mismo para describir lo que se busca. Para calcular la similitud entre dos objetos, se debe definir una función de distancia que permita describir realmente la disimilitud entre ellos.

El modelo habitual para las búsquedas por similitud es el de espacios métricos; dado que, además de brindar un marco formal, es independiente del dominio de la aplicación. Un espacio métrico se compone de un universo $\mathcal{U}$ de objetos y una función de distancia $d : \mathcal{U} \times \mathcal{U} \longrightarrow \mathbb{R}^+$, la cual cumple con las propiedades de una métrica. Sobre una *base de datos* $\mathcal{S} \subseteq \mathcal{U}$, se pueden considerar dos tipos de búsqueda por similitud: la *búsqueda por rango* y la *búsqueda de los k vecinos más cercanos*. La función de distancia permite medir el mínimo esfuerzo (costo) necesario que se debe realizar para transformar un objeto en otro. Dependiendo de los tipos de datos no estructurados, el cálculo de la distancia puede ser muy costoso. por lo tanto, se busca ahorrar cálculos de distancia y ello se logra generalmente gracias a que la funci'on de distancia cumple con la desigualdad triangular.

Si se considera que la base de datos $\mathcal{S}$ posee n objetos, trivialmente cualquier consulta se puede responder con n evaluaciones de distancia. Sin embargo, en la mayoría de las aplicaciones sobre conjuntos de datos masivos, como las distancias son costosas de computar (por ej.: comparación de huellas digitales), no es factible aplicar la solución trivial. Así, para responder consultas con la menor cantidad de cálculos de distancia posibles 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, en algunos casos admitir respuestas no exactas y cuando sea posible utilizar técnicas paralelas.

Así, en este contexto se considera como objetivo obtener herramientas de recuperación de información, desarrollando nuevas técnicas y aplicaciones que soporten la interacción con el usuario, diseñando índices capaces de manipular eficientemente grandes volúmenes de datos no estructurados y facilitando la realización de diferentes tipos de consultas, de modo de contribuir al desarrollo de aplicaciones reales para problemas de big data.

3. Líneas de Investigación

Como se pretende investigar sobre distintos aspectos de los sistemas de recuperación de información sobre grandes volúmenes de datos no estructurados, se ha considerado el diseño de nuevos índices y la resolución de distintas consultas sobre estos tipos de bases de datos y cómo lograr eficiencia y escalabilidad en las soluciones al considerar grandes volúmenes de datos.

Índices

Los índices que resultan apropiados, para luego realizar búsquedas sobre bases de datos conteniendo datos no estructurados, son los índices métricos [3]. En todos ellos se aprovecha que la función de distancia debe cumplir la propiedad de desigualdad

triangular para ahorrar algunos cálculos de distancia y de esta manera tiempo, gracias a que la desigualdad triangular permite estimar la distancia entre cualquier objeto de consulta q y los objetos de la base de datos, si se mantienen algunas distancias precalculadas enttre los elementos de la base de datos y elementos distinguidos. Los dos enfoques más comunes 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 y si por el contrario 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 enfocamos en 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; 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 (HPC) para 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* [3]; 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).

Como nuestro interés está puesto sobre conjuntos de datos masivos que contienen datos no estructurados, los volúmenes de información con los que se debe trabajar (por ejemplo, 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 las búsquedas realicen el menor número de cálculos de distancia sino también, dado el costo de las operaciones de E/S, se efectúe la menor cantidad posible de operaciones sobre el disco. Por ello, esta línea se dedica a diseñar índices especialmente adaptados para trabajar en memoria secundaria, cuyo desempeño en las búsquedas sea bueno. 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) [2], 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.

El *Árbol de Aproximación Espacial Distal* (DiSAT), basado en el *Árbol de Aproxiación Espacial* [6], es un índice estático que no necesita sintonizar ningún parámetro y es muy eficiente gracias a definir una partiición de hiperplanos con muy buenas características [4]. La raíz elegida para el DiSAT define una partición sobre el espacio, donde las zonas que se obtienen son muy compactas y los hiperplanos que las definen permiten diferenciarlas muy bien. Por ello, se busca aprovechar la información que brindan distintas particiones sobre el espacio para clasificar los elementos de acuerdo a las zonas en las que cada elemento cae en las distintas par-

ticiones consideradas. En este caso, a cada elemento se le asigna una secuencia de bits, denominada "sketch", donde cada bit indica de qué lado del hiperplano considerado se encuentra el elemento. Este conjunto de "sketches" constituye el índice en sí mismo. Cuando se considera una consulta, se calcula el sketch del elemento de consulta q y se lo compara con los sketches de todos los elementos de la base de datos, sin calcular realmente distancias entre objetos sino entre sketches y se revisan luego los objetos más prometedores primeros. En este caso, se espera que un elemento similar a q estará en una partición similar en el espacio. En este caso, se puede limitar de antemano el número de distancias reales que se permiten calcular, logrando una respuesta aproximada a la consulta por similitud con poco costo.

Existen en la actualidad pocas medidas que permitan reflejar adecuadamente la dimensionalidad intrínseca de los espacios métricos [3]. 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].

Por otra parte, se está estudiando cómo aprovechar los índices para búsquedas por similitud sobre bases de datos masivos de datos no estructurados, para solucionar un problema de estacionamiento de vehículos, usando en este caso los índices como herramienta de apoyo en un sistema de recuperación de datos e información.

En esta línea de investigación se están desarrollando dos tesis de maestría y un trabajo final.

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 [10].

Existen diferentes variantes para el join por similitud, dependiendo del criterio de similitud utilizado, pero ellas tienen en común que se aplican entre dos bases de datos A y B, ambas subconjuntos del mismo universo del espacio métrico $\mathcal{U}$ 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 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 (x, y) es verdadero (se satisface el criterio de similitud 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 [9], 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 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 [5].

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 [11]; es decir, un operador de join por similitud que asegure un conjunto más pequeño, más diversificado 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 publicado la evaluación experimental de un conjunto de estimadores de la dimensión intrínseca de un espacio métrico [7], que permitió establecer que los mejores estimadores de dimensión son el exponente de distancia y el estimador basado en correlación.

Actualmente se está evaluando experimentalmente la versión paralela del índice *DSC*, que trabaja con grandes volúmenes de datos, diseñada especialmente para memoria secundaria, que admite inserciones y eliminaciones de elementos y que permitirá responder eficientemente a lotes de consultas por similitud. Además, se encuentra también en proceso de evaluación la propuesta de sketches basados en el *DiSAT*. 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 Maestría en Ciencias de la Computación:

1. "Estructuras Eficientes sobre Datos Masivos para Búsquedas en Espacios Métricos",
2. "Cómputo Aproximado del Grafo de Todos los k-Vecinos",
3. "Sistema Administrador para Bases de Datos Métricas".

Además, está en su etapa inicial el desarrollo de un trabajo final de la Ingeniería en Computación.

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 and G. Navarro. A compact space decomposition for effective metric indexing. *Pattern Recognition Letters*, 26(9):1363–1376, 2005.

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

[4] Edgar Chávez, Verónica Ludueña, Nora Reyes, and Patricia Roggero. Faster proximity searching with the distal {SAT}. *Information Systems*, pages–, 2016. In Press, Available online.

[5] F. Kasián and N. Reyes. Búsquedas por similitud en PostgreSQL. In *Actas del XVIII Congreso Argentino de Ciencias de la Computación (CACiC)*, pages 1098–1107, Bahía Blanca, Argentina, October 2012. Universidad Nacional del Sur.

[6] G. Navarro. Searching in metric spaces by spatial approximation. *VLDBJ*, 11(1):28–46, 2002.

[7] Gonzalo Navarro, Rodrigo Paredes, Nora Reyes, and Cristian Bustos. An empirical evaluation of intrinsic dimension estimators. *Information Systems*, 64:206 – 218, 2017.

[8] Gonzalo Navarro and Nora 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] C. Rong, C. Lin, Y. N. Silva, J. Wang, W. Lu, and X. Du. Fast and scalable distributed set similarity joins for big data analytics. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1059–1070, April 2017.

[11] Lucio F. D. Santos, Luiz Olmes Carvalho, Willian D. Oliveira, Agma J.M. Traina, and Jr. Traina, Caetano. Diversity in similarity joins. In Giuseppe Amato, Richard Connor, Fabrizio Falchi, and Claudio Gennaro, editors, *Similarity Search and Applications*, volume 9371 of *Lecture Notes in Computer Science*, pages 42–53. Springer International Publishing, 2015.