Contribuciones a las Bases de Datos Métricas

J. Arroyuelo, Maria E. Di Genaro, A. Grosso, V. Ludueña, C. Martínez, N. Reyes
Dpto. de Informática, Fac. de Cs. Físico-Matemáticas y Naturales, Universidad Nacional de San Luis
{*bjarroyu, digeme, agrosso, vlud, nreyes*}@unsl.edu.ar, *cintiavmartinez*@hotmail.com
Edgar Chávez
Centro de Investigación Científica y de Educación Superior de Ensenada, México
elchavez@cicese.mx
Karina Figueroa
Fac. de Cs. Físico-Matemáticas, Universidad Michoacana de San Nicolás de Hidalgo, México
karina@fismat.umich.mx
Rodrigo Paredes
Dpto. de Cs. de la Computación, Fac. de Ingeniería, Universidad de Talca, Chile
raparede@utalca.cl

Resumen

Claramente, los nuevos modelos de bases de datos, capaces de contener y manejar todo tipo de datos no estructurados: imágenes, videos, música, secuencias biológicas, etc., no tienen la madurez y versatilidad que presentan las bases de datos convencionales. Estas nuevas bases de datos deben ser capaces de adaptarse al gran volumen de datos digitales, que son generados constantemente por fuentes muy disímiles; al igual que al tipo de requerimientos al que son sometidas, que pueden ser tan dispares como el tipo de datos administrados, debido que éstos pertenecen a campos muy diferentes.

Por esto, se hace necesario optimizar estos depósitos especializados, o desarrollar nuevos, y utilizar formas más sofisticadas de búsqueda sobre los mismos, que permitan enfrentar tales requerimientos. La administración del espacio disponible también se vuelve crucial debido a la gran cantidad de datos que se debe manipular para lograr respuestas adecuadas y eficientes. Esto obliga a los índices utilizados para acceder a este tipo de base de datos, a ser *conscientes de la jerarquía de memoria*.

Esta investigación pretende contribuir a la madurez de este nuevo modelo de bases de datos considerando distintas perspectivas. Para ello utiliza un modelo en el cual se puede utilizar métodos de acceso que contemplen estos aspectos, y que se adapta a tales requerimientos: las *Bases de Datos Métricas*.

Palabras Claves: bases de datos métricas, índices, búsquedas por proximidad.

Contexto

El presente trabajo se realizó en el marco de la línea *Bases de Datos no Convencionales*, del Proyecto Consolidado *Tecnologías Avanzadas de Bases de Datos*, (Cód. 03-2218 y en Programa de Incentivos 22-F814) de la Universidad Nacional de San Luis. En colaboración con investigadores de otros grupos de: Universidad de Talca (Chile), Universidad Michoacana de San Nicolás de Hidalgo y Centro de Investigación Científica y de Educación Superior de Ensenada (México).

La investigación que se realiza en este ámbito, está enfocada en lograr la consolidación de las Bases de Datos Métricas. Se espera contribuir a estos sistemas obteniendo índices que resulten más eficientes para memorias jerárquicas, dinámicos, con E/S eficiente y escalables (capaces de manejar grandes volúmenes de datos). Esto incluye además, plantear nuevas arquitecturas del procesador que mejoren a muy bajo nivel los administradores de estas bases de datos. Se espera contribuir en diferentes campos de aplicación: sistemas de información geográfica, robótica, visión artificial, diseño asistido por computadora, computación móvil, entre otros.

Introducción

El uso masivo de internet y la disponibilidad de dispositivos electrónicos en diversos ámbitos, como el productivo, artístico, laboral, recreativo, científico, de la salud, etc., ha generado una significativa aceleración tanto en el crecimiento del volumen de datos generados y almacenados, como la variedad de tipos de datos que aparecen. Este escenario ha exigi-

do que las bases de datos sean capaces de adaptarse tanto a los diferentes entornos, como a la gran variedad de usuarios de las mismas. Para ello deben administrar eficientemente todo tipo de datos (no estructurados) y responder consultas sobre los mismos de una manera totalmente diferente a la tradicional. Si se necesita encontrar las huellas digitales más similares a una dada, las búsquedas tradicionales (exactas) carecen de sentido. En la mayoría de estos casos, sobre estos tipos de datos, las *búsquedas por similitud* resultan más adecuadas, que las tradicionales.

El modelo habitual para las búsquedas por similitud es el de *espacios métricos*, y a pesar de la variedad de estas aplicaciones, todas comparten ciertas características que permiten la utilización de este modelo como su marco formal. Se define un espacio métrico como un universo de objetos $\mathbb{U}$ y una función de distancia definida entre ellos, $d : \mathbb{U} \times \mathbb{U} \longmapsto \mathbb{R}^+$, que mide cuán diferentes son estos objetos. Resolver este tipo de búsquedas puede ser tan sencillo como realizar una examinación secuencial del conjunto de datos, pero hacerlo eficientemente hace necesario el uso de los llamados *Métodos de Acceso Métricos* (MAMs). Sin embargo, debido a la diversidad de ámbitos en los que se aplica el modelo, es esencial la actualización y optimización de los MAMs, permitiendo su mejor adaptación a cada caso, además de la solución de problemas como la posibilidad de admitir actualizaciones (inserciones/eliminaciones), el soporte de conjuntos masivos de datos y la resolución de búsquedas complejas. Estos avances se reflejan en áreas como: reconocimiento de voz, reconocimiento facial, bases de datos médicas, minería de datos, biología computacional, entre otros.

Otro enfoque analizado es el desempeño de los administradores de bases de datos (DBMS) a bajo nivel. En ese sentido se está intentando caracterizar nuevas arquitecturas que permitan reducir el flujo de bits entre el procesador y la memoria, en relación a la cantidad de datos utilizados por cada programa, para mejorar el desempeño de los mismos.

Líneas de Investigación y Desarrollo

Arquitecturas de Procesadores Orientadas a Bases de Datos

Según algunos autores, se puede distinguir entre arquitectura, implementación y realización. Conforme a esta distinción, el conjunto mínimo de propiedades que determinan qué programas correrán y qué resultados producirán sobre el procesador, es lo que se denomina la arquitectura de una computadora. Es decir, es la interfaz entre el software y el hardware. La implementación está conformada por organización básica del flujo de datos y el control. Por último, la estructura física que comprende la implementación, conforma la realización [1].

Actualmente, la investigación sobre la implementación de procesadores ha reemplazado la investigación sobre arquitecturas de procesadores. La mayoría de los trabajos se ha enfocado en mejorar técnicas de sincronización y comunicación de procesadores (núcleos) a través de mensajes y/o memoria compartida, al igual que técnicas de predicción (tanto de control como de datos). Muchas de estas técnicas, surgidas en los años 60, que se han incorporado a los diseños de nuevos microprocesadores, se pueden aplicar a todo tipo de arquitecturas; tanto a una arquitectura RISC[1] (que intenta acercar el lenguaje de máquina al hardware del procesador), como a una arquitectura que se aleje del hardware e intente disminuir el tráfico de bits entre procesador y memoria. Si bien las arquitecturas RISC compitieron en desempeño con las arquitecturas CISC [2], las mismas poseen un alto tráfico de bits entre el procesador y la memoria para una determinada traza de ejecución. Esto finalmente favoreció a las CISC sobre las RISC, una vez que las CISC mejoraron sus técnicas de implementación.

Con el objetivo el plantear nuevas arquitecturas, que minimicen el tráfico de bits entre el procesador y la memoria, se está construyendo un simulador del set de instrucción AMD-64 o x86-64. Esto permitirá evaluar el tráfico de bits para benchmarks como Specint y Specfp para la arquitectura x86. A continuación, se evaluará el tráfico de bits para la arquitectura propuesta sobre los mismos benchmarcks, lo que implica construir no sólo el simulador de la arquitectura sino también el compilador C para la misma. Finalmente, se pretende aprovechar el conocimiento adquirido para, desde bajo nivel, mejorar el desempeño de los DBMSs.

Bases de Datos Métricas

Como se mencionó, se utilizarán los espacios métricos para modelizar aquellas bases de datos ca-

[1] acrónimo del inglés “Reduced Instruction Set Computer”.
[2] acrónimo del inglés “Complex Instruction Set Computer”.

paces de gestionar imágenes, videos, texto libre, secuencias de ADN o de proteínas, audio, etc. Por lo que se hará uso de MAMs a fin de responder eficientemente consultas por similitud sobre las mismas. Debido a lo costoso que resultan los cálculos de distancia, el número de cálculos realizados al crear el índice o al realizar búsquedas, es usado como medida de complejidad. Por ello, el objetivo aquí es optimizar los MAMs, analizando aquellos que han mostrado buen desempeño en las búsquedas para reducir su complejidad considerando, cuando sea necesario, la jerarquía de memorias. En general, dada una base de datos $X \subseteq \mathbb{U}$ y una consulta $q \in \mathbb{U}$, las consultas por similitud son de dos tipos: por rango o de k-*vecinos más cercanos* (k-NN).

Grafo de los k Vecinos

Entre las consultas por similitud en espacios métricos, una que resulta muy útil es la que obtiene los k-vecinos más cercanos de *todos* los elementos de la base de datos (*All-k-*NN). Esta consulta relaciona cada elemento $u \in X$, con los k objetos en X- $\{u\}$ que tengan la menor distancia a él. La forma ingenua de resolverlo es comparar cada objeto en la base de datos con todos los demás y devolver los k más cercanos a él. Esta solución tiene una complejidad de n^2 cálculos de distancia ($|X| = n$). Una solución más eficiente es preprocesar la base de datos construyendo un índice y luego buscando en el mismo los k-NN de cada elemento del conjunto.

Sin embargo, existen situaciones en las cuales el costo de la construcción del índice, para luego realizar n consultas del tipo k-NN, puede resultar excesivo. Este es el caso de una base de datos masiva, o cuando la función de distancia es demasiado costosa de calcular, o si se está trabajando con espacios métricos de alta dimensión. Estos casos pueden requerir revisar la base de datos completa, a pesar de la estrategia utilizada. Otro factor a considerar son los requerimientos de algunas aplicaciones particulares, que priorizan la velocidad de respuesta sobre la precisión de la misma [13, 7, 14, 8]. Para hacer frente a éstas circunstancias es que se han considerado las llamadas *búsquedas por similitud aproximadas*. Este tipo de consultas mejoran su complejidad aceptando algunos "errores" en la respuesta.

Sabemos que resolver el problema *All-k-*NN permite construir el *Grafo de los k-vecinos más cercanos* (kNNG)[12]. Dada una colección de objetos de un espacio métrico, el grafo de k vecinos más cercanos asocia cada nodo a sus k vecinos más cercanos. El kNNG resulta ser un índice eficiente, que admite mejoras y permite resolver búsquedas por similitud. Por ello hemos propuesto nuevas técnicas para resolver el problema de *All-k-*NN, que *no utiliza ningún índice* para buscar en él, y que permiten computar una aproximación del kNNG. Éstas conectan cada objeto u de la base de datos con k vecinos *cercanos*, relajando la condición que exige que no haya, en toda la base de datos, algún objeto más cercano a u que los k vecinos devueltos. Esto puede ocasionar que se pierda algún objeto muy cercano y en su lugar se devuelva otro un poco más lejano, pero a cambio la respuesta será más rápida. A este grafo se lo denominó *Grafo de vecinos cercanos* (knNG) [5].

Una primera aproximación aprovecha el profundo conocimiento que se tiene del *DiSAT* para plantear un enfoque novedoso. Aquí se consideró un caso particular del problema (k = 1) obteniendo el 1nNG. Esta propuesta utiliza la información obtenida durante la *construcción* del *DiSAT* para construir el 1nNG, conectando a cada elemento con un elemento cercano de la base de datos, que puede ser, o no, su vecino más cercano [5]. Esta propuesta permite recuperar el 1nNG con bajo costo, muy buena precisión y un error bajo, logrando un buen compromiso calidad/tiempo, y *sin realizar ninguna búsqueda.*

Las otras propuestas abordadas se enfocan en responder a los *All-k-*nN y computar el knNG. Estos planteos no utilizan el apoyo de ningún índice, no sólo no buscan en ellos, sino que ni siquiera recurren a la información provista por su construcción. El propósito de estos desarrollos es aprovechar de manera ingeniosa las propiedades de la *función de distancia*. En ellos se sugieren distintas maneras de seleccionar muestras de la base de datos, a partir de las cuales se obtiene un conjunto de distancias que serán el punto de partida de este proceso; analizando diferentes maneras de utilizar la información. En algunos casos se calculan los vecinos exactos [4] y en otros los aproximados, para todos los objetos de la base de datos, utilizando propiedades como la simetría o la desigualdad triangular. Los resultados de estas propuesta se muestran muy prometedores.

Métodos de Acceso Métricos

Muchas veces, debido al tamaño de los objetos almacenados en una base de datos, o su gran cantidad, los índices no caben en memoria principal. Entonces surge la necesidad de diseñar índices que

se almacenan en memoria secundaria. Muchos de estos índices se basan en "agrupar elementos". Teniendo esto en consideración, se han diseñado dos nuevos índices basados en la *Lista de Clusters*(*LC*) [7] que son totalmente dinámicos, es decir, admiten inserciones y eliminaciones de objetos y están especialmente diseñados para trabajar sobre grandes volúmenes de datos [11]. La *Lista de Clusters Dinámica* (*DLC*), tiene buen desempeño en espacios de alta dimensión, una buena ocupación de página y operaciones eficientes tanto en cálculos de distancia como en operaciones de I/O. Sin embargo, durante las búsquedas se debe recorrer completamente la lista de centros de los clusters, elevando los costos. El *Conjunto Dinámico de Clusters* (*DSC*), también mantiene los clusters en memoria secundaria, pero organiza los centros de clusters en un *DSAT* en memoria principal, permitiendo que las búsquedas realicen menos cálculos de distancia y accedan a menos páginas/clusters. Durante las inserciones, también se aprovecha la información de ese *DSAT* también, mejorando los costos en cálculos de distancia y manteniendo los costos de acceso a disco bajos. Ambos, *DLC* y *DSC*, han demostrado tener una razonable utilización de páginas de disco y son competitivas respecto a las alternativas representativas del estado del arte.

La calidad de los clusters generados es otro aspecto a tener en cuenta. El mismo se considera en una variante de la *DSC*, que en lugar de insertar los elementos en el índice a medida que van llegando, demora la incorporación de cada nuevo elemento a un cluster hasta tener varios elementos y poder determinar así un mejor agrupamiento de los objetos. Esto permite reducir el costo de construcción del índice, porque se realiza una escritura de un cluster en disco luego de varias inserciones y además implícitamente puede mejorar los costos de búsqueda al lograr clusters más compactos y que aseguran una total ocupación de la página del disco, achicando el tamaño del archivo y reduciendo los tiempos de acceso.

El dinamismo es otra característica necesaria en los MAM's. Esta necesidad provocó el desarrollo del *árbol de Aproximación Espacial Dinámico (DSAT)* [11] que permite realizar inserciones y eliminaciones. Está basado en el *árbol de Aproximación Espacial* (*SAT*) que, a pesar de ser uno de los índices de mejor desempeño en espacios de mediana a alta dimensión, es totalmente estático. El *DSAT* conserva el muy buen desempeño en las búsquedas, pero agrega un parámetro a sintonizar. Otra variante del *SAT*, el *árbol de Aproximación Espacial Distal* (*DiSAT*) [6], a pesar de ser estática, logra optimizar las búsquedas respecto de ambos (SAT y DSAT) y no necesita parámetros. Por ello, se ha propuesto la *Foresta de Aproximación Espacial Distal* (*DiSAF*) [3], basada en él pero que es dinámica. Es para memoria principal y aplica la técnica de dinamización de Bentley y Saxe al *DiSAT*, aprovechando el profundo conocimiento que se tiene sobre la aproximación espacial para mejorar al máximo su desempeño.

Otra faceta que hay que tener en cuenta, son los requerimientos de algunas aplicaciones que priorizan la rapidez en las respuestas aunque sea a costa de perder algunos elementos: se intercambia precisión (devolviendo sólo algunos objetos relevantes) por velocidad en la respuesta. Este tipo de búsquedas se denominan *aproximadas*. Para conjuntos de datos masivos, las búsquedas por similitud aproximadas permiten obtener un buen balance entre el costo de las búsquedas y la calidad de la respuesta obtenida. El *Algoritmo Basado en Permutaciones* (*PBA*) [2]. es uno de los mejores representantes de este tipo de consultas, logrando una respuesta de alta calidad a un bajo costo. Por esta razón, se ha utilizado como base del diseño de la *Lista Dinámica de Permutaciones Agrupadas* (*DLCP*) [9], que es dinámica y para memoria secundaria. Este índice, que combina *LC* con *PBA*, agrupa por distancia entre las permutaciones de los objetos, en lugar de por distancia entre objetos y se le puede indicar cuántos cálculos de distancia y/o operaciones de I/O utilizar, para obtener una respuesta rápida, aunque menos precisa. Además, se están considerando nuevas variantes para obtener mejores resultados.

Resultados y Objetivos

Las investigaciones realizadas sobre el modelo de espacios métricos, han permitido mejorar el desempeño de los MAMs estudiados, y los resultados obtenidos conducen a intentar aplicarlos a otros métodos de acceso [4, 5, 10, 6, 11, 3].

Se espera brindar nuevas herramientas eficientes de administración para bases de datos métricas, que logren acercar su desarrollo al de los modelos tradicionales de base de datos. Para ello, se buscará profundizar en el estudio de nuevos diseños de estructuras de datos, buscando incrementar su eficien-

cia en espacio y en tiempo: que se adapten mejor al nivel de la jerarquía de memorias donde se almacenarán y a las características de los datos a ser indexados. modelos tradicionales de base de datos. Se continuará estudiando diferentes técnicas que sin utilizar de índices, permitan resolver consultas eficientemente. Además, se espera mejorar el desempeño de las operaciones de bajo nivel en los DBMS, mediante una nueva arquitectura del procesador.

Actividades de Formación

Dentro de esta línea de investigación se forman alumnos y docentes-investigadores en:
Doctorado en Cs. de la Computación: una tesis sobre expresividad de lenguajes lógicos de consulta.
Maestría en Cs. de la Computación: una tesis sobre búsqueda por similitud aproximada (concluida) y otra sobre un índice dinámico eficiente.
Maestría en Informática: una tesis, de la Universidad Nacional de San Juan, sobre un índice dinámico para búsquedas aproximadas en disco (concluida).

Referencias

[1] G. Blaauw and F. Brooks, Jr. *Computer Architecture: Concepts and Evolution*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1997.

[2] E. Chávez, K. Figueroa, and G. Navarro. Effective proximity retrieval by ordering permutations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(9):1647–1658, Sept 2008.

[3] E. Chávez, M. Di Genaro, N. Reyes, and P. Roggero. Decomposability of disat for index dynamization. *Computer Science & Technology*, pages 110–116, 2017.

[4] E. Chávez, V. Ludueña, and N. Reyes. Solving all-k-nearest neighbor problem without an index. In *Procs. del XXV Congreso Argentino de Ciencias de la Computación CACIC 2019)*, pages 567–576. UniRío editora, 2019.

[5] E. Chávez, V. Ludueña, N. Reyes, and F. Kasián. All near neighbor graph without searching. *Computer Science & Technology*, 18:61–67, 2018.

[6] E. Chávez, V. Ludueña, N. Reyes, and P. Roggero. Faster proximity searching with the distal {SAT}. *Information Systems*, 59:15 – 47, 2016.

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

[8] P. Ciaccia and M. Patella. Approximate and probabilistic methods. *SIGSPATIAL Special*, 2(2):16–19, 2010.

[9] K. Figueroa, C. Martínez, R. Paredes, N. Reyes, and P. Roggero. Dynamic list of clustered permutations on disk. In *Computer Science and Technology Series: XXI Argentine Congress of Computer Science Selected Papers*, pages 201–211. EDULP, 2016.

[10] A. Camarena-Ibarrola L. Valero-Elizondo K. Figueroa, N. Reyes. Improving the list of clustered permutation on metric spaces for similarity searching on secondary memory. In *10th Mexican Conference on Pattern Recognition (MCPR2018)*, volume 10880, pages 82–92, 2018.

[11] G. Navarro and N. Reyes. New dynamic metric indices for secondary memory. *Information Systems*, 59:48 – 78, 2016.

[12] R. Paredes, E. Chávez, K. Figueroa, and G. Navarro. Practical construction of k-nearest neighbor graphs in metric spaces. In *Proc. 5th Workshop on Efficient and Experimental Algorithms (WEA)*, LNCS 4007, pages 85–97, 2006.

[13] H. Samet. *Foundations of Multidimensional and Metric Data Structures (The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[14] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. XVIII, 220 p., Hardcover ISBN: 0-387-29146-6.