Bases de Datos no Convencionales

M. D. Alba, J. Arroyuelo, M. E. Di Genaro, A. Grosso, M. Jofré, V. Ludueña, N. Reyes, D. Welch
Dpto. de Informática, Fac. de Cs. Físico-Matemáticas y Naturales, Universidad Nacional de San Luis
{*bjarroyu, digeme, agrosso, vlud, nreyes,dwelch*}@unsl.edu.ar,{*mdaniela.alba, monicajofre*}@gmail.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

El avance que han tenido las nuevas tecnologías en todos los ámbitos de la vida, ha ocasionado la producción de infinidad de datos digitales tan disímiles como lo son sus fuentes. Así, se generan datos provenientes de sensores de todo tipo, resultados de estudios médicos, datos científicos, secuencias biológicas, información obtenida de satélites, etc.; datos complejos que necesitan ser procesados para atender a los variados requerimientos del mundo actual. Este contexto motivó que las bases de datos debieran adaptarse, para ser capaces de administrar eficientemente grandes volúmenes de estos nuevos y diversos tipos de datos y así atender esos requerimientos.

El enfoque que se ha vuelto popular en los últimos años, para manejar las diversas bases de datos emergentes de objetos complejos y no estructurados, es el de *espacios métricos* y es el utilizado en las investigaciones presentadas en este artículo. Las mismas pretenden contribuir a la madurez de este modelo de bases de datos, *Bases de Datos Métricas*, considerando distintas perspectivas: administración del espacio disponible (crucial por la gran cantidad de datos); formas más sofisticadas de búsqueda sobre las mismas; optimización de estos depósitos, o desarrollo de nuevos, entre otros.

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

Contexto

Las investigaciones mencionadas se realizan en el marco de la línea *Bases de Datos no Convencionales* perteneciente al Proyecto Consolidado *Tecnologías Avanzadas de Bases de Datos* que se desarrolla en la Universidad Nacional de San Luis (Código 03-2218) y en el Programa de Incentivos (Código 22-F814) finalizado en diciembre de 2022. Actualmente se realizó una nueva presentación para su continuación. Dicho proyecto está incorporado al Laboratorio de Investigación y Desarrollo en Bases de Datos (LaBDa) en el cual colaboran investigadores de otros grupos de la región: Universidad de Talca (Chile), Universidad Michoacana de San Nicolás de Hidalgo y Centro de Investigación Científica y de Educación Superior de Ensenada (México).

El presente artículo expone el trabajo que se desarrolla en este ámbito, orientado a contribuir al modelo de Bases de Datos Métricas, dado que todavía no se ha alcanzado todo el potencial en las soluciones, el problema de las búsquedas sobre estos tipos de bases de datos es un problema abierto en varios aspectos. Para ello se trabaja, obteniendo nuevas estructuras de datos (índices) que resulten eficientes para memorias jerárquicas, que sean dinámicos, manejen grandes volúmenes de datos (escalables), con eficiencia en E/S y adecuados a la variedad de datos no estructurados existentes. De esta manera, se espera aportar a diferentes campos de aplicación: diseño asistido por computadora, sistemas de información geográfica, robótica, visión artificial, biología computacional, computación móvil, entre otros.

Introducción

La generación de grandes cantidades de datos digitales, provocada por el uso generalizado de dispositivos capaces de producirlos, aumenta constan-

temente. Así, tanto la cantidad como la variedad de datos provenientes de ámbitos tan diferentes como el de la salud, el educativo, productivo, laboral, recreativo, científico, etc. ha crecido de manera exponencial. En este contexto los repositorios especializados en datos *no estructurados* se vuelven indispensables. Como respuesta, las bases de datos han debido adaptarse rápidamente, tanto a la cantidad de datos que deben administrar, como a su variedad y disimilitud.

Más aún, son diferentes los requerimientos que las aplicaciones actuales plantean sobre estos datos; encontrar las huellas digitales más similares a una dada, u obtener composiciones semejantes a un trozo de una melodía, son algunos ejemplos. En estos casos las búsquedas tradicionales (exactas) carecen de sentido, en cambio las *búsquedas por similitud* resultan más adecuadas; en ellas, se suele dar un objeto como "ejemplo" de lo que se quiere recuperar y se busca en la base de datos los objetos que sean suficientemente similares a la muestra. Estas búsquedas son coincidas como consultas por contenido o consultas mediante un ejemplo (query by example).

Un modelo adecuado para la variedad de aplicaciones que utilizan búsquedas por similitud es el de *espacios métricos*, ya que, a pesar de las necesidades tan diversas de las mismas, todas comparten ciertas características que hacen a este modelo el más adecuado como su marco formal. Este modelo es determinado por un universo de objetos y una función de distancia definida entre ellos, que mide cuán diferentes son. Cualquier tipo de objetos no estructurados, que admita la definición de una medida que indique cuán diferentes son dos objetos, admite ser modelizado de esta manera. La única restricción es que esa medida cumpla con las propiedades que la hagan una métrica. En general, esas medidas son provistas por expertos (por ejemplo: distancias para comparar huellas dactilares).

Sin embargo, para responder eficientemente a los requerimientos tan dispares que se realizan sobre los objetos no estructurados almacenados en estas bases de datos, son necesarios los llamados *Métodos de Acceso Métricos* (MAMs), que permiten resolver este tipo de búsquedas, sin realizar una examinación secuencial del conjunto de datos [6, 9]. La diversidad de ámbitos en los que se aplica el modelo, vuelve esencial la actualización y optimización de los MAMs, el adaptarlos a cada caso particular y afrontar retos como soportar conjuntos masivos de datos, permitir actualizaciones (inserciones/eliminaciones) y resolver búsquedas más complejas.

Líneas de Investigación y Desarrollo

Bases de Datos Métricas

Como se expresó anteriormente, las *bases de datos no convencionales*, aquellas capaces de administrar vídeos, imágenes, texto libre, secuencias de proteínas o ADN, audio, etc., son modelizadas utilizando el *modelo de espacios métricos*, por lo que también se las conoce como *bases de datos métricas*. En este modelo, debido a lo costoso que suele resultar calcular la distancia entre dos objetos, el número de cálculos realizados durante alguna operación se usa habitualmente como medida de complejidad. Estos cálculos son necesarios tanto para crear un índice como para realizar búsquedas sobre él. Por lo tanto, para responder eficientemente a los distintos requerimientos sobre las mismas, evitando comparaciones exhaustivas sobre la base de datos, se requiere el uso de índices especializados.

Ésta es la razón por la cual optimizar los mismos se ha vuelto un objetivo prioritario, analizando aquellos que han mostrado buen desempeño, principalmente en las búsquedas, para reducir su complejidad. Para ello, cuando sea necesario, se considera el nivel de la memoria en la que se lo alojará, su capacidad de ser dinámico y de ser escalable.

Formalmente, un espacio métrico está definido por un universo de objetos $\mathbb{U}$ y una función de distancia entre ellos $d : \mathbb{U} \times \mathbb{U} \longmapsto \mathbb{R}^+$, la cual permite medir la disimilitud entre los objetos del universo. En particular, d cumple con las propiedades de una métrica (reflexividad, positividad estricta, simetría y desigualdad triangular), lo que resulta muy útil al momento de resolver consultas por similitud. De sus propiedades, las que permiten ahorrar cálculos de distancia son la simetría y la desigualdad triangular. Conceptualmente, una base de datos X es un conjunto $X \subseteq \mathbb{U}$. Además, una consulta por similitud se expresa brindando un elemento $q \in \mathbb{U}$ y, en general, suele ser de dos tipos: *por rango* (q, r) o de *k-vecinos más cercanos* (k-NN(q)).

Métodos de Acceso Métricos

Como se comentó anteriormente, optimizar los índices métricos se ha vuelto un objetivo prioritario, y teniendo en cuenta el variado espectro de aplicación de los mismos, se deben considerar diferentes puntos de vistas para hacerlo. Hay que considerar su dinamismo; su escalabilidad; analizar si se adaptan adecuadamente al almacenamiento en memoria secundaria; entre otros. En este último caso, se de-

be considerar un cambio en la medida de complejidad, y que no sólo se debe tener en cuenta el número de cálculos de distancia, sino también el número de operaciones de E/S necesarias.

En relación al dinamismo, un buen representante en este sentido es el *Árbol de Aproximación Espacial Dinámico (DSAT)* [13], que es la versión dinámica de uno de los índices de mejor desempeño en espacios métricos de mediana a alta dimensión el *Árbol de Aproximación Espacial* (*SAT*). El *DSAT* mantiene un buen desempeño en las búsquedas pero agrega un parámetro a sintonizar. Siguiendo la idea del *DSAT*, se emprendió la tarea de proveer de dinamismo al *Árbol de Aproximación Espacial Distal* (*DiSAT*) [5], que además de no necesitar ningún parámetro extra, lograba mejorar las búsquedas respecto de sus antecesores: *SAT* y *DSAT*, pero también era estático. Así, en una primera instancia se propuso el desarrollo de la *Foresta de Aproximación Espacial Distal* (*DiSAF*) [2], que es dinámica y para memoria principal. Este índice aplica la técnica de dinamización de Bentley y Saxe al *DiSAT* y aprovecha el profundo conocimiento que se tiene sobre la aproximación espacial para mejorar al máximo su desempeño. Sin embargo, los costos de construcción son altos debido a la necesidad de reconstruir subárboles luego de cada inserción.

Para solucionar este problema, se diseñó el *Árbol de Aproximación Espacial Distal Dinámico* (*DDiSAT*) [7]. Esta nueva versión utiliza *inserción perezosa* para amortizar los costos de reconstrucción, a la vez que mejora su desempeño en las búsquedas, sin usar espacio adicional. La propuesta es retrasar la inserción de nuevos elementos, asignándolos (en una bolsa) a su objeto más cercano en el árbol. Los objetos se insertan en las bolsas hasta que la cantidad de elementos pendientes iguala al número de nodos del árbol. En ese momento, se reconstruye el árbol completo. El (DDiSAT) reduce significativamente los costos de construcción con respecto a *DiSAF* y, sorpresivamente, obtiene un mejor rendimiento de búsqueda que sus parientes cercanos estáticos (*SAT* y *DiSAT*) y dinámico (*DSAT*). Ello se debe a la buena partición del espacio y a la buena distribución de elementos que hace el índice. Así, se sigue trabajando en posibles mejoras, por ejemplo, diseñando un algoritmo de carga masiva para crear el DDiSAT, si se conoce de antemano un subconjunto de elementos, evitando algunas reconstrucciones y combinándolo con la inserción diferida.

Al considerar el almacenamiento de un índice, muchas veces sucede que los mismos no caben en memoria principal debido al tamaño de los objetos almacenados en él o la gran cantidad de objetos que almacena. Entonces surge la necesidad de diseñar MAMs que se puedan almacenar en memoria secundaria sin perder eficiencia. En este contexto, se está trabajando en lograr una versión dinámica del *DDiSAT* para memoria secundaria, que además de amortizar los costos de reconstrucción entre varias inserciones y mantener un buen desempeño en las búsquedas, se adapte a memoria secundaria realizando un buen uso de las páginas de disco para minimizar el número de operaciones de E/S. Para lograrlo, se debe considerar no sólo que logre un desempeño comparable al de la versión de memoria principal en cantidad de cálculos de distancia, sino que las páginas en disco mantengan una buena ocupación, que la cantidad de operaciones de E/S necesarias sea baja y que se mantenga la localidad en los accesos.

Otra faceta a considerar se relaciona con aplicaciones que priorizan la rapidez en las respuestas, a costa de perder algunos elementos de la misma. Este tipo de búsquedas, en las que se intercambia precisión (devolviendo sólo algunos objetos relevantes) por velocidad en la respuesta (esos objetos se devuelven más rápido), se conocen como *búsquedas aproximadas*. Cuando se trabaja con conjuntos de datos masivos, las búsquedas por similitud aproximadas permiten obtener un buen balance entre costo de las búsquedas y calidad de la respuesta obtenida. En este contexto, se propuso la *Lista Dinámica de Permutaciones Agrupadas* (*DLCP*) [11], que además de ser dinámica y para memoria secundaria, permitía realizar consultas aproximadas. Sin embargo, algunos aspectos de la estructura eran mejorables [12]. Así, se está trabajando en combinar el *Conjunto Dinámico de Clusters* (*DSC*) [13], un índice para memoria secundaria con muy buen desempeño; con uno de los mejores representantes para consultas aproximadas, el *Algoritmo Basado en Permutaciones* (*PBA*) [1], para lograr un nuevo índice para búsquedas aproximadas, con buena calidad de respuesta y eficiente en memoria secundaria.

Además, se espera poder incorporar estos nuevos índices, tipos de búsqueda y otras operaciones de interés en un DBMS que trabaja sobre bases de datos métricas, el cual se basa en PostgreSQL. Esto permitirá aumentar su capacidad de administración y procesamiento para grandes bases de datos.

Búsqueda 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 un elemento dado; este es el tipo de consulta que utilizamos cuando se quiere encontrar los restaurantes más cercanos a un lugar en particular. Una variación muy útil de la misma, es la *All-k*-NN, que 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 tan excesivo como la solución ingenua. Este es el caso de administrar 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 [15, 6, 16, 10]. Para hacer frente a éstas circunstancias es que se han considerado las llamadas *búsquedas por similitud aproximadas*, las mismas mejoran su complejidad aceptando algunos "errores" en la respuesta.

Se sabe que al resolver el problema *All-k*-NN permite construir el *Grafo de los k-vecinos más cercanos* (kNNG)[14]. 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 provocar que la respuesta pierda algún objeto muy cercano, devolviendo uno más lejano en su lugar, a cambio de una respuesta más rápida. Este grafo se denominó *Grafo de vecinos cercanos* (knNG) [4].

Una primera propuesta, utilizando un enfoque novedoso, surge aprovechando el profundo conocimiento que se tiene del *DiSAT* [5]. 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 [4]. Con este algoritmo se consigue recuperar el 1nNG logrando un buen compromiso calidad/tiempo; los costos de las consultas resultan ser bajos, se alcanza muy buena precisión en la respuesta con un error bajo y todo esto *sin realizar ninguna búsqueda en el índice.*

Se realizaron otras tres propuestas que, a diferencia de la primera, se enfocan en el problema más general al responder a los *All-k*-nN y computar el knNG. Estos desarrollos 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. La base de estos algoritmos es aprovechar de manera ingeniosa las propiedades que cumple la *función de distancia*. Para hacerlo, se proponen diferentes 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 para resolver la consulta. Se analizan diferentes maneras de utilizar la información obtenida para calcular, en algunos casos, los vecinos exactos [3] y en otros los aproximados de todos los objetos de la base de datos, utilizando las propiedades de simetría o desigualdad triangular [8].

Resultados y Objetivos

Las investigaciones realizadas sobre el modelo de espacios métricos, permitieron mejorar el desempeño de los MAMs analizados, los resultados obtenidos conducen a estudiar su aplicación a otros métodos de acceso [7, 8, 3, 4, 12, 2].

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. Así, se buscará profundizar en el estudio de nuevos diseños de estructuras de datos, buscando incrementar su eficiencia 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. Del mismo modo, se continuará indagando sobre técni-

cas innovadoras que, sin utilizar índices, permitan resolver consultas eficientemente.

Actividades de Formación

Dentro de esta línea de investigación se forman alumnos y docentes-investigadores participando en:

- **Maestría en Cs. de la Computación** (UNSL): tesis sobre una versión dinámica eficiente del *DiSAT*.
- **Maestría en Ing. de Software** (UNSL): tesis sobre una aplicación de estos índices a un DBMS métrico.
- **Maestría en Informática** (UNSJ): tesis sobre la evaluación del knNG para búsquedas por similitud.
- **Maestría en Informática** (UNSJ): tesis sobre una versión dinámica y para memoria secundaria del *DDiSAT*.

Referencias

[1] 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.

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

[3] 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*, pages 567–576. UniRío editora, 2019.

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

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

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

[7] Edgar Chávez, María E. Di Genaro, and Nora Reyes. An efficient dynamic version of the distal spatial approximation trees. In *Actas del XXVIII Congreso Argentino de Ciencias de la Computación*, pages 468–477, Oct. 2022.

[8] Edgar Chávez, Verónica Ludueña, and Nora Reyes. Heuristics for computing k-nearest neighbors graphs. In Patricia Pesado and Marcelo Arroyo, editors, *Computer Science – CACIC 2019*, pages 234–249, Cham, 2020. Springer.

[9] Lu Chen, Yunjun Gao, Xuan Song, Zheng Li, Yifan Zhu, Xiaoye Miao, and Christian S. Jensen. Indexing metric spaces for exact similarity search. *ACM Comput. Surv.*, 55(6), dec 2022.

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

[11] 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.

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

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

[14] 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*, LNCS 4007, pages 85–97, 2006.

[15] 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.

[16] 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.