Contribuciones a las Bases de Datos no Convencionales

Jorge Arroyuelo, Maria E. Di Genaro, Susana Esquivel, Alejandro Grosso, Veró nica Ludueñ a, Cintia Mart´ınez, Nora Reyes
Dpto. de Informática, Fac. de Cs. Físico-Matemáticas y Naturales, Universidad Nacional de San Luis
{bjarroyu, mdigena, esquivel, 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@computo.fismat.umich.mx
Rodrigo Paredes
Dpto. de Cs. de la Computación, Fac. de Ingeniería, Universidad de Talca, Chile
raparede@utalca.cl

Resumen

El advenimiento de las ciencias de la computación a todos los ámbitos de la vida moderna, ha exigido el desarrollo de aplicaciones que satisfagan los requerimientos de distintos tipos de usuarios, desde campos muy dispares, adaptándose a todo tipo de exigencias para lograr un alcance masivo. Claramente, esto implica lograr manipular eficientemente datos no convencionales muy dis´ımiles como: huellas digitales, imágenes, audio, secuencias de ADN, texto, video, etc. Como las soluciones tradicionales no suelen hacer frente a tales requerimientos, es necesario utilizar depósitos especializados y búsquedas no exactas sobre estos tipos dedatos.

Además de proveer una respuesta rápida y adecuada a dichas demandas, es necesario un uso eficiente del espacio disponible, y al considerar bases de datos masivas, las estructuras en particular serán *estructuras de datos con I/O eficiente*. Las *Bases de Datos Métricas* son uno de los modelos generales en los cuales se pueden utilizar estructuras de datos especializadas que contemplen estos aspectos. Los lenguajes de consulta no siempre poseen el poder expresivo necesario para reflejar las consultas consideradas de interés. Así, nuestra investigación pretende este proyecto, abarca aspectos relacionados con lograr que las bases de datos no convencionales, destinadas a manipular datos no estructurados, alcancen la madurez de las bases de datos tradicionales. Es- to incluye la expresividad de los lenguajes de con- sulta, los operadores necesarios para responder pre- guntas de interés, y el análisis de aspectos teóricos, emp´ıricos y aplicativos de los mismos; contribuyen- do así a distintos campos de aplicació n: sistemas de información geográfica, computación móvil, robótica, visión artificial, motores de búsqueda en internet, diseño asistido por computadora, etc.

Nuestras investigaciones se realizan en la colaboración con investigadores de otros grupos de: Universidad de Talca (Chile), Universidad Michoacana de San Nicolás de Hidalgo (México) y Centro de Investigación Científica y de Educación Superior de Ensenada (México).

Palabras Claves: bases de datos no convencionales, expresividad, ´ındices, lenguajes de consulta.

Contexto

El desarrollo del presente trabajo se enmarca en la línea de investigación *Bases de Datos no Convencionales* del Proyecto Consolidado 30314, *Tecnolog´ıas Avanzadas de Bases de Datos*, de la Universidad Nacional de San Luis La investigación que se realiza en

Introducción

La evolución de la tecnología ha permitido que la [illegible] los diferentes aspectos de la vida humana, como el laboral, productivo, recreativo, de la salud, etc. Esto se ha logrado gracias al desarrollo de aplicaciones capaces de adaptarse tanto a estos nuevos entornos como a los diversos usuarios de las mismas. Para ello, las bases de datos han debido evolucionar hasta ser capaces de administrar todo tipo de datos y responder consultas sobre los mismos de una manera totalmente diferen- te a la tradicional, muchas veces más intuitiva. Estos

avances contribuyen en aplicaciones como: comparación de huellas digitales, bases de datos médicas, reconocimiento de voz, reconocimiento facial, reconocimiento de imágenes, recuperación de texto, biología computacional, minería de datos, clasificación y aprendizaje automático, etc.

A pesar de su diversidad, todas estas aplicaciones tienen caracter´ısticas comunes, que pueden englobarse en el modelo de *espacios métricos*. Formalmente un espacio métrico consiste de un universo de objetos $\mathbb{U}$ y una función de distancia definida entre ellos $d : \mathbb{U} \times \mathbb{U} \rightarrow \mathbb{R}^+$ que mide la disimilitud entre los objetos. Este escenario es propicio para resolver demandas tales como, introducir una imagen en un buscador, esperando obtener aquellas que sean similares a la muestra. Por ello, las *bú squedas por similitud*, provistas por este modelo, son más naturales sobre estos tipos de datos.

Los *Métodos de Acceso Métricos* (MAMs) permiten responder eficientemente a este tipo de búsquedas, evitando la examinación secuencial de los datos. Sin embargo, es esencial su optimización, ya que la mayoría de estos métodos no están diseñados para soportar conjuntos masivos de datos, ni operaciones de búsqueda complejas, y además no admiten dinamismo. El trabajo con bases de datos masivas, o con objetos muy grandes, da lugar también a investigar sobre diseñar MAMs más eficientes para memorias jerárquicas, al cambiar el modelo de costos.

Otras áreas de investigación exploradas buscan incrementar la expresividad de los lenguajes de consulta para expresar consultas más precisas y 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 en administradores de bases de datos (DBMS) a bajo nivel.

Líneas de Investigación y Desarrollo

Expresividad de los Lenguajes

Si se piensa en una base de datos simplemente como una estructura finita, se pueden utilizar las lógicas para expresar consultas sobre éstas. El empleo de lógicas para expresar consultas (o problemas) da origen a la complejidad descriptiva, que clasifica a los problemas según el uso de recursos lógicos tales como el número de variables, cuantificadores, operadores, etc. Existe una relación estrecha entre estos dos tipos de complejidades para clases que se identifican con la computación factible, pero el dominio de las estructuras debe estar ordenado. En ese caso la clase de complejidad P es capturada por FO (*First-Order Logic*) extendida con un operador de punto fijo. Aún así, estas lógicas todavía resultan incompletas, ya que ninguna caracterización lógica de computación factible es conocida para estructuras cuyo dominio no esta´ ordenado.

A. Dawar demuestra en [9] que ninguna extensión de la lógica de punto fijo, con un número finito de cuantificadores generalizados, captura la clase de complejidad P y en [10] que ciertos problemas NP completos sobre inequivalencia de autómatas finitos son expresables en el fragmento existencial de la lógica SO^{ω} mientras que el problema NP completo 3-coloreablidad no lo es. Es importante utilizar diferentes lógicas para separar problemas que en complejidad clásica son vistos como similares.

En esta línea de investigación se continúa con la l´ınea estudiada por Dawar en SO^{ω}, la cual plantea una restricción semántica a la SO, donde la valuación de las variables relacionales para los cuantificadores de segundo orden son cerrados bajo $\equiv_k$. Se define una nueva lógica de tercer orden (TO), denominada TO^{ω}. É sta intenta caracterizar y estudiar clases de complejidad relacionales (temporales) de lógicas de orden superior. Una relación se dice cerrada bajo $\equiv_k$ si todas las tuplas del dominio sobre el que trabaja, que satisfacen las mismas formulas de FO^k, están en la relación. Se definió una variación de una máquina relacional no deterministica, denotada como 3-NRM, donde se permiten relaciones de tercer orden en el *relational store*; esto permite asociar TO^{ω} a una clase de complejidad temporal. Esa clase de complejidad fue llamada $\mathrm{NEXPTIME}_{3,r}$, como la clase de máquinas 3-NRM que trabajan en tiempo exponencial de acuerdo al tamaño de la entrada. La clase $\mathrm{NEXPTIME}_{3,r}$ es exactamente caracterizada por el fragmento existencial de TO^{ω} [1].

Arquitecturas de Procesadores Orientadas a Bases de Datos

La arquitectura del procesador es la funcionalidad que se le provee al programador en lenguaje de máquina, modos de direccionamiento, operaciones, interrupciones y entrada-salida [2]. En ella se distinguen: la organización básica del flujo de datos y el control (*implementación*) y la estructura física que comprende la implementación (*realización*). El lenguaje de máquina actual no es ni un lenguaje de aplicación ni un lenguaje de hardware, sino algo intermedio. Entonces, ¿por que´ no interpretar directamente un lenguaje de alto nivel en lugar de compilar a un lenguaje intermedio? o ¿por que´ no darle acceso

directo al programador, o compilador, al hardware en lugar de restringirlo al lenguaje de máquina?

Se puede elegir la estrategia de "impulso ha- cia arriba"; es decir subir el nivel, para mejorar el desempeño de la máquina, además de facilitar el uso del lenguaje de máquina. Un aspecto a considerar en este caso es el tráfico de bits y la forma usual de reducirlo es tener una arquitectura que haga lo más posible con cada búsqueda de instrucción, abandonando la arquitectura de bajo nivel y yendo tan al- to como el software lo permita. El otro aspecto es explotar la concurrencia, porque si una implementación conoce más sobre lo que debe ser hecho entonces es posible que a menudo realice varias acciones simultánemente. El implementador posee varias técnicas para aumentar la concurrencia: paralelizar, segmentar (pipelining), adelantar, poner a un lado (cache look-aside), adivinar y corregir (control and data prediction). La otra estrategia es considerar el "impulso hacia abajo". Aún si todas las aplicaciones fueran escritas en lenguaje de alto nivel, hay razones para definir una arquitectura de computadora de ni- vel más bajo, pues existe conflicto de intereses entre usuario e implementador: el usuario desea expresar en forma simple y breve, haciendo uso del contexto, y el implementador desea que cada instrucción sea interpretada independientemente delresto.

Por lo tanto, es importante definir una arquitectura cuando se construye una computadora. En la actualidad la investigación sobre arquitecturas de procesadores ha sido desplazada por la de implementación de procesadores. La mayor´ıa de los trabajos de investigación se dedican a mejorar técnicas de predicción (tanto de control como de datos), técnicas pa- ra sincronizar y comunicar procesadores (núcleos) mediante mensajes y/o memoria compartida. Muchas de estas técnicas de implementación surgieron en los años 60 y hoy se han incorporado a los diseños de microprocesadores actuales. Sin embargo, estas técnicas de implementación se podrían aplicar a todo tipo de arquitectura, desde una arquitectura RISC [1] (que intenta acercar el lenguaje de máquina al hardware del procesador) a una arquitectura que se aleje del hardware e intente disminuir el tráfico de bits entre procesador y memoria. El objetivo en esta área es plantear nuevas arquitecturas que minimicen el tráfico de bits entre el procesador y la memoria. Se está construyendo un simulador del set de instruc- ciones AMD-64 o x86-64, como "benchmark", para evaluar el tráfico de bits, como Specint y Specfp para la arquitectura x86. Luego, se evaluará el tráfico de bits para la arquitectura propuesta sobre los mismos benchmarcks, lo que implica construir tanto el simulador de la arquitectura como el compilador C para la misma. Finalmente, se pretende aprovechar al conocimiento adquirido para, desde bajo nivel, mejorar el desempeño de los DBMSs.

[1] Acrónimo del inglés "Reduced Instruction Set Computer".

Bases de Datos Métricas

Las bases de datos no convencionales, que gestionan imágenes, videos, texto libre, secuencias de ADN, audio, etc., pueden modelarse con espacios métricos generales. 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 aqu´ı es optimizar los MAMs, necesarios al momento de responder las diversas consultas a una base de datos, analizando aquellos que han mostra- do buen desempeño en las búsquedas para reducir su complejidad considerando, cuando sea necesario, la jerarqu´ıa 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).

Métodos de Acceso Métricos

Como se dijo anteriormente, una de las optimzaciones necesarias a los MAM's es el dinamismo. Por ejemplo, considerando el *Árbol de Aproximación Espacial* (*SAT*), un índice con muy buen desempeño en espacios de mediana a alta dimensión, pero totalmente estático, se desarrolló el *Árbol de Aproximación Espacial Dinámico (DSAT)* [12] que permite realizar inserciones y eliminaciones, conservan- do muy buen desempeño en las búsquedas, pero que agrega un parámetro a sintonizar. El *Árbol de Aproximación Espacial Distal* (*DiSAT*) [6], una variante también estática del *SAT* y sin parámetros, logra optimizar las búsquedas respecto de ambos (SAT y DSAT). Por ello, se ha propuesto la *Foresta de Aproximación Espacial Distal* (*DiSAF*) [4], que es dinámica, para memoria principal y que para lograr mejorar al máximo su desempeño, aplica la técnica de dinamización de Bentley y Saxe al *DiSAT* y aprovecha el profundo conocimiento que se tiene sobre la aproximación espacial.

Sin embargo, muchas veces los ´ındices no ca- ben en memoria principal, ya sea porque adminis- tran una base de datos masiva, o porque los obje- tos de la misma son muy grandes. Entonces surge la necesidad de diseñar índices para memoria secun- daria. Muchos de estos ´ındices se basan en "agru-

par elementos"; y para analizar cuán buenos son los agrupamientos que logran, se pueden utilizar estrategias de optimización basadas en heurísticas bioinspiradas. 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 elementos y están especialmente diseñados para trabajar sobre grandes volúmenes de datos [13]. La *Lista de Clusters Dinámica* (*DLC*), tiene buen desempeño en espacios de alta dimensión, con buena ocupación de página y operaciones eficientes tanto en cálculos de distancia como en operaciones de I/O. Sin embargo, las búsquedas en ella deben 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. La información de ese *DSAT* también se aprovecha en las inserciones, mejorando los costos de las operaciones en cálculos de distancia y manteniendo los bajos costos de acceso a disco. Ambos, *DLC* y *DSC*, han demostrado tener una razonable utilización de páginas de disco y son competitivas respecto a las alternativas representativas del estado delarte.

Algunas aplicaciones requieren que las respuestas sean aún más rápidas, aunque sea a costa de algunos errores: 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*) [3]. es uno de los mejores representantes de este tipo de consultas, logrando una respuesta de alta calidad a un bajo costo. Por ello, se ha diseñado la *Lista Dinámica de Permutaciones Agrupadas* (*DLCP*) [11] (combina *LC* con *PBA*), que además es dinámica y para memoria secundaria. Este índice 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.

All-k-NN Aproximados

Entre las aplicaciones que pueden modelizarse con *espacios métricos*, se encuentran algunas como la predicción de funciones, la clasificación y aprendizaje automático, la cuantificación y compresión de imágenes; que utilizan las búsquedas por similitud k-NN. Dado un elemento $u \in \mathbb{U}$ y sea $X \subseteq \mathbb{U}$ la base de datos, k-NN(u) obtiene los k elementos en $X - \{u\}$ que tengan la menor distancia d a u. Una variante menos analizada de esta primitiva, denominada *All-k*-NN, es la búsqueda de los k-vecinos más cercanos de *todos* los elementos de X; es decir, obtiene los k-NN(u_i) para cada $u_i \in X$. Si $|X| = n$, se pretende resolver este problema realizando menos de n^2 cálculos de distancia, preprocesando los datos para reducir el número de cálculos de distancia (la complejidad en este modelo).

Se han propuesto soluciones a este problema para espacios métricos generales [14], basadas en la construcción del *Grafo de los k-vecinos más cercanos* (kNNG). El kNNG indexa un espacio métrico y luego se emplea en la resolución de las consultas por similitud. Su desempeño en las búsquedas supera al de las técnicas clásicas basadas en pivotes. La solución habitual para obtener el kNNG suele construir un índice y luego realizar búsquedas de los k vecinos más cercanos, para todos los elementos de la base de datos. Una aproximación al kNNG puede ser el *Grafo de vecinos cercanos* (knNG) [5]. Un caso particular se obtiene cuando $k = 1$ y así el 1nNG sería el grafo que conecta a cada elemento con un elemento cercano, que puede ser, o no, su vecino más cercano.

Las *búsquedas por similitud aproximadas* se requieren en muchas aplicaciones que priorizan la velocidad sobre la precisión [15, 7, 16, 8]. También resultan útiles en espacios métricos de alta dimensión donde, resolver consultas requiere revisar casi todo el conjunto de datos sin importar la estrategia utilizada. Lo mismo sucede cuando la función de distancia es demasiado costosa de calcular, o si se tiene una base de datos masiva. En estos casos además, puede surgir otro problema: el costo de la construcción de un índice, para luego obtener los vecinos cercanos, puede resultar excesivo.

Considerando estas situaciones y que gracias al conocimiento profundo que se tiene del DiSAT se sabe que la información que se obtiene durante la construcción del índice puede aprovecharse para la obtención del 1nNG, se ha propuesto en [5] un enfoque novedoso al problema: cada objeto es relacionado con el elemento más cercano de la base de datos con el que se comparó durante la construcción. Más aún, la construcción del DiSAT retorna una aproximación del 1-NNG, la cual puede mejorarse median-

te reconstrucciones adicionales. Es posible obtener así un adecuado 1nNG, sin realizar búsquedas y utilizando una fracción del costo de la solución habitual. La nueva propuesta permite recuperar el 1nNG con bajo costo, una muy buena precisión y un error bajo, logrando un buen compromiso calidad/tiempo.

Resultados y Objetivos

Los estudios realizados sobre el modelo de espacios métricos permitirán mejorar el desempeño de los MAMs analizados y estudiar la aplicación de los resultados obtenidos a otros [4, 5, 6, 13, 11].

Se profundizará el estudio del diseño 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 caracter´ısticas de los datos a ser indexados. Se espera brindar nuevas herramientas de administra- ció n eficiente para las bases de datos métricas, que permitan que su desarrollo se acerque al que tienen los modelos tradicionales de base dedatos.

Se continuará analizando la expresividad de las distintas lógicas, para lograr caracterizar la clase de las consultas computables sobre bases de datos no convencionales. Además, se espera mejorar el desempeño de las operaciones de bajo nivel que realizan los DBMS, mediante la propuesta de una nueva arquitectura delprocesador.

Actividades de Formación

Formación de investigadores dentro de la línea:

Doctorado en Cs. de la Computació n: una tesis sobre expresividad de la lógica como lenguaje de consulta.

Maestría en Cs. de la Computación: una tesis sobre búsqueda por similitud aproximada 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 por similitud aproximadas en memoria secundaria.

Referencias

[1] J. Arroyuelo and J. M. Turull Torres. The existential fragment of third order logic and third order relational machines. In *Proc. del XX CACIC*, 324–333, 2014.

[2] G. Blaauw and F. Brooks, Jr. *Computer Architecture: Concepts and Evolution*. Addison-Wesley Longman Pub. Co., 1st edition, 1997.

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

[4] E. Chávez, M. Di Genaro, N. Reyes, and P. Roggero. Distal dynamic spatial approximation forest. In *XXII CACIC*, 804–813, 2016.

[5] E. Chávez, V. Ludueña, N. Reyes, and F. Kasián. Approximate nearest neighbor graph via index construction. In *Proc. del CACIC*, 824–833, 2016.

[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. Marroqu´ın. Searching in metric spaces. *ACM Comp. Surveys*, 33(3):273–321, 2001.

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

[9] A. Dawar. *Feasible Computation Through Model Theory*. PhD thesis, University of Pennsylvania, 1993. UMI Order No. GAX93-21378.

[10] A. Dawar. A restricted second order logic for finite structures. *Information and Compu- tation*, 143(2):154 –174, 1998.

[11] K. Figueroa, C. Mart´ınez, R. Paredes, N. Re- yes, and P. Roggero. Dynamic list of clustered permutations on disk. *Computer Science and Technology*, 201–211, 2016.

[12] G. Navarro and N. Reyes. Dynamic spatial approximation trees. *Journal of Exp. Algorithmics*, 12:1–68, 2008.

[13] G. Navarro and N. Reyes. New dynamic me- tric indices for secondary memory. *Informa- tion 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 WEA*, LNCS 4007, 85–97, 2006.

[15] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kauf- mann Publishers Inc., 2006.

[16] P. Zezula, G. Amato, V. Dohnal, and M. Bat- ko. *Similarity Search: The Metric Space Ap- proach*. Springer-Verlag New York, Inc., 2005.