Lenguajes y Operaciones para Bases de Datos no Convencionales

Jorge Arroyuelo, Susana Esquivel, Alejandro Grosso, Verónica Ludueña, Nora Reyes
Dpto. de Informática, Fac. de Cs. Físico-Matemáticas y Naturales, Universidad Nacional de San Luis
{ *bjarroyu, esquivel, agrosso, vlud, nreyes*}*@unsl.edu.ar*

Gonzalo Navarro
Departamento de Ciencias de la Computación, Universidad de Chile.
gnavarro@dcc.uchile.cl

Resumen

Los avances tecnológicos han producido una gran cantidad de información que debe ser almacenada y procesada. Los diferentes tipos y tamaños de datos provenientes de diversas fuentes como revistas, transacciones financieras, fotografías, música, etc., han dado lugar al desarrollo de depósitos no estructurados de información, *Bases de Datos no Convencionales*, en los que se almacenan y consultan nuevos tipos de datos (texto libre, imágenes, audio, vídeo, etc.). Estos grandes volúmenes de información exigen dispositivos de almacenamiento capaces de mantenerlos y además proveer un acceso eficiente y efectivo a los mismos. Esta nueva realidad requiere un modelo más general tal como las *Bases de Datos Métricas*, con un nivel de madurez similar al de las bases de datos tradicionales.

Por otro lado la necesidad de reducir la brecha entre los tiempos de CPU y los de I/O provocó el desarrollo de memorias más rápidas y de gran capacidad, promoviendo así la aparición de estructuras de datos que tienen en cuenta estas arquitecturas como las *estructuras de datos compactas* y las *estructuras de datos con I/O eficiente*. Nuestra investigación pretende contribuir a la madurez de este nuevo modelo de bases de datos.

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

Contexto

En el marco del Proyecto Consolidado 330303 "Tecnologías Avanzadas de Bases de Datos" se encuentra la línea *Bases de Datos no Convencionales*, la cual motiva esta presentación. Este proyecto pertenece a la Universidad Nacional de San Luis y se encuentra dentro del Programa de Incentivos a la Investigación (Código 22/F014). El ámbito de este proyecto ha permitido el estudio y tratamiento de objetos de diversos tipos, útiles en distintos campos de aplicación: sistemas de información geográfica, robótica, visión artificial, computación móvil, diseño asistido por computadora, motores de búsqueda en internet, computación gráfica, entre otras, y que se relacionan en tales bases de datos. Se consideran como actividades centrales de esta línea el análisis de distintos tipos de bases de datos, la investigación de aspectos empíricos, teóricos y aplicativos derivados de la administración de una base de datos que maneja tipos de datos no convencionales, la expresividad de los lenguajes de consulta, los operadores necesarios para responder consultas de interés, y también las estructuras y operaciones necesarias para responderlas eficientemente.

El contacto permanente con investigadores de otros países permite nuevas perspectivas en nuestras investigaciones. Esto ha sido posible gracias a la participación de nuestros integrantes en proyectos conjuntos de cooperación internacional con: Universidad de Chile, Universidad de Massey (Nueva Zelanda), Universidad Michoacana de San Nicolás de Hidalgo (México) y Universidade da Coruña (España).

Introducción

Para lograr aprovechar la información brindada por la diversidad de datos existentes, nos centraremos en aquellas estructuras capaces de manejar datos tales como: secuencias, textos, espacios métricos, entre otros. Sabiendo que las búsquedas exactas sobre estos datos carecen de sentido, se hace necesario un modelo más general, como el de *espacios métricos*, donde las *búsquedas por similitud*, más naturales sobre estos tipos de datos, son posibles.

Dado que la brecha entre los tiempos de CPU y los de I/O se ha mantenido creciente, se ha hecho cada vez más atractivo el uso de estructuras de datos que ocupen poco espacio. Esto puede lograrse comprimiendo la información sobre la que actúan. Si bien trabajar sobre esta información compacta es más la-

borioso, la aparición de nuevos niveles en la jerarquía de memoria (cachés de tamaño cada vez más considerable) la convierte en una alternativa ventajosa al poder mantenerla en una memoria más rápida, frente a las implementaciones clásicas. Este entorno da lugar a líneas de investigación que, conscientes de estas diferencias de costos, diseñan estructuras de datos más eficientes (en espacio, en I/O, u otras medidas de eficiencia) para memorias jerárquicas, usando la compacticidad o la I/O eficiente.

La "*maldición de la dimensionalidad*" describe el fenómeno por el cual el desempeño de los índices existentes se deteriora exponencialmente con la dimensión del espacio. Este fenómeno, presente en los espacios de vectores (representación común para datos multimedia), aún no está completamente analizado cómo afecta la dimensión a los índices para espacios métricos. De las numerosas estructuras que existen para búsquedas por similitud en espacios métricos, sólo unas pocas trabajan eficientemente en espacios de alta o mediana dimensión, y la mayoría no admiten dinamismo, ni están diseñadas para trabajar sobre conjuntos masivos de datos; es decir, en memoria secundaria. Por lo tanto, nos dedicamos a estudiar distintas maneras de optimizarlas.

Otros aspectos que se están investigando dentro de la línea son, por ejemplo, el dinamismo en una estructura, operaciones de búsqueda complejas, obtener una mayor expresividad en los lenguajes utilizados para expresar consultas y caracterizar la clase de consultas computables.

Líneas de Investigación y Desarrollo

Bases de Datos Métricas

Tomando como modelo para las bases de datos no convencionales a los espacios métricos, surge la necesidad de responder consultas por similitud eficientemente haciendo uso de *métodos de acceso métricos* (MAMs). En espacios métricos generales la complejidad usualmente se mide como el número de cálculos de distancias realizados. Por ello, se analizan aquellos MAMs que han mostrado buen desempeño en las búsquedas, con el fin de optimizarlos más, considerando la jerarquía de memorias.

Métodos de Acceso Métricos

El estudio del *Árbol de Aproximación Espacial* [11], que había mostrado un muy buen desempeño en espacios de mediana a alta dimensión, pero totalmente estático, nos permitió el desarrollo de un nuevo índice llamado *Árbol de Aproximación Espacial Dinámico (DSAT)* [12] que permite realizar inserciones y eliminaciones, conservando su buen desempeño en las búsquedas, lo cual es importante porque pocos índices son completamente dinámicos.

El *DSAT* es una estructura que realiza una partición del espacio considerando la proximidad espacial; pero, si el árbol agrupara los elementos que se encuentran muy cercanos entre sí, lograría mejorar las búsquedas al evitar recorrerlo para alcanzarlos. Podemos pensar entonces que construimos un *DSAT*, en el que cada nodo representa un grupo de elementos muy cercanos ("clusters") y relacionamos los clusters por su proximidad en el espacio. Cada nodo mantiene el centro del cluster correspondiente, y almacena los k elementos más cercanos a él; cualquier elemento a mayor distancia del centro que los k almacenados, pasa a formar parte de otro nodo en el árbol [2]. Sin embargo, falta aún analizar es cuán bueno es el agrupamiento o "clustering"que logra esta estructura, lo cual podría analizarse haciendo uso de nuevas estrategias de optimización de funciones a través de heurísticas bioinspiradas, que han mostrado ser útiles en detección de clusters.

Al trabajar sobre base de datos métricas, puede surgir la necesidad de hacer uso de la memoria secundaria. Es posible que la base de datos no pueda almacenarse en memoria principal por ser masiva o porque sus objetos son muy grandes, o que el índice no quepa en memoria principal, o ambas cosas. Por lo tanto, existe la necesidad de diseñar los índices especialmente para memoria secundaria. Así, en [13] se presentaron versiones preliminares del *DSAT* (*DSAT+* y *DSAT**) especialmente diseñadas para memoria secundaria: índices con buena ocupación de página y eficientes tanto en el número de cálculos de distancia y de operaciones de I/O para cada operación, y se están analizando variantes que mejoren aún más su desempeño. Además, se está adaptando para memoria secundaria la *Lista de Clusters* [3], por su buen desempeño en espacios de alta dimensión; pero volviéndola dinámica, lo que la haría aplicable a más situaciones reales en donde no se conoce de antemano la base de datos.

Join Métricos

El modelo de espacios métricos permite cubrir muchos problemas de búsqueda por similitud, aunque en general se deja fuera de consideración al operador de ensamble o "join" por similitud, otra primitiva importante [5].

De hecho, a pesar de la atención que esta primitiva ha recibido en las bases de datos tradicionales y

aún en las multidimensionales, no han habido grandes avances para espacios métricos generales. Nos hemos planteado resolver algunas variantes del problema de join por similitud: (1) *join por rango:* dadas dos bases de datos de un espacio métrico y un radio r, encontrar todos los pares de objetos (uno desde cada base de datos) a distancia a lo sumo r, (2) *k–pares más cercanos:* encontrar los k pares de objetos más cercanos entre sí (uno desde cada base de datos). Para resolver estas operaciones de manera eficiente hemos diseñado un nuevo índice métrico, llamado *Lista de Clusters Gemelos* (*LTC*) [14], éste se construye sobre ambas bases de datos conjuntamente, en lugar de indexar una o ambas bases de datos independientemente. Esta nueva estructura permite además resolver las consultas por similitud clásicas en espacios métricos sobre cada una de las bases de datos independientemente.

A pesar de que esta estructura ha mostrado ser competitiva y obtener buen desempeño en relación a las alternativas más comunes para resolver las operaciones de join, queda mucho por mejorar para que se vuelva una estructura práctica y mucho más eficiente para trabajar con grandes bases de datos métricas. A la fecha se está analizando la construcción de otra clase de índice basada en "permutantes" para resolver el join aproximado de dos bases de datos métricas; es decir que permita rápida y eficientemente encontrar los pares de elementos más similares entre ambas bases de datos, aunque no los obtenga a todos. Así sería posible extender apropiadamente el álgebra relacional como lenguaje de consulta y diseñar soluciones eficientes para nuevas operaciones, considerando aspectos de memoria secundaria, de concurrencia, de confiabilidad, etc. Algunos de estos problemas ya poseen solución en bases de datos espaciales, pero no en bases de datos métricas.

Lenguajes de Consulta

La relación existente entre lógica y teoría de bases de datos es muy estrecha y natural, ya que es posible pensar en una base de datos simplemente como una estructura finita, y utilizar las lógicas para expresar consultas sobre éstas. Esto les da una posición central como modelo computacional para el análisis del poder expresivo de los lenguajes de consultas que nos permiten obtener información de una base de datos, siendo relevante como marco teórico para el estudio de las bases de datos relacionales.

La mayoría de los lenguajes de consulta sobre bases de datos es equivalente, en su poder expresivo, a FO (First-Order logic). El principal problema es que la expresividad de FO no es lo suficientemente poderosa, porque no alcanza para reflejar ciertas consultas, como por ejemplo *clausura transitiva* o *paridad* (sobre el tamaño del dominio de una base de datos). Esto ha llevado a la búsqueda de una mayor expresividad por medio de diferentes mecanismos de extensión sobre FO utilizados como herramientas de construcción de lógicas más poderosas. Uno de ellos corresponde a la incorporación de cuantificadores que no pueden ser expresados en FO, como *clausura transitiva* y *punto fijo*, entre otros, los que han sido ampliamente estudiados. La idea de agregar cuantificadores es generalizada mediante la noción de *cuantificadores generalizados de Lindström* (ver [7]). Aún así, estas lógicas todavía resultan incompletas, por lo que se analizan lógicas de orden superior, SO (Second-Order Logic), y algunos de sus fragmentos (restricciones) que han demostrado poseer propiedades interesantes sobre las estructuras finitas. Un resultado importante de R. Fagin fue la caracterización de $SO\exists$ (fragmento existencial de SO) [6]. Allí se establece que las propiedades de las estructuras finitas que son definidas por sentencias existenciales de segundo orden coinciden con las propiedades que pertenecen a la clase de complejidad NP, lo cual fue extendido por Stockmeyer [15], estableciendo una relación cercana entre la lógica SO y la jerarquía de tiempo polinomial (PH).

Actualmente existen muchos resultados igualando la expresividad lógica a la complejidad computacional, pero requieren estructuras ordenadas (ver [8], [9]). Estas relaciones entre la complejidad computacional (cantidad de recursos necesarios para resolver un problema sobre algún modelo de máquina computacional) y la complejidad descriptiva (el orden de la lógica que se necesita para describir el problema), han llevado a que los resultados obtenidos en alguno de estos campos sea transferido de manera inmediata al otro.

En uno de nuestros trabajos de investigación se ha introducido la definición de una restricción de SO, que consiste en limitar las relaciones que pueden tomar los cuantificadores de SO, considerando a la lógica como uno de los lenguajes de consulta a base de datos. El tipo de relaciones a los que estos cuantificadores pueden referirse son relaciones cerradas bajo $FO-type$. Esta lógica(SOF) es un intento de lograr una lógica de mayor poder expresivo que la lógica definida por Dawar(SO^w) en la que los cuantificadores sólo pueden tomar relaciones cerradas bajo $FO-k$ tipos. Se demostró que nuestra lógi-

ca incluye estrictamente la definida por Dawar [4]. Se ha podido definir una nueva clase de complejidad descriptiva (NPF), que caracteriza el fragmento existencial de nuestra lógica mediante una modificación de las máquinas relacionales.

En otro de nuestros trabajos se estudia el impacto que ocasiona el aumento del orden de las variables en las lógicas. Se continúa con el estudio del poder expresivo de las lógicas HO (High-Order logic) y en particular de los fragmentos de la lógica VO (Variable-Order logic) definida en [10], que nace debido a que ninguna de las lógicas de orden superior cubre la clase completa de consultas computables (CQ)[1], es decir que no son completas. Además, si consideramos la unión de todas las lógicas de orden superior, es decir $\bigcup_{i\geq 2} HO^i$ (HO^i representa la lógica de orden i), tampoco obtenemos una lógica completa. De aquí, se define VO permitiendo el uso de variables de orden variable, mediante el uso de cuantificadores de orden. Las restricciones más importantes sobre VO que se estudian aquí son: sobre la cantidad de alternaciones de cuantificadores, sobre la aridad de las variables de orden variable, sobre los valores que pueden asignarse a las variables de orden en función del tamaño del dominio, sobre el rango de cuantificadores y restricciones sobre la cantidad de variables, de valuación y de orden.

Bases de Datos de Texto

El *texto* es uno de los tipos de datos sobre los que se está trabajando en esta línea. Este tipo de dato debe ser manejado en forma adecuada para lograr un ahorro en el espacio de almacenamiento y a la vez permitir que los accesos a él sean eficientes.

Una herramienta clave para el manejo eficiente de grandes cantidades de texto son los índices textuales, que proveen un rápido acceso al mismo permitiendo buscar en él. Una variante de los índices clásicos son los llamados índices *compactos* que almacenan sus datos y el texto en forma eficiente tratando de ocupar el menor espacio posible y de aprovechar la existencia de la jerarquía de memorias, como también responder a consultas en forma rápida.

Entre las consultas que se realizan sobre el texto encontramos las búsquedas por similitud; este tipo de búsqueda tiene aplicaciones tales como la biología computacional, comunicaciones de datos, data mining, bases de datos textuales, recuperación de errores (en reconocimiento óptico de caracteres, spelling), entre otras. El problema general de la *búsqueda aproximada* puede verse como: sea $T = T[1, n]$ un *texto*, y $P = P[1, m]$ un *patrón* sobre el alfabeto Σ (con $m << n$) y un entero k, se desea encontrar y devolver todos los *substrings* en el texto T que sean una ocurrencia aproximada de P, con a lo más k diferencias. La diferencia entre dos strings α y β ($d(\alpha, \beta)$) se obtiene con la *distancia de edición* o *distancia de Levenshtein* d (mínimo número de inserciones, eliminaciones y/o sustituciones de caracteres a realizar para convertir β en α).

Los índices compactos se diferencian de la compresión pura en su capacidad de manipular los datos en forma comprimida, es decir, sin tener que descomprimirlos primero; se puede operar sin descomprimir los datos. En la actualidad los índices compactos pueden manipular secuencias de bits o de símbolos generales, grafos, colecciones de texto, sumas parciales, árboles en general, búsqueda por rango en una y más dimensiones, permutaciones y mapping, etc. La variedad de índices comprimidos existentes es grande y entre ellos encontramos los basados en listas de ocurrencia de q-gramas.

Un índice de q-gramas es una estructura de datos que permite encontrar rápidamente en el texto todas las ocurrencias de un q-grama dado (subsecuencia de tamaño q de un texto). Existen distintas implementaciones de índices para q-gramas. La básica es un arreglo de punteros de tamaño $|\Sigma|^q$. Cada posición del arreglo referencia a la lista de ocurrencias en el texto del q-grama correspondiente. Otra perspectiva usa una estructura de trie construido con los distintos q-gramas que aparecen en el texto. Cada hoja del trie contiene un puntero a la lista de ocurrencias del correspondiente q-grama en el texto. Las implementaciones anteriores encuentran todas las ocurrencias de un q-grama dado en tiempo óptimo en función del número de ocurrencias encontradas, pero sufren del mismo inconveniente: el tamaño del índice se vuelve impráctico al crecer la longitud del texto.

Existe un índice para q-gramas que comprime las listas de ocurrencias de q-gramas. En él, la estructura trie para los distintos q-gramas, llamada *índice primario*, es aún necesaria para proveer un punto de comienzo para las búsquedas. Éstas pueden dividirse en dos etapas: seleccionar las regiones del texto en las que podrían estar presentes todos los q-gramas del patrón, y verificar la existencia del patrón buscado en las regiones de texto seleccionadas. Durante la implementación del índice se pueden analizar distintas técnicas de compresión de las listas de ocurrencias de q-gramas seleccionando la mejor.

1. Resultados y Objetivos

Como trabajo futuro de esta línea de investigación se consideran varios aspectos relacionados al diseño de estructuras de datos que, consciente de la jerarquía de memorias y de las características particulares de los datos a ser indexados, saquen el mejor partido haciendólas eficientes en espacio y en tiempo. Se trabajará con estructuras de datos compactas para textos, implementando un índice basado en q-gramas y estudiando su comportamiento respecto de los mejores auto-índices del sitio *Pizza&Chili*.

Respecto de los lenguajes de consulta se continuará analizando la expresividad de distintas extensiones de FO y posibles restricciones de SO, para lograr caracterizar la clase de las consultas computables sobre bases de datos no convencionales.

En el caso de bases de datos métricas, se intentará que los índices se adapten mejor al espacio métrico particular considerado, gracias a la determinación de su dimensión intrínseca, y también al nivel de la jerarquía de memorias en que se deba almacenar. Es importante destacar que estos estudios sobre espacios métricos y sobre algunas estructuras de datos particulares permitirán no sólo mejorar el desempeño de las mismas sino también aplicar, eventualmente, muchos de los resultados que se obtengan a otros índices para bases de datos métricas.

2. Actividades de Formación

Dentro de esta línea de investigación se están formando docentes-investigadores de acuerdo al siguiente detalle:

Tesis de Doctorado en Ciencias de la Computación: dos integrantes de la línea se encuentran desarrollando su tesis sobre la expresividad de la lógica como lenguaje de consulta. Otro integrante desarrolla su tesis sobre bases de datos métricas.

Tesis de Maestría en Ciencias de la Computación: un investigador de la línea está desarrollando su tesis en la temática bases de datos de texto sobre búsqueda en texto utilizando índices comprimidos.

Además, se están dirigiendo actualmente tres trabajos finales de alumnos de la Licenciatura en Ciencias de la Computación.

Referencias

[1] D. Harel A. K. Chandra. Computable queries for relational data bases. *Journal of Computer ans System Sciences*, 21(2):156–178, 1980.

[2] M. Barroso, N. Reyes, and R. Paredes. Enlarging nodes to improve dynamic spatial approximation trees. In *Procs. of the 3rd Int. Conf. on Similarity Search and Applications*, 41–48. ACM Press, 2010.

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

[4] A. Dawar. A restricted second order logic for finite structures. *Information and Computation*, 143:154–174, 1998.

[5] V. Dohnal, C. Gennaro, P. Savino, and P. Zezula. Similarity join in metric spaces. In *Proc. 25th European Conf. on IR Research*, LNCS 2633, pages 452–467, 2003.

[6] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of Computation*, 7:43–73, 1974.

[7] J. Flum. H. Ebbinghaus. Finite model theory, second edition. *Springer*, 1999.

[8] N. Immerman. Descriptive and computational complexity. *Computational Complexity Theory*, 38:75–91, 1989.

[9] N. Immerman. Descriptive complexity. *Springer*, 1998.

[10] J. M. Turull Torres L. Hella. Computing queries with higher-order logics. *Theoretical Computer Science*, 355:197–214, 2006.

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

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

[13] G. Navarro and N. Reyes. Dynamic spatial approximation trees for massive data. In *Procs. of the 2nd Int. Conf. on Similarity Search and Applications*, 81–88. IEEE Comp. Society, 2009.

[14] R. Paredes and N. Reyes. Solving similarity joins and range queries in metric spaces with the list of twin clusters. *J. of Discrete Algorithms*, 7(1):18–35, 2009.

[15] L. Stockmeyer. The polynomial-time hierarchy. *Theoret. Comput. Sci.*, 3:1–22, 1976.