Indexación y Búsqueda sobre Datos no Estructurados

Norma Herrera, Dar´ıo Ruano, Paola Azar, Susana Esquivel
Departamento de Informática
Universidad Nacional de San Luis, Argentina
{nherrera, dmruano, epazar, esquivel}@unsl.edu.ar

Anabella De Battista, Andrés Pascal
Departamento Ingeniería en Sistemas de Información
FRCU, Universidad Tecnológica Nacional
Entre R´ıos, Argentina
{anadebattista, andrespascal22}@gmail.com

Abstract

Las bases de datos actuales han incluido la capacidad de almacenar datos no estructurados tales como imágenes, sonido, texto, video, etc. La problemática de almacenamiento y búsqueda en estos tipos de base de datos difiere de las bases de datos clásicas, dado que no es posible organizarlos en registros y campos, y aun cuando pudiera hacerse, la búsqueda exacta carece de interés. Es en este contexto donde surgen nuevos modelos de bases de datos capaces de cubrir las necesidades de almacenamiento y búsqueda de estas aplicaciones. Nuestro interés se basa en el diseño de ´ındices eficientes para estas nuevas bases de datos.

1 Contexto

El presente trabajo se desarrolla en el ámbito de la línea Técnicas de Indexació n para Datos no Estructurados del Proyecto Tecnolog´ıas Avanzadas de Bases de Datos (22/F414), cuyo objetivo es realizar investigación básica en problemas relacionados al manejo y recuperació n eficiente de informació n no tradicional.

2 Introducción

La mayor´ıa de los administradores de bases de datos actuales están basados en el modelo relacional, presentado por Edgard F. Codd en 1970. Bajo el modelo relacional, cada elemento de la base de datos puede ser almacenado como un registro (tupla) y cada registro a su vez dividido en campos (atributos). La mayor´ıa de las consultas que se realizan a una base de datos relacional (conocidas también como bases de datos tradicionales) se corresponden con *búsquedas exactas*, esto significa obtener todos los registros cuyos campos coinciden exactamente con los campos aportados durante la búsqueda. También se pueden realizar búsquedas por rango sobre valores numéricos, y búsquedas de sub-cadenas sobre campos alfabéticos; en estos casos debe existir una relación de orden sobre los campos consultados.

La información disponible en formato digital aumenta día a día su tamaño de manera ex-

ponencial. Gran parte de esta información involucra el uso de datos no estructurados tales como imágenes, sonido, texto, video, etc. Debido a que no es posible organizar estos tipos de datos en registros y campos, las tecnolog´ıas tradicionales de bases de datos para almacenamiento y búsqueda de información no son adecuadas en este ámbito.

Es en este contexto donde surgen nuevos modelos de bases de datos capaces de cubrir las necesidades de almacenamiento y búsqueda de estas aplicaciones. Nuestro interés se basa en el diseño de índices para estas nuevas bases de datos, centrándonos en bases de datos textuales y en espacios métricos.

Bases de Datos Textuales (BDT) Una base de datos de texto es un sistema que mantiene una colección grande de texto, y provee acceso rápido y seguro al mismo. Sin pérdida de generalidad, asumiremos que la base de datos de texto es un único texto T posiblemente almacenado en varios archivos. Las búsquedas en la que el usuario ingresa un *patrón de búsqueda* y el sistema retorna todas las posiciones del texto donde el patrón ocurre, es una de las búsquedas más comunes en este tipo de bases de datos.

Mientras que en bases de datos tradicionales los ´ındices ocupan menos espacio que el conjunto de datos indexado, en las bases de datos de texto el índice ocupa más espacio que el texto, pudiendo necesitar de 4 a 20 veces el tamaño del mismo [8, 14]. Una alternativa para reducir el espacio ocupado por el ´ındice es buscar una representación compacta del mismo, manteniendo las facilidades de navegación sobre la estructura. Pero en grandes colecciones de texto, el índice aún comprimido suele ser demasiado grande como para residir en memoria principal [9, 10]. Por esta razón, el estudio de ´ındices comprimidos y en memoria secundaria para búsquedas en texto es un tema de creciente interés en la comunidad de bases de datos.

Espacios Métricos El modelo de espacios métricos permite formalizar el concepto de búsqueda por similitud en bases de datos no tradicionales [4].

Un espacio métrico está formado por un conjunto de objetos $\mathcal{X}$ y una función de distancia d definida entre ellos que mide cuan diferentes son. La base de datos sera´ un subconjunto finito $U \subseteq \mathcal{X}$.

Una de las consultas más comunes en este modelo de bases de datos es la *búsqueda por rango*. En esta búsqueda dado un elemento $q \in \mathcal{X}$, al que llamaremos *query* y un radio de tolerancia r, la búsqueda por rango consiste en recuperar los objetos de la base de datos cuya distancia a q no sea mayor que r. Para evi- tar examinar exhaustivamente la base de datos, se preprocesa la misma por medio de un *algoritmo de indexación* con el objetivo de construir una *índice*, diseñado para ahorrar cálculos en el momento de la búsqueda. En [4] se presenta un desarrollo unificador de las soluciones existentes en la temática.

Bases de datos temporales

En las bases de datos temporales [19, 12] los datos asociados al tiempo forman parte de la relevancia de sus registros. El modelo temporal permite almacenar y recuperar datos que dependen del tiempo. Mientras que las bases de datos tradicionales tratan al tiempo como otro tipo de dato más, este modelo incorpora al tiempo como una dimensión. Se distinguen dos tipos de tiempo, el *tiempo válido*, y el *tiempo transaccional*. El tiempo válido es el per´ıodo en el cual un hecho existe y el tiempo transaccional es el periodo en el cual el he- cho es registrado en la base de datos. Estos tiempos no necesariamente tiene que coincidir, por ejemplo, algunos determinados sucesos del siglo XX pueden haberse ingresado a unabase de datos durante el siglo XXI.

Bases de datos métrico-temporales (BDMT) Este modelo permite almacenar objetos no es- tructurados con tiempos de vigencia asocia- dos y realizar consultas por similitud y por tiempo en forma simultánea. Formalmente un *Espacio Métrico-Temporal* es un par (U, d), donde $U = O \times N \times N$, y la función d es de la forma $d : O \times O \rightarrow R^+$. Cada elemento $u \in U$ es una triupla (obj, t_i, t_f), donde obj es un objeto (por ejemplo, una imagen, sonido, cadena, etc) y $[t_i, t_f]$ es el intervalo de vigen- cia de obj. La función de distancia d, que mide la similitud entre dos objetos, cumple con las propiedades de una métrica (positivi- dad, simetr´ıa y desigualdad triangular).

Un nuevo tipo de consulta son las denomi- nadas métrico-temporales que se definen for- malmente en s´ımbolos como:

$(q, r, t_{iq}, t_{fq})_d = \{o / (o, t_{io}, t_{fo}) \in X \wedge d(q, o) \leq r \wedge (t_{io} \leq t_{fq}) \wedge (t_{iq} \leq t_{fo})\}$

La consulta implica buscar todos los objetos o de la parte finita X del universo U que estén a una distancia a lo más r de q, y que su tiempo asociado t coincida(osesolape) conentiempo de la consulta.

Varios índices métrico-temporales se han propuesto en este ámbito, todos estos índices fueron desarrollados para ser eficientes en memoria principal.

3 Líneas de Investigación

3.1 Indices Comprimidos en Memoria Secundaria para BDT

Como ya mencionamos, el principal proble- ma que surge al indexar una bases de datos de texto es el espacio ocupado por el ´ındice.

Una forma de tratar con este problema es buscar una representación compacta del ´ındice, manteniendo las facilidades de nave- gació n sobre la estructura. Esto significa en- contrar una representación que ocupe menos espacio que la representación clásica, pero que permita navegar sobre el ´ındice sin necesidad de descomprimirlo [6, 7, 9, 10, 13, 15, 16, 18]. Un ***trie de sufijos*** es un ´ındice que per- mite resolver eficientemente las operaciones de búsquedas en texto pero que necesita en es- pacio 10 veces el tamaño del texto indexado. En [17] se presenta una nueva representació n de un trie de sufijos que permite reducir el espacio necesario para almacenar el ´ındice, eliminando la necesidad de mantener los pun- teros explícitos a los hijos. Esta representación surge como una extensió n a árboles r-arios de la técnica presentada en [11] y tiene la ventaja de permitir un posterior proceso de paginado para manejar eficientemente el trie de sufijos en memoria secundaria [20].

Hemos realizado una implementación que mejora en espacio a la anterior en un 40%, sin afectar los tiempos de búsqueda. Esta nueva versión compacta del trie de sufijos consiste en usar códigos *DAC* (*Directly Addressable Variable-Length Code* [3]), para los arreglos que representan la secuencia de saltos y de grados. La navegación sobre esta nueva re- presentació n sigue los lineamientos generales propuestos en [17], adaptándolo a los códigos *DAC*.

Estamos trabajando en integrar esta nueva representación con la técnica de paginado propuesta en [17], a fin de lograr un ´ındice comprimido en memoria secundaria. Nos en- contramos en la etapa de implementación de esta nueva propuesta.

3.2 Indices en Memoria Secundaria para BDMT

Varios índices métrico-temporales se han propuesto en este ámbito, todos estos índices fueron desarrollados para ser eficientes en memoria principal; dos de ellos son el $H - FHQT$ [5] y el $NewH - FHQT$ [2].

El H-FHQT consiste en una lista de los in- stantes válidos de tiempo, donde cada celda de la lista contiene un ´ındice FHQT [1] con el que indexa todos los objetos vigentes en dicho instante. Esta estructura es eficiente en bases de datos métrico-temporales donde los objetos tienen vigencia en un sólo instante de tiempo. El New H-FHQT está basado también en el uso del FHQT como estructura métrica y el en- foque temporal se ha abordado mediante el uso de una l´ınea de tiempo, del mismo modo que en el H-FHQT.

Este ´ındice consiste en una lista compuesta por los instantes válidos de tiempo. Para cada instante de la lista que posee obje- tos vigentes, se construye un FHQT para in- dexar los objetos. La principal diferencia con el ´ındice antes propuesto se da en la etapa de construcción de los FHQTs. En este caso se van tomando los primeros pivotes disponibles de la lista, que se considera una lista circu- lar, de tal manera que el FHQT del instante i, este´construido con pivotes diferentes a los de los instantes $i - 1$ e $i + 1$. La construcción de FHQTs consecutivos con diferentes grupos de pivotes da a la estructura mayor poder de filtrado de elementos desde el punto de vista métrico. Esta idea se plantea debido a que los objetos a indexar tienen un intervalo de vigencia asociado, por lo que pueden estar presentes en varios FHQTs consecutivos. Con este enfoque se logra que los objetos pasen por varios filtros, se eliminen la mayor cantidad de objetos mediante la firma y la desigualdadtriangular y se reduzcan as´ı la cantidad necesaria de evaluaciones de la función de distancia al momento de la ejecución de la consulta.

Estos ´ındices se desarrollaron bajo el supuesto de que la memoria principal tienecapacidad suficiente como para mantener tanto el ´ındice como la base de datos. En este contexto nuestro objetivo es adecuarlos para que los mismos también resulten eficientes en memoria secundaria. Cabe señalar que no existe hasta el momento ningún índice en memoria secundaria para este tipo de base de datos.

4 Resultados Esperados

Se espera obtener ´ındices eficientes, tanto en espacio como en tiempo, para el procesamiento de consultas en bases de datos textuales y en espacios métricos. Los mismos serán evaluados tanto analíticamente como emp´ıricamente.

5 Recursos Humanos

El trabajo desarrollado en esta l´ınea forma parte del desarrollo de un Trabajo Final de la Licenciatura, dos Tesis de Maestr´ıa y una Tesis de Doctorado, todas ellas en el área temática de Ciencias de la Computación, en la Univer- sidad Nacional de San Luis.

References

[1] R. Baeza-Yates, W. Cunto, U. Manber, and S. Wu. Proximity matching using fixed-queries trees. In *Proc. 5th Combinatorial Pattern Matching (CPM'94)*, LNCS 807, pages 198–212, 1994.

[2] A. De Battista, A. Pascal, N. Herrera, and G. Gutierrez. Metric-temporal access methods. *Journal of Computer Science & Technology,*, 10(2):54–60, 2010.

[3] Nieves R. Brisaboa, Susana Ladra, and Gonzalo Navarro. Directly addressable variable-length codes. In *SPIRE*, pages 122–130, 2009.

[4] E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroqu´ın. Searching in met- ric spaces. *ACM Computing Surveys*, 33(3):273–321, September 2001.

[5] A. De Battista, A. Pascal, G. Gutierrez, and N. Herrera. Un nuevo índice métrico-temporal: el historical-fhqt. In *Actas del XIII Congreso Argentino de Ciencias de la Computació n*, Corrientes, Agentina, 2007.

[6] P. Ferragina and G. Manzini. Indexing compressed text. *J. ACM*, 52(4):552–581, 2005.

[7] P. Ferragina, G. Manzini, V. Mäkinen, and G. Navarro. Compressed representations of sequences and full-text indexes. *ACM Trans. Algorithms*, 3(2):20, 2007.

[8] G. H. Gonnet, R. Baeza-Yates, and T. Snider. *New indices for text: PAT trees and PAT arrays*, pages 66–82. Prentice Hall, New Jersey, 1992.

[9] R. González and G. Navarro. A compressed text index on secondary memory. In *Proc. 18th International Workshop on Combinatorial Algorithms (IWOCA)*, pages 80–91. College Publications, UK, 2007.

[10] R. González and G. Navarro. Compressed text indexes with fast locate. In *Proc. 18th Annual Symposium on Combinatorial Pattern Matching (CPM)*, LNCS 4580, pages 216–227, 2007.

[11] N. Herrera and G. Navarro. Árboles de sufijos comprimidos en memoria secundaria. In *Proc. XXXV Latin American Conference on Informatics (CLEI)*, Pelotas, Brazil, 2009.

[12] C. S. Jensen. A consensus glossary of temporal database concepts. *ACM SIGMOD Record*, 23(1):52–54, 1994.

[13] V. Mäkinen and G. Navarro. *Compressed Text Indexing*, pages 176–178. Springer, 2008.

[14] U. Manber and G. Myers. Suffix ar- rays: A new method for on-line string searches. *SIAM Journal of Computing*, 22(5):935–948, 1993.

[15] G. Navarro. Indexing text using the ziv-lempel trie. *Journal of Discrete Algorithms (JDA)*, 2(1):87–114, 2004.

[16] G. Navarro and K. Sadakane. *Compressed Tree Representations*. Springer, 2nd edition, 2015.

[17] D. Ruano and N. Herrera. Repre- sentació n secuencial de un trie de sufi- jos. In *XX Congreso Argentino de Cien- cias de la Computació n*, Buenos Aires, Argentina, 2014.

[18] K. Sadakane. New text indexing functionalities of the compressed suffix arrays. *J. Algorithms*, 48(2):294–313, 2003.

[19] B. Salzberg and V. J. Tsotras. A comparison of access methods for temporal data. *ACM Computing Surveys*, 31(2), 1999.

[20] J. Vitter. External memory algorithms and data structures: Dealing with mas- sive data. *ACM Computing Surveys*, 33(2):209–271, 2001.