Indexando Objetos Métrico-Temporales

Anabella De Battista , Andrés Pascal
Departamento de Sistemas de Información
Universidad Tecnológica Nacional
Fac. Reg. Concepción del Uruguay
Entre Ríos, Argentina
{debattistaa, pascalj}@frcu.utn.edu.ar

Norma Edith Herrera
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Argentina
nherrera@unsl.edu.ar

Gilberto Gutierrez
Facultad de Ciencias Empresariales
Universidad del Bio-Bio
Chillán, Chile
ggutierr@ubiobio.cl

Resumen

El modelo de bases de datos métrico-temporal permite abordar aquellas situaciones en las que resulta necesario realizar búsquedas por similitud teniendo en cuenta también la componente temporal. En este modelo se combinan los espacios métricos con las bases de datos temporales permitiendo así procesar consultas por similitud restringidas a un intervalo o a un instante de tiempo. Varios índices han sido propuestos para procesar eficientemente consultas métrico-temporales, los cuales han demostrado ser competitivos en memoria principal. En este trabajo estamos interesado en el diseño de índices eficientes para el procesamiento de consultas métricos temporales, considerando también el desempeño de los mismos en memoria secundaria.

Palabras Claves: Espacios Métricos, Bases de Datos Temporales, Bases de Datos Métrico-Temporales, Índices

1. Contexto

El presente trabajo se desarrolla en el ámbito del Grupo de Investigación en Bases de Datos (Proy. Nro 25-D040) perteneciente al Departamento de Sistemas de la Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay, cuyo objetivo principal es el estudio de métodos de acceso, procesamiento de consultas y aplicaciones de bases de datos no tradicionales.

2. Introducción

Las operaciones de búsquedas en una base de datos requieren de algún soporte y organización especial a nivel físico. En el caso de las bases de datos clásicas, la organización de la información se basa en el concepto de búsqueda exacta sobre datos estructurados. Esto significa que la información se organiza en registros con campos completamente comparables. Una búsqueda en la base retorna todos aquellos registros cuyos campos coinciden con los apor-

tados en la consulta (búsqueda exacta). Otra característica importante de las bases de datos clásicas es que capturan sólo un estado de la realidad modelizada, usualmente el más reciente. Por medio de las transacciones, la base de datos evoluciona de un estado al siguiente descartando el estado previo.

Actualmente las bases de datos han incluido la capacidad de almacenar otros tipos de datos tales como imágenes, sonido, texto, video, datos geométricos, etc. La problemática de almacenamiento y búsqueda en estos tipos de base de datos difiere notablemente de las bases de datos clásicas en tres aspectos: primero los datos no son estructurados, esto significa que es imposible organizarlos en registros y campos, segundo la búsqueda exacta carece de interés y tercero resulta de interés mantener todos los estados de la base de datos y no sólo el más reciente a fin de poder consultar el instante o intervalo de tiempo de vigencia de dichos objetos. Como solución a esta problemática surgen modelos que permiten procesar esta clase de datos. Entre estos nuevos modelos encontramos los siguientes:

Espacios métricos [1, 4], que permiten almacenar objetos no estructurados y realizar búsquedas por similitud sobre los mismos. Un espacio métrico es un par (U, d) donde U es un universo de objetos y $d : U \times U \rightarrow R^+$ es una función de distancia definida entre los elementos de U que mide la similitud entre ellos. Una de las consultas típicas en este nuevo modelo de bases de datos es la búsqueda por rango, denotado por $(q, r)_d$, que consiste en recuperar los objetos de la base de datos que se encuentren como máximo a distancia r de un elemento q dado.

Bases de datos temporales [8, 5], que incorporan al tiempo como una dimensión, por lo que permiten asociar tiempos a los datos almacenados. Existen tres clases de bases de datos temporales en función de la forma en que manejan el tiempo: *de tiempo transaccional (transaction time)*, donde el tiempo se registra de acuerdo al orden en que se procesan las transacciones; *de tiempo vigente*, que almacenan el momento en que el hecho ocurrió en la realidad (puede no coincidir con el momento de su registro) y *bitemporales*, que integran la dimensión transaccional y la dimensión vigente a través del versionado de los estados, es decir, cada estado se modifica para actualizar el conocimiento de la realidad pasada, presente o futura, pero esas modificaciones se realizan generando nuevas versiones de los mismos estados.

Bases de datos métrico-temporales (BDMT) [6, 7], que permiten almacenar objetos no estructurados con tiempos de vigencia asociados 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 vigencia de obj. La función de distancia d, que mide la similitud entre dos objetos, cumple con las propiedades de una métrica (positividad, simetría y desigualdad triangular). Como un ejemplo de aplicación podemos mencionar una base de datos de rostros de delincuentes y cada foto tiene una intervalo de vigencia asociado, que representa el intervalo de tiempo en que el delincuente tenía el aspecto representado en esa foto; en este caso sería de interés, dada una foto y un intervalo de tiempo, poder recuperar de la base todos aquellos rostros parecidos al dado en el intervalo de tiempo especificado. Formalmente una *consulta métrico-temporal* se define como una 4-upla $(q, r, t_{iq}, t_{fq})_d$, tal que $(q, r, t_{iq}, t_{fq})_d = \{o/(o, t_{io}, t_{fo}) \in X \wedge d(q, o) \leq r \wedge (t_{io} \leq t_{fq}) \wedge (t_{iq} \leq t_{fo})\}$

3. Índices en BDMT

Una forma trivial de resolver una consulta métrico-temporal, sin realizar un barrido secuencial sobre todos los elementos de la bases de datos, es construir un índice métrico agregándole a cada objeto el intervalo de tiempo de vigencia del mismo. Luego, ante una consulta $(q, r, t_{iq}, t_{fq})_d$ primero se utiliza el índice métrico para descarta aquellos objetos obj que están a distancia mayor que r de q; posteriormente se realiza un barrido secuencial sobre el conjunto de elementos no descartados por el paso anterior a fin de determinar cuáles objetos son realmente respuesta a la consulta, es decir, cuáles tienen un intervalo de vigencia que se superpone con $[t_{iq}, t_{fq}]$.

La desventaja que tiene esta solución trivial es que no se usa la componente temporal para mejorar el filtrado en el índice; en este proceso sólo se aprovecha la componente métrica. Una mejor estrategia es que durante el proceso de búsqueda se utilice tanto la componente métrica como la componente temporal para descartar elementos.

Varios índices métrico-temporales se han propuesto en este ámbito. El más reciente es el *Pivot-FHQT* propuesto en [6] como una mejora del *Historical-FHQT*[2]. Ambos toman como base el *Fixed Height Queries Tree*[1], un índice para espacios métricos.

El *Fixed-Height FQT* (FHQT) construye un árbol a partir de un elemento p (pivote) que puede ser elegido arbitrariamente, o mediante algún procedimiento de selección de pivotes [3], del universo U. Para cada distancia i se crea el conjunto C_i formado por todos aquellos elementos de la base de datos que están a distancia i de p. Luego, para cada C_i no vacío se crea un hijo del nodo correspondiente a p, con rótulo i, y se construye recursivamente un FHQT teniendo en cuenta que todos los subárboles del mismo nivel usarán el mismo pivote como raíz. Este proceso recursivo se continúa hasta lograr que todas las hojas estén en un mismo nivel y tengan menos de b elementos, siendo b un valor fijado previamente. La figura 1 muestra un ejemplo de un FHQT conjunto de 15 elementos en los que se ha elegido u_{11} como pivote en el primer nivel y u_5 como pivote del segundo nivel. Ante una consulta $(q, r)_d$, se comienza por la raíz y se descartan todas aquellas ramas con rótulo i tal que $i \notin [d(p, q) - r, d(p, q) + r]$ siendo p el pivote utilizado en la raíz. La búsqueda continúa recursivamente en todos aquellos subárboles no descartados, utilizando el mismo criterio.

El *Historical-FHQT* (H-FHQT) consiste en una lista de instantes válidos donde cada uno contiene un FHQT correspondiente a todos los objetos vigentes en dicho instante. Esta estructura es eficiente en bases de datos métrico-temporales en las que los objetos tienen vigencia en un solo instante de tiempo. Los FHQT tienen distintas profundidades en función de la cantidad de elementos que deban indexar. La cantidad de pivotes utilizada en un árbol se calcula como $\lceil log_2(|o_i|) \rceil$ donde $|o_i|$ es la cantidad de objetos vigentes en el instante i. De esta manera se evita que haya árboles con mayor profundidad de la necesaria, con el fin de que la estructura no tenga un costo excesivo en almacenamiento. Las consultas métrico-temporales se efectúan de la siguiente manera: en primer lugar se seleccionan los instantes incluidos en el intervalo de consulta. Luego se realizan consultas por similitud usando cada uno de los FHQT correspondientes, y finalmente se unen los conjuntos resultantes.

Si bien en un H-FHQT la cantidad de pivotes en distintos instantes de tiempo varía, siempre se trabaja sobre el mismo conjunto base de pivotes; esto significa que si en el instante i necesito k_i pivotes y en el instante j necesito k_j pivotes con $k_i < k_j$, entonces los primeros k_i pivotes de ambos instantes son iguales. Con esto se evita que una query deba compararse con grupos distintos de pivotes en distintos instantes, lo que implicaría mayor cantidad de evaluaciones de la función de distancia d para poder calcular la firma de q en cada instante.

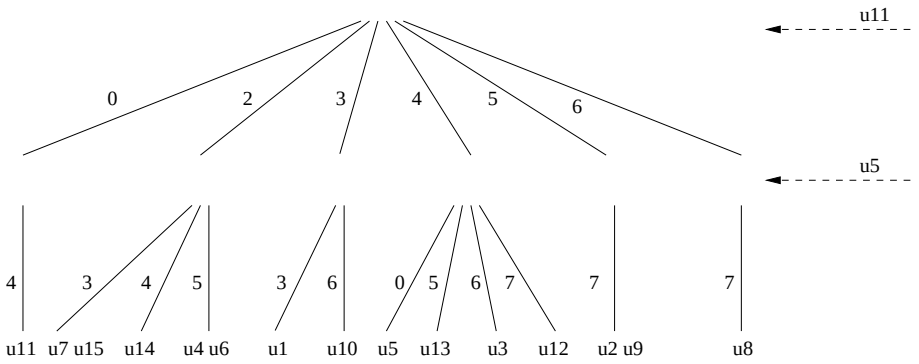

Figura 1: *Un ejemplo de un FHQT sobre un conjunto de 15 elementos*

Pero supongamos que un objeto o que no pertenece al resultado de una consulta $(q, r, t_{iq}, t_{fq})_d$, está vigente en varios instantes de tiempo i que pertenecen al intervalo de la consulta. Si el objeto o no pudo ser descartardo por el *fhqt*$_i$, entonces la única posibilidad de que sea descartado por el *fhqt*$_{i+1}$ es que k_{i+1} sea mayor que k_i y que esos pivotes adicionales del *fhqt*$_{i+1}$ logren eliminar a o (recordar la regla de eliminación del algoritmo de búsqueda). Esto significa que la capacidad de filtrado del *fhqt*$_{i+1}$ frente a ese objeto o se reduce a la capacidad de filtrado de los pivotes adicionales, si es que éstos existen.

Lo anterior nos indica que usar conjuntos disjuntos de pivotes para *fhqt* consecutivos, si bien aumenta la cantidad de evaluaciones de distancias al momento de calcular la firma de la query q, también aumenta la probabilidad de disminuir la cantidad de falsos positivos en la lista de candidatos con los que deberá compararse q. Estas ideas fueron la base que permitieron diseñar el Pivot-FHQT que empíricamente mostró ser más competitivo que su antecesor, el H-FHQT, en la totalidad de las consultas por intervalo y en la mayoría de las consultas instantáneas Las mejoras observadas en este índice se deben al mayor poder de filtrado que se logró generando *fhqt*$_i$ consecutivos con diferentes grupos de pivotes.

.

4. Líneas de Investigación

Nuestra principal línea de estudio e investigación es el desarrollo de índices métrico-temporales eficientes.

Los índices desarrollados hasta el momento se basan en el supuesto de que la memoria principal tiene capacidad suficiente como para mantener tanto el índice como la base de datos. Si esto no es así, la cantidad de accesos a memoria secundaria realizados durante el proceso de búsqueda es un factor crítico en la performance del índice [9]. Nos proponemos explorar técnicas de paginado que sean aplicables a los índices métrico-temporales a fin de lograr que los mismos resulten eficientes también en memoria secundaria. Un buen punto de partida para este problema es estudiar las técnicas de paginado que existen en otras áreas para árboles r-arios y/o binarios [9] y adaptar las mismas a nuestro problema. Otra posibilidad para lograr un índice métrico-temporal eficiente en memoria secundaria es reemplazar el FHQT por algún índice métrico en memoria secundaria, que permita dinamismo.

Finalmente, en las aplicaciones en las que el modelo métrico-temporal tiene interés, existen otros tipos de consultas que resultan interesantes, tales como: búsqueda del vecino o de los k-vecinos más cercanos en un intervalo de tiempo, consultas instantáneas puras como por ejemplo *encontrar todas las fotografías vigen-*

tes en un instante de tiempo y consultas por clave, por ejemplo *encontrar las diferentes fotografías de una persona a lo largo del tiempo*. Para cada una de ellas se hará necesario estudiar si los índices propuestos en este trabajo las pueden resolver o si son necesarios cambios en el diseño de los mismos.

5. Resultados Esperados

Se espera contar con un índice e métrico-temporal en memoria secundaria que sea eficiente para resolver las consultas planteadas tanto en los tiempos de respuesta como en el espacio ocupado por el mismo.

6. Formación de Recursos Humanos

El trabajo desarrollado hasta el momento forma parte del desarrollo de dos Tesis de Maestría en Ciencias de la Computación, una de ellas fue defendida y aprobada en marzo del 2009. Se cuenta con el asesoramiento del Dr. Gilberto Gutiérrez, de la Universidad del Bio Bio, Chile. El grupo cuenta además con cinco alumnos becarios que están iniciando su formación en estas temáticas.

Referencias

[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] 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 Computación*, Corrientes, Agentina, 2007.

[3] B. Bustos, G. Navarro, and E. Chávez. Pivot selection techniques for proximity searching in metric spaces. In *Proc. of the XXI Conference of the Chilean Computer Science Society (SCCC'01)*, pages 33–40. IEEE CS Press, 2001.

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

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

[6] A. Pascal, A. De Battista, G. Gutierrez, and N. Herrera. Métodos de acceso para bases de datos métrico - temporales. In *Actas del XV Congreso Argentino de Ciencias de la Computación*, pages 1061–1070, Jujuy, Argentina, 2009.

[7] A. Pascal, De Battista, G. Gutierrez, and N. Herrera. Procesamiento de consultas métrico-temporales. In *XXIII Conferencia Latinoamericana de Informática*, pages 133–144, San José de Costa Rica, 2007.

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

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