Estudio del estado del arte de índices de búsqueda en GML (Geographical Markup Language)

José Eduardo Córcoles
Universidad de Castilla-La Mancha - España
corcoles@dsi.uclm.es

ABSTRACT

En la medida que aumenta la cantidad de documentos escritos en XML, se hace más crítico contar con un mecanismo que permita buscar información dentro de estos documentos. De cara al usuario, estos mecanismos deberían permitir ejecutar una consulta en la menor cantidad de tiempo posible. Para acelerar este procesamiento y obtener una respuesta en los tiempos adecuados, las estrategias de indexación se convierten en la clave del preprocesamiento. En este trabajo se presenta un resumen de cuáles son los tipos de índices que diversos investigadores proponen para su implementación. La mayoría de las propuestas están orientadas hacia XML sin tener en cuenta las características particulares de los documentos GML, por lo que aún queda trabajo por desarrollar en ésta área.

1. INTRODUCCIÓN

Como lo indicara Serge Abiteboul et al. el año 2000[9]: "La combinación de Datos Semi-Estructurados y XML. dará como resultado una nueva tecnología para la administración de múltiples fuentes de datos y datos web los cuales harán desaparecer los esquemas rígidos".

Una de las características de XML es permitir una flexibilidad importante a la hora de desarrollar aplicaciones web. Esa flexibilidad va desde la presentación de los datos al usuario hasta el almacén de la información que es consultada[3].

La flexibilidad de XML, sumado a otras ventajas importantes, permitió su amplia adopción como un estándar para la descripción e intercambio de datos, lo cual generó el crecimiento explosivo de documentos escritos en este formato. Tal crecimiento ha generado el desarrollo de diversas tecnologías en diversos ámbitos de la informática, entre las cuales no podía estar ausente la de Sistemas de Información.

Hoy en día existen Bases de Datos XML Nativas (NXD por Native XML Database) y Extensiones de Bases de Datos para XML. El objetivo primordial de estas bases de datos no es otro que permitir la misma funcionalidad de los Sistemas Gestores de Bases de Datos clásicos, sobre documentos XML. Sin embargo, almacenar documentos XML conlleva la necesidad de contar con mecanismos de recuperación eficientes.

En principio, las primeras Bases de Datos XML Nativas utilizaron XPath (XML Path Language) para interrogar a la base de datos, con extensiones para consultar a través de colecciones. Sin embargo, XPath no fue diseñado como un lenguaje de consulta y no cumplía con las expectativas cuando se le utilizaba como tal.[1]

Más próximo en el tiempo, el desarrollo de lenguajes con mayor expresividad, y basados sobre modelos formales de datos, como XQuery (XML Query), proporcionan un avance significativo para el desarrollo de las bases de datos. A partir de un modelo formal, las entradas y salidas de una consulta se convierten en una instancia del modelo de datos[3].

Sin embargo, dada la naturaleza semiestructurada de los documentos XML, las técnicas de procesamiento de consulta más generales no trabajan bien con este tipo de documentos. Los investigadores han propuesto métodos de indexado especializados que ofrecen un acceso más eficiente[2].

Otro desarrollo importante en el comienzo del presente siglo, ha sido la generación, a partir de la gramática de XML, del Lenguaje de Marcado Geográfico(GML por Geographic Markup Language), que fue desarrollado por el grupo OpenGis, ahora el Consorcio Geo-espacial Abierto (OGC por Open Geospatial Consortium), para solucionar los problemas relacionados con la interoperabilidad de datos en la esfera científica geográfica. Su importancia radica en que a nivel informático se ha constituido en la lengua franca para el manejo e intercambio de información entre los diferentes software que hacen uso de este tipo de datos, como los Sistemas de Información Geográfica(GIS por Geographic Information System).

GML ha sido usado principalmente como un estándar para transportar datos geográficos, pero también puede ser un formato útil para almacenarlo. Ya que GML es una aplicación del estándar XML a datos geográficos, los sistemas de bases de datos XML también pueden ser usa-

dos para la administración de GML. Sin embargo, la naturaleza de datos que GML representa y las técnicas de consultas que estos requieren es completamente diferente de otras lenguas basadas en XML. Hasta el año 2004, por ejemplo, sólo dos especificaciones para lenguajes de consulta GML habían sido propuestas[8].

En la segunda sección se presentan los conceptos previos para entender el objetivo de las estrategias de indexación para consultas en XML y en la tercera sección se presentan estas estrategias. La mayor parte de estas dos secciones están realizadas sobre el trabajo de Barbara Catania et al.[2].

En la cuarta sección se presenta el uso de R-tree como estructura de indexación para datos de tipo GML. En la quinta sección se presentan las conclusiones y en la sexta están las referencias del trabajo
GML se diseñó a partir de la especificación abstracta producida por el grupo OpenGIS, ahora Open Geospatial Consortium (OGC, www.opengeospatial.org), y de la serie de documentos ISO 19100. Tal como XML, el contenido de los datos geográficos está separado de su presentación. Ya que este lenguaje está basado sobre el estándar XML, puede ser fácilmente leído y transformado en cualquier variedad de formatos, que incluyen gráficos vectoriales, gráficos raster, texto y sonido. La presentación de salida gráfica más común de GML es como un mapa.

Sintácticamente, un documento GML es como un documento XML, pero con algunas etiquetas especiales que pertenecen a un conjunto de esquemas predefinidos, como en el siguiente ejemplo:

GML 2.0 está basado en esquemas XML, lo cual le permite soportar una gran cantidad de características deseables para el propósito de la representación de datos geográficos. Entre estas características se pueden mencionar las siguientes: herencia de tipos, integración de esquemas distribuidos y espacios de nombres, tipos de datos primitivos como string, boolean, float, etc. y la construcción de tipos de datos complejos definidos por el usuario.

GML 3.0 ha sido ampliado para representar fenómenos geoespaciales. Además de las características lineales simples en dos dimensiones, esta versión permite representar características de geometría tridimensional compleja y no lineal, topologías bidimensionales, propiedades temporales, características dinámicas, cobertura y observaciones. Por sobre todo, GML 3.0 representa sistemas de referencia espaciales y temporales, unidades de medida e

información estandarizada.

Los esquemas en GML 2.0 pueden ser clasificados en dos tipos: esquemas bases y esquemas de aplicación. Hay tres esquemas bases en GML: Esquema de Características GML (feature.xsd), Esquema de Geometría GML (geometry.xsd) y Esquema XLinks (xlinks.xsd). Los primeros dos fueron definidos y desarrollados por la OGC y el último por W3C. Los esquemas de aplicación son particulares a un dominio de aplicación, por lo tanto son desarrollados por los propios usuarios.

El GML 3.0 requiere, además de los esquemas bases, otros esquemas adicionales para obtener características de codificación como el soporte de geometrías tridimensionales, topologías, dimensión del tiempo, etc. Por ejemplo, si se deben codificar características con propiedades topológicas, entonces se debe tener el esquema topology.xsd.

3. ÍNDICES EN XML

De forma amplia, las técnicas de indexación se pueden clasificar basándose en los tipos de consulta que pueden ser ejecutadas mirando en un único índice. La clasificación sería:

1. Índices de Resumen.

2. Índices de Reunión Estructural.

3. Índices Basados en Secuencia.

3.1 Índices de Resumen

Los índices de resumen indexan atributos y elementos basado en las rutas del documento XML que ellos han identificado. Luego, una expresión de ruta simple envuelve sólo relaciones padre-hijo que pueden ser ejecutadas con una sencilla mirada al índice. Por otro lado, con pocas excepciones, las consultas con bifurcaciones generalmente requieren un procesamiento adicional porque a menudo deben ser descompuestas en un conjunto de consultas de ruta ejecutadas de manera independiente, combinando luego todos los resultados. Esta aproximación no puede manejar eficazmente las expresiones de rutas que contienen relaciones de antepasado-descendiente, incluidas las consultas de correspondencia parcial, fallando en el apoyo directo a las condiciones de selección sobre nodos internos del árbol.

En su forma simple, un índice de resumen asocia las rutas de un documento XML con el conjunto de elementos que estas rutas pueden alcanzar. Puede ser implementado, por lo tanto, como un árbol, en el cual cada nodo representa un nombre de etiqueta y está asociado con todas las posiciones de los elementos que se pueden alcanzar desde la raíz hasta el nombre de la etiqueta. Se requiere de estructuras de datos más complejas, sin embargo, se adapta no sólo para expresiones de ruta, sino también con tipos de consulta más generales, como consultas de bifurcación y basadas en contenido.

3.2 Índices de Reunión Estructural

Los índices de reunión estructural indexan atributos y elementos con un nombre particular o aquellos cuyos contenidos satisfacen una condición dada. En general, los desarrolladores usan estos índices en el contexto de procesamiento basado en reuniones, en el cual el procesador de consultas determina primero los elementos que corresponden a los nodos del árbol de la consulta. Luego reúne los conjuntos obtenidos con un algoritmo de reunión estructural, usando esquemas de localización específicos para mejorar el procesamiento. La reunión estructural es un punto clave en la optimización de consultas XML, y los investigadores han propuesto varias técnicas, desde variaciones de los algoritmos de mezcla-reunión relacional hasta técnicas para reducir la computación de resultados intermedios sin valor, posiblemente confiando en el uso de índices de resumen totales. Los desarrolladores pueden usar procesamiento basado en reuniones para resolver consultas de bifurcación y de rutas, así como consultas de correspondencia total y parcial, sin cambios en el procesamiento o el rendimiento.

Los índices de reunión estructural se pueden clasificar en tres grupos: simples, de texto completo y basados en estructuras.

3.2.1 Índices Simples

Estos índices regresan el conjunto de elementos y atributos que satisfacen una cierta condición que puede ser chequeada localmente contra ellos mismos. La condición debe envolver el contenido asociado con un elemento o un atributo (índices basados en valor) o su propio nombre de etiqueta (índices de nombre). Típicamente los desarrolladores construyen índices simples utilizando las tecnologías de índices relacionales, tales como los B-tree.

3.2.2 Índices de Texto Completo

Estos índices regresan el conjunto de elementos que satisfacen una cierta condición sobre el contenido textual. Los índices de texto completo típicamente confían en los índices invertidos, los cuales asocian cada palabra con su posición en el documento donde aparecen. Los desarrolladores pueden implementar un índice de texto completo como un B-tree. El esquema de localización que ellos utilizan puede cambiar el número de consultas de texto completo que la aplicación puede ejecutar. Los esquemas de localización basados en rutas , por ejemplo, no soportan consultas de proximidad (consultas que preguntan por grupos de palabras que tienen una cierta distancia lexicográfica unas de otras).

3.2.3 Índices Basados en Estructuras

Estos índices regresan elementos basados en sus relaciones estructurales (antepasado-descendiente o padre-hijo). Las aplicaciones pueden, por ejemplo, saltar elementos de tipo antepasado o de tipo descendiente que no participen en la reunión. Algunos índices basados en estructuras garantizan un buen rendimiento durante las modificaciones, incluso cuando ellos confían en esquemas basados en posición.

3.2.4 Ejemplos de Índices de Reunión Estructural

- XML Indexing and Storage System soporta índices de nombres simples.

- XML Region Tree reduce el número de elementos recuperados, dependiendo de las relaciones requeridas. Los investigadores también han propuesto un nuevo algoritmo de reunión estructural usando XR-tree.

- The Boxes methods usan un esquema de etiquetado basado en la posición y estructuras de datos ad hoc para conseguir una mejor consulta que reduzca el costo.

- Lazy Join ejecuta modificaciones en las rutas y en los modelos de los documentos XML como un árbol de segmentos de documentos XML, luego los indexa.

3.3 Índices Basados en Secuencia

En los índices basados en secuencia, los documentos XML y las consultas de bifurcación son representadas como secuencias, y las correspondencias de subsecuencias proporcionan la respuesta a la consulta. De esta manera los índices XML basados en secuencia utilizan la es-

tructura del árbol como la unidad de consulta así evitan la necesidad de desensamblar una consulta estructurada en múltiples subconsultas. De la misma manera que ocurre con los índices de resumen, los índices basados en secuencia no soportan directamente las consultas con condiciones de selección sobre nodos internos.

Estos índices pueden generar falsos resultados, cuando los aciertos encontrados por una subsecuencia no siempre corresponden a los aciertos de un subárbol, entre la consulta y los documentos. En tales casos, se agrega un paso de refinamiento para eliminar los falsos resultados. Los investigadores han definido métodos de secuencia específicos para evitar este problema.

3.3.1 Ejemplos de Índices Basados en Secuencias

Virtual Suffix Tree codifica los documentos XML y las consultas como secuencias de pares, cada uno representando un nodo y la ruta (incluyendo el contenido del nodo) para alcanzarlo, de acuerdo a una visita en preorden del árbol. El peor caso de almacenamiento es lineal sobre el número total de elementos.

Prüfer sequences para indexación XML codifica los documentos XML y las consultas como secuencias de etiquetas, correspondientes a las secuencias Prüfer. El peor caso de almacenamiento es lineal sobre el número total de elementos.

Wang/Meng method: el método de Haixun Wang y Xiaofeng Meng usa clases de métodos de secuencia para preservar la equivalencia de la consulta entre una correspondencia estructural y una correspondencia de subsecuencia. Ellos proponen algoritmos para indexar las secuencias resultantes.

3.4 Sistemas de Indexado Comerciales y Prototipos

Los desarrolladores pueden administrar documentos XML eficientemente usando uno o dos tipos de sistemas administradores de bases de datos

- Extensiones de Bases de Datos para XML: Estas bases de datos (usualmente relacionales) proporcionan interfaces para transformar datos desde XML a un modelo interno y viceversa. La representación puede ser estructurada (si el modelo de dato subyacente representa

documentos XML) o no estructurada (o nativa, cuando los documentos XML son representados como texto usando tipos específicos XML). Ejemplos de estas bases de datos son Oracle, Microsoft SQL Server y IBM DB2.

- Bases de Datos XML Nativas: Estas bases de datos almacenan y recuperan documentos XML de acuerdo a un modelo de datos propietario. La única interfaz para datos XML está basada en tecnologías XML (como SAX, DOM, XPath, XQuery, y otras). Ejemplos de estas bases de datos son dbXML, eXist, XIndice, Ipedo, Timber y Tamino.

Como las extensiones XML descansan sobre sus tecnologías de bases de datos (generalmente relacionales), proporcionan al usuario eficiencia, escalabilidad, control de concurrencia y confianza. Sin embargo, como es una representación estructurada, dado el contenido de un documento XML éste es factorizado en varias estructuras de acuerdo al modelo de datos subyacente, lo cual puede derivar en un menor rendimiento en el caso de procesar consultas sobre este tipo de datos. En contraste, las representaciones no-estructuradas de las NXD, usan técnicas de almacenamiento propietarias, proporcionando una alta flexibilidad y una eficiente utilización del espacio. Sin embargo, las NXD aún no logran proporcionar toda la funcionalidad de una base de datos tradicional.

Típicamente, las extensiones para XML usan índices estructurales simples (B-trees) para su representación estructurada. De la misma manera, las NXD indexan sólo los contenidos de atributos y elementos, además de los nombres de etiquetas. En ambos casos las tecnologías típicas adoptan índices de texto completo (invertido) para indexar contenidos textuales o rutas. Microsoft SQL Server 2005 implementa la reunión estructural y los índices resumidos basados en Ordpath, un esquema de numeración basado en la posición.

Para las extensiones XML, la distancia entre la teoría y los propósitos comerciales es debido probablemente al hecho que los desarrolladores pueden implementar fácilmente índices simples y de reunión estructural de texto completo usando B-trees e índices de texto, estructuras de datos que la arquitectura subyacente ya soporta. En el caso de las NXD, probablemente su tecnología aún no es lo suficiente madura para soportar técnicas de indexación sofisticadas.

4. ÍNDICES EN GML

La minería de datos espacial analiza las relaciones entre los atributos de un objeto espacial, almacenado en la base de datos, y los atributos de los vecinos. Las preguntas típicas requeridas por esta clase de análisis son: encontrar todos los objetos que se traslapan con el punto de la consulta o encontrar todos los objetos que tienen al menos un punto en común con una ventana de consulta.

Los métodos de acceso para datos multidimensionales pueden ser clasificados en Métodos de Acceso de Punto (PAM por Point Access Methods) y Métodos de Acceso Espaciales (SAM por Spatial Access Methods). Los métodos de acceso de punto, como su nombre lo indica, están diseñados expresamente para realizar búsquedas en bases de datos de puntos. Estos métodos, por lo general, administran los datos como cubos, cada uno correspondiente a una pá- gina de disco. Los cubos son indexados por estructuras de datos planas o jerárquicas las estructuras planas son usadas en métodos de partición multidimensional como los grid file y EXCELL, las estructuras jerárquicas son usadas en métodos de acceso jerárquicos como quadtree, KD-tree y KD- B-tree. Los métodos de acceso espacial manejan objetos con propiedades espaciales como área y forma. Los métodos de acceso en SAM son a menudo extensiones de PAM para manejar objetos con un grado espacial. Tales métodos incluyen el R-tree, R*-tree y Multi-layer grid file [6].

Al relacionar los términos "R-tree" y "GML" en la web, ésta recupera un número significativo de investigaciones con las cuales se puede suponer que las técnicas de indexación basadas en esta estructura son aplicables a GML. En el caso de otros nombres de índices prácticamente no aparece ninguna relación, lo cual no significa que no existan, pero por no encontrarse ninguna aplicación que los relacionara directamente se ha desarrollado esta sección en torno a R-Tree.

4.1 Índice R-Tree

Las técnicas basadas en R-tree y sus derivaciones, son utilizadas para mejorar el rendimiento de las consultas basadas en datos espaciales, de los sistemas comerciales y abiertos, que dan soporte a los GIS. De igual forma, los desarrolladores del ámbito geoespaciales, al parecer lo han seleccionado como la estructura más adecuada para los sistemas basados en GML.

Uno de los trabajos más recientes es "Un modelo para indexado y consultas espaciales sobre GML basado en XQuery", de Shuangfeng Wei et al.[11]. Este modelo separa el documento GML en estructura y contenido, en concordancia con la naturaleza semiestructurada de XML, y los optimiza individualmente. Por una parte, se adopta un esquema de codificación ampliado, basado en un árbol k-ario completo, que cuantifica los nodos que presentan la información sobre ru tas GML, luego optimizan las expresiones de caminos con el Modelo Extensible Orientado a Ruta (POEM por Path Oriented Extensible Model) para mejorar la eficacia de las consultas. Por otra parte, los contenidos GML son almacenados por separado como conjuntos de nodos, y se indexan mediante R-tree. Para integrar las características de datos espaciales, se da un conjunto de funciones de consulta extendidas, incluidas funciones de operación básicas, operaciones de relaciones espaciales y funciones de análisis espaciales. Finalmente se amplía XQuery para realizar un sistema de consulta basado en un software de código fuente abierto: XQEngine.

Otra aproximación interesante es el "Sistema de Consultas para Web Geoespaciales basado en Ontologías", de Nancy Wiegand y Naijun Zhou[12]. Este modelo proporciona soporte a conjuntos de datos geoespaciales distribuidos en la web. El trabajo está hecho en el contexto de una propuesta para todo el estado de un sistema de información terrestre, que consiste tanto de datos espaciales como de no espaciales que permanecen en servidores locales. Uno de los obstáculos principales en las consultas de datos geoespaciales distribuidos es la heterogeneidad semántica. El trabajo se concentra en la resolución de la heterogeneidad semántica, a nivel de valores, para acomodar los conjuntos de datos distribuidos que tienen atributos conceptualmente similares, en los cuales los valores son obtenidos desde diversos dominios. Se describe el sistema de consulta y se incluye un acercamiento dinámico para la integración de ontologías entre una ontología central y los dominios locales. El sistema acomoda los datos geoespaciales, y propone diseños para las consultas. En general, este trabajo extiende las capacidades de consultas de las bases de datos web XML a datos heterogéneos y geoespaciales. La propuesta se basa en una arquitectura que extiende las capacidades del motor de búsqueda de una base de datos mediante la inclusión de bases de datos con ontologías e índices de metadatos. La indexación de datos GML incluye técnicas de indexación para datos geoespaciales y no geoespaciales, utilizando listas para los primeros, y listas invertidas para los segundos. Propone la utilización

de un indexado basado en R+-tree para indexar los límites de los objetos que contienen coordenadas.

5. CONCLUSIONES

Según [2] se puede establecer, en base a las clasificaciones, que cuando no existe información sobre el conjunto de datos de las consultas objetivos, los índices basados en secuencia y los de reunión estructural son la mejor elección para procesar las consultas XML. Cuando la estructura de los documentos XML y las consultas están definidas a priori, los índices de resumen son la opción más conveniente.

Las técnicas de indexación mostradas en este trabajo podrían proporcionar, a los sistemas comerciales de bases de datos, un mayor rendimiento de las consultas sobre datos XML, por estar basadas en la naturaleza semiestructurada de dichos documentos. Para las NXD significaría disminuir la distancia, en funcionalidad, entre éstas y las bases de datos tradicionales, al proporcionar tiempos de respuesta mucho mayores.

Para el caso de GML: las bases de datos, los lenguajes de consultas y las técnicas de acceso, son tópicos de estudio que están en pleno desarrollo y a simple vista se puede apreciar que aún queda mucho camino por recorrer. Entre tanto se puede conjeturar que así como las técnicas basadas en B- tree son las más usuales en NXD y extensiones de XML, lo más probable es que las técnicas de indexación basadas en R-tree sean las más utilizadas en GML.

5. Bibliografía

[1] C. Brys. Xml y bases de datos. Revista Científica. Visión de Futuro, 1(1), Junio 2004.

[2] B.Catania, A.Maddalena, and A.Vakali. Xml document indexes: a classification. Internet Computing, IEEE, 9(5):64–71, 2005.

[3] J.Córcoles. Apuntes del curso tecnologías de software orientada a objetos: Introducción a xml. Programa de Doctorado: Arquitectura y Gestión de la Información y del Conocimiento en Sistemas en Red, 2007.

[4] J.Córcoles and P.González. A specification of a spatial query language over gml. Proceedings of the

9th ACM international symposium on Advances in geographic informa-

tion systems, pages 112–117, 2001.

[5] J.Córcoles and P.González. Analysis of different approaches for storing gml documents. Proceedings of the tenth ACM international symposium on Advances in geographic information systems, pages 11–16, 2002.

[6] M.Frailis, A.DeAngelis, and V.Roberto. Data management and mining in astrophysical databases. Arxiv preprint cs.DB/0307032, 2003.

[7] Y.Li, J.Li, and S.Zhou. Gml storage: A spatial database approach. Proc. of ER, 4.

[8] B.Shrestha. XML Database Technology and its use for GML. PhD thesis, Master Thesis, International Institute for Geo-information Science and Earth Observation (ITC), The Netherlands, 2004.

[9] E.Stefanakis. Geographic databases and xml.

Lecture, Muenster, http://www.dbnet.ece.ntua.gr, August 2002.

[10] R.Vatsavai. Gml-ql: A spatial query language specification for gml. Department of Computer Science and Engineering, University of Minnesota

http://www. cobblestoneconcepts. com/ucgis2summer2002/vatsavai/vatsavai. htm.

[11] S.Wei, D.Li, Z.Xiao, and L.Zheng. A model for xquery-based gml spatial index and query. Proceedings of SPIE, 6420:642006, 2006.

[12] N.Wiegand and N.Zhou. Ontology-based geospatial web query system. Next Generation Geospatial Information: From Digital Image Analysis to

Spatio-Temporal Databases, ISPRS Book series. Balkema, Taylor & Francis, 2005.

[13] A.Zipf and S.Krüger. Tgml: extending gml by temporal constructs-a proposal for a spatiotemporal framework in xml. Proceedings of the 9th ACM international symposium on Advances in geographic information systems, pages 94–99, 2001.

SOCIEDAD DE LA INFORMACION

www.sociedadelainformacion.com

Edita:

Director: José Ángel Ruiz Felipe
Jefe de publicaciones: Antero Soria Luján
D.L.: AB 293-2001
ISSN: 1578-326x