Universidad Simón Bolívar
Web Semántica
Exámen

Fabiola Di Bartolo
Carnet: 09-87324

11 de diciembre de 2009

En base a la tendencia actual que existe en la Web Semántica de publicar grandes *datasets* en el Çloud of Linked Data", identifique las bondades y limitaciones del proyecto que esta desarrollando que permitan o limiten la manipulación de este tipo de *datasets*.

El Cloud of Linked Data actualmente es de gran importancia para la Web Semántica, ya que según [1] la Web de Linked Data, que surge de la conexión de los datos de diferentes fuentes vía links RDF, puede ser entendida como un único *dataspace* globalmente distribuido. Por lo tanto, consultar este *dataspace* abre posibilidades anteriormente no concebibles: la información de distintas fuentes puede ser integrada para alcanzar una vista más completa [2]. Es por esto, que ha surgido el reto de desarrollar nuevas técnicas que permitan consultar estos *datasets*, ya que debido a la amplitud del *dataspace*, no es posible conocer por anticipado todas las fuentes de dados que pueden ser relevantes para responder la consulta.

Diversos trabajos se han realizado sobre el área, algunos trabajan con fuentes de datos distribuidas: en [3] proporcionan algunos conceptos desarrollados en el area, en [4] proponen la aplicación DARQ que descompone la consulta en subconsultas, que son ejecutadas y luego integradas, en [5] proponen SemWIQ, que transparente distribuye la ejecución de la consulta SPARQL y en [2] proponen SQUIN. SQUIN permite realizar descubrimientos de fuentes de datos relevantes y devolver los datos que va encontrando durante el tiempo de ejecución de la consulta, usando un iterador pipeline, materialización de los patrones y rechazos de solicitudes, permitiendo retornar los datos sin esperar que la consulta haya sido evaluada en su totalidad.

Otros investigadores en el area, se basan en el estudio de la realización de búsquedas sobre un subconjunto de Linked Data que ha sido copiado de las diferentes fuentes y centralizado previamente con la finalidad de indexar los datos y retornar resultados más completos tales como las aplicaciones: Sindice [6], Swoogle [7] y Watson [8].

Similar a las aplicaciones anteriores se encuentra OneQL [9], la cual permite ejecutar consultas sobre *datasets* centralizados (localmente), haciendo un especial énfasis en encontrar un buen plan de ejecución y en realizar una evaluación eficiente ya que se sabe que la tendencia en la Web Semántica es trabajar con volúmenes de datos cada vez más grandes y por lo tanto, son necesarias técnicas eficientes en las aplicaciones que permiten consultas

SPARQL para que la ejecución (optimización y evaluación) de consultas sea escalable al aumentar la cantidad de tripletas y fuentes de datos.

Sin embargo, algunas de las aplicaciones diseñadas para esto RDF3x [10], Jena [11] y OneQL [9], tienen como precondición que los documentos a ser consultados deben ser descargados del *site* en donde se encuentren y posteriormente cargados en la aplicación que nos permitirá realizar las consultas. Esto ocurre porque estas aplicaciones no proveen la plataforma para buscar los datos en el Web necesarios para responder la consulta, es por esto que se debe prestar atención al proceso de carga de los documentos RDF ya que puede ser muy costoso en tiempo y espacio. Para probar el comportamiento de OneQL con *datasets* de gran tamaño se descargó el *dataset* de Yago que ocupa aproximadamente 4GB, una vez descargados los documentos RDF, se procedió a traducir las tripletas RDF al formato entendible por OneQL y para esto se necesitó 16 GB de memoria RAM, excluyendo los casos de traducciones de documentos que fallaban por falta de memoria. Luego, se depuraron estos documentos traducidos ya que tenían caracteres que hacían que los predicados no fueran entendidos por Swi-Prolog (el lenguaje en que se encuentra desarrollado OneQL) y finalmente se pasó a eliminar los duplicados y cargarlos. Este proceso se llevó alrededor de unas 5 semanas de trabajo. Si OneQL proporcionará una facilidad para optimizar y evaluar consultas sin la necesidad de tener los datos almacenados de forma local podría resultar más rápido y viable al momento de querer consultar documentos publicados en el Web, mas bien, se podría invertir ese tiempo en descubrimientos relevantes.

Aún cuando puede ser una necesidad realizar las búsquedas navegando en el Web sin tener los datos precargados, puede ser más importante aún realizar las búsquedas de forma eficiente, que arrojen resultados en un tiempo razonable. Y estos tiempos están relacionados directamente por el plan escogido para evaluar la consulta, por lo tanto el optimizador de la consulta juega un papel fundamental en la eficiencia de la ejecución. OneQL utiliza estadísticas relacionadas con la selectividad y cardinalidad de los datos, así como también, emplea técnicas de muestreo para encontrar un buen plan, y esto es una gran ventaja sobre SQUIN, ya que este sólo se enfoca en obtener resultados rápidos mediante la ejecución pipeline, pero no emplea buenas técnicas de optimización para hallar un mejor plan. Tendría que realizarse un estudio de extensión OneQL y ejecutar estas tareas sobre documentos publicados y distribuidos en toda la web y así medir la efectividad de las técnicas propuestas contra los otros evaluadores disponibles como SQUIN.

Una vez que el plan es escogido, debe ser evaluado de una forma eficiente,

es por esto que algunas aplicaciones diseñadas para tales fines han implementado índices sobre los datos, en particular OneQL propone los índices Bypher (basados en técnicas sobre hipergrafos). Y si se observan los resultados de los experimentos basados en la ejecución de consultas utilizando OneQL, es notable la diferencia en cuanto a tiempo de ejecución de la utilización de estos índices en la evaluación de consultas. Por lo tanto, se debería hallar una forma de incluir una variante de estas estructuras, tal vez más flexible, en el Cloud of Linked Data. Aunque éstas ocupen un gran espacio pueden ser de gran utilidad para hacer ejecuciones más eficientes.

Una limitación es que al realizar consultas en el Cloud of Linked Data, como los documentos están enlazados sin ninguna restricción y se desconocen los documentos, los datos en ellos y los enlaces entre los mismos, no es factible saber a priori las fuentes que participarán en el resultado de la consulta, por lo que los algoritmos de optimización (Programación dinámica y Simulated Annealing) y las técnicas de muestreo presentadas para OneQL no funcionarán para este tipo de consultas porque este optimizador no trabaja a ciegas. Estas técnicas se podrían utilizar si previamente se realiza un proceso de análisis de los datos y creación de los diccionarios de datos o si son adaptadas al enfoque de datos distribuidos.

Por otro lado, como OneQL no cuenta con un iterador pipeline para ejecutar las consultas, aunque pueda retornar la respuesta completa en un menor tiempo, al usuario le puede resultar mejor tener una respuesta parcial en un corto tiempo. Sin embargo, esta respuesta parcial puede variar según la disponibilidad de los datos en el Web, ya que por diversas causas como: trafico de datos, servidores caídos, conexiones lentas, pueden hacer que datos relevantes sean inaccesibles para el evaluador, problema que no ocurriría con una consulta ejecutada con OneQL teniendo los *datasets* locales en el servidor. Otro aspecto importante, es que se debería definir claramente y formalmente la manera en que la aplicación que consulta el cloud rechazará las peticiones para que esto no se realice aleatoriamente y se le pueda sacar un mayor provecho.

Finalmente, dependiendo de las necesidades del usuario y la disponibilidad de los recursos (disco, memoria RAM, procesador) puede ser mejor utilizar aplicaciones que permitan realizar consultas sobre el Cloud of Linked Data en lugar de las que sólo permiten ejecutar consultas sobre *datasets* locales en el servidor de la aplicación, o viceversa. Queda entonces de parte del usuario evaluar la mejor forma de consultar documentos RDF considerando los puntos tratados anteriormente.

Bibliografía

[1] M. Franklin, A. Halevy, and D. Maier, "From databases to dataspaces: a new abstraction for information management," *SIGMOD Rec.*, vol. 34, pp. 27–33, December 2005.

[2] O. Hartig, C. Bizer, and J.-C. Freytag, "Executing sparql queries over the web of linked data," in *8th International Semantic Web Conference (ISWC2009)*, 2009.

[3] A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and autonomous databases," *ACM Computing Surveys*, vol. 22, pp. 183–236, 1990.

[4] B. Quilitz and U. Leser, "Querying distributed rdf data sources with sparql," in *Proceedings of the 5th European Semantic Web Conference* (M. Hauswirth, M. Koubarakis, and S. Bechhofer, eds.), LNCS, (Berlin, Heidelberg), Springer Verlag, June 2008.

[5] A. Langegger, W. Wöß, and M. Blöchl, "A semantic web middleware for virtual data integration on the web," in *5th European Semantic Web Conference (ESWC2008)*, pp. 493–507, 2008.

[6] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello, "Sindice.com: a document-oriented lookup index for open linked data," *Int. J. Metadata Semant. Ontologies*, vol. 3, no. 1, pp. 37–52, 2008.

[7] L. Ding, T. Finin, A. Joshi, R. Pan, S. R. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, "Swoogle: a search and metadata engine for the semantic web," in *CIKM '04: Proceedings of the thirteenth ACM conference on Information and knowledge management*, (New York, NY, USA), pp. 652–659, ACM Press, 2004.

[8] M. D'aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, and D. Guidi, "Toward a new generation of semantic web applications," *Intelligent Systems, IEEE*, vol. 23, no. 3, pp. 20–28, 2008.

[9] T. Lampo, E. Ruckhaus, J. Sierra, M. E. Vidal, and A. Martynez, "Oneql: An ontology-based architecture to efficiently. query resources on the semantic web,"

[10] "Rdf-3x: A risc-style engine for rdf,"

[11] "Jena tdb. http://jena.hpl.hp.com/wiki/tdb,"