Procesamiento de Consultas en Motores de Búsqueda: Diseño y Evaluación en Términos de Consumo de Energía

Verônica Gil Costa[(1)] and Mauricio Marin[(2)]

[(1)] LIDIC, Departamento de Informática
Facultad de Ciencias Físico, Mateáticas y Naturales
Universidad Nacional de San Luis

[(2)] Yahoo! Research Latin America
Santiago, Chile

RESUMEN

Actualmente los centros de datos accedidos por los buscadores web junto con las computadoras personales consumen el 10% de la energía mundial, y de ese porcentaje aproximadamente el 2% es consumido sólo por los buscadores y sus centros de datos. Sin embargo, es de esperar que en los próximos años estos porcentajes se incrementen en un 30% o 40% debido a que el tamaño de la Web tiende a duplicarse cada ocho meses, la cantidad de usuarios que se conectan a ésta sigue creciendo y los buscadores satisfacen la creciente demanda incrementando el hardware utilizado.

En este trabajo se presentan los objetivos y los desafíos de una línea de investigación que abarca los problemas de consumo de energía que deben solucionar actualmente los grandes centros de cómputos y de datos, en particular los buscadores Web.

Palabras clave: *Buscadores Web, Consumo de energía, Métricas de evaluación, Simuladores.*

CONTEXTO

El estudio de los motores de búsqueda Web y sus optimizaciones requiere estudiar técnicas de paralelización y optimización de algoritmos en arquitecturas multi-core; así como estudiar las ventajas de utilizar estructuras de datos comprimidas que permitan acelerar la resolución de consultas y la intersección de documentos en las que aparecen los términos de una misma consulta.

La línea de investigación presentada en este trabajo recurre a dos grandes proyectos de investigación de la Universidad Nacional de San Luis, a) el proyecto de sistemas distribuidos y paralelos; b) el proyecto de recuperación de la información y estructuras de datos.

1. INTRODUCCION

La Web se duplica en tamaño cada seis u ocho meses y con ello más y más personas ingresan a la Web con sus computadores. Además, los grandes buscadores como Google, Yahoo! o Bing dan acceso a sus servicios a través de centros de datos repartidos por distintos lugares del planeta. Se estima que actualmente la cantidad de computadores en los centros de datos es de alrededor de 50 millones. Por otra parte, los computadores personales están por sobre los mil millones de unidades. Todo lo cual hace que en total estos computadores sean responsables de alrededor del 10% del consumo de energía eléctrica a nivel mundial. Los centros de datos son responsables del 2% de dicho consumo. Indudablemente, también, el aspecto económico respecto de consumo de energía de un centro de datos es un tema relevante para un motor de búsqueda que atiende a cientos o miles de millones de usuarios.

En este contexto, resulta relevante investigar y desarrollar estrategias de procesamiento de consultas en motores de búsqueda que hagan que una misma carga de trabajo sea servida por menos computadores. Para esto es necesario desarrollar algoritmos eficientes de búsqueda que trabajen en paralelo y sobre datos distribuidos en miles de computadores. En particular, un aspecto importante a investigar es la cantidad de energía que consumen estos computadores al procesar consultas de usuarios a tasas de miles por segundo.

En los últimos años se han desarrollado varias estrategias de procesamiento paralelo de consultas, las cuales involucran nuevos diseños

de índices distribuidos para texto, estrategias de caché de resultados y otros. Invariablemente la métrica de interés ha sido el throughput definido como la cantidad de consultas por unidad de tiempo que el motor de búsqueda es capaz de resolver. La literatura técnica en estas áreas es abundante [Moffat2006, Moffat07, Risvik03, Tang2004] y da cuenta de una gran actividad en investigación orientada a lograr motores de búsqueda más eficientes respecto de esta métrica.

Pareciera que un motor de búsqueda que es capaz de lograr un throughput dado utilizando menos computadores, es también capaz de consumir menos energía durante su operación diaria. Esto sería cierto si la cantidad de energía consumida por un computador fuese directamente proporcional a la cantidad de trabajo ejecutado sobre éste. Sin embargo, este supuesto dista mucho de la realidad puesto que la mayoría de los dispositivos de hardware que componen los computadores de un centro de datos consumen alrededor del 50% de su energía máxima, es decir, la energía que consumen cuando están operando al 100% de utilización.

Las curvas de consumo de energía versus carga de trabajo para los dispositivos de hardware puede variar significativamente dependiendo del tipo de dispositivo. Por otra parte, las distintas estrategias de procesamiento de consultas originan cargas de trabajo muy diferentes sobre los dispositivos de hardware. Por ejemplo, una estrategia que utilice compresión de datos como parte central de su algoritmo de resolución de consultas podría lograr menos accesos a memoria secundaria respecto de otra que no realice tal compresión. Sin embargo, la primera requiere un mayor uso de procesador si se requiere descomprimir los datos antes de procesarlos mientras que la segunda es más eficiente en uso de procesador pero requiere más accesos a memoria secundaria.

En algunos casos, una memoria Ram dinámica puede consumir más energía que un Disco, y un procesador multi-core Intel de última generación puede consumir más energía que uno más convencional pero tiene una mejor curva de eficiencia energética. Operando a cero carga este procesador puede llegar a consumir sólo el 30% de la energía que consume a máxima carga.

Por lo tanto, no es evidente que un mejor throughput conduzca a un menor consumo de energía, y por tanto no es claro cuales de las estrategias de procesamiento de consultas desarrolladas hasta ahora son las más eficientes en consumo de energía.

Adicionalmente, enfocar el diseño de estrategias de procesamiento de consultas en la métrica de consumo de energía de dispositivos puede dar lugar a estrategias completamente nuevas y capaces de orquestar el uso de los recursos de hardware para lograr un menor consumo de energía según la carga de trabajo a la que están sometidas durante un periodo dado del tiempo.

Las cargas de trabajo en los motores de búsqueda para la Web dependen del comportamiento de sus usuarios. A ciertas horas del día el tráfico de consultas crece significativamente y se pueden presentar subidas muy drásticas de tráfico ante la ocurrencia de eventos de interés masivo. Por esta razón, los motores de búsqueda operan en régimen permanente a una utilización promedio de hardware de alrededor del 30% para responder eficientemente ante subidas de tráfico de consultas. Es decir, respecto de eficiencia en consumo de energía, los buscadores operan en el rango en que los dispositivos de hardware son menos eficientes respecto de sus curvas de energía consumida versus carga de trabajo.

2. TREABAJO PREVIO

Los motores de búsqueda para la Web utilizan listas invertidas distribuidas [Risvik03, Barroso03, BR99] para manejar eficientemente el tráfico alto de consultas. Las listas invertidas son una estructura de datos utilizadas como índices para determinar rápidamente los top-R documentos más relevantes para una consulta. Estas listas invertidas están compuestas por una tabla de vocabulario y un conjunto de posting lists. La tabla de vocabulario contiene los términos encontrados en la colección de documentos indexada, y cada término tiene asociada una posting list. Las posting lists tienen información utilizada durante la operación de ranking para detectar los documentos importantes para las consultas. Básicamente las posting lists tienen el

identificador del documento donde aparece el término y la frecuencia con la que el término aparece en dicho documento. Para procesar una consulta, se recuperan las posting lists de los términos que aparecen en la consulta y se realiza una operación de ranking para seleccionar los top-R documentos.

Existen diferentes métodos para distribuir las listas invertidas en un conjunto de P procesadores y se han propuesto diferentes estrategias de procesamiento de consultas sobre estas particiones [BBRZ2001, Jeong95, MMR2000, Moffat07, RB1998, Stan90, TOM96]. La distribución o particionado del índice consiste en dividir la colección de documentos y/o el índice mismo sobre un conjunto de procesadores. Pero existen básicamente dos estrategias estándar: partición basada en documentos o partición local, y partición basada en términos o partición global. En la primera, la colección completa de documentos se distribuye entre todos los procesadores del servidor y luego cada procesador construye su propio índice local utilizando el sub-conjunto de documentos recibidos. Por lo tanto las operaciones de construcción y actualización no tienen costo en términos de comunicación. Durante el procesamiento de una consulta, una máquina broker recibe esta consulta y la envía a todos los procesadores del servidor. Es decir que en un instante de tiempo dado, una única consulta accede a todos los recursos del sistema para ser resuelta. Luego, los resultados obtenidos para la consulta utilizando los P procesadores, se envían a la máquina broker quien se las entregará al usuario correspondiente.

En la técnica de particionado por términos, se construye un único índice secuencial y luego cada término junto con su posting list se asigna a un único procesador. Por lo tanto las operaciones de construcción y actualización tienen un costo elevado en términos de comunicación. Sin embargo, para procesar una consulta usando esta estrategia, la consulta se envía sólo a un sub-conjunto de procesadores del servidor, sólo a aquellos que tienen los términos de la consulta. Es decir que el particionado por términos permite obtener una mayor escalabilidad reduciendo el hardware utilizado por cada consulta. Algunas variantes de estas dos técnicas que se enfocan en la optimización de situaciones particulares del procesamiento de consultas han sido presentadas en [MM-cikm, Tang2004hybrid, Xi02b].

En el trabajo presentado en [FMMGB2009] se introduce una distribución novedosa de las listas invertidas, que consiste en ver el conjunto de procesadores como una arreglo de dos dimensiones, aplicando una partición por términos sobre las filas y una partición por documentos al nivel de las columnas. En ese trabajo se mostró que seleccionando en número adecuado de columnas y filas es posible obtener mejoras significativas en la performance del sistema.

En la práctica, el conjunto de P procesadores es replicado D veces para incrementar el throughput y tener un sistema tolerante a fallas. Algunos trabajos adicionales sobre el indexamiento de documentos pueden ser encontrados en [Zhang2008, Reynolds2003, Falchi2007, Zhang2007, Chaudhuri2007, Moffat2006, Suel03].

3. RESULTADOS OBTENIDOS/ESPERADOS

Los resultados obtenidos hasta el momento son:

- Diseño e implementación de algoritmos eficientes que permiten resolver consultas ingresadas por los usuarios en un sistema de búsqueda. El sistema ha sido implementado para soportar consultas de texto y multimediales sobre un cluster de computadoras[Gil08a, Gil09, MM-cikm].
- Los sistemas de búsquedas pueden operar en forma síncrona y asíncrona[MM2008a].
- Se ha implementado un simulador basado en dos capas: 1) capa de software, que simula la ejecución de un intérprete (kernel) para realizar las tareas involucradas en cada uno de los comandos ingresados por el usuario. 2) Capa de bajo niel que modela una máquina BSP[Val90].

Los resultados esperados son:

- Contar con un modelo de evaluación de consumo de energía que esté construido sobre la combinación de técnicas de simulación discreta, modelo de computación paralela y carga de trabajo proveniente de trazas generadas por

usuarios reales en un motor de búsqueda.

- Lograr estrategias de procesamiento de consultas que sean eficientes en términos de consumo de energía, ya sea por la vía del diseño de nuevas estrategias que resuelvan las falencias de las estrategias existentes o la combinación de ellas para obtener una mejor respecto de la métrica de eficiencia en consumo de energía.

4. BIBLIOGRAFIA

[Barroso03] L. Barroso, J. Dean and U. Hölzle. Web Search for a Planet: The Google Cluster Architecture. IEEE Micro 23(2):22-28, 2003.

[BBRZ2001] C. Badue, R. Baeza-Yates, B. Ribeiro, and N. Ziviani. Distributed query processing using partitioned inverted files. In SPIRE, pp. 10-20, 2001 (IEEE-CS).

[BR99] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley. 1999.

[Chaudhuri2007] S. Chaudhuri, K. Church, A. C. Konig, and L.Sui. Heavy-tailed distributions and multi-keyword queries. In SIGIR, pp. 663-670, 2007 (ACM).

[Falchi2007] F. Falchi, C. Gennaro, F. Rabitti, and P. Zezula. Mining query logs to optimize index partitioning in parallel web search engines. In INFOSCALE, pp. 1-9, 2007.

[FMMGB2009] Esteban Feuerstein, Mauricio Marin, Michel Mizrahi, Veronica Gil-Costa and Ricardo A. Baeza-Yates. Two-Dimensional Distributed Inverted Files. SPIRE, 2009. pp. 206-213.

[Gil08a] V. Gil-Costa and M. Marin. Distributed Sparse Spatial Selection Indexes. In 16th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP'08), pp. 440–444. IEEE-CS Feb. 2008.

[Gil09] V. Gil-Costa, M. Marin, and N. Reyes. Parallel Query Processing on Distributed Clustering Indexes. Journal of Discrete Algorithms, 7:03–17, 2008.

[Jeong95] B.S. Jeong and E. Omiecinski. Inverted file partitioning schemes in multiple disk systems. IEEE Trans. Parallel and Distributed Systems, 16(2):142-153, 1995.

[Lempel03] R. Lempel and S.Moran. Predictive caching and prefetching of query results in search engines. In WWW, pp. 19--28, 2003.

[MMR2000] A. MacFarlane, J. McCann, and S. Robertson. Parallel search using partitioned inverted files. In SPIRE'02, pp. 209-220., 2000 (IEEE CS).

[MM-cikm] M. Marin and V. Gil-Costa. High-performance distributed inverted files. 16th ACM Conference on Information and Knowledge Management (CIKM 2007), 2007. pp. 935-938,ACM.

[MM2009] M. Marin, F. Ferrarotti, M. Mendoza, C. Gomez and V. Gil-Costa. Location Cache for Web Queries. CIKM 2009, pp. 1995-1998.

[MM2008a] Mauricio Marín, Carlos Gomez-Pantoja, Senen Gonzalez, Veronica Gil-Costa: Scheduling Intersection Queries in Term Partitioned Inverted Files. Euro-Par 2008: 434-443.

[MM2008b] Mauricio Marín, Veronica Gil-Costa, Carolina Bonacic: A Search Engine Index for Multimedia Content. Euro-Par 2008: 866-875

[MM2007] Mauricio Marín, Carolina Bonacic, Veronica Gil-Costa, Carlos Gomez: A Search Engine Accepting On-Line Updates. Euro-Par 2007: 348-357.

[Moffat2004] A. Moffat and J. Zobel. What does it mean to measure performance? In Conf. Web Informations Systems, pp. 1-12, 2004.

[Moffat2006] A. Moffat, W. Webber, and J. Zobel. Load balancing for term-distributed parallel retrieval. SIGIR'06, pp. 348-355, 2006 (ACM).

[Moffat07] A. Moffat and W. Webber and J. Zobel and R. Baeza-Yates. A pipelined architecture for distributed text query

evaluation. Information Retrieval 10(3): 205-231, 2007. Issn 1386-4564.

[Puppin2007] D. Puppin, F. Silvestri, R. Perego, and R. Baeza-Yates. Load-balancing and caching for collection selection architectures. In INFOSCALE, pp. 1-10, 2007.

[Puppin2009] D. Puppin, F. Silvestri, R. Perego, and R. Baeza-Yates. Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load. To appear in TOIS, 2009.

[RB1998] R. Ribeiro-Neto and R. Barbosa. Query performance for tightly coupled distributed digital libraries. ACM Conference on Digital Libraries, pp. 182-190, 1998.

[Reynolds2003] P. Reynolds and A. Vahdat. Efficient Peer-to-Peer Keyword Searching. In Proc. Middleware pp.21-40, 2003.

[Risvik03] K. Risvik, Y. Aasheim, and M. Lidal. Multi-tier architecture for web search engines. In First Latin American Web Congress, pages 132–143, 2003.

[Stan90] C. Stanfill. Partitioned posting files: a parallel inverted file structure for information retrieval. 13th Conference on Research and Development in Information Retrieval (SIGIR). pp 413-428, 1990

[Suel03] T. Suel, C. Mathur, J. Wen Wu, J. Zhang, A. Delis, M. Kharrazi, X. Long, and K. Shanmugasundaram. ODISSEA: A peer-to-peer architecture for scalable web search and information retrieval. In WWW'03, pp. 67-72, 2003.

[SuelWWW09] Q. Gan, T. Suel. Improved Techniques for Result Caching in Web Search Engines. In WWW, pp. 431-440, 2009.

[Tang2004] C. Tang and S. Dwarkadas. Hybrid global-local indexing for efficient peer-to-peer information retrieval, 1st conference on Symposium on Networked Systems Design and Implementation, 2004. pp 16-26.

[TOM96] A. Tomasic and H. García-Molina. Performance issues in distributed shared-nothing information-retrieval systems. Information Processing & Management. 32(6):647-665, 1996.

[Val90] L. Valiant. A bridging model for parallel computation. Communication of the ACM, Vol 33. pp 103-111. 1990.

[Xi02b] W. Xi and O. Sornil and M. Luo and E.A. Fox, Hybrid Partition Inverted Files: Experimental Validation. 6th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 02), pp. 422-431, London, UK, 2002

[Zhang2007] J. Zhang and T. Suel. Optimized inverted list assignment in distributed search engine architectures. In IPDPS, pp. 1-10, 2007.

[Zhang2008] J. Zhang, X. Long, and T. Suel, Torsten. Performance of compressed inverted list caching in search engines. In Proc. WWW08, pp. 387-396, 2008.