CORE
Metadata, citation and similar papers at core.ac.uk
Provided by El Servicio de Difusión de la Creación Intelectual

Grandes Datos y Algoritmos Eficientes para Búsquedas de Escala Web

Gabriel H. Tolosa[1,2], Santiago Banchero[1], Esteban A. Ríssola[1],
Tomás Delvechio[1] y Esteban Feuerstein[2]
{tolosoft, sbanchero, earissola, tdelvechio}@unlu.edu.ar; efeurest@dc.uba.ar

[1]Departamento de Ciencias Básicas, Universidad Nacional de Luján
[2]Departamento de Computación, FCEyN, Universidad de Buenos Aires

Resumen

El acceso a la información en tiempo y forma es un factor esencial en muchos procesos que ocurren en dominios diferentes: la academia, la industria, el entretenimiento, entre otros. En la actualidad, el enfoque más general para acceder a la información en la web es el uso de motores de búsqueda de gran escala. Éstos sistemas enfrentan constantes desafíos debido al crecimiento explosivo de contenido en la web y también de la cantidad de nuevos usuarios. Principalmente, aparecen nuevas necesidades de almacenamiento y procesamiento para satisfacer estrictas restricciones de tiempo: las consultas deben ser respondidas en pequeñas fracciones de tiempo, típicamente, milisegundos.

Esta problemática tiene aún muchas preguntas abiertas y – mientras se intentan resolver cuestiones – aparecen nuevos desafíos. Existen necesidades puntuales de los servicios que recolectan y utilizan esta información tal como nuevas estructuras de datos y algoritmos altamente eficientes lo que brinda oportunidades únicas para avances científico/tecnológicos en áreas como algoritmos, estructuras de datos, sistemas distribuidos y procesamiento de datos a gran escala, entre otras.

En este proyecto se estudian, proponen, diseñan y evalúan estructuras de datos y algoritmos eficientes junto con el análisis de grandes datos que permitan aumentar procesos internos de un motor de búsqueda con el objetivo de mejorar su performance y escalabilidad.

Palabras clave: motores de búsqueda, estructuras de datos, algoritmos eficientes, grandes datos.

Contexto

Esta presentación se encuentra enmarcada en el proyecto de investigación "Algoritmos Eficientes y Minería Web para Recuperación de Información a Gran Escala" del Departamento de Ciencias Básicas (UNLu) en el cual los autores son integrantes (Disp. CD-CB Nº 327/14). Complementariamente, el primer autor desarrolla su tesis de doctorado en el Depto. de Computación de la FCEyN (UBA) en esta temática.

Introducción

En los últimos años el número y complejidad de documentos en la web ha crecido exponencialmente, convirtiéndola en el mayor repositorio de información en el mundo. Esto crea nuevas necesidades de almacenamiento, procesamiento y búsquedas, expandiendo los límites del trabajo en una sola máquina y unos pocos algoritmos al trabado distribuido, paralelo y altamente eficiente. En este escenario existen por un lado, necesidades puntuales de los servicios que recolectan y utilizan información de la más diversa y compleja y por el otro, aparecen oportunidades únicas para avances científico/tecnológicos en áreas como algoritmos, estructuras de datos, sistemas distribuidos y procesamiento de datos a gran escala.

El acceso a la información en tiempo y forma es un factor esencial en muchos procesos que ocurren en dominios diferentes: la academia, la industria, el entretenimiento, entre otros. En la actualidad, el enfoque más general para acceder a la información en la web es el uso de motores de búsqueda, a partir de consultas basadas en las necesidades de información de los usuarios. De forma simple, los motores

de búsqueda intentan satisfacer la consulta de los usuarios realizando procesos de recuperación sobre una porción del espacio web que "conocen", es decir, que han recorrido, recopilado y procesado [2]. Esto es así dado que existen porciones de la web a las cuales no se puede acceder debido a su dinámica o restricciones de acceso. De aquí que su tamaño no se puede determinar de manera precisa[1]. Además, el número de usuarios crece permanentente [25] y éstos no solamente *buscan* en la web para satisfacer sus necesidades de información sino que - además - realizan tareas cotidianas (por ejemplo, organizar un viaje, comprar cosas, etc.). Además, todas estas aplicaciones operan con estrictas restricciones de tiempo: las consultas deben ser respondidas en pequeñas fracciones de tiempo, típicamente, milisegundos. Los motores de búsqueda se han convertido en herramientas indispensables en la Internet actual y las cuestiones relacionadas con su eficiencia (escalabilidad) y eficacia son temas de muy activa investigación [4].

En su arquitectura interna, las máquinas de búsqueda de gran escala presentan un grado de complejidad desafiante [6], con múltiples oportunidades de optimización. Como la web es un sistema dinámico que en algunos casos opera en tiempo real, las soluciones existentes dejan de ser eficientes y aparecen nuevas necesidades. Paralelamente, en los últimos años se ha popularizado el uso de técnicas estadísticas y de *machine learning* para lograr extraer modelos útiles a partir de repositorios de datos [14] complejos. Esta disciplina, conocida como minería de datos (o minería web en este contexto), es una etapa de un proceso más complejo, el de descubrimiento de conocimiento, que puede ser altamente útil en el ámbito de los motores de búsqueda [24].

Además, el crecimiento de los repositorios y de las diferentes fuentes de generación de información (redes sociales, sensores, etc.) han agregado mayor complejidad y la necesidad de dar respuestas en tiempo real. Se ha redoblado la apuesta y gran parte de los problemas que se trataban desde la óptica de la minería de datos pasaron a ser problemas de "Grandes Datos" [27] (Big Data), donde las soluciones a éstos son significativamente más complejas ya que los volúmenes de información son muy grandes, llegan de manera continua y requieren respuestas en tiempo real [20].

Los problemas de Grandes Datos requieren de soluciones complejas que sobre arquitecturas que puedan escalar de manera flexible [21], tanto en cómputo como almacenamiento. En las grandes organizaciones el conocimiento y manejo de grandes volúmenes de datos permite tomar mejores decisiones a partir de evidencia, es decir, algunas soluciones surgen de los datos (*data driven*) y no de la intuición [16].

Las técnicas para descubrimiento de conocimiento son transversales a cualquier disciplina científica, por lo que se considera que existe un amplio abanico de soluciones de optimización aún no exploradas para los motores de búsqueda a gran escala que pueden ser tratadas siguiendo una metodología de minería de datos. Principalmente, en problemáticas que abarcan desde el análisis profundo de query logs en buscadores y query recommendation hasta políticas para la optimización de caches [24].

Líneas de investigación y desarrollo

En este proyecto se continúan líneas de I+D del grupo que incorporan análisis de grandes datos para rediseñar algunos procesos internos de un motor de búsqueda web que permitan aumentar sus prestaciones. Existen oportunidades de investigación en temas poco explorados por la comunidad científica que permiten mejorar y/o rediseñar los algoritmos internos y las estructuras de datos usadas principalmente para recuperacion de información de gran escala. En especial, las líneas de I+D principales son:

a. Estructuras de Datos

1. Distribuidas: Los sistemas de búsqueda en texto utilizan como estructura de datos básica un índice invertido. De forma simple, este índice está formado por un vocabulario (V) con todos los posibles términos de búsqueda y un conjunto de *posting lists*, L, con información acerca de los documentos donde aparece cada término junto con datos extra (por ejemplo, la frecuencia del término i en el documento j). Como los sistemas de búsqueda a gran escala se ejecutan en clusters de computadoras, es necesario distribuir los documentos entre los nodos. Para ello, los dos enfoques clásicos [2] son:

[1] De acuerdo al sitio World Wide Web Size (http://www.worldwidewebsize.com/), se estiman unos 48.000 millones de documentos en la web superficial (accedida por los motores de búsqueda)

*** Particionado por documentos:** La colección, C, es dividida entre P procesadores, los cuales almacenan solo una porción del índice $\frac{C}{P}$.

*** Particionado por términos:** Cada nodo mantiene información de las listas de posting completas de solamente un subconjunto de los términos. El vocabulario V es dividido entre los P nodos y a cada uno de éstos se le asignan $\frac{V}{P}$ listas.

*** Estrategias híbridas:** En estos modelos, como el índice 2D introducido en [9] y el 3D en [8], el índice se particiona por términos y documentos al mismo tiempo (2D) e incluso, agregando réplicas (3D). La idea de estas arquitecturas es explotar el *trade-off* entre los costos de comunicación y procesamiento que se requieren para resolver consultas.

Además, los nodos de un motor de búsqueda almacenan su porción del índice en memoria (total o parcialmente), lo que modifica los modelos de costos. Resultados experimentales muestran que es posible obtener mejoras si se incorpora la arquitectura del cluster (cantidad de nodos, procesadores y núcleos) en la optimización.

2. Escalables: Para tratar con el crecimiento en la cantidad de información que generan algunos servicios como los sitios de microblogging y redes sociales, junto con la necesidad de realizar búsquedas en tiempo real, son necesarios algoritmos y estructuras de datos escalables. Los aspectos principales a tener en cuenta en este escenario son la tasa de ingestión de documentos, la disponibilidad inmediata del contenido y el predominio del factor temporal [3] [1].

Para satisfacer estas demandas, resulta indispensable mantener el índice invertido en memoria principal. Dado que este es un recurso limitado, se trata de mantener solamente aquella información que permita alcanzar prestaciones de efectividad razonables (o aceptables) [5].

Un primer aporte del grupo [20] consiste en un conjunto de invalidadores de entradas en el índice invertido. Siguiendo esta línea, se propone el desarrollo de una familia de algoritmos de invalidacion y poda selectiva (dinámica) [17] de las entradas en el índice, tanto a nivel del vocabulario como de las *posting lists*. Esto se logra con estrategias que monitorizen la evolución y dinámica del vocabulario.

3. Algoritmos Eficientes: Entre las técnicas más utilizadas para mejorar la performance en motores de búsqueda a gran escala se encuentran las técnicas de *caching*, las cuales se basan en la idea fundamental de almacenar en una memoria de rápido acceso los ítems que van a volver a aparecer en un futuro cercano, de manera de obtenerlos sin incurrir en acceso a disco.

En una arquitectura típica de un motor de búsqueda se implementan caches para resultados [18], posting lists [28], intersecciones [15] y documentos [22]. Nuestro grupo se enfoca en el problema de las intersecciones. Aquí se propone diseñar políticas de admisión y reemplazo que consideren el costo de ejecutar una consulta [10], y no solamente *hit-ratio*. Esta línea es particularmente interesante en escenarios actuales ya que cuando el índice invertido se encuentra en memoria principal el cache de listas pierde sentido. Por otro lado, integrar diferentes caches permite optimizar el uso de espacio, lo que impacta positivamete en las prestaciones [23].

Otra dirección posible es tratar de optimizar la estrategia de caching incorporando información proveniente de redes sociales. En trabajos previos del grupo se ha mostrado que los temas que son tendencia en redes sociales guardan relación con el aumento de la popularidad de una consulta relacionada al mismo [19] y permiten mejorar la performance del cache. Esta línea de trabajo es prometedora ya que el uso de esta clase de información ha mostrado resultados positivos en otros ámbitos (por ejemplo, para mejorar el rendimiento de CDNs).

b. Grandes Datos en Motores de Búsqueda

Los motores de búsqueda son probablemente uno de los primeros ejemplo del uso de Grandes Datos. Las demandas de recolección de documentos, almacenamiento, analisis, gestión y búsqueda requieren de sofisticados algoritmos que operan sobre arquitecturas paralelas y distribuidas. Además, la información generada por las búsquedas de los usuarios (consultas, clicks, etc.) se convierte en información muy valiosa a partir de la cual es posible encontrar patrones de comportamiento y obtener estadísticas acerca de cómo los usuarios interactúan con los buscadores. Algunos trabajos [11] [12] ya mostraron la potencialidad de estas técnicas.

Esta propuesta global propone optimizar procesos internos de un buscador por lo que se considera que existen oportunidades de optimización que abren nuevos problemas y temas de investigación.

c. Indexación Distribuida

Este es un problema real en una máquina de búsqueda de escala web. Los documentos son procesados de forma distribuida y el resultado final debe ser un índice invertido particionado por algún criterio (como se mencionó anteriormente) que pueda ser implementado en un cluster. En los últimos años, además, se han propuesto nuevas estructuras de datos avanzadas que ofrecen un mejor rendimiento en la recuperación (en algunos contextos), como Block-Max [7] y Treaps [13].

Esta línea de investigación se centra en estudiar, diseñar y evaluar algoritmos de construcción de índices sofisticados como los mencionados utilizando estrategias comunmente utilizadas en el ámbito de Grandes Datos (por ejemplo, sobre el framework Hadoop [26]) y tratar de determinar cómo influyen algunos parámetros como el tamaño de la colección y la arquitectura del cluster a utilizar.

Resultados y objetivos

El objetivo principal en este proyecto es estudiar el problema de las búsquedas a gran escala e incorporar el análisis de datos masivos para mejorar las prestaciones de los sistemas de búsqueda. Para ello, se pretende desarrollar, aplicar, validar y transferir modelos, algoritmos y técnicas que permitan construir herramientas y/o arquitecturas para abordar algunas de las problemáticas relacionadas con las búsquedas en Internet, en diferentes escenarios, donde la masividad, velocidad y variedad de los datos es una característica. Se propone profundizar sobre el estado del arte y proponer nuevos enfoques, en particular:

a) Diseñar estructuras de datos eficientes con base en los modelos recientemente propuestos, para soportar índices invertidos distribuidos (en memoria primaria o secundaria).

b) Utilizar análisis de datos masivos para optimizar los algoritmos de búsquedas, a partir de comprender de mejor manera la dinámica de las consultas y sus costos asociados.

c) Diseñar nuevas técnicas de caching complementándolas con información del análisis de redes sociales para mejorar tanto las políticas de admisión como las de reemplazo.

d) Diseñar y evaluar estrategias de indexación distribuida para estructuras de datos avanzadas usando frameworks del ecosistema de Grandes Datos.

e) Diseñar arquitecturas para aplicaciones específicas de búsquedas ad-hoc para problemas concretos, donde una solución de propósito general no es la más eficiente.

f) Adaptar y transferir las soluciones a diferentes dominios de aplicación como motores de búsqueda de propósito general, verticales y empresariales, redes sociales y servicios móviles, principalmente.

Formación de Recursos Humanos

Este proyecto brinda un marco para que algunos docentes auxiliares y estudiantes lleven a cabo tareas de investigación y se desarrollen en el ámbito académico. Junto con el doctorado del primer autor hay en finalización una tesis de la maestría en "Exploración de Datos y Descubrimiento de Conocimiento", DC, FCEyN, Universidad de Buenos Aires.

Actualmente, se están dirigiendo cuatro trabajos finales correspondientes a la Lic. en Sistemas de Información de la Universidad Nacional de Luján en temas relacionados con el proyecto. Además, hay dos pasantes alumnos y se espera dirigir al menos dos estudiantes más por año y presentar dos candidatos a becas de investigación.

Referencias

[1] N. Asadi, J. Lin, and M. Busch. Dynamic memory allocation policies for postings in real-time twitter search. *CoRR*, abs/1302.5302, 2013.

[2] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval - the concepts and technology behind search, 2nd ed.* Pearson Education Ltd., 2011.

[3] M. Busch, K. Gade, B. Larson, P. Lok, S. Luckenbill, and J. Lin. Earlybird: Real-time search at twitter. In *Proceedings of the 28th International Conference on Data Engineering*, ICDE '12. IEEE Computer Society, 2012.

[4] B. B. Cambazoglu and R. A. Baeza-Yates. Scalability and efficiency challenges in large-scale web search engines. In *Proceedings of the*

Eighth ACM International Conference on Web Search and Data Mining, WSDM, 2015.

[5] C. Chen, F. Li, B. C. Ooi, and S. Wu. Ti: An efficient indexing mechanism for real-time search on tweets. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. ACM, 2011.

[6] B. Croft, D. Metzler, and T. Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, 1st edition, 2009.

[7] S. Ding and T. Suel. Faster top-k document retrieval using block-max indexes. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11. ACM, 2011.

[8] E. Feuerstein, V. G. Costa, M. Marín, G. Tolosa, and R. A. Baeza-Yates. 3d inverted index with cache sharing for web search engines. In *18th International Conference, Euro-Par 2012, August 27-31, 2012.*, 2012.

[9] E. Feuerstein, M. Marín, M. J. Mizrahi, V. G. Costa, and R. A. Baeza-Yates. Two-dimensional distributed inverted files. In *16th International Symposium of String Processing and Information Retrieval, SPIRE'09, August 25-27*, 2009.

[10] E. Feuerstein and G. Tolosa. Cost-aware intersection caching and processing strategies for in-memory inverted indexes. In *In Proc. of 11th Workshop on Large-scale and Distributed Systems for Information Retrieval*, LSDS-IR'14, 2014.

[11] Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, and Q. Zheng. Mining query subtopics from search log data. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012.

[12] P. Kaushik, S. Gaur, and M. Singh. Use of query logs for providing cache support to the search engine. In *International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2014.

[13] R. Konow, G. Navarro, C. L. Clarke, and A. López-Ortíz. Faster and smaller inverted indices with treaps. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13. ACM, 2013.

[14] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[15] X. Long and T. Suel. Three-level caching for efficient query processing in large web search engines. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 2005.

[16] A. McAfee and E. Brynjolfsson. Big data: the management revolution. *Harvard business review*, (90), 2012.

[17] A. Ntoulas and J. Cho. Pruning policies for two-tiered inverted index with correctness guarantee. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

[18] R. Ozcan, I. S. Altingovde, and O. Ulusoy. Cost-aware strategies for query result caching in web search engines. *ACM Trans. Web*, 5(2), May 2011.

[19] S. Ricci and G. Tolosa. Efecto de los trending topics en el volumen de consultas a motores de búsqueda. In *XVII Congreso Argentino de Ciencias de la Computación, CACIC.*, 2013.

[20] E. Rissola and G. Tolosa. Inverted index entry invalidation strategy for real time search. In *Proceedings of the XXI Congreso Argentino en Ciencias de la Computación*, CACIC '15, 2015.

[21] E. E. Schadt, M. D. Linderman, J. Sorenson, L. Lee, and G. P. Nolan. Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics*, 11(9), 2010.

[22] T. Strohman and W. B. Croft. Efficient document retrieval in main memory. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.

[23] G. Tolosa, L. Becchetti, E. Feuerstein, and A. Marchetti-Spaccamela. Performance improvements for search systems using an integrated cache of lists+intersections. In *Proceedings of 21st International Symposium of String Processing and Information Retrieval, SPIRE'14*, 2014.

[24] G. Tolosa and E. Feuerstein. Using big data analysis to improve cache performance in search engines. In *Proceedings of the Simposio Argentino de GRANdes DAtos (1st ed.) at 44 JAIIO - 44th Argentine Conference on Informatics*, AGRANDA '15, 2015.

[25] A. Trotman and J. Zhang. Future web growth and its consequences for web search architectures. In *CoRR, vol abs/1307.1179, 2013*, 2013.

[26] T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, Inc., 1st edition, 2009.

[27] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding. Data mining with big data. *Knowledge and Data Engineering, IEEE Transactions on*, 26(1), 2014.

[28] J. Zhang, X. Long, and T. Suel. Performance of compressed inverted list caching in search engines. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08. ACM, 2008.