Algoritmos Eficientes para Búsquedas a Gran Escala Integrando Datos Masivos

Gabriel H. Tolosa[1,2], Santiago Banchero[1], Esteban A. Ríssola[1], Tomás Delvechio[1], Santiago Ricci[1] y Esteban Feuerstein[2]
{tolosoft, sbanchero, earissola, tdelvechio, sricci}@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 crecimiento explosivo de contenido en la web crea nuevas necesidades de almacenamiento, procesamiento y propone múltiples desafíos a los sistemas de búsquedas. Por un lado, existen necesidades puntuales de los servicios que recolectan y utilizan esta información 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, entre otras.

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. Éstos son sistemas distribuidos de altas prestaciones que se basan en estructuras de datos y algoritmos altamente eficientes ya que operan bajo 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 .

En este proyecto se estudian y evalúan estructuras de datos y algoritmos eficientes junto con el análisis de datos masivos para mejorar procesos internos de un motor de búsqueda.

Palabras clave: motores de búsqueda, estructuras de datos, algoritmos eficientes, datos masivos, big data.

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 N° 327/14). Complementariamente, el primer autor desarrolla su tesis de doctorado en el Depto. de Computación de la FCEyN (UBA) en esta temática.

Introducción

El crecimiento explosivo de contenido en la web crea nuevas necesidades de almacenamiento, procesamiento y búsquedas. Por un lado, existen necesidades puntuales de los servicios que recolectan y utilizan esta información 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, entre otras.

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 mas general para acceder 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 [3]. Pero, además, estas aplicaciones operan con estric-

tas restricciones de tiempo: las consultas deben ser respondidas en pequeñas fracciones de tiempo, típicamente, milisegundos.

La cantidad, diversidad y dinamismo en la información disponible en la web es cada día más compleja [29], lo que exige que se investiguen y desarrollen nuevas ideas, modelos y herramientas computacionales que permitan satisfacer más eficientemente las necesidades de acceso, tanto desde la perspectiva de tiempo y espacio como de precisión en los resultados. 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 [5].

En su arquitectura interna, las máquinas de búsqueda de gran escala presentan un grado de complejidad desafiante [7], 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 [17] complejos. Esta disciplina, conocida como minería de datos, es una etapa de un proceso más complejo, el de descubrimiento de conocimiento.

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 Big Data [31]. Donde las soluciones a estos problemas son significativamente más complejas ya que los volúmenes de información son muy grandes, llegan de manera continua y requieren respuestas en tiempo real. Los problemas de big data requieren de soluciones más complejas que involucran cómputo paralelo, almacenamiento distribuido, en otras palabras, necesitan de arquitecturas que puedan escalar de manera flexible [25]. En las grandes organizaciones el conocimiento y manejo de big data en manos de los tomadores de decisiones permite que estos actúen a partir de evidencia, es decir que las soluciones surjan de los datos (*data driven*) y no a través de la intuición [20].

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.

Líneas de investigación y desarrollo

Este proyecto continúa líneas de I+D iniciadas por el grupo y propone la incorporación de técnicas de análisis de datos masivos (algunas provenientes del área de mineria web y actualmente sobre conceptos de Big Data) para mejorar los procesos internos de un motor de búsqueda web. Existen múltiples oportunidades de investigación en temas no explorados aún que permiten mejorar y/o rediseñar los algoritmos internos y las estructuras de datos usadas principalmente para recuperacion de información de gran escala. En particular, las líneas de I+D principales son:

a. Estructuras de Datos

a.1. Distribuidas

Los sistemas de recuperación de información utilizan el índice invertido como estructura de datos básica. De forma simple, esta contiene un vocabulario (V) con todos los términos extraídos de los documentos y – asociada a éstos – una lista de los documentos (posting list) donde aparece dicho término (junto con información adicional). 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 [3] son:

*** Particionado por documentos:** El conjunto de documentos (C) es dividido entre P procesadores, los cuales almacenan una porción del índice $\frac{C}{P}$. En esta estrategia todos los nodos participan de la resolución de la consulta.

*** Particionado por términos:** Cada nodo mantiene información de las listas de posting completas de solamente un subconjunto de los términos. De la forma más trivial, el vocabulario V es dividido entre

los P nodos y a cada uno de éstos se le asignan $\frac{V}{P}$ listas. Para la resolución de la consulta solo participan aquellos nodos que poseen la información de los términos involucrados.

Esquemas más sofisticados como los enfoques híbridos denominados índice 2D [11] y 3D [10] también son posibles. En el primer caso, se organizan los P procesadores en un array bidimensional (C columnas $\times R$ filas) en el cual se aplica el particionado por documentos en cada columna y el particionado por términos a nivel de filas. Los resultados de esta estrategia muestran que se pueden obtener mejoras si se selecciona adecuadamente el número de filas y columnas del array. Esto se debe a que existe un *trade-off* entre los costos de comunicación y procesamiento que se requieren para resolver un conjunto de consultas. El índice 3D agrega una dimensión (D) de procesadores que trabajan como réplicas.

Sobre la arquitectura 2D se está realizando una implementación optimizada que incorpora para su configuración óptima la arquitectura del cluster: cantidad de nodos, de procesadores por nodo y núcleos por procesador, principalmente.

a.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 [4] [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) [6].

En esta línea, se propone el desarrollo de algoritmos y estrategias que monitorizen cómo evoluciona el vocabulario de esta clase de servicios con el fin de que - selectivamente – se invaliden y desalojen aquellas entradas que no aportan sustancialmente o de forma perceptible la efectividad en la búsqueda. Para ello se proponen estrategias de poda dinámica [21] y de invalidación de entradas adaptadas a este escenario.

b. Algoritmos Eficientes para Búsquedas

Entre los enfoques mas utilizados para aumentar la performance en motores de búsqueda a gran escala se encuentran las técnicas de caching. De forma simple, se basan en la idea fundamental de almacenar en una memoria de rápido acceso los ítems (objetos) que van a volver a aparecer en un futuro cercano, de manera de poder obtenerlos desde ésta sin incurrir en costos de CPU o acceso a disco.

Típicamente, se implementan caches para resultados de búsqueda [22], listas de posting [32], intersecciones [18] y documentos [26]. Si bien se han propuesto diversos enfoques para cada caso, ninguno está completamente resuelto y aún existen oportunidades de optimización [19].

En el caso de las intersecciones, se propone diseñar políticas de admisión y reemplazo que consideren el costo de ejecutar una consulta, y no solamente *hit-ratio* (métrica habitualmente utilizada) [12]. En este sentido, combinar información de comportamiento del stream de consultas permite optimizar el algoritmo que implementa la política. 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 [28].

Por otro lado, se puede optimizar la estrategia usada en caching de resultados incorporando información proveniente de redes sociales. 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). Además, en trabajos anteriores [23] se han obtenido indicios de que los temas que son tendencia en redes sociales guardan relación con el aumento de la popularidad de una consulta relacionada al mismo [24]. Además, en otros trabajos se ha observado que las características de las tendencias pueden funcionar como buen indicador sobre los temas que las personas están buscando en la Web en un determinado momento [16, 27, 2].

c. Big Data en Motores de Búsqueda

Como se ha mencionado, existe información muy valiosa en los logs de queries de los motores de búsqueda y a partir de ella es posible, a través de técnicas de descubrimiento de conocimiento, en-

contrar patrones de comportamiento y obtener estadísticas acerca de cómo los usuarios interactúan con los buscadores. Esto es de mucha utilidad ya que permite mejorar las tasas de acierto, obtener mejoras en el rendimiento y en los tiempos de respuestas. En [13] se utiliza análisis por clusters sobre los logs de consultas para determinar cuáles son los subtemas, o las facetas, que pueden ser de interés para el usuario a partir de los clics que este realiza en los enlaces retornados de la consulta realizada. Por otro lado, [14] utiliza otra conocida técnica de la mineria de datos como son las reglas de asociación para mejorar el tiempo de respuesta. Del análisis de los archivos de log se extraen reglas que permiten encontrar consultas muy relacionadas entre si.Para este subconjunto de consultas se propone generar una cache parcial dinámica.

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.

d. Indexación Distribuida

Sobre el índice invertido clásico se han desarrollado estructuras avanzadas, que en determinados contextos, presentan un mejor rendimiento en la recuperación. En un escenario dinámico, con la cantidad y variedad de información a procesar en la web, una evolución posible es incorporar estrategias comunmente utilizadas en el ámbito de Big Data a los procesos de construcción y recuperación de índices. Un ejemplo clásico es el paradigma MapReduce [8] y el framework Hadoop [30].

En este contexto, resulta de interés evaluar estrategias de indexación distribuida sobre estructuras de índice avanzadas como Block-Max [9] y Treaps [15].

Se requiere adaptar, analizar y comparar los diversos algoritmos de construcción de índices y determinar como influye la variación del tamaño de la colección, la cantidad de tareas (mappers/reducers), y la cantidad de nodos en el cluster.

Resultados y objetivos

El objetivo principal de este proyecto, el cual es continuación de los trabajos del grupo, es estudiar, 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.

Se pretende optimizar los procesos de búsqueda en casos donde la masividad, velocidad y variedad de los datos es una característica. Se propone profundizar sobre el estado del arte y proponer nuevos enfoques incorporando el análisis de datos masivos a los procesos internos de los motores de búsqueda. En particular:

a) Diseñar estructuras de datos eficientes con base en los modelos recientemente propuestos, aplicando alguna de las ideas anteriormente descriptas.

b) Optimizar los algoritmos de búsquedas incorporando estrategias provenientes de la extracción de relaciones entre los objetos del sistema mediante procesos de análisis de datos masivos.

c) Diseñar nuevas técnicas de caching, incorporando la información del análisis de redes sociales a las políticas de admisión y reemplazo.

d) Diseñar y evaluar estrategias de indexación distribuida para estructuras de datos avanzadas usando frameworks del área de Big Data.

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, Buscadores verticales, Redes Sociales y Búsquedas móviles.

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 curso una tesis de la maestría en "Exploración de Datos y Descubrimiento de Conocimiento", DC, FCEyN, Universidad de Buenos Aires.

Actualmente, se están dirigiendo tres trabajos finales correspondientes a la Lic. en Sistemas de Información de la Universidad Nacional de Luján en temas relacionados con el proyecto. Además, hay un Becario CIN (Becas Estímulo a las Vocaciones Científicas) y dos pasantes alumnos. Se espera dirigir al menos dos estudiantes más por año e incorporar al menos otro pasante al grupo dentro del proyecto principal.

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] S. Asur, B. A. Huberman, G. Szabó, and C. Wang. Trends in social media : Persistence and decay. *CoRR*, abs/1102.1402, 2011.

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

[4] M. Busch, K. Gade, B. Larson, P. Lok, S. Luckenbill, and J. Lin. Earlybird: Real-time search at twitter. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 1360–1369. IEEE Computer Society, 2012.

[5] 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*, pages 411–412, 2015.

[6] 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*, pages 649–660. ACM, 2011.

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

[8] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 10–10. USENIX Association, 2004.

[9] 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, pages 993–1002. ACM, 2011.

[10] 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.*, pages 272–284, 2012.

[11] 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*, pages 206–213, 2009.

[12] 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.

[13] 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*, pages 305–314. ACM, 2012.

[14] 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)*, pages 819–824. IEEE, 2014.

[15] 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, pages 193–202. ACM, 2013.

[16] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, pages 591–600. ACM, 2010.

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

[18] 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*, pages 257–266. ACM, 2005.

[19] M. Marin, V. Gil-Costa, and C. Gomez-Pantoja. New caching techniques for web search engines. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 215–226. ACM, 2010.

[20] A. McAfee and E. Brynjolfsson. Big data: the management revolution. *Harvard business review*, (90):60–6, 2012.

[21] 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*, pages 191–198, 2007.

[22] R. Ozcan, I. S. Altingovde, and O. Ulusoy. Cost-aware strategies for query result caching in web search engines. *ACM Trans. Web*, 5(2):9:1–9:25, May 2011.

[23] S. Ricci. Impacto de las redes sociales en la popularidad de las consultas a motores de búsqueda. In *In 42 Jornadas Argentinas de Informática (Trabajos Estudiantiles)*, 2013.

[24] 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.

[25] 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):647–657, 2010.

[26] 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*, pages 175–182, 2007.

[27] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: a comparison of microblog search and web search. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM'11*, pages 35–44, 2011.

[28] 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*, pages 227–235, 2014.

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

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

[31] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding. Data mining with big data. *Knowledge and Data Engineering, IEEE Transactions on*, 26(1):97–107, 2014.

[32] 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, pages 387–396. ACM, 2008.