Algoritmos Eficientes para Búsquedas A Gran Escala Integrando Datos Masivos

Gabriel Tolosa[1,2], Santiago Banchero[1], Esteban Ríssola[1], Tomás Delvechio[1], Santiago Ricci[1] y Esteban Feuerstein[2]
{tolosoft, sbanchero, earissola, tdelvechio, sricci}@unlu.edu.ar; efeuerst@dc.uba.ar
[1]Departamento de Ciencias Básicas, Universidad Nacional de Luján
[2] Departamento de Computación, FCEyN, Universidad de Buenos Aires

Introducción

El acceso a Información en Internet plantea múltiples desafíos debido a su cantidad, diversidad y dinamismo [2]. Los datos son cada vez mas ricos, complejos y se utilizan y varían en tiempo real. Los motores de búsqueda se han vuelto indispensables en este escenario.

La arquitectura interna de una máquina de búsqueda de gran escala presenta un grado de complejidad desafiante pero – además – múltiples oportunidades de optimización [7]. Como operan sobre un sistema dinámico y en constante evolución, las soluciones existentes pueden ya no ser eficientes a futuro y nuevas necesidades aparecen constantemente. Sin embargo, dos requerimientos son indispensables: eficiencia (responder en una fracción de segundo a millones de usuarios) y efectividad (que las respuestas sean relevantes).

Contexto y Formación de RRHH

Proyecto de Investigación "*Algoritmos Eficientes y Minería Web para Recuperación de Información a Gran Escala*" del Depto. De Ciencias Básicas (UNLu) y tesis de doctorado del primer autor.

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

Dirección de tres trabajos finales correspondientes a la Lic. en Sistemas de Información de la Universidad Nacional de Luján y un Becario CIN (Becas Estímulo Vocación Científica). Dos pasantes alumnos realizan tareas con el grupo dentro del proyecto principal. Se espera dirigir al menos dos estudiantes más por año e incorporar al menos otro pasante.

Proponemos optimizar los algoritmos de búsqueda incorporando estrategias provenientes de la extracción de relaciones entre los objetos del sistema mediante procesos de análisis de datos masivos

Buscamos definir nuevas técnicas de caching, incorporando la información del análisis de RS a las políticas de admisión y reemplazo.

Planteamos el diseño de arquitecturas para aplicaciones específicas de búsquedas ad-hoc para problemas concretos, donde una solución de propósito general no es la más eficiente.

Procuramos diseñar y evaluar estrategias de indexación distribuida para estructuras de datos avanzadas usando frameworks del área de BD.

Líneas de Investigación

1) Estructuras de Datos

Distribuidas: Índices invertidos [1, 2] particionados por enfoques clásicos (documentos o términos) e híbridos (índice 2D [3] y 3D [4]), donde se organizan los nodos en un array bidimensional (C col x R filas) aplicando particionado por documentos en cada columna y por términos a nivel de filas (y se replican en el 3D).
Escalables: Mecanismos que proporcionen soporte para búsquedas en tiempo real sobre índices invertidos en memoria [5], considerando la dinámica del vocabulario para evaluar aquella información que resultará relevante a la búsqueda.

2) Algoritmos Eficientes para Búsquedas

Basados principalmente en técnicas de *caching*. En motores de búsqueda, habitualmente se implementan caches [10] para resultados de búsqueda, listas de posting, intersecciones y documentos. Incorporando complementariamente, información de redes sociales [6] (RS) para optimizar las técnicas propuestas.

3) Big Data en Motores de Búsqueda

Utilizar la información presente en logs de consultas de motores de búsqueda para encontrar patrones de comportamiento empleando técnicas de descubrimiento del conocimiento [7, 8], con el fin de aplicar dicha información para para optimizar procesos internos de un buscador.

4) Indexación Distribuida

Incorporar estrategias comunes al ámbito de Big Data [9, 10], a los procesos de construcción y recuperación de índices, con particular interés en la evaluación de técnicas de indexación distribuida sobre estructuras de índice avanzadas [11, 12].

Referencias

[1] C. Badue, R. Baeza-yates, B. Ribeiro-Neto, N. Ziviani. Distributed query processing using partitioned inverted files. SPIRE, 2001.
[2] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval: The Concepts and Technology behind Search (2nd Ed). Addison-Wesley Professional. 2011.
[3] E. Feuerstein, M. Marín, M. Mizrahi, V. Gil Costa y R. A. Baeza-Yates. Two-dimensional distributed inverted files. SPIRE, 2009.
[4] Feuerstein, E; Gil-Costa, V.; Marin, M.; Tolosa G. y Baeza-Yates, R. 3D Inverted Index with Cache Sharing for WSE. Euro-Par, 2012.
[5] Busch, M., Gade, K., Larson, B., Lok, P., Luckenbill, S., Lin, J., 2012. Earlybird: Real-time search at twitter. ICDE '12.
[6] Ricci, S., and Tolosa, G. Efecto de los trending topics en el volumen de consultas a motores de búsqueda. CACIC, 2013.
[7] Hu, Y., Qian, Y., Li, H., Jiang, D., Pei, J., Zheng, Q. Mining query subtopics from search log data. SIGIR, 2012.
[8] Kaushik, P., Gaur, S., Singh, M. Use of query logs for providing cache support to the search engine. INDICAcom, 2014.
[9] Dean, J. and Ghemawat, S. Mapreduce: Simplified data procesessing on large clusters. OSDI '04.
[10] White, T. Hadoop: The definitive Guide. O'Reilly Media, Inc., 1st edition, 2009.
[11] Ding, S., Suel, T. Faster top-k document retrieval using block-max indexes. SIGIR, 2011.
[12] Konow, R., Navarro, C., Clarke, C. L., López.Ortíz, A. Faster and smaller inverted indices with treaps. SIGIR, 2013.