Mejoras Algorítmicas para Problemas de Búsquedas en Datos Masivos

Gabriel Tolosa[1,2], Tomás Delvechio[1], Pablo Lavallén[1], Andrés Giordano[1], Agustín González[1], Claudia Reinaudi[2], Santiago Ricci[1], Tomás Juran[1,2], Esteban A. Ríssola[1,3]
{tolosoft, tdevechio, plavallen, agiordano, agonzalez, creinaudi, sricci, tjuran}@unlu.edu.ar
esteban.andres.rissola@usi.ch

[1]Departamento de Ciencias Básicas, Universidad Nacional de Luján
[2]CIDETIC, Universidad Nacional de Luján
[3]Faculty of Informatics, Università della Svizzera italiana

Resumen

El procesamiento de datos masivos propone a diario nuevos desafíos, debido tanto a cuestiones vinculadas a los datos mismos como a la variedad de aplicaciones y soluciones que requieren los usuarios. En el primero de los casos, los datos crecen a tasas exponenciales pero también existen diversidad de nuevas fuentes a considerar, incluyendo aquellas en las cuales se producen en flujos en tiempo real. Estas características exigen mayores capacidades de hardware a los proveedores de servicios e imponen restricciones a los usuarios en la facilidad de acceso.

En este escenario, los algoritmos que resuelven problemas de búsquedas (en sentido amplio) requieren de mejoras tanto conceptuales como ingenieriles que les permitan escalar con el tamaño del problema. La eficiencia es un requerimiento fundamental para procesar datos masivos, debido al tamaño, la complejidad y la dinámica de las fuentes actuales de información digital.

Este proyecto presenta el abordaje de problemas relacionados con dos escenarios actuales. Por un lado, el procesamiento de colecciones masivas de documentos, para la construcción de motores de búsqueda de escala web. Por otro lado, el procesamiento de grafos en cuanto a las métricas de distancias, para aplicar, por ejemplo, a búsquedas de caminos más cortos entre usuarios de redes sociales. Las líneas de investigación enfatizan el estudio, diseño y evaluación de algoritmos eficientes (y estructuras de datos asociadas) que permitan aumentar las prestaciones de los sistemas de búsqueda haciendo un uso racional de los recursos de hardware.

Palabras clave: algoritmos eficientes, búsquedas web, grafos, datos masivos.

Contexto

Esta presentación se encuentra enmarcada en el proyecto de investigación “Estrategias y Algoritmos para Problemas de Búsquedas a Gran Escala” (Disposición CD-CB Nº 350/19) del Departamento de Ciencias Básicas (UNLu).

Introducción

La cantidad, variedad y velocidad a la que se produce información digital desafía día a día a los sistemas de búsquedas. Además, el número de usuarios que interactúa con diversas plataformas online también se incrementa y, en general, se deben ofrecer respuestas a estos usuarios con restricciones de tiempo. Muchas de estas respuestas están basadas en diferentes formas de *búsquedas*, ya sea sobre documentos, bases de datos estructuradas, grafos, flujos en tiempo real, entre otras. Contar con herramientas eficientes que aborden este tipo de problemas es un requerimiento [24].

La Recuperación de Información es una de las áreas de las Ciencias de la Computación que ofrece un ámbito para la investigación y abordaje de estos problemas, en particular siendo aplicada en el contexto de Datos Masivos (*Big Data* [13]). Así, es posible enfocarse en temas de representación, almacenamiento y procesamiento que permitan ofrecer a los usuarios resultados relevante en tiempo y forma [4]. Esto aplica, por ejemplo, a motores de búsqueda de escala web pero, además, muchas de sus técnicas (como indexación y compresión) se usan también junto con algoritmos que manejan grandes estructuras de datos como los grafos masivos, estructura subyacente las redes sociales.

En general, este tipo de problemas presenta características particulares como la masividad de datos (documentos que forman la web o millones de nodos/aristas de un grafo social), los tiempos de respuesta acotados, la necesidad de estructuras de datos específicas y combinaciones de algoritmos sofisticados que permitan el procesamiento eficiente, considerando también parámetros de eficacia.

Para abordar este tipo de problemas, el área de RI se complementa con técnicas de disciplinas relacionadas, lo que, por un lado, ha complejizado las soluciones pero, por el otro, ha abierto nuevos problemas y temas de investigación y transferencia para abordar. Por ejemplo, se han incorporado a la disciplina estrategias basadas en aprendizaje automático para clasificar o rankear documentos [19] y técnicas de estimación y muestreo (no aleatorio) para seleccionar porciones de un grafo masivo que puedan ser procesadas en un tiempo prudencial [2]. En este contexto, aparecen nuevas oportunidades de soluciones que exploran el *tradeoff* entre la eficacia y la eficiencia y que, además, alguna de éstas puedan derivar en soluciones ingenieriles que sean transferibles a problemas concretos.

En el caso de los motores de búsqueda de escala web, el procesamiento de consultas es uno de los desafíos más difíciles de manejar debido al constante crecimiento tanto en datos como en usuarios [15]. Mantener las prestaciones requiere de algoritmos que combinen diversas técnicas tales como poda dinámica [22], caching [20], ranking [3] (como *Learning to Rank*), compresión de las estructuras de datos [17] o selección de recursos [11], entre otras. Dadas las características de estos problemas también se requieren estrategias de distribución de la carga de trabajo optimizadas para cada caso [24].

Los problemas de eficiencia en búsquedas son continuamente identificados como uno de los más importantes en RI [7] y reciben atención permanentemente, tanto de la academia como de la industria [21]. Por lo tanto, esta propuesta considera este tipo de problemas en escenarios de datos masivos. La idea general de aumentar la eficiencia en las búsquedas permite procesar mayor cantidad de datos con menos recursos, impactando positivamente en el mantenimiento de las infraestructuras de hardware (*datacenters*) en los cuales se ejecutan estos sistemas, disminuyendo costos operativos y mejorando el impacto ambiental.

Líneas de I+D

Las líneas de I+D del grupo se enfocan a mejoras algorítmicas y representaciones basadas en colecciones de documentos (índices invertidos) o elementos relacionados (grafos).

a. Búsquedas a Gran Escala

La estructura de datos comúnmente utilizada para soportar la recuperación eficiente es el índice invertido. De forma simple, está compuesto por un vocabulario (V) con todos los términos extraídos de los documentos y, por cada uno de éstos, una lista de los documentos (*posting list*) donde aparece dicho término junto con información usada para el ranking. En el caso de los algoritmos, la eficiencia está dada por analizar la menor cantidad de documentos para satisfacer una consulta, o bien, poder seleccionar adecuadamente un subconjunto de nodos que puedan responderla.

Algoritmos para Top-k: Existen dos estrategias predominantes para recorrer un índice invertido: DAAT (Document-at-a-Time) y TAAT (Term-at-a-Time). Dado un *query* con n términos ($q = \{t_1, t_2 \dots t_n\}$), DAAT recorre las n listas en paralelo intentando determinar en qué momento detener la evaluación sin llegar al final de todas (*dynamic pruning*). En el caso de TAAT, las listas de los términos se procesan una a la vez, siguiendo la misma idea. No obstante, las dos estrategias predominantes actualmente siguen el criterio DAAT (Maxscore [22] y WAND [5]). En ambos casos, la idea subyacente es contar con un valor umbral (*upper bound*) que permita determinar en qué momento finalizar la evaluación. La evolución sobre éstas consiste en combinarlas con una estructura de índice particular basada en bloques fijos [8] o de longitud variable [14].

En esta línea se trabaja en una extensión de MaxScore en la cual se almacenan múltiples valores umbrales en una estructura similar a una *skip list* [6], dotando al algoritmo de más información para mejorar la eficiencia del procesamiento. Resultados preliminares muestran que la evaluación de documentos se puede reducir hasta un 50 % favoreciendo a consultas con términos muy populares.

Por otro lado, se aborda una propuesta que combina una estructura de datos donde la *posting list* completa está divida en dos secciones: una primera sección con documentos ordenados por la frecuencia del término en el documento y una segunda sec-

ción ordenada por identificador de documento. Cada sección se recorre usando técnicas DAAT y TAAT según corresponda, usando particiones variables en cada caso, de acuerdo a propiedades estadísticas de las listas.

Búsquedas sobre Flujos: Las búsquedas sobre flujos de información en tiempo real (como en redes sociales) desafían las arquitecturas de procesamiento distribuido, los algoritmos y las estructuras de datos empleadas en los motores de búsqueda. El reto radica en que millones de usuarios (por ejemplo, más de 300 millones en Twitter) publican *documentos cortos* desde diferentes tipos de dispositivos (generalmente, móviles). A su vez, estos documentos deben estar disponibles casi de inmediato, lo que implica una dinámica no presente en las búsquedas web clásicas. Así, este problema puede ser abordado particionando la colección en porciones denominadas *shards*, distribuyendo los mismos en los nodos de procesamiento disponibles y, a la hora de la búsqueda, enviando la consulta solo a un número reducido de nodos que sean los más *adecuados* para resolverla. A este enfoque, se lo denomina "búsquedas selectivas" (*selective search*) e implica el diseño, estudio y desarrollo de métodos y estrategias de partición de la colección, selección de los recursos adecuados, estrategias de *caching* y de fusión de resultados.

En esta línea de investigación se intenta mejorar la eficiencia en la recuperación de información de gran escala sobre flujos de documentos en tiempo real mediante el enfoque de búsquedas selectivas. Los problemas a abordar incluyen los criterios de actualización del índice invertido (particionado), implementar estrategias de caché y definir estrategias de selección de recursos para el algoritmo de búsqueda.

Compresión de Índices: El tamaño de un índice invertido es un factor a considerar no solo en cuanto al medio de almacenamiento persistente, sino también a la posibilidad de que resida en memoria principal (parcial o completamente). Aplicar técnicas de compresión específicas inciden en un menor uso de espacio y son requerimiento en sistemas de gran escala actuales. El objetivo es lograr un *tradeoff* adecuado entre espacio final y tiempo de decodificación acorde a la aplicación particular.

Existe un gran cuerpo de literatura en el tema que presenta enfoques para la compresión de *posting lists* con diferentes propiedades. Una clasificación posible es entre aquellos algoritmos (*codecs*) que se consideran *libres de parámetro* (codifican cada entero de forma individual) [17, 12, 26] y los *adaptativos a lista* (considera porciones de la lista para comprimir) [26, 12, 25, 23, 16].

En esta línea se aborda la compresión de un índice invertido en bloques pero usando múltiples codecs de acuerdo a cada porción particular y sus propiedades. Por ejemplo, se pueden combinar métodos como *Varint* (eficiente en tiempo) e *Interpolative* (eficiente en espacio). Trabajos preliminares del grupo [10] en los que se propone un esquema *multicompresión* muestran que es posible diseñar una solución de compromiso entre el espacio que ocupa un índice invertido y el tiempo de resolución de consultas, contemplando la partición de las listas tanto de forma uniforme como variable.

b. Procesamiento de Grafos

La representación de datos en estructuras de redes o grafos sigue siendo relevante dada su versatilidad para organizar datos de muy variada naturaleza. Una de las métricas más importantes que se calcula sobre grafos es la distancia entre dos nodos. Como los grafos son utilizados para representar, por ejemplo, a redes sociales (siendo los nodos los usuarios, y las aristas las relaciones), el cálculo de la distancia o la búsqueda del camino más corto (*Shortest Path*) sirve, entre otras cosas, para saber qué tan conectada se encuentra esa red. La masividad de las redes sociales actuales hace que ese cálculo sea muy costoso de realizar online y sobre la totalidad de los nodos.

Estimación de Distancias: El caso de las métricas de centralidad o de distancias en grafos, es un ejemplo donde algoritmos clásicos y que aún al día de hoy se consideran estado del arte en grafos pequeños (en relación a la baja cantidad de aristas y nodos presentes), no responden de manera óptima en entornos distribuidos y para grafos masivos. Un enfoque derivado de lo anterior es el de calcular de forma aproximada el valor de una métrica (por ejemplo, mediante métodos basados en *landmarks* [18]). Una de las líneas de trabajo se basa en métodos de cálculo aproximados que permiten estimar el valor de la distancia reduciendo el grado de error asociado [9] y mejorando los tiempos de respuestas.

Corrección de la Estimación: Complementando la línea anterior, cuando se llega a los límites en la performance de un método de estimación de la distancia es posible aplicar estrategias de corrección

de la misma. Esto se puede realizar, por ejemplo, mediante el ajuste de funciones que modelen la distribución de los errores y permitan corregir la estimación inicial. Continuando una línea de trabajo del grupo [9] se busca obtener la o las funciones que mejor ajusten los valores de distancia estimados con los reales, o incluso combinaciones de funciones según intervalos de error. Se proponen nuevas funciones de ajuste y su aplicación en datasets heterogéneos, intentando establecer si alguna característica estructural del grafo influye en el valor del error. Además, se busca comprender cómo utilizar esta información para una mejor corrección.

Estrategias de Partición y Procesamiento: Una forma eficiente de abordar el procesamiento de grafos masivos es hacerlo mediante procesamiento distribuido. Para esto, una opción es particionar el grafo en porciones que puedan ser derivadas a diferentes nodos de un cluster. El problema de la partición de un grafo no es nuevo, pero el área se mantiene activa debido a los múltiples factores a considerar [1]. Por último, una problemática que resulta de interés es la de calcular estas métricas en grafos que pueden evolucionar en el tiempo (grafos dinámicos). El desafío aquí consiste en lograr métodos de cálculo que puedan reutilizar procesamientos previos y no tener que realizar cálculos completos (dada la escala del grafo). En esta línea se investigan estrategias de partición que permitan realizar el cálculo de métricas de un grafo que evoluciona en el tiempo minimizando el intercambio de datos.

Resultados y objetivos

Los problemas de eficiencia en las búsquedas sobre datos masivos siguen siendo desafiantes y ofrece múltiples oportunidades para desarrollos científico/tecnológicos [7]. En este sentido, estas líneas de investigación proponen estudiar, desarrollar, evaluar y transferir modelos, algoritmos y técnicas específicas de búsquedas y procesamiento de datos masivos en documentos y grafos. En particular, se propone:

- Diseñar y evaluar variantes de los algoritmos para recuperación *top-k* usando umbrales de poda/corte dinámicos para reducir el número de elementos procesados.
- Desarrollar estructuras de datos (y algoritmos de recorrido) que combinen ordenamientos mixtos de las listas basados en frecuencias e identificadores de documentos, de forma variable.
- Desarrollar esquemas multicompresión de listas usando combinaciones de codecs y parámetros específicos para diferentes problemas.
- Diseñar y evaluar estrategias de indexación distribuida (y políticas de asignación de documentos) y resolución de consultas para flujos en tiempo real. Esto incluye métodos de selección de nodos y técnicas de *caching*.
- Definir y evaluar estructuras y modelos de cómputo distribuido sobre clusters de *hardware commodity* para problemas de escalabilidad en el cálculo exacto de métricas sobre grafos masivos.
- Considerar y analizar el impacto en el rendimiento y escalabilidad de utilizar estrategias de procesamiento para grafos masivos evolutivos. De forma similar, analizar el efecto de utilizar técnicas de particionado en estos entornos distribuidos.
- Proponer y evaluar nuevas estrategias de estimación de distancias entre nodos de un grafo masivo para problemas de búsqueda. Complementariamente, se propone extender el estudio de estrategias de corrección de la estimación de distancias mediante diferentes familias de funciones que ajusten la distribución de los errores.

Formación de Recursos Humanos

En el marco de estas líneas de investigación se están dirigiendo tres tesis de Licenciatura en Sistemas de Información (UNLu). Además, asociados al proyecto de investigación hay una Beca Estímulo a las Vocaciones Científicas (CIN) y dos pasantías internas en la UNLu.

Referencias

[1] Z. Abbas, V. Kalavri, P. Carbone, and V. Vlassov. Streaming graph partitioning: an experimental study. *VLDB Endowment*, 11(11), 2018.

[2] N. K. Ahmed, N. Duffield, T. L. Willke, and R. A. Rossi. On sampling from massive graph streams. *Proc. VLDB Endow.*, 10, 2017.

[3] Q. Ai, T. Yang, H. Wang, and J. Mao. Unbiased learning to rank: Online or offline? *ACM Trans. Inf. Syst.*, 39(2), Feb. 2021.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley Publishing Company, 2nd edition, 2011.

[5] A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien. Efficient query evaluation using a two-level retrieval process. In *Proc. of the 12th Int. Conf. on Information and Knowledge Management*, CIKM '03, 2003.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

[7] J. S. Culpepper, F. Diaz, and M. D. Smucker. Research frontiers in information retrieval (swirl 2018). *SIGIR Forum*, 52(1), 2018.

[8] S. Ding and T. Suel. Faster top-k document retrieval using block-max indexes. In *Proc. of the 34th Int. Conf. on Research and Development in Information Retrieval*. ACM, 2011.

[9] A. Giordano and G. Tolosa. Improved landmark-based shortest path length estimation in large graphs with distance correction. In *IEEE/WIC/ACM Int. Joint Conf. on Web Intelligence and Intelligent Agent Tech.*, 2020.

[10] A. González and G. Tolosa. Multicompresión de grandes listas de enteros para sistemas de búsquedas. In *Simposio Argentino de GRANdes DAtos*. JAIIO, 2020.

[11] Y. Kim, J. Callan, J. S. Culpepper, and A. Moffat. Efficient distributed selective search. *Information Retrieval Journal*, 20(3), June 2017.

[12] D. Lemire and L. Boytsov. Decoding billions of integers per second through vectorization. *Softw. Pract. Exper.*, 45(1), Jan. 2015.

[13] S. Madden. From databases to big data. *IEEE Internet Computing*, 16(3), 2012.

[14] A. Mallia, G. Ottaviano, E. Porciani, N. Tonellotto, and R. Venturini. Faster blockmax wand with variable-sized blocks. In *Proc. of the 40th Int.Conf. on Research and Development in Information Retrieval*, 2017.

[15] A. Mallia and E. Porciani. Faster blockmax wand with longer skipping. In L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, and D. Hiemstra, editors, *Advances in Information Retrieval*. Int. Publishing, 2019.

[16] G. Ottaviano and R. Venturini. Partitioned elias-fano indexes. In *37th Int. ACM SIGIR Conf. on Research & Development in Information Retrieval*, SIGIR '14, 2014.

[17] G. E. Pibiri and R. Venturini. Techniques for inverted index compression. 53(6), 2020.

[18] M. Potamias, F. Bonchi, C. Castillo, and A. Gionis. Fast shortest path distance estimation in large networks. In *18th ACM Conf. on Information and knowledge management*, 2009.

[19] N. Tax, S. Bockting, and D. Hiemstra. A cross-benchmark comparison of 87 learning to rank methods. *Information Processing & Management*, 51, 2015.

[20] G. Tolosa, E. Feuerstein, L. Becchetti, and A. Marchetti-Spaccamela. Performance improvements for search systems using an integrated cache of lists + intersections. *Information Retrieval. Journal*, 20(3):172–198, 2017.

[21] A. Trotman, J. Degenhardt, and S. Kallumadi. The architecture of ebay search. In *SIGIR 2017 Workshop on eCommerce (ECOM '17)*, 2017.

[22] H. Turtle and J. Flood. Query evaluation: Strategies and optimizations. *Inf. Process. Manage.*, 31(6), Nov. 1995.

[23] S. Vigna. Quasi-succinct indices. In *Sixth ACM Int. Conf. on Web Search and Data Mining*, WSDM '13, 2013.

[24] Y. Wang, L. Wu, L. Luo, Y. Zhang, and G. Dong. Short-term internet search using makes people rely on search engines when facing unknown issues. *PloS One*, 12(4), 2017.

[25] H. Yan, S. Ding, and T. Suel. Inverted index compression and query processing with optimized document ordering. In *18th Int. Conf. on World Wide Web*, 2009.

[26] J. Zhang, X. Long, and T. Suel. Performance of compressed inverted list caching in search engines. In *17th Int. Conf. on World Wide Web*, 2008.