Correction to: Two-dimensional indexing to provide one-integrated-memory view of distributed memory for a massively-parallel search engine

Tae-Seob Yun¹ · Kyu-Young Whang¹ · Hyuk-Yoon Kwon² · Jun-Sung Kim¹ · Il-Yeol Song³

Published online: 11 April 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Correction to: World Wide Web (2018)

<https://doi.org/10.1007/s11280-018-0647-1>

After our paper was published on-line, the authors have learned existence of the paper by Feuerstein et al. (2009). We regret that items 1 and 3 in Section 1.4 should be corrected to read as follows:

– (Item 1) We adopt the notion of two-dimensional distributed inverted files proposed by Feuerstein et al. (2009) with the purpose of optimizing the entire *system cost* including computation, communication, and fixed overheads of the system. Feuerstein et al. (2009) allocated index fragments in a two-dimensional array of processors through keyword (or term) partitioning (defined in Section 2.2) and document partitioning (defined in Section 2.2). In this paper, we adopt this basic scheme with a different purpose of minimizing the average *query response time* by simulating collective distributed (main) memory as one integrated memory. Then, we extend this scheme so that each row of the two-dimensional array stores a replicated (document-partitioned only) index shard on disk for efficient processing of multiple-keyword queries. We call this extended scheme and associated algorithms *two-dimensional indexing*.

The online version of the original article can be found at <https://doi.org/10.1007/s11280-018-0647-1>

✉ Kyu-Young Whang
kywhang@mozart.kaist.ac.kr

¹ Department of Computer Science, KAIST, Daejeon, South Korea

² Department of Global Fusion Industrial Engineering, Seoul National University of Science and Technology, Seoul, South Korea

³ College of Information Science and Technology, Drexel University, Philadelphia, PA, USA

Figure 2 shows the concept of providing a one-integrated-memory view of collective memories by using two-dimensional indexing over $n \times m$ two-dimensional distributed memory. Here, the horizontal axis of the array indicates n index shards that are partitioned from the entire index; the vertical axis in each column of the array indicates m index fragments that are partitioned from each index shard as in Feuerstein et al. (2009). In addition, each index fragment in main-memory is associated with a full (non-keyword-partitioned) index shard on disk. The two-dimensional indexing architecture is scalable as it uses a shared-nothing architecture. Moreover, we can achieve low query response time as it allows processing queries in main memory.

– (Item 3) Two-dimensional indexing allows us to process multiple-keyword queries efficiently. For that purpose, we propose the notion of *pre-join* that can handle a multiple-keyword query just like a single-keyword query (Section 5.1). Then, we propose the notion of *semi-memory join* that eliminates the costly inter-node communication among the nodes at a cost of some disk accesses to the entire index shared stored on disk (Section 5.2). Further, we propose methods to achieve dynamic index partitioning for load balancing and for dynamic update of in-memory multiple-keyword sets to cope with an environment where query keywords change dynamically in a real-life search engine (Section 5.5).

Reference

1. Feuerstein, E., Marin, M., Mizrahi, M., Gil-Costa, V., Baeza-Yates, R.: Two-dimensional distributed inverted files. In: Proc. 16th String Processing and Information Retrieval Symposium (SPIRE 2009), pp. 206–213, Springer LNCS 5721 (2009)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.