I/O-efficient algorithms for localized bisimulation partition construction and maintenance on massive graphs

Yongming Luo[*][1], George H.L. Fletcher[1], Jan Hidders[2], Yuqing Wu[3], and Paul De Bra[1]

[1] *Eindhoven University of Technology, The Netherlands* [†]
[2] *Delft University of Technology, The Netherlands* [‡]
[3] *Indiana University, Bloomington, USA* [§]

In this talk, we present, to our knowledge, the first known I/O efficient solutions for computing the k-bisimulation partition of a massive graph, and performing maintenance of such a partition upon updates to the underlying graph.

Bisimulation is a robust notion of node equivalence which is ubiquitous in the theory and application of graph data. It defines an intuitive notion of nodes in a graph sharing fundamental structural features. In data management, for instance, bisimulation partitioning (i.e., grouping together bisimilar nodes) is often a basic step in indexing semi-structured datasets [6], and also finds fundamental applications in RDF [7] and general graph data (e.g., compression [1, 3], query processing [4], data analytics [2]). Inspired by these applications, in this talk, we consider the problem of computation and maintenance of k-bisimulation, which is the standard variant of bisimulation where the topological features of nodes are only considered within a local neighborhood of radius $k \geqslant 0$.

The I/O cost of our partition construction algorithm is bounded by $O(k \cdot |E_t| \cdot \lceil log_{B-1} \lceil \frac{|E_t|}{B} \rceil \rceil + k \cdot |N_t| + |N_t| \cdot \lceil log_{B-1} \lceil \frac{|N_t|}{B} \rceil \rceil)$, while our maintenance algorithms are bounded by $O(k \cdot |E_t| \cdot \lceil log_{B-1} \lceil \frac{|E_t|}{B} \rceil \rceil + k \cdot |N_t| \cdot \lceil log_{B-1} \lceil \frac{|N_t|}{B} \rceil \rceil)$. Here, $|E_t|$ and $|N_t|$ are the number of disk pages occupied by the input graph's edge set and node set, resp., and B is the maximum number of disk pages which can fit in internal memory. Empirical analysis on a variety of massive real-world and synthetic graph datasets shows that our algorithms not only perform efficiently, but also scale gracefully as graphs grow in size [5]. During the talk, we will explain the basic idea that leads to the design of our algorithms by one running example, and will discuss some interesting observations during the empirical study of the algorithms on massive graph datasets.

References

[1] P. Buneman, M. Grohe, and C. Koch. Path queries on compressed XML. In *VLDB*, pages 141–152, Berlin, Germany, 2003.

[2] W. Fan. Graph pattern matching revised for social network analysis. In *ICDT*, pages 8–21, Berlin, Germany, 2012.

[3] W. Fan, J. Li, X. Wang, and Y. Wu. Query preserving graph compression. In *SIGMOD*, pages 157–168, Scottsdale, AZ, USA, 2012.

[4] R. Kaushik, P. Shenoy, P. Bohannon, and E. Gudes. Exploiting local similarity for indexing paths in graph-structured data. In *ICDE*, pages 129–140, San Jose, CA, USA, 2002.

[5] Y. Luo, G. H. L. Fletcher, J. Hidders, Y. Wu, and P. De Bra. I/O-efficient algorithms for localized bisimulation partition construction and maintenance on massive graphs. *CoRR*, abs/1210.0748, 2012.

[6] T. Milo and D. Suciu. Index structures for path expressions. In *ICDT*, pages 277–295, Jerusalem, Israel, 1999.

[7] F. Picalausa, Y. Luo, G. H. L. Fletcher, J. Hidders, and S. Vansummeren. A structural approach to indexing triples. In *ESWC*, pages 406–421, Heraklion, Greece, 2012.

[*] Yongming Luo is the prospective speaker

[†] {y.luo, g.h.l.fletcher, P.M.E.d.Bra}@tue.nl

[‡] {a.j.h.hidders}@tudelft.nl

[§] {yuqwu}@cs.indiana.edu