

Adaptive Partitioning for Large-Scale Dynamic Graphs

Luis Vaquero*, Félix Cuadrado*, Dionysios Logothetis**, Claudio Martella***
Queen Mary University of London, **Telefónica I+D, ***VU University Amsterdam
luis.vaquero@ieee.org, felix@eecs.qmul.ac.uk, dl@tid.es, c.martella@vu.nl

Abstract

Mining large-scale graphs is increasingly important, as it provides a powerful way of extracting useful information from real-world data. Efficient processing of that volume of information requires partitioning the graph across multiple nodes in a distributed system. However, traversing edges across distributed partitions results in significant performance penalty due to the additional cost of inter-partition communication. Minimising the number of cut edges between partitions improves communication cost between neighbouring vertices; balanced graph partitioning is required for load balancing [2].

These large graphs represent real-world information, which is inherently dynamic. Recent systems such as Kinograph [1] can process changing graphs, but they do not consider the impact of dynamism in graph partitioning. To illustrate this impact, we built a call graph from mobile Call Detail Records data, with a sliding window defining the creation and removal of nodes and edges. The graph was partitioned using three different techniques: *modulo hash* (HSH), the most popular partitioning technique because of its high scalability to produce balanced partitions, [2]; a state of art streaming partitioning technique (*deterministic greedy*, DTG) [3]; and our *adaptive repartitioning* heuristic, (ADP). Figure 1 shows the evolution of the partitioning (expressed as the ratio of edges that cut across different partitions). While a good partitioning strategy significantly improves the initial ratio of cuts, the quality of the partitioning degrades over time, resulting in higher communication penalty.

Copyright © 2013 by the Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of portions of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page in print or the first screen in digital media. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

SoCC'13, 1–3 Oct. 2013, Santa Clara, California, USA.
ACM 978-1-4503-2428-1.
<http://dx.doi.org/10.1145/2523616.2525943>

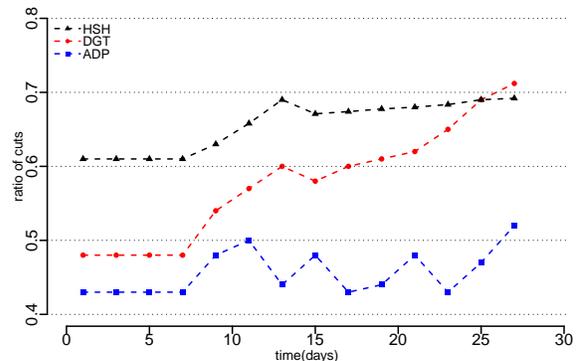


Figure 1: Evolution of the ratio of cuts over time on a dynamic graph generated by processing CDR calls on a sliding window.

In order to prevent this performance degradation, current approaches would require a full graph repartition, which can be extremely costly with large-scale graphs, and generate downtime gaps in the system. While this problem does not deeply affect batch processing systems, it can greatly impact throughput and latency of graph processing systems requiring faster response times. We propose an adaptive approach, where the graph is optimised with every change, over computation execution.

We improve graph partitioning in a scalable manner by applying a local decision heuristic, based on decentralised, iterative vertex migration. The heuristic [4] migrates vertices between partitions trying to minimise the number of cut edges, while at the same time keeping partitions balanced upon structural changes at run time.

We tested this approach in a system that processes dynamic graphs and adapts to graph changes by applying the iterative vertex migration algorithm. While continuous migrations bring added overhead to the computation, we observed in several experiments that the total execution time was reduced by over 50%. A more detailed analysis of the system and experiments is available at [4].

References

- [1] R. Cheng, J. Hong, A. Kyrola, Y. Miao, X. Weng, M. Wu, F. Yang, L. Zhou, F. Zhao, and E. Chen.

Kineograph: taking the pulse of a fast-changing and connected world. In *EuroSys*, 2012.

- [2] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: a system for large-scale graph processing. In *PODC*, 2009.
- [3] I. Stanton and G. Kliot. Streaming graph partitioning for large distributed graphs. In *KDD*, 2012.
- [4] L. Vaquero, F. Cuadrado, D. Logothetis, and C. Martella. xdg: A dynamic graph processing system with adaptive partitioning. <http://arxiv.org/abs/1309.1049>, 2013, 1309.1049.