Comparison of One-Phase and Two-Phase Approaches for Replicated Hypergraph Partitioning

R. Oguz Selvitopi[a], Volkan Yazici[c], Ata Turk[b], Cevdet Aykanat[a]

[a] *Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey*
[b] *Yahoo! Labs, Barcelona, Spain*
[c] *Department of Computer Science, Ozyegin University, Istanbul 34794, Turkey*

Hypergraphs find their applications in wide range of domains that include VLSI circuit design, scientific computing [1], information retrieval and database systems. They are successfully adopted and used in these domains to model problems with different types of relations. These relations can broadly be categorized as directed or undirected according to the requirements of the application being modeled. In undirected relations, the relation among data items is equally shared, whereas in directed relations, there exists an input/output relation among data items being modeled. In information retrieval and database systems, replication is a useful method to provide fault tolerance, enhance parallelization and improve processing performance. In this study, we target vertex replication to further improve the objective of hypergraph partitioning for hypergraphs that employ undirected relations. The applications that benefit from hypergraph models and replication can utilize the methods and techniques that are proposed for replicating vertices of undirected hypergraphs.

There are two possible approaches to solve the replicated hypergraph partitioning problem: one-phase and two-phase. In [2], we propose a one-phase approach where replication of vertices in the hypergraph is performed during the partitioning process. This is achieved during the uncoarsening phase of the multilevel methodology by proposing a novel iterative-improvement-based heuristic which extends the Fiduccia-Mattheyses (FM) algorithm [3] by also allowing vertex replication and unreplication. In [4], we describe a two-phase approach in which replication of vertices is performed after partitioning is completed. The replication phase of this two-phase approach utilizes a unified approach of coarsening and integer linear programming (ILP) schemes. The one-phase approach has the possibility of generating high quality solutions since it achieves replication during the partitioning process and it considers unreplication of vertices as well. The advantages of using two-phase approach are the flexibility of using any of the hypergraph partitioning tools available and being able to work on the partitions that already contain replicated vertices.

In the one-phase approach, to perform replication of vertices during the partitioning process, the FM heuristic is extended to allow replication and unreplication of vertices as well as move operations. This extended heuristic has the same linear complexity as the original FM heuristic and is able to replicate vertices in a two-way partition by new vertex states and gain definitions. This heuristic is later utilized in a recursive bipartitioning framework to enable K-way replicated partitioning of the hypergraph and it supports the two widely used cut-net and connectivity cutsize metrics. It is integrated into the successful hypergraph partitioning tool PaToH as the refinement algorithm in the uncoarsening phase.

The two-phase approach utilizes Dulmage-Mendelsohn [5] decomposition to find replication sets for each part by only considering boundary vertices. The replication sets are bounded by a maximum replication capacity, and they are arranged in such a way that the imbalance of the given original partition is not disturbed, or even is sometimes improved. The balancing constraint is enforced by proposing a part-oriented method to determine the amount of replication to be performed on each part in a particular order.

We compare the mentioned one-phase and two-phase approaches for achieving vertex replication in undirected hypergraphs. We present the results of the quality of the partitions on hypergraphs from different domains with varying setups.

References

[1] O. Selvitopi, M. Ozdal, C. Aykanat, A novel method for scaling iterative solvers: Avoiding latency overhead of parallel sparse-matrix vector multiplies, Parallel and Distributed Systems, IEEE Transactions on PP (99) (2014) 1–1. doi:http://dx.doi.org/10.1109/TPDS.2014.2311804.
URL `http://www.computer.org/csdl/trans/td/preprint/06766662-abs.html`

[2] R. Oguz Selvitopi, A. Turk, C. Aykanat, Replicated partitioning for undirected hypergraphs, J. Parallel Distrib. Comput. 72 (4) (2012) 547–563. doi:10.1016/j.jpdc.2012.01.004.
URL `http://dx.doi.org/10.1016/j.jpdc.2012.01.004`

[3] C. M. Fiduccia, R. M. Mattheyses, A linear-time heuristic for improving network partitions, in: Proceedings of the 19th Design Automation Conference, DAC '82, IEEE Press, Piscataway, NJ, USA, 1982, pp. 175–181.
URL `http://portal.acm.org/citation.cfm?id=800263.809204`

[4] V. Yazici, C. Aykanat, Constrained min-cut replication for k-way hypergraph partitioning, INFORMS Journal on Computing, 10.1287/ijoc.2013.0567 0 (0) (0) null. arXiv:http://pubsonline.informs.org/doi/pdf/10.1287/ijoc.2013.0567, doi:10.1287/ijoc.2013.0567.
URL `http://pubsonline.informs.org/doi/abs/10.1287/ijoc.2013.0567`

[5] A. Pothen, C.-J. Fan, Computing the block triangular form of a sparse matrix, ACM Trans. Math. Softw. 16 (4) (1990) 303–324. doi:10.1145/98267.98287.
URL `http://doi.acm.org/10.1145/98267.98287`