Recent Trends in Graph Partitioning for Scientific Computing

Burkhard Monien (University of Paderborn),
Henning Meyerhenke (Georgia Institute of Technology)

Abstract

Graph partitioning is a widespread technique in computer science, scientific computing, and related fields. The most common formulation of the graph partitioning problem (GPP) for an undirected graph G = (V,E) asks for a division of V into k pairwise disjoint subsets (parts) with size approximately |V|/k such that the edge cut, i.e., the total number of edges having their incident nodes in different subsets, is minimized.

Among others, applications that employ graph partitioning methods include dynamical systems, VLSI circuit layout, and image segmentation. In this talk, however, we focus mainly on applications in the area of parallel scientific computing. Here the graph's vertices usually model computations and its edges dependencies among these computations. For an efficient parallelization, the graph needs to be partitioned such that dependencies across different processors are minimized.

An early very influential approach for graph partitioning is based on spectral methods. In the 1990s the introduction of the multilevel method for graph partitioning led to further advances in terms of quality and running time. Multilevel algorithms with the local search heuristic Kernighan-Lin (KL) for local improvement yielded several fast graph partitioning software packages that are still in widespread use today.

The KL heuristic and its variants have the drawback, however, that they focus too strongly on the edge cut. Moreover, the edge-cut is a summation norm, while often the maximum norm is of higher importance (e.g., for parallel numerical solvers, the worst part determines the overall application time). Finally, for some applications, the shape of the partitions, in particular small aspect ratios, but also connectedness and smooth boundaries, plays a significant role.

In this talk we survey a number of techniques that have emerged in recent years to alleviate some of the problems of multilevel KL in the context of scientific computing. With very different approaches, these techiques usually invest more running time to obtain a better parallelization and/or solutions with a higher quality. Among the novelties are improved multilevel mechanisms, as well as local improvement strategies related to random walks, based on flows or – for reasons of shape optimization – based on diffusion.