Partitioning a Call Graph

Rob H. Bisseling*, Jarosław Byrka, Selin Cerav-Erbas,
Nebojša Gvozdenović, Mathias Lorenz, Rudi Pendavingh,
Colin Reeves, Matthias Röger, Arie Verhoeven

Abstract

Large software systems used by today's organizations such as banks, health care providers, or government agencies, have become costly to maintain. To reduce the costs, the systems need to be split into smaller, more manageable modules with minimum total interface size.

A software system can be described by a call graph, which is a directed graph where vertices represent programs and edges represent programs calling another program. Interface vertices represent programs called by programs in a different module.

Here, we present two methods for partitioning a call graph: an optimal solution based on integer linear programming suitable for small and medium-sized problems, and a faster approximate solution based on multilevel hypergraph partitioning with the cut-net metric. We apply the methods to Java and Cobol problems provided by the Software Improvement Group in the Netherlands, and solved all given problems (with up to 1398 vertices) to optimality. Comparing the results of the two methods for these real-world examples gives an indication of the quality of the multilevel hypergraph partitioning heuristic.

*presenter