A DISTRIBUTED BRANCH AND BOUND METHOD FOR BOINC DESKTOP GRIDS

Bo Tian, M.A. Posypkin

Moscow State University, Moscow, Russia

yesyestian@gmail.com

The Branch-and-Bound method (B&B) is a very efficient and well-known technique to solve combinatorial optimization problems such as Traveling Salesman Problem, Knapsack, Integer Programming, Vertex Covering and many others. This algorithm allows to reduce-considerably the computation time required to explore the entire solution space associated with the problem being solved. However, the exploration time remains considerable, using parallel or distributed processing is one of the major and popular ways to reduce it. Many parallel B&B approaches have been proposed so far, such as [1]. However distributed grid-oriented B&B implementations are not well studied.

Normally, B&B algorithm consists of four main parts:

1. The branching rule that consists in a strategy for expanding a parent (sub) problem into child sub-problems. The leaf-nodes of the expanded tree represent all possible solutions;
2. The selection rule that chooses some of the current evaluated sub-problems for expansion using a search heuristic, e.g. Best-First, Depth-First, Breadth-First or others;
3. The bounding rule that consists in reducing the search space by eliminating sub-problems that do not yield to the optimal solution;
4. The termination condition which takes place when all sub-problems are either de-composed or eliminated.

This work discusses the design and the deployment of this algorithm on a computational grid. We parallelize the B&B algorithm in order to make it suitable for the coarse-grained work distribution. This approach was implemented using BOINC - an open source volunteer computing platform. Our implementation works as follows:

1. Master computer (BOINC server) performs server B&B steps and generates the first possible solution from search space then stops calculating immediately and goes to the second step.
2. Master computer creates a collection of work-units based on the terminal nodes of the explored sub-tree and sends work-units to the clients. Load balancing and sending work-units are handled by BOINC.
3. A client receives work-units, extracts and processes sub-problems, performing the given number of steps or less. After that the client returns the result (best obtained solution and the remaining sub-problems which need to be processed by other clients) back to master computer.
4. Master computer collects all results from clients, chooses the best from the collected solutions or re-allocates sub-problems to free clients.
5. This process goes to end unless all sub-problems have been processed. Then master computer chooses the final best solution.

We evaluate and demonstrate the efficiency of the proposed approach on large-scale knapsack instances. In the future we plan to try different strategies to compose work-units and study its impact on the efficiency of the algorithm. We also plan to use this framework for more practical problems like transportation problems, managing network flows, engineering design etc.

[1] P.Laursen, Simple approaches to parallel branch and bound, Parallel Computing, 19(1993), pp.143-152.