PARALLEL ALGORITHMS FOR SOLVING LARGE MULTIDIMENSIONAL ASSIGNMENT PROBLEMS WITH DECOMPOSABLE COSTS

Allocation: Illinois/210 Knh
PI: Rakesh Nagi[1]
Co-PI: Ketan Date[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

The objective of this research is to develop fast and scalable GPU-based parallel algorithms that can solve large instances of the Multidimensional Assignment Problem with Decomposable Costs (MDADC). MDADC has important applications in information fusion, resource planning, and multitarget tracking domains. Typically, these problems are solved using tree search procedures, in which the lower bounds are calculated at each tree node by solving a Lagrangian dual problem, for which we propose fast and scalable GPU-accelerated algorithms. For this project, Blue Waters' computational resources are extremely beneficial due to the availability of a large number of GPU-enabled processors. These can significantly speed up the lower bound calculations at each node, allowing us to solve large problems that have not been attempted to date.

RESEARCH CHALLENGE

The axial Multidimensional Assignment Problem (MAP) is a generalization of the Linear Assignment Problem (LAP) to $K \geq 3$ dimensions, and therefore, it is nothing but a minimum-cost K-partite matching problem. To be more specific, if we have a K-partite graph with N vertices in each partition (or dimension), then the problem is to find N vertex disjoint subsets of size K (one vertex per dimension), such that the sum of the costs of the subsets is minimized. Different variants of MAP can be conceived for different strategies of defining the subset costs in the objective function [1]. The MDADC [2] is one such variant in which each edge connecting a pair of nodes from two different partitions has a certain weight, and the goal is to minimize the sum of the weights of the pairwise edges included in the subsets (basically a "clique" cost).

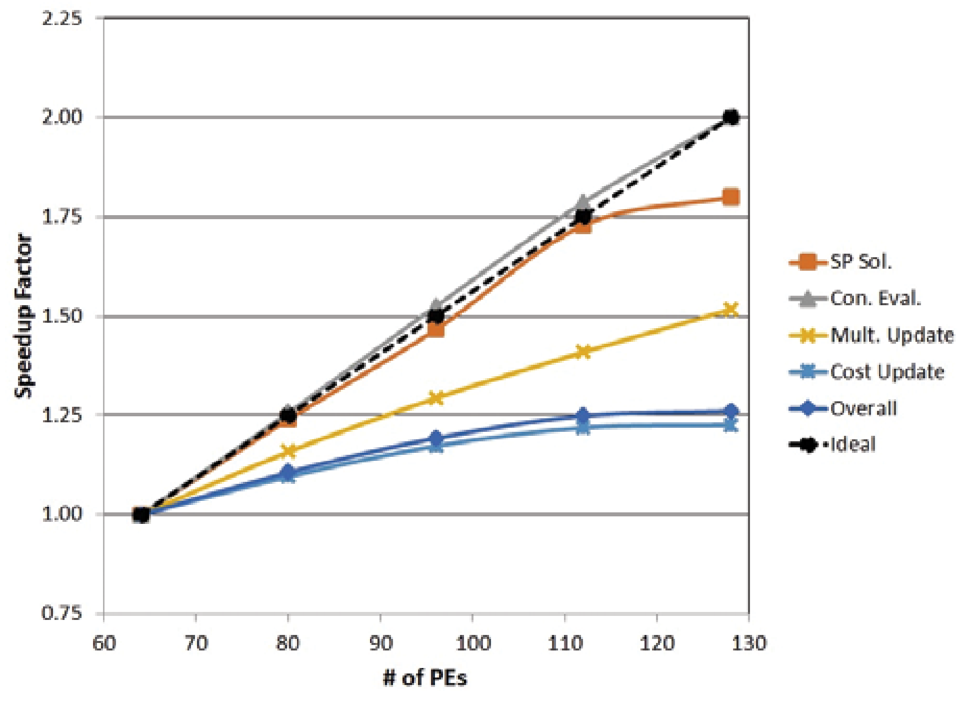

Figure 1: Speedup factors for MDADC instance with N=500 and K=500.

MDADC is routinely used to model the Data Association Problem [3], which is a fundamental problem in data science applications involving multiple data sources. Data gathered by these sources may be in different formats, and the goal of data association is to merge these data into a cumulative evidence aggregate that can be used for sense-making tasks or to obtain information about the current state of the real world. MDADC provides a systematic way of modeling and solving these important problems and has tremendous potential in information fusion and multitarget tracking models.

METHODS & CODES

We chose to parallelize the Lagrangian subgradient search heuristic for the MDADC formulation similar to [4], in which we need to solve $O(K^2)$ LAPs of size $N \times N$ and adjust $O(K^3 N^3)$ Lagrange multipliers to obtain strong lower bounds. We designed a parallel Lagrangian heuristic for solving MDADC using hybrid MPI+CUDA architecture. The $O(K^2)$ LAPs are split across multiple GPU nodes and solved using our GPU-accelerated Hungarian algorithm [5], while the $O(K^3 N^3)$ Lagrange multipliers are updated by multiple CUDA threads in parallel. This problem is primarily memory bound due to the presence of a large number of Lagrange multipliers. However, we found that many of these multipliers have a value of zero as a consequence of the corresponding constraint being inactive. This important observation led us to design our innovative algorithm, which identifies and stores only the nonzero multipliers and alleviates the large memory requirement.

RESULTS & IMPACT

With our method, we were able to obtain strong lower bounds on problem instances with N=500 and K=500. To solve this problem, we needed at least 64 GPUs, but the number of GPUs can be increased to achieve additional parallel speedup. We performed strong scalability studies with up to 128 GPUs. Fig. 1 shows we obtained good speedup in the initial stages. However, as we continued to increase the number of GPUs in the system, we got diminishing returns in the execution times because some of the steps require excessive MPI communication.

Our single-GPU experiments revealed that the proposed algorithm is over 200 times faster than the Map-Reduce implementation of [4] for some of the smaller problem instances. The larger MDADC instance with N=500 and K=500 contains over 3.1188×10^{10} binary variables and over 7.7657×10^{15} constraints (and corresponding Lagrange multipliers). To the best of our knowledge, problems of this magnitude have not been attempted in the literature, and we are at the frontier of this research domain.

WHY BLUE WATERS

In a typical MDADC instance, we need to solve a large number of LAPs and adjust a large number of Lagrange multipliers in order to obtain a tight lower bound. This problem is memory bound and, therefore, we need a large number of processors that can handle the LAPs and adjust the multipliers in parallel. As a result, the GPU-accelerated Lagrangian heuristic benefits from the host of powerful GPU-enabled processors available on Blue Waters. We have used over 128 XK compute nodes, which would have incurred significant costs on the proprietary systems such as the AWS. We are grateful to Blue Waters and the project staff for providing this invaluable service to the scientific community.

PUBLICATIONS & DATA SETS

Natu, S., K. Date, and R. Nagi, GPU-accelerated Lagrangian Heuristic for Multidimensional Assignment Problems with Decomposable Costs. *Parallel Computing*, submitted (2018).