GA

MI

PARALLEL ALGORITHMS FOR SOLVING LARGE ASSIGNMENT PROBLEMS

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 our project is to develop fast and scalable algorithms for obtaining strong lower bounds and exact solutions for large instances of the Quadratic Assignment Problem (QAP), using Graphics Processing Unit (GPU) clusters. The QAP is an NP-Hard problem, in the strong sense. To solve a linearized model of the QAP using branch-and-bound, lower bounds must be calculated using the Lagrangian dual technique, in which a large number of Linear Assignment Problems (LAP) are solved efficiently, using our GPU-accelerated Hungarian algorithm. Additionally, in a branch-and-bound scheme, a large number of nodes must be explored in order to find a provable optimal solution. To this end, we have used Blue Waters to develop a GPU-accelerated Lagrangian dual ascent heuristic for obtaining lower bounds on the QAP, which is used in the parallel branch-and-bound scheme to solve large QAPs to optimality.

RESEARCH CHALLENGE

The Quadratic Assignment Problem (QAP) was introduced by [1] as a mathematical model to locate indivisible economical activities (such as facilities) on a set of locations so as to minimize a quadratic cost function. Typical applications of the QAP may be found in facility layout (re)design in manufacturing, distribution systems, services (retail outlets, hospital floors, etc.), and electronic circuit design. QAP may also serve as a specialization to many "harder" optimization problems, such as the Graph Association (GA), Traveling Salesman Problem (TSP), Vehicle Routing Problem (VRP), etc., in alternative formulations. Therefore, to solve these problems efficiently, we need to solve the QAP sub-problems efficiently. As a result, a fast and scalable QAP solver can be a powerful tool for researchers working on such NP-hard problems, or problems that are exceeding difficult to computationally solve on a finite resource in a finite time. The sequential QAP solver can become computationally intensive and, therefore, the algorithm can benefit from parallelization on an appropriate parallel architecture, such as Blue Waters.

METHODS & CODES

We chose to parallelize the Lagrangian dual ascent algorithm for Level-2 Refactorization-Linearization Technique (RLT2) proposed by [2], in which we need to solve $O(n^4)$ LAPs and adjust $O(n^6)$ Lagrange multipliers to obtain a strong lower bound on the QAP. We designed a parallel Lagrangian dual ascent heuristic for solving RLT2 using hybrid MPI+CUDA architecture. The $O(n^4)$ LAPs are split across these GPUs and solved using our GPU-accelerated Hungarian algorithm [3], while the $O(n^6)$ Lagrange multipliers are updated by multiple CUDA threads in parallel.

We used this GPU-accelerated dual ascent algorithm in a branch-and-bound scheme to solve QAP instances to optimality. For a node in the search tree, we fix a facility to a location and solve the corresponding RLT2 sub-problem, whose objective value provides a lower bound on the QAP. If this value is greater than the incumbent solution then the node is fathomed; otherwise, it is branched further. Each node is processed using a bank of GPUs. By using multiple such banks, we can process multiple nodes in parallel. The algorithms were tested on the problem instances from the QAPLIB [4].

RESULTS & IMPACT

Lower bounds: The results for lower bounding tests for the various problem instances are summarized in Table 1. With our architecture, we are able to obtain strong lower bounds on problem instances with up to 42 facilities, which is a tremendous achievement.

Problem	GPUs	Optimal Value	Lower Bound	% GAP	Itn Time (s)
Nug25	4	3744*	3610	3.58	7.99
Nug27	7	5234*	5076	3.02	8.49
Nug30	12	6124*	5846	4.54	9.08
Tai25a	4	1,167,256*	1,091,480	6.49	7.75
Tai30b	12	637,117,113*	620,444,000	2.62	10.73
Tho40	71	240,516	213,372	11.29	18.41
Sko42	95	15,812	14,741	6.77	21.54

Table 1: RLT2 dual ascent lower bounds on QAPLIB instances (selective)

Scalability study: Although there is a minimum required number of PEs for applying accelerated RLT2 dual ascent to a QAP of specific size, the number of GPUs can be increased and the LAPs can be solved in parallel on multiple GPUs. This allows us to achieve some parallel speedup. We performed strong scalability with 1 to 32 GPUs. We obtain good speedup in the initial stages. However, as we continue to increase the number of GPUs in the system, we get diminishing returns in the execution times, due to increased MPI communication.

Parallel branch-and-bound: The results for the parallel branch-and-bound tests are shown in Table 2. We can see that the number of nodes explored and the completion times increase exponentially with the problem size. The most challenging Nug30 problem instance required more than four days to solve optimally, using 300 GPU banks with 4 GPUs each, which is a significant achievement.

PUBLICATIONS AND DATA SETS

Date, K., and R. Nagi, GPU-accelerated Hungarian algorithms for the Linear Assignment Problem. *Parallel Computing*, 57 (2016), pp. 52–72, DOI: 10.1016/j.parco.2016.05.012.

Problem	Optimal Value	GPU Banks	GPUs/Bank	Nodes Explored	Time (d:hh:mm:ss)
Nug25	3744*	50	2	3868	0:02:44:24
Nug27	5234*	100	2	55761	1:02:28:32
Nug30	6124*	300	4	840273	4:14:06:21
Tai25a	1,167,256*	100	2	523005	3:13:53:33
Tai30b	637,117,113*	60	4	30523	2:09:55:17

Table 2: Branch-and-bound results for QAPLIB instances (selective)

Impact: With our architecture, we are able to obtain strong lower bounds on problem instances with up to 42 facilities, and optimally solve problems with up to 30 facilities, using only a modest number of GPUs. To the best of our knowledge, this is a first-of-its-kind study that has pushed the RLT2 formulation to solving such large problems. With some additional work in memory management and CPU + GPU collaboration, our proposed algorithms may be used effectively to solve truly large QAPs with over 30 facilities, which has been impossible so far.

WHY BLUE WATERS

In a typical branch-and-bound tree, we need to explore a large number of nodes in order to find an optimal solution. As the problem size grows, the number of nodes that need to be explored grows exponentially. Therefore, we need a large number of processors that can explore the solution space in parallel. Additionally, the GPU-accelerated dual ascent procedure benefits from the large number of powerful GPU-enabled processors available at the Blue Waters facility. As seen in the results, we have used over 1,200 XK compute nodes for over 110 hours to solve Nug30 problems, 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.