Finishing GPU Jobs running on a Multi-GPU Batch-Queue Node-Sharing System Earlier with Remote GPU Execution and Migration

Pak Markthub, Akihiro Nomura and Satoshi Matsuoka
Tokyo Institute of Technology

Problem

Problem: Multi-GPU batch-queue systems usually have large number of idle GPUs despite having jobs waiting.

Main Cause: The scattered idle-GPU problem, which is one form of resource assignment fragmentation problem, prevents efficient GPU assignment.

Objective & Proposed Solution

Objective: To increase GPU occupancy and reduce job lifetime (waiting + execution time) without changing the systems’ scheduling policies.

Evaluation

• Applied our solution to the first-come-first-serve (FCFS) scheduling algorithm, which is called “MRQ” (use same scheduling policy as FCFS).
• Simulated various job sets with various job characteristics to see how well MRQ performs compared with FCFS.

Solution Overview

Allow the system to serve more jobs concurrently by enabling GPU jobs to access unoccupied GPUs on remote nodes.

Comparison of the job lifetime on FCFS and MRQ when varying GPU communication intensity (above) and when varying the number of requested GPUs per node (below)

Case Study

• Used the recorded job set from TSUBAME2.5 (G-GPU) queue during Aug. 1-15, 2015 (a busy period) — 574 jobs on 480 nodes, 10 hours/job on avg.
• Varied number of GPU invocations (gpu_invocation) and total GPU data transferred (gnodata) since they are not recorded.
• The result shows about 5% decrease of job lifetimes on average — aligns with the result from simulated job characteristics because about 85% of the jobs on that period requested three GPUs per node (all GPUs on a node).

Conclusions & Future Work

• Ultimate Goal: To improve the utilization of multi-GPU systems.
• Research Contribution toward the Goal: Enable multi-GPU batch-queue node-sharing systems to serve more GPU jobs concurrently.

Research Conclusions:
• Current resource assignment strategies on multi-GPU batch-queue node-sharing systems are not optimal because of the scattered idle-GPU problem.
• Our solution, virtually consolidating unoccupied GPUs while minimizing the negative impact of remote GPU communication with remote GPU execution and migration, can decrease a huge improvement of job lifetime up to 30% on average without changing the scheduling policy nor rewriting the code (work seamlessly).
• Future Work: Improve communication between remote GPUs, and research about oversubscribing GPUs with dynamic GPU assignment (both local & remote).

References


This research was supported by JST, CREST (Research Area: Advanced Core Technologies for Big Data Intensive).