Hierarchical Scheduling Frameworks for Heterogeneous Clusters with GPUs
Kittisak Sajjapongse, PhD candidate, Michela Becchi, Advisor
University of Missouri - Columbia

Abstract

Graphical Processing Units have increasingly been adopted for a wide-range of scientific applications both in industry and academia, and have recently become part of HPC clusters. Distributed GPU applications typically offload computation to GPUs using CUDA and OpenCL and distribute tasks across compute nodes through MPI and SHMEM. Despite the availability of these parallel programming frameworks, coding distributed GPU applications is still a specialized task. In addition to the use of batch schedulers to handle these applications on shared clusters often leads to resource underutilization and performance degradation.

In our research, we propose the design of hierarchical scheduling frameworks to provide GPU virtualization and improve the performance of concurrent applications in shared cluster environments. In our previous work we have designed and implemented efficient node- and cluster-level runtime systems to support distributed GPU applications in heterogeneous CPU-GPU clusters. In particular, our proposed node-level runtime enables GPU virtualization, GPU sharing across applications, flexible scheduling mechanisms, and virtual memory support. Our proposed cluster-level scheduler allows the user to customize and define new scheduling policies and configure the GPU sharing mechanism used at the node level. Our results show that our hierarchical scheduling framework outperforms existing batch schedulers while improving the overall GPU utilization. Our current work focuses on increasing the programmability of hybrid nodes while enabling effective sharing and load balancing mechanisms.

Overall Design

Node-level runtime

CUDA

Native Operating System

GPU

GPU

Node-0

Node-1

Node-N

Dynamic Global Address Space (DGAS) Framework

Abstract view

Virtual Memory Space

Physical view

Memory and resource virtualization

Transparency: Scheduling

Process Elements (PEs) will be scheduled transparently relative to programmers

Heterogeneity & load-balancing

Cluster- and node-level schedulers will distribute PEs to nodes and GPUs according to the nodes and GPUs performance capabilities.

Locality

Each physical node will have local copy of data objects which can be accessed by PEs on the node. The local-copy will be merged back to the master copy on the root node when consistency is enforced.

Scheduling Framework for MPI applications

Node-level Scheduler

Connection Manager

Connection handling

Context Guesser

Guesses context filters

Queue Monitor

Communicates with different sub-systems

CUDA driver/runtime

Virtual GPUs

Non-blocking

GPU Sharing


DGAS Cluster

DGAS Scheduler

DGAS Cluster Scheduler

DGAS Scheduler

DGAS Scheduler

Multi-process applications scheduling

Batch scheduling

A(1)

A(2)

A(3)

A(4)

A(5)

A(6)

Preemptive sharing improves performance in the presence of synchronization by better utilizing GPU resources.