Portable and Performant GPU/Heterogeneous Asynchronous Many-Task Runtime System

Thesis proposal

Brad Peterson

bpeterson@sci.utah.edu

October 24, 2017

Abstract

Asynchronous many-task (AMT) frameworks are maturing as a model for computing simulations on a diverse range of architectures at large-scale. The Uintah AMT framework is driven by a philosophy of maintaining an application layer distinct from the underlying runtime while operating on an adaptive mesh grid. This model has enabled task developers to focus on writing task code while minimizing their interaction with MPI transfers, halo processing, data stores, concurrency of simulation variables, and proper ordering of task execution. Uintah is also exploring portability through task code written using Kokkos constructs and a generalized Uintah API.

Nvidia GPUs introduce numerous challenges in maintaining this clear separation between the application and runtime layer. Specifically, Nvidia GPUs require code adhere to a proprietary programming model, use separate high capacity memory, utilize asynchrony of data movement and execution, and partition execution units among many streaming multiprocessors. Abstracting these GPU features into an application layer requires numerous novel solutions to both Uintah and Kokkos.

The focus of this research is largely split into two main parts, performance and portability. Runtime performance comes from 1) preparing and executing a heterogeneous mixture of tasks to keep compute node processing units busy and 2) minimizing runtime overhead when preparing tasks for execution. Uintah's target problems heavily rely on halo data dependencies, and so automated halo processing receives significant emphasis. In addition, this work covers automated data movement of simulation variables between host and GPU memory as well as distributing tasks throughout a GPU for execution. This work does not describe strategies for optimizing task code, as that is an application developer responsibility. Rather, this work enables application developers to achieve good wall time performance with low runtime overhead.

Portability is becoming an productivity necessity for Uintah application developers as simulations can require a compute node process thousands of tasks with thousands of simulation variables and potentially hundreds of thousands of data dependencies across multiple adaptive mesh refinement layers. Application developers struggle to maintain one set of CPU tasks for single CPU core execution, another set of GPU tasks written in CUDA code, and a third set of code for Xeon Phi parallel execution. Uintah seeks a portable solution through 1) unifying Uintah API for CPU and GPU tasks, and 2) adopting Kokkos as a portability layer enabling developers to write task code once while maintaining performance. Currently, Kokkos only supports synchronous GPU data copies and code execution, and Kokkos itself must be modified for asynchrony to performantly execute GPU tasks in Uintah's AMT runtime. This research will cover both Uintah API and Kokkos changes and demonstrate results by applying these changes to production ready tasks.

An overview of other runtimes and parallel tools is given in Chapter 2. Chapters 3 describes the prior state of Uintahs GPU engine. Chapter 4 outlines work completed to date. Chapter 5 provides remaining work required to meet the full goal of this thesis. Chapter 6 outlines the proposed thesis format. The remainder of this document contains the Conclusion, references, and a list of my publications.

Contents

1 Existing Runtime Systems and Parallel Tools **1**

2 Uintah GPU Support Prior to This Research **1**

3 Current Work and Preliminary Results **1**

4 Work Plan and Implications **1**

4.1 Thesis Format 1

1 Existing Runtime Systems and Parallel Tools

This part talks about related work of your proposal.

2 Uintah GPU Support Prior to This Research

The content of your proposal. Each topic occupies one section, each with their own conclusion and future work.

3 Current Work and Preliminary Results

The content of your proposal. Each topic occupies one section, each with their own conclusion and future work.

4 Work Plan and Implications

Provide an overview of what you have done and what need to be done.

4.1 Thesis Format

References