Ph.D Research Proposal

Doctoral Program in Scientific Computing

Concurrent and Performant GPU/Heterogeneous Runtime System For Many-Task Computational Framework

Bradley Ryan Peterson

[bradpeterson@gmail.com](mailto:bradpeterson@gmail.com)

Advisor:

Dr. Martin Berzins

School of Computing

The University of Utah

**ABSTRACT**

Asynchronous-many task frameworks are maturing as a model for computing simulations on a diverse range of architectures at large scale. As more architectures are introduced, both runtime and application developers are increasingly challenged to ensure their code enables performant execution on various central processing units (CPUs), Intel Xeon Phis, and graphic processing units (GPUs). Nvidia GPUs exasperate these challenges by requiring code adhere to a proprietary programming model, using GPU specific high capacity memory, asynchronicity of data movement and execution, and partitioning execution units among many streaming multiprocessors. The Uintah framework’s approach to these challenges is largely driven by a philosophy to maintain a runtime layer separate from an application layer, shielding application developers from the complexities of data dependency scattering, gathering, and task preparation. Recently, Uintah has been adopting the Kokkos programming model for application developers to easily introduce code portability among various parallel architectures. However, both the Uintah framework and the Kokkos programming model had difficulty providing under-the-hood solutions to better allow application developers to achieve full GPU performance without employing restrictive GPU-specific development strategies. The research in this work demonstrates several novel solutions for GPU/heterogenous task scheduler design, concurrent data store implementation, and execution strategies to better enable Uintah to run existing task code while utilizing numerous unique features of Nvidia GPUs. This work required improving both the Uintah runtime system and the Kokkos programming model. The work is targeted to a wide range of applications, including simulations utilizing a few short-lived tasks with minimal data halos to full-scale problems production requiring hundreds of tasks per node and global data dependencies on thousands of nodes. This research concludes by demonstrating Uintah can realize both performance and portability among various architectures, especially GPUs, while avoiding burdening application developers with architecture specific development strategies.

**TABLE OF CONTENTS**

TABLE OF CONTENTS

[CHAPTER 1 – INTRODUCTION 5](#_Toc488678100)

[CHAPTER 2 – STATE OF THE ART 5](#_Toc488678101)

[CHAPTER 3 – RESEARCH OBJECTIVES AND APPROACH 5](#_Toc488678102)

[CHAPTER 4 – CURRENT WORK AND PRELIMINARY RESULTS 5](#_Toc488678103)

[CHAPTER 5 – WORK PLAN AND IMPLICATIONS 5](#_Toc488678104)

[CHAPTER 6 – CONCLUSIONS 5](#_Toc488678105)

[REFERENCES 5](#_Toc488678106)

# **CHAPTER 1 – INTRODUCTION**

# **CHAPTER 2 – STATE OF THE ART**

# **CHAPTER 3 – RESEARCH OBJECTIVES AND APPROACH**

# **CHAPTER 4 – CURRENT WORK AND PRELIMINARY RESULTS**

# **CHAPTER 5 – WORK PLAN AND IMPLICATIONS**

Limitations

# **CHAPTER 6 – CONCLUSIONS**

# **REFERENCES**