Computer Science Clinic

Statement of Work for
Harvey Mudd College

Statement of Work

September 25, 2014

Team Members

Brett Collins (Project Manager)
Alex Gruver
Ellen Hui
Tyler Marklyn

Advisor

Jeff Amelang

Liaisons

Carter Edwards
Robert Hoekstra

1 Problem Statement

Sandia National Laboratories currently has many libraries that do not utilize new manycore architectures. Using their Kokkos library, which was created to make it possible and easy to port code to new architectures (specifically manycore architectures), we will make their tensor contractions library and, as time permits, other libraries thread scalable.

2 Background

Sandia National Laboratories is a federally funded research and development center owned and managed by the Sandia Corporation. The laboratory's primary focus is the maintenance, management, and development of the United States' nuclear arsenal. Sandia also performs research in the fields of supercomputing and scientific computing. The Trilinos Project, developed and maintained by Sandia, is a collection of open source libraries intended for use in large-scale scientific applications.

One of the packages in Trilinos is Kokkos, which is designed to aid portability and performance in software written for manycore architectures. Well designed and well implemented parallelism can yield significant speedup in certain processing-heavy applications, including many in the scientific computing domain. However, optimizing an application to take advantage of many processors is highly dependent on the architecture on which the application is to run. In addition, highly parallel code may suffer from race conditions between threads, causing either incorrect results or slower runtimes resulting from the steps necessary to ensure correctness.

Kokkos attempts to mitigate these issues by allowing programmers to write their code once, and compile for optimization on a variety of manycore architectures. This is possible because the library manages the allocation of memory for the programmer and optimizes storage for speed on the current architecture. Kokkos also allows users to write thread scalable software by providing an API for using fast, non-locking data structures.

However, the Kokkos package is still new and relatively untested. No large-scale projects have yet been written with it. Some small kernels have been rewritten using Kokkos, but there are still many kernels in Trilinos that would benefit from increased thread scalable parallelization. We hope that we will be able to both improve the quality of the

3 Goal

The goal of this clinic is to rewrite several Trilinos kernels to be efficient and thread-scalable on manycore architectures, using Kokkos. These redesigned and reimplemented kernels will be integrated into Trilinos' production code, as both a performance improvement and as a further test for Kokkos.

The two kernels we plan to work on are Intrepid, a tensor manipulation library, and

4 Objectives

As stated, the primary goal of this clinic is to transform Sandia's tensor manipulation and *X* library to be thread scalable using Kokkos. This goal can be broken down into a couple different objectives. First, our team needs to get familiar with Sandia's Kokkos library. Learning the syntax benefits and disadvantages of Kokkos will allow us to optimize the libraries. Our second major objective will apply the knowledge that is learned from completing the first objective: rewritting and refactoring Sandia's tensor contraction to be thread scalable using Kokkos. The third objective, which is essentially the same as the second, is to make the *x* library thread scalable also. All of these objectives are big and very time consuming, which is why they will be broken down more in the Tasks section.

4.1 Optional Objectives

The optional objectives presented in this section are to be done if time permits. Firstly, if the team has finished, or has made significant progess towards making Sandia's tensor manipulation library thread scalable, then we will give a seminar on the use of Kokkos to a group of Sandia's employees during the site visit. After making the tensor manipulation and *TODO* libraries thread scalable, there are a couple different objectives that can be pursued, depending on the priority given by the liaisons.The first possible objective is to continue making more of Sandia's libraries thread scalable. This benefits Sandia by making more code run faster. While another possibility is to code versions of the tensor manipulation and *TODO* libraries in CUDA, or another framework such as OpenCL, in order to compare their performance with Kokkos' performance. The performance tests can show the strengths and weaknesses of Kokkos and give information that is use-

ful to the developers of Kokkos. The performance tests and examples of Kokkos' simplicity can be useful examples for helping Sandia try to integrate Kokkos into the C-standard.

5 Tasks

Our first task for this semester is to acquire a desktop with suitable computational capabilities. In practice, this translates to either an NVIDIA K20X tesla card or an NVIDIA Titan (Jeff please confirm) . This will allow us to test the thread scalability of our programs and ensure that they meet Sandia's needs.

After this, we will try to familiarize ourselves with massively multithreaded programming by working through a series of example problems in a variety of environments. So far, we have identified numerical integration, matrix multiplication, and histogram calculations as likely candidates for practice problems. We will try to implement these problems in Intel's Threaded Building Blocks, CUDA, and OpenMP.

After familiarizing ourselves with TBB, CUDA and OpenMP, we will have a background to understand the ideal way to write code using Kokkos. At this point, we will begin to write solutions to our practice problems using Kokkos. We can then compare the speedup gains on Kokkos compared to the other libraries, and also the ease of writing thread scalable code using Kokkos compared to the other libraries. This will be good practice because ideally, we will also test a few of the kernels we write so that we can see differences is ease of coding and speedup in them as well.

At this point, we will shift our focus to our first kernel. While we have a significant mathematics background none of us have any experience with tensor contraction. We will need to explore the main algorithms used for tensor contraction and determine if there are existing procedures for parallelizing the process. The original Kokkos source code, along with online materials should help us with this.

Once we are familiar with Kokkos, and we feel we have enough of a background with the technologies, we will begin working on improving the tensor contraction library. At this point in the project it is difficult to create an accurate timeline for this process, but we hope that we will be able to present our progress at the end of semester site visit at Sandia.

After the tensor contraction library ..... TODO