Statement of Research

Rahulkumar Gayatri

April 26, 2018

1 General Introduction

1.1 Motivation

Many high performance research centers are involved in building an exascale machine, i.e., a machine capable of performing a billion billion calculations per second. While hardware to perform such computing is available, a close integration between software programs and hardware is necessary. To this end prallel programming models such as OpenMP and MPI are used to take advantage of the powerful hardware. However, porting a code into a parallel framework is a difficult task, especially for legacy codes that are written by multiple developers over a long period of time. Furthermore, the complexity involved in writing a portable code makes this task even more challenging. Most of the big machines are a combination of a CPU and a multi-core processor or an accelerator such as Nvidia's graphical processing unit (GPU) or an field programmable gate arrays (FPGA) or an Intel's Xeon Phi. An application code optimized for a single architecture may not necessarily elicit us the best performance on another. In this regard, my current work is focussed on using different programming models such as OpenMP{3.0, 4.5}, Kokkos, Cuda, Raja and HPX to assess:

1. the effort needed to write code using them
2. portability provided by each of the models

As describe earlier it is challenging to write portable codes, which take advantage of the underlying hardware to provide good performance. Building on my expertise acquired during my doctoral work, I wish to continue my association in the parallel application and tools development, which are imperative to take advantage of the big machines of the future.

1.2 Research Statement

The need for models and tools which aid in the development of applications, to effectively use the underlying hardware is imminent. To this end, I am presently working towards improving performance of applications. I would also like to work on the development of tools which offer insights into the behaviour of the applications.

2 Qualifications

2.1 PostDoc

As a PostDoc at Lawrence Berkeley National Lab, I am involved in the following two projects

1. SW4 - Seismic Waves of 4th order accuracy.
   It is an Exascale Computing Project (ECP), where my role is to optimize the performance of the code on Intel's Knights Landing (KNL) processors. For this, I use techniques such as cache-blocking, vectorization and reducing the overhead incurred due to OpenMP directives. We are currently working on running large scale simulations of SW4 on the Cori supercomputer. For this the main objective is to use all 9K KNL nodes available on Cori.

2. Performance Portability - I also work on implementing portable application codes using programming models such as OpenMP{3.0, 4.5}, Kokkos, Cuda, Raja. The aim is to determine the effort required and the performance achieved when using these programming models. I am currently working on porting Berkeley GW (BGW), a set of material science application kernels using the above-mentioned programming models.

2.2 PhD Degree

I graduated with a PhD from Barcelona Supercomputing Center (BSC), Barcelona, Spain in March, 2015 under the supervision of Rosa M.Badia and Eduard Ayguade. During this period, I was a part of the Programming Models group that worked on the OMPSs framework. My Doctoral thesis was focused on speculative synchronization techniques for OMPSs, a task-based programming model. For my Doctoral thesis, I extended the framework to speculatively update shared memory locations using Software Transactional Memory (STM) instead of the traditional lock and mutex based mechanisms. An extension to speculative memory updates is the speculative execution of tasks, where tasks can be scheduled before their presence in the execution flow can be confirmed. I implemented a lighweight rollback mechanism which could undo the updates of tasks in case of speculation failure. The idea of greedy task execution improved the performance by an average of 20% for a select category of applications. At BSC, I also worked on porting applications from the domain of linear iterative solvers and graph algorithms using SMPSs, OMPSs implementation for SMPs. These applications are now a part of their application repository.