CSE523
Enabling Distributed OpenMP Codes to Share Data

Prerequisites

- Basic understanding of multicore systems
- An interest in programming languages and their implementation
- Preferable is some experience in parallel programming
- Enrolled MS/PhD student
- Proficiency in C/C++

Motivation

Many applications use the OpenMP programming interface to create parallel versions of their C/C++ programs that can exploit the multiple cores in modern computer systems. It can also be used to run codes across CPUs and attached GPUs. Applications running on cell phones through the largest computers on the planet use OpenMP to enable the compute cores to collaborate to solve problems quickly. There is interest in expanding OpenMP so that it can also help in the creation of programs that run across several compute “nodes” of a High Performance Computing (HPC) system. One possible means to enable this would rely on creating a special kind of memory for storing variables that may be shared by parts of an OpenMP program that are running on different nodes.

Overview

OpenMP is a directive-based parallel programming model, maintained by an organization called the OpenMP Architecture Review Board (OpenMP ARB), that has bindings for the C, C++ and Fortran programming languages. Information about it, along with its specification, can be found at the official website [http://www.openmp.org](http://www.openmp.org). It was originally designed to enable a straightforward means of creating shared memory parallel programs that can run across multiple CPUs that share memory, as well as on multi-core platforms. It was subsequently expanded to also support the programming of GPUs in a portable manner. Supported by many
vendors, it is available in a number of compilers and offers ease of programming as well as portability.

OpenMP allows the application developer to map the computation in a program to a team of collaborating threads, and to synchronize the threads as needed. It also allows for the creation of tasks that can be executed asynchronously on available resources. We have been experimenting with new features and implementation strategies for OpenMP, relying on the LLVM compiler infrastructure for our practical work. Stony Brook is a member of the OpenMP ARB and we engage extensively in both the OpenMP and LLVM communities. Our recent work has included providing optimizations for offloading code and data to GPUs, and exploring extensions to better support systems that have multiple GPUs configured within a node.

One of the recent introductions to OpenMP is a means to map data to memories with specific attributes. This has been used e.g. to create code that specifies which variables should be stored in high bandwidth memory, when available. We would like to expand this mechanism to create "global" memory that can be used to share data among OpenMP tasks that may be running on different GPUs or even across different nodes of a platform. This data would be managed by using calls to a library called OpenSHMEM. We have also worked with OpenSHMEM for a number of years and are thoroughly familiar with its use.

Project

This project will investigate how to extend OpenMP’s memory interface to include “global memory” and a means of allocating data in it. Since the data stored in global memory may not be local to the thread using or updating it, special support will be needed for these operations if the details are to be transparent to the application developer. The project will experiment with the use of single-sided communication provided by the OpenSHMEM programming library to manage these operations. Our group creates a reference implementation of OpenSHMEM that will be used in this work.
The outcomes are:

- A written report on the work done that includes a description of the proposed extensions to OpenMP
- The code created
- Presentation to our research group
- Demonstration of the use of these features on a simple code

For More Information
● Our group
  ○ https://you.stonybrook.edu/exascallab/
● http://www.openmp.org
● http://www.openshmem.org/