Tutorial

Title: An Introduction to OpenMP and its use on clusters

Presenter(s):

Tim Mattson
Intel Corporation

Tutorial Abstract:

OpenMP is an industry-standard API for programming shared memory computers. It is available on most if not all commercially available shared memory computers.

With OpenMP, you direct the compiler to create multi-threaded blocks of code by adding compiler directives to your program. It is easy to use and in many cases supports the incremental addition of parallelism to a program.

So why should a shared memory API such as OpenMP be of interest to the cluster computing community? There are two reasons. First, many clusters are built from shared memory nodes. OpenMP can be used to exploit parallelism on a node while a distributed memory API is used between nodes. Second, OpenMP is evolving to address non-uniform memory architecture (NUMA) computers. A cluster running some sort of distributed shared memory (DSM) is an extreme case of NUMA. Hence, in some cases, you can program a cluster using OpenMP.

This tutorial will consist of three modules. In the first module, we will introduce OpenMP. We will cover its history, the API itself, and the infrastructure in place to make sure it evolves to meet the changing needs of parallel programmers.

In the second module, we will discuss how to use OpenMP and MPI together to program clusters of shared memory computers. Some simple examples will be giving to show the rules that must be followed to safely mix the two programming models.

Finally, we will discuss some of the ways OpenMP might evolve to support programs running across the nodes of a cluster.

Detailed Description:

The availability of high-speed networks and increasingly powerful commodity microprocessors are making the usage of clusters, or networks, of computers an appealing vehicle for cost effective parallel computing. Clusters, built using commodity-of-the-shelf (COTS) hardware components as well as free, or commonly used, software, are playing a major role in redefining the concept of supercomputing. In this tutorial, we discuss the motivation for the transition from using dedicated parallel supercomputers, to COTS-based cluster supercomputers. We also describe the enabling technologies and then present a number of case studies of cluster-based projects to support our discussion. Finally, we summarise our findings and draw a number of conclusions relating to the usefulness and likely future of cluster computing.

The commercial success of clusters has pushed them into mainstream general purpose computing. Today clusters are not only used for HPC, they have been used in the areas of mission-critical, web serving, and database applications. A number of commercial applications are ex-

ploiting commercial applications. In this tutorial we will discuss architectural model of such applications with case studies.

This tutorial will start by discussing the motivation for cluster computing and endeavour to make a clear distinction between it and parallel/distributed computing. We then focus on cluster computer architectures, enabling technologies, hardware and software structures with an emphasis on recipes for building ones own high performance cluster system based different OS, networking, and middleware technologies with example systems. The remaining part of the tutorial focuses on architectural issues related to cluster middleware, the provision of environments with a Single System Image and various programming paradigms; these include distributed shared memory models, message passing as well as Java for HPC. During the discussion of each of these topics, we will present latest developments in that area and possible future issues that researchers and developers need to address.

The tutorial concludes with an open discussion of the work being carried out by various international groups on cluster computing. It is hoped that this discussion will help stimulate interest in the widespread take up of COTS-based clusters for high performance computing.

Schedule:

Part 1 (45%): OpenMP Introduction
Part 2 (25%): Mixed mode OpenMP/MPI programming
Part 3 (30%): OpenMP and DSM on clusters

Note - % indicates the percent of overall time dedicated to each topic.

Duration:

Half day tutorial.

Level:

40% Introductory, 40% Intermediate, and 20% Advanced.

Required experience:

The audience should have experience programming computers using C or Fortran. No parallel programming experience is required.

Presenter's profile:

Tim Mattson
Intel Corporation

Tim Mattson, Intel Corporation, Senior Research Scientist and OpenMP Program Manager
Dr. Mattson earned a PhD. in Chemistry for his work on quantum molecular scattering theory. This was followed by a Post-doc at Caltech where he ported his molecular scattering software to the Caltech/JPL hypercubes. Since then, he has held a number of commercial and academic positions with computational science on high performance computers as the common thread.

Dr. Mattson joined Intel in 1993 to work on a variety of parallel computing problems. This included benchmarking, system performance modeling, and applications research. He was one of the lead scientists on Intel's ASCI teraFLOPS project: a project that resulted in the first computer to run MPLINPACK in excess of one teraFLOPS.

Currently, he is a member of Intel's Microcomputer Software Laboratory where he is working

on technologies to make parallel computing more accessible. He is one of the authors of OpenMP - an industry standard for programming shared memory computers. In addition to his OpenMP work, Dr. Mattson is actively involved with defining software middleware to support technical computing on clusters of workstations.

Instructor experience

I have given OpenMP tutorials at conferences all over the world. This includes SC'98, SC99 and SC'00; the OpenMP workshops in Europe (EWOMP99) and the U.S. (WOMPAT'2000), and at Para2000. I think I am good at presenting OpenMP tutorials. Given that requests for my OpenMP tutorials keep coming-in, others seem to think they are good as well.

As for my qualifications, I am one of the founding members of the OpenMP Architecture Review Board (the ARB). I was an active participant in the creation of each of the OpenMP specifications and I led the OpenMP 2.0 project. Finally, I chair the ARB committee on the future of OpenMP where we regularly discuss how to move OpenMP onto clusters.

Finally, I have a great deal of experience with cluster computing. I built my first cluster in 1989 and helped organize the FSU cluster computing workshops in 1991 and 1992. I worked extensively with different API's for programming clusters throughout the 90's. I am currently leading an industry-wide effort to create a common Open Source cluster development kit (a project conducted in collaboration with IBM, SGI, ORNL, NCSA, and MS-software).