Extending OpenMP to Clusters

OpenMP is a well-known parallel programming paradigm for shared-memory multiprocessors. In the past, OpenMP has been confined to Symmetric Multi-Processing (SMP) machines and teamed with Message Passing Interface (MPI) technology to make use of multiple SMP systems. A new system, Cluster OpenMP, is an implementation of OpenMP that can make use of multiple SMP machines without resorting to MPI. This advance has the advantage of eliminating the need to write explicit messaging code, as well as not mixing programming paradigms. The shared memory in Cluster OpenMP is maintained across all machines through a distributed shared-memory subsystem. Cluster OpenMP is based on the relaxed memory consistency of OpenMP, allowing shared variables to be made consistent only when absolutely necessary.

Performance Considerations for Cluster OpenMP

Some memory operations are much more expensive than others. To achieve good performance with Cluster OpenMP, the number of accesses to unprotected pages must be as high as possible, relative to the number of accesses to protected pages. This means that once a page is brought up-to-date on a given node, a large number of accesses should be made to it before the next synchronization. In order to accomplish this, a program should have as little synchronization as possible, and re-use the data on a given page as much as possible. This translates to avoiding fine-grained synchronization, such as atomic constructs or locks, and having high data locality.

Cluster OpenMP does not perform well for all types of programs, but programs with certain characteristics can achieve reasonably good performance on a cluster, compared with attainable performance on a hardware shared memory machine.