MPI Communication in SMP Clusters
PENTTI HUTTUNEN, JOUNI IKONEN, JARI PORRAS
Department of Information Technology
Lappeenranta University of Technology
P.O.Box 20, FIN-53851 Lappeenranta
FINLAND

Abstract: - The recent years have seen a considerable increase in the number of cluster systems. These systems provide a very good performance-cost ratio. However, in order to meet the requirements for ever-increasing computing power of present day applications, several SMP cluster systems have emerged. By deploying two or more processors per workstation can increase the performance of a cluster significantly. The introduction of SMP workstations to a cluster adds a new dimension to the communication between processors. The purpose of this paper is to discuss the impact of SMP workstations in a cluster with respect to the communication. The communication mechanisms are studied and their performances are evaluated. In addition, performance results from a special communication library called MPIT, which is optimized to handle communication operations in SMP clusters, are compared to the traditional communication schemes available for the SMP cluster environments.

Key-Words: - SMP, Clusters, Communication, MPI, Networks of workstations, message passing, performance evaluation

1 Introduction
As the speed of processors has doubled every 18 months, it has put pressure on memory and network designers. Both memory and network access times have fallen behind the speeds of processors. Especially, in high performance computing, where the number of processors is in the order of thousands, has suffered from the discrepancy between processor speeds and memory and network access times. This article focuses on studying the impact of the network access time (subsequently called merely communication) on the performance of parallel applications.

There are studies about communication in SMP clusters [1][2]. As the majority of the research in literature has focused on the overall performance or on a small detail of an SMP cluster, this article aims at giving insight to the multi-level communication schemes on SMP clusters.

Communication on an SMP cluster is studied with respect to the performance of the MPI communication library. The MPI (Message Passing Interface) library has become a de facto standard for implementation communication operations in parallel applications. MPI itself is merely a standard that has spawned several implementations. Perhaps, the most popular ones are MPICH [3][4] and LAM-MPI [5].

This article evaluates the performance of various MPI implementations on an SMP cluster with three Ethernet and Myrinet networks. The cluster is made of 6 dual Pentium III 750MGz Linux workstations with 1-2 GB of RAM. However, only two machines are used at a time due to the type of measurements required. The study considers various levels of communication ranging from an intra-SMP message transfer to an inter-SMP communication. The results include latency and bandwidth measurements from the generic MPICH library [3][4] used in the Ethernet networks, the MPICH-GM library [6] optimized for the Myrinet network, and the MPIT library [7] implemented by the authors to provide a low-overhead MPI communication paradigm on SMP clusters.

The article is structures as follows: Subsection 2 gives a short introduction to SMP clusters, their characteristics, and interconnection networks. Communication libraries are discussed in subsection 3, whereas subsection 4 explores the communication schemes in an SMP cluster. The results from the performance evaluation of an SMP cluster are depicted in subsection 5. Subsection 6 includes discussion about the performance results, and the conclusions are drawn in subsection 7.
2 SMP Clusters
An SMP cluster is comprised of one or more multiprocessor workstations. A cluster is called heterogeneous, if it has workstations with different computing capabilities. In high performance computing, most dedicated clusters are homogeneous (all workstation identical with respect to their computing capabilities) to simplify scheduling and load balancing.

2.1 SMP Workstations
An SMP workstation has at least two processors that are connected to each other with a network. In most cases, the network is a bus that also connects the processors to a global memory. If the workstation has a very large number of processors, a different kind of network topology is required due to the poor congestion characteristics of the bus topology.

The global memory is used by the processors to communicate. The communication operation, in the existence of a global memory, is a memory reference operation; the sender saves data to a predefined memory location where it is read by the receiving processor. Since the global memory is accessible by both the processors, a synchronization method is required to prevent concurrent memory access to any given location; the main cause for the synchronization method is the need to protect two or more write operations to the same memory location.

The performance of an SMP machine is determined by the CPU speeds, memory characteristics (caches, amount, network), and software (operating system, compilers). In addition to these factors, the communication between the processors on an SMP has to be considered. Even though the communication occurs transparently through memory references, it still affects the performance of a parallel application. Therefore, as part of this study measurements were carried out to evaluate the latency and bandwidth of the network connecting two processors within an SMP workstation.

2.2 Interconnection Networks
When a cluster is created by connecting a number of SMP workstations with an interconnection network, new performance characteristics are introduced. The performance of an SMP cluster is not only dependent on the aforementioned characteristics but also on the interconnection network, and the software to utilize the network.

There are a wide range of interconnection network technologies available. The most common, and inexpensive, interconnection network is Ethernet. The Ethernet standard defines three levels of performance in regards to the network speed: 10, 100, and 1000 Mbps. However, only the Gigabit Ethernet (1000 Mbps) network has gained popularity in the field of the high performance computing because of its relatively good performance. Among the more efficient interconnection networks are Myrinet [8], QsNet [9], and InfiniBand[10]. Myrinet and QsNet are proprietary networks by Myricom and Quadrics, respectively. InfiniBand is a standard that has been implemented by several manufactures such as Mellanox. Table 1 summarizes the characteristics of the four interconnection network technologies.

<table>
<thead>
<tr>
<th>Network</th>
<th>Bandwidth</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet 1</td>
<td>10 Mbps</td>
<td>Ethernet</td>
</tr>
<tr>
<td>Ethernet 2</td>
<td>100 Mbps</td>
<td>Ethernet</td>
</tr>
<tr>
<td>Ethernet 3</td>
<td>1 Gbps</td>
<td>Ethernet</td>
</tr>
<tr>
<td>Myrinet</td>
<td>2 Gbps</td>
<td>GM</td>
</tr>
<tr>
<td>QsNet</td>
<td>2.7 Gbps</td>
<td>Elan</td>
</tr>
<tr>
<td>InfiniBand</td>
<td>10 Gbps</td>
<td>Verbs</td>
</tr>
</tbody>
</table>

The common denominator of all the three fastest interconnection networks is that they aim at excluding the kernel from communication operations. Through kernel bypass these interconnection networks provide significant performance improvements over the Ethernet networks that require kernel involvement. All the networks have their own low-level communication libraries that allow for the programmer to access the network interface card without having to go through the kernel space.

3 Communication Libraries
In order to obtain the best performance from any given network communication operations should use the native communication library. However, in most cases the native communication library is excessively complex and hard to use. For example, it often requires explicit connection establishment and termination operations. The complexity of the native communication libraries has led to the emergence of high-level communication libraries. The most well-known and widely adapted communication library is the Message Passing Interface (MPI) [10]. MPI is actually a standard that
a set of primitives to send and receive messages to and from processors. The standard also includes primitives for collective communication and synchronization operations. Many network manufacturers provide their own implementations of the MPI standard that is built on top of native communication library. This is the case for Myrinet, QsNet, and InfiniBand. In addition to these highly optimized versions of the MPI standard, there are several generic implementations that aim to provide portable programming methods to the field of parallel computing. One of such libraries is called MPICH, which is commonly used as a base implemented by the network equipment manufacturers to build their optimized implementations. The layered design of the MPICH library allows for the network manufacturers to replace the generic network layer (called Abstract Device Interface) with a version optimized for their network. The generic and optimized versions of the MPICH library are used to evaluate the performance of an SMP cluster in this article.

4 Communication in SMP Clusters
The communication occurs on two levels in an SMP cluster. If the communicating processors are located on the same workstation, the communication is denoted as intra-SMP communication, whereas otherwise the communication is known as inter-SMP communication.

4.1 Intra-SMP Communication
Since the communication over a global memory is faster than over an interconnection network, it is prudent to take advantage of the global memory whenever possible. However, this requires that an MPI implementation can detect and act upon the knowledge that the communicating processors are located on the same workstation.

Fig. 1 shows a comparison of thread and MPI communication operations in the presence of a global memory. The MPI results have been obtained from an implementation that does not support the communication through the global memory. The results indicate MPI latencies between two processors on the same workstation. The thread communication times were obtained by measuring the time from a memory write operation (to a specific memory location) by a thread until the data was read by another thread. The figure clearly indicates that the utilization of threads and the global memory, in particular, to communicate between processors is an order of magnitude faster than utilizing a generic MPI implementation. Note that two sets of results are shown for threads; one with synchronization and one without it. The only way to properly handle communication through the global memory is to deploy a synchronization method. Therefore, the results for the thread communication without synchronization are provided as a reference to illustrate the overhead of the synchronization method.

The MPICH implementation has the capability of utilizing the global memory. The communication occurs with the help of a shared memory region providers by the operating system. The involvement of the operating system is required, since the processors execute different MPI processes. Should there be a single process with multiple threads to utilize the processors, the communication would become a memory reference. In fact, the MPIT library, created by the authors, deploys such an approach which is discussed later. In addition, due to the process level approach and the fact that one of the design goals of the MPICH was to be portable, the shared memory support is not as optimized as it is in the LAM-MPI implementation [5].

![Fig. 1. Thread vs. MPI communication through a global memory.](image-url)
operations to transfer data over a network, such as connection establishments and terminations.

It is up to the programmer to determine when and how the processors communicate. The communication pattern and its timing have an impact on the performance of the communication operations. Therefore, it is essential that the communication requirements and implementation are carefully considered. After all, the communication is overhead introduced by parallel computing, and decrease the maximum achievable speedup of a parallel application.

5 Performance Evaluation

The MPI performance measurements were performed between two dual Pentium III 750MHz Linux workstations in three Ethernet and Myrinet networks. The generic version of the MPICH library was used in the Ethernet networks, whereas the MPICH-GM library was available for the Myrinet network. It is important to realize that the MPICH-GM implementation supports shared memory communication between processors within an SMP workstation.

For each network, latency and bandwidth measurements were obtained. The latency indicates the one-way communication delay, whereas the bandwidth describes the transfer rate of a network and communication library. Performance results were gathered for various message sizes to determine the impact of message size on the latency and bandwidth.

There are a multitude of applications to obtain the MPI latency and bandwidth metrics. NetPIPE proved to be the most simple and effective benchmark tool for our purposes [11]. All measurements were conducted on two levels: intra-SMP and inter-SMP. The following subsections describe test configurations and provide latency and bandwidth results.

5.1 Intra-SMP Performance

As earlier discussed the global memory provides a means for the communication when the processors are located on the same workstation. The intra-SMP performance was measured with an MPI implementation that utilizes shared memory regions to pass messages between processors. This implementation still creates one process for each processor on the SMP workstation. Therefore, subsequent results were obtained from the MPIT library, which created only one process per workstation [7]. The library creates a user-defined number of threads inside the process to utilize all the processors available on an SMP workstation. A detail description of the MPIT library can be found in [7].

Fig. 2. The communication latencies of MPICH implementations for Gigabit Ethernet and Myrinet, and the MPIT library over the global memory.

Fig. 2 depicts the latencies of MPI communication operations. The MPI implementations for Gigabit Ethernet and Myrinet network are capable of handling the communication over the global memory. The Ethernet results are approximately 10-times faster than in the case where the shared memory cannot be utilized. However, the latencies are still significantly inferior to Myrinet and the MPIT library. In fact, the results indicate that the Myrinet communication scheme is very fast (1.5 µsec). The low latency is contributed to the kernel bypass and the deployment of the memory on the Myrinet network interface card (instead of the global memory of the SMP workstation). The MPIT library performs nearly as good as the Myrinet network. The lowest latency observed was 4.0 µsec. The difference is explained by the fact that the MPIT library is implemented in the user-space and requires one memory allocation and two memory copy operations, i.e. kernel involvement is necessary. On the other hand, with message sizes from 2 KB to 128 KB the MPIT library outperforms the Myrinet network. It should also be noted that the main goal of the MPIT library is to provide a means to communication and computation to overlap.
5.2 Inter-SMP Performance

In order for two processors on separate workstations to communicate the messages have to traverse over an interconnection network. The network along with the message passing library defines the message characteristics such as latency and bandwidth. Naturally, the operating system is involved to a certain extent in the communication, but the tendency is to minimize the involvement of the operating system. Systems such as Myrinet provide direct access from the user space to the network interface. The elimination of the kernel (operating system) has been shown to improve the communication performance. In fact, most of the proprietary high speed interconnection network vendors have systems that bypass the kernel in order to keep up with the ever-increasing demand for lower latencies and higher bandwidths.

Fig. 3 and Fig. 4 plot the MPI latencies and bandwidths measured on the Ethernet and Myrinet networks. The Myrinet network clearly outperforms other networks when latency and bandwidth results are considered. Due to the proprietary technology the Myrinet communication bypasses the operating system, which results in significantly lower latencies and higher bandwidths.

6 Discussion

Table 1 shows the best results from various tests conducted. Also, references results from other high speed interconnection networks are displayed for comparison.

<table>
<thead>
<tr>
<th>Network</th>
<th>Library</th>
<th>Latency</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>MPICH</td>
<td>25.8 µsec</td>
<td>1238 Mbps</td>
</tr>
<tr>
<td>Memory</td>
<td>MPICH-GM</td>
<td>1.5 µsec</td>
<td>2081 Mbps</td>
</tr>
<tr>
<td>Memory</td>
<td>MPIT</td>
<td>4.0 µsec</td>
<td>3248 Mbps</td>
</tr>
<tr>
<td>Gigabit Eth</td>
<td>MPICH</td>
<td>147 µsec</td>
<td>360 Mbps</td>
</tr>
<tr>
<td>Myrinet</td>
<td>MPICH-GM</td>
<td>11 µsec</td>
<td>1788 Mbps</td>
</tr>
<tr>
<td>Myrinet</td>
<td>MPICH-GM</td>
<td>8.1 µsec</td>
<td>1824 Mbps</td>
</tr>
<tr>
<td>QsNet</td>
<td>MPICH</td>
<td>5.0 µsec</td>
<td>2456 Mbps</td>
</tr>
<tr>
<td>InfiniBand</td>
<td>MVAPICH</td>
<td>6.8 µsec</td>
<td>6648 Mbps</td>
</tr>
</tbody>
</table>

As the results indicate there are significant differences in the MPI performance depending on the network and message passing library. It was not a major surprise to observe that the Myrinet network provides the lowest latency and highest bandwidth. However, the performance results for the Myrinet network are also relatively better than the Gigabit Ethernet; the MPI communication over the Myrinet network achieves nearly 90% of the theoretical maximum bandwidth, whereas the corresponding figure in the Gigabit Ethernet network is 36%. This can be explained by the fact that the MPI implementation for Myrinet bypasses the kernel, which is not the case with the Gigabit Ethernet. Similar results are seen when the communication occurs between two processors on the same workstation. Although Myrinet provides the best performance out of the networks tested, it still falls behind when compared to Quadrics’ QsNet and InfiniBand networks.
7 Conclusions

The purpose of this paper was to discuss the MPI communication in an SMP cluster. As the SMP cluster imposes a new level of communication, i.e. intra-SMP, it is important to understand its implications. This paper provides results from various networks and MPI implementation to illustrate the significance of choosing the most suitable network and MPI implementation.

A generic MPI implementation was tested in Ethernet networks, whereas an optimized version of the MPICH library (MPICH-GM) was evaluated on a proprietary Myrinet network. Both implementations included a support for shared memory communication in the present of a global memory. The results obtained showed that the global memory is a substantially faster medium to transfer data, which in a sense was not a surprise. However, the MPICH-GM library for the Myrinet exhibited extremely low latencies when communicating between two processors on the same workstation (1.5 µsec).

The MPIT library developed by the authors to allow communication and computation to overlap in an SMP cluster was also evaluated. The MPIT results were very close to the ones observed in the Myrinet network. The difference in favor of the Myrinet network is explained by the architectural design of the library. Even though, the library supports intra-SMP communication, its main objectives are elsewhere, such as in providing overlaps of communication and computation as well as computation and scheduling.

In conclusion, it is quite obvious that proprietary high speed interconnection networks with supporting software provide an order of magnitude better performance than traditional networks. Unfortunately, the proprietary networks are also relatively much more expensive. Regardless of their higher cost, such proprietary networks are required to prevent the network and communication from becoming a bottleneck while the speeds of processors continue to increase.

References:


