1 Introduction

Sandia University of New Mexico Operating System (SUNMOS) is a highly portable operating system for massively parallel distributed memory systems. SUNMOS was originally developed for the cCUBE-2 and was ported to the Intel Paragon in 1993. SUNMOS is designed to allow the application programmer as much of the hardware bandwidth as possible. As a result, SUNMOS has limited functionality but typically achieves better performance than other operating systems. If its functionality is insufficient for an application, the user must supply the needed functionality in the application program or by using library functions; this usually avoids degrading the overall performance in a multi-user environment. SUNMOS supports a large subset of the C library from the Open Software Foundation (OSF)/1 AD [6] with functions such asstdio library and the UNIX I/O system calls, but sockets, signals, pipes, pthreads and functions involving multi-tasking are not provided by SUNMOS.

Two operating systems, OSF/1 AD and SUNMOS, are available on Sandia’s Paragon. OSF/1 AD is the vendor supplied operating system for the Paragon; it takes about 8 MB of memory on each node to load OSF and the Mach micro kernel on which it is based. It sends messages at a bandwidth of 30-35 MB per second rather than the peak advertised rate (200 MB per second), with latencies of 100 microseconds, using Intel NX message passing calls. In contrast, SUNMOS takes only 250KB of memory on each node and messages can be sent at a bandwidth of 170MB per second with latencies of 70 microseconds [1]. SUNMOS is not 100% compatible with NX message passing, but it is close to it. However, a possible portability concern as a result of the design of SUNMOS is that SUNMOS does not have message-packet flow control. In OSF, NX uses handshaking to check the overflow status of the buffer on the receiving end, while SUNMOS is designed to send the non-packetized data across the mesh regardless of the buffer availability. SUNMOS applications that require high bandwidth of communication or I/O must be tuned carefully by the programmers because of the absence of the flow-control.

For communication between nodes under SUNMOS, the kernel transfers the entire message across the network in one single burst since no message packetization is used [4]. This scheme allows SUNMOS to achieve up to ninety-seven percent of the hardware bandwidth for large messages. However, for a bandwidth intensive application, a problem may arise when an application monopolizes access to communication channels by sending extremely long messages over one or more links; hence, other independent applications may get very poor performance. We can expect the same effects when an application requires heavy I/O operations.

To quantify the effects of such channel monopolization by the bandwidth intensive applications, we developed tests on the Intel Paragon and investigated the bandwidth intensive application’s interference on another applications. In Section 2, we give details on the bandwidth intensive application’s nature on the Intel Paragon. Section 3 explains the mesh configurations we used for the experiment and the methods of the testing we have performed. The detailed performance figures on which we base our conclusions are given at the end of the paper.

2 Applications on the Paragon

2.1 Paragon Hardware

Paragon systems are arrays of 8 to more than 1800 interconnected nodes. Each node can serve either as a compute node on which an application resides or as a service node. A service node can be a root node or an interface to the outside world or run interactive jobs such as shells and editors. The Intel Paragon at Sandia involved in our experiments has 1834 SUNMOS compute nodes and 11 service nodes; no OSF compute nodes are currently available. Each node is a single board with two processors and at least 16MB of memory. Both processors are i860XP microprocessors rated at 50MFLOPs peak double precision. One of the processor is used for general processing while the other is used as a message coprocessor. Each processor has a 16KB instruction cache and a 16KB data cache. The memory bus is fully coherent and is capable of a peak transfer rate of 400MB/s. Nodes are arranged in a two-dimensional mesh using the Hippi interface [1].

2.2 Bandwidth Intensive Applications

SUNMOS has many advantages over OSF, most importantly its portability and size. Loading data from memory to the processor under SUNMOS is much faster than under OSF. The performance of applications under SUNMOS is generally better than that under OSF. However, SUNMOS does not packetize messages. When the kernel transfers a message, the entire message is sent across the network in one single burst; this allows SUNMOS to get high bandwidth for large messages. It is possible for a bandwidth intensive application to monopolize access to communication channels by sending extremely large messages over one or more links. SUNMOS is well suited to computationally intensive applications that do not require heavy I/O operations. For the SUNMOS operating system that runs on the Intel Paragon mesh architecture, I/O intensive applications are a potential threat, since I/O messages are sent across the machine, possibly through nodes that are running unrelated applications, to I/O nodes that reside in the service partition. Thus, an I/O application could be robbing unrelated applications of access to communication channels.

To this end, we study the following problems. First, what is an acceptable size of the message in SUNMOS which doesn’t interfere with other applications? Second, how sensitive to overall network traffic are the throughput of the independent applications? Last, is packetization really needed to improve the overall performance under SUNMOS? In the next section, we quantify the effects of a bandwidth intensive application’s interference on another application and study their relations in terms of throughput through the common