Wide-Area ATM Networking for Large-Scale MPPs *

Philip M. Papadopoulos   G.A. Geist, II †

Abstract

This paper presents early experiences with using high-speed ATM interfaces to connect multiple Intel Paragons on both local and wide area networks. The testbed includes the 1024 and 512 node Paragons running the OSF [4] operating system at Oak Ridge National Laboratory and the 1840 node Paragon running the Puma [5] operating system at Sandia National Laboratories. The experimental OC-12 (622 Mbits/sec) interfaces are built by GigaNet and provide a proprietary API for sending AAL-5 encapsulated packets. PVM is used as the messaging infrastructure and significant modifications have been made to use the GigaNet API, operate in the Puma environment, and attain acceptable performance over local networks. These modifications are described along with a discussion of roadblocks to networking MPPs with high-performance interfaces. Our early prototype utilizes approximately 25 percent of an OC-12 circuit and 80 percent of an OC-3 circuit in send plus acknowledgment ping-pong tests.

1 Introduction

The idea of networking supercomputers to make an even larger "metacomputer" is not new. However, the availability of inexpensive off-the-shelf high-speed networking hardware has removed one roadblock to making metacomputing a reality. Long distance providers are just now making the networking infrastructure (for a price!) widely available at DS-3 speeds (45 Mbits/sec). OC-3 (155 Mbits/sec) and OC-12 (622 Mbits/sec) connections are available in limited areas. We are experimenting with OC-12 speed ATM boards that are interfaced as I/O nodes on Intel Paragons. Our testbed consists of a 512 node GP node machine (xps35, OSF operating system), a 1024 node MP node machine (xps150, OSF), and a 1840 node GP machine (xps140, Puma). The xps35 and xps150 are physically housed at Oak Ridge and are connected directly through an OC-12 switch. The xps140 housed at Sandia is connected to Oak Ridge via a DS-3 line which is shared with ES/Net IP traffic.

PVM was chosen as the messaging passing interface because of its ability to handle heterogeneous machine configurations and its "daemon-based" design where inter-machine traffic is easily separated from intra-machine traffic. This project is ongoing and Pittsburgh Supercomputing Center's Cray T3D will be added soon to the testbed thus making transparent heterogeneous messaging a necessity. PVM provides an excellent base, yet several practical modifications were made to support AAL-5 intermachine messaging, enable more efficient messaging, and function in the Puma environment. All modifications

*Research supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research, U.S. Department of Energy, under contract DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation

†Authors are with the Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6367

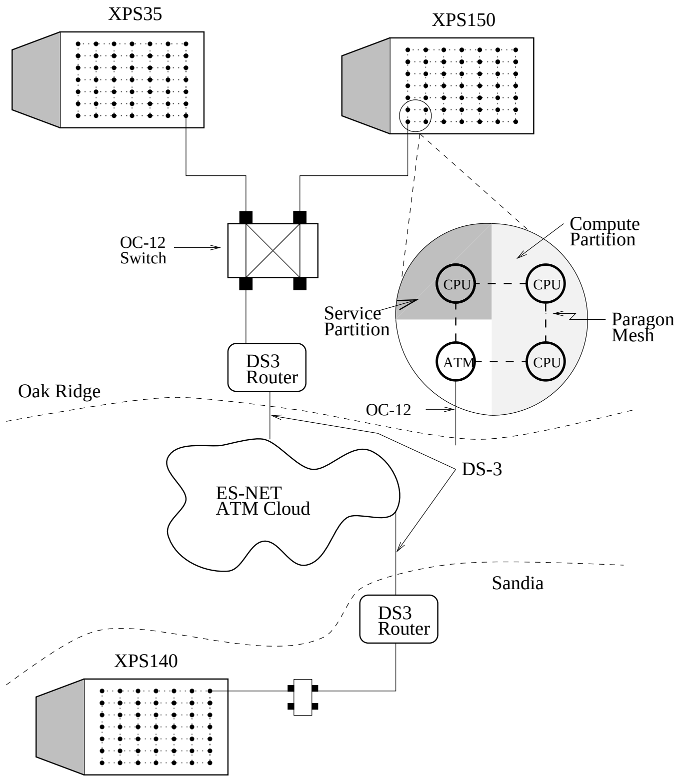

FIG. 1. *Paragon ATM testbed environment*

maintain the interoperability of ATM-connected Paragons with IP-connected workstations, making the high-speed networking support an added feature rather than a preemptive one. Section 2 describes the testbed in greater detail, Section 3 describes the modifications made to PVM and where some further improvements can be made, Section 4 points to some practical issues of networking MPPs, Section 5 presents some timing results and Section 6 presents conclusions and future directions.

2 Testbed Environment

The testbed environment is illustrated in Figure 1. The Paragon xps35 and xps150 are connected through an OC-12 switch. Connections between sites go through the ES/Net ATM cloud at DS-3 speeds via ES/Net DS-3 routers. Both Paragons at Oak Ridge are running OSF version 1.4.2, while the Sandia Paragon uses the Puma operating system, developed at Sandia. Each machine has an OC-12 ATM interface card built by GigaNet, Inc. GigaNet provides a user-level API that presents the AAL-5 (ordered, unreliable) encapsulation layer. The GigaNet design required OSF kernel modifications which have been folded into the 1.4 OSF releases. Their software interfaces to the machine at the mesh level making full-rate OC-12 transfers possible from any node in the Paragon. Essentially, nodes are given the illusion that the ATM interface is "local." The ATM hardware is designed so that multiple nodes may concurrently access the ATM hardware as long as the Virtual Circuit Identifiers (VCIs) used are unique. The software also provides a runtime adjustable "rate control" to clock bits onto slower networks at ratios of 1/4, 1/8, or 1/16 of the OC-12 line. This provides compatibility with slower physical networks and primitive traffic shaping without requiring extensive buffering in hardware switches.

Two different Paragon operating systems are used in the testbed: OSF and Puma. However, the ATM drivers operate only under OSF which implies that off-machine messages must be routed through a node running OSF. The structure of the Paragon operating environment divides the CPUs into three partition types: compute, service, and I/O. Puma is essentially a lightweight kernel for the compute partition. Some operating system components such as file system I/O are handled by a service node on behalf of the compute nodes. Under Puma, messages *must* go through a service node to ATM board. Puma provides a native messaging interface based on portals and supports a limited subset of the Paragon NX message passing library.

3 PVM Modifications

PVM is used for creating and controlling the metacomputer and implements a multi-protocol messaging layer to provide a familiar API to the programmer. Several important modifications were made to PVM to include the ability to operate through firewalls, utilize ATM inter-machine connections, and support the Puma OS. The basic structure of PVM provides a single daemon, the pvmd, which runs in the service partition of each Paragon. Each pvmd manages its own compute partition by spawning tasks and handling inter-machine messages. A collection of communicating daemons make up an entire virtual machine. Task-to-task messages that are to stay on a particular machine, bypass the pvmd and use the underlying scalable (NX, in this case) MPP message layer. There are two distinct message components in the MPP version of PVM: daemon communication for both intermachine and daemon-to-local-task communication; and intramachine communication over native messaging.

3.1 Intermachine Communication

The standard PVM daemon-to-daemon communication utilizes UDP as its basic transport. UDP is an unsequenced, unreliable transport so PVM provides its own positive acknowledgement with retransmission [2] to maintain intermachine links. For this testbed, UDP is replaced by AAL-5 packet encapsulation which is also unreliable but sequenced. For simplicity, the same PVM reliability algorithms are used for both ATM and UDP. The GigaNet API [3] allows any CPU to bind to a particular virtual circuit identifier (VCI) and communicate to the "other end" over this VCI. The testbed currently utilizes permanent virtual circuits (PVCs) because no provision is available in either the OC-12 switch or in the ES/Net cloud for switched virtual circuits (SVCs). Essentially, virtual circuits are hardwired in the network routers and PVM must communicate using these hand-configured connections. Because of local site security policies, a virtual machine can not be brought together without first using Ethernet-based authentication (manual startup with s/key one-time passwords). This means that daemons are configured initially to use UDP as the intermachine transport. ATM then replaces UDP by giving a single configuration command that specifies the machines that should connect together on a particular VC. With this architecture, the Paragon daemons become multi-protocol routers that switch messages among sockets (UDP, TCP), ATM, and native communication. Sockets are used to talk to non-ATM capable machines and to local service node tasks. Native communication is used for the daemon to talk to its local tasks on the compute partition. This protocol switching requires the daemon to enter a busy wait polling loop because the three protocols cannot be multiplexed in a single blocking wait call. The ATM-capable daemons maintain interoperability with standard PVM and keeps true to the idea that messages should travel

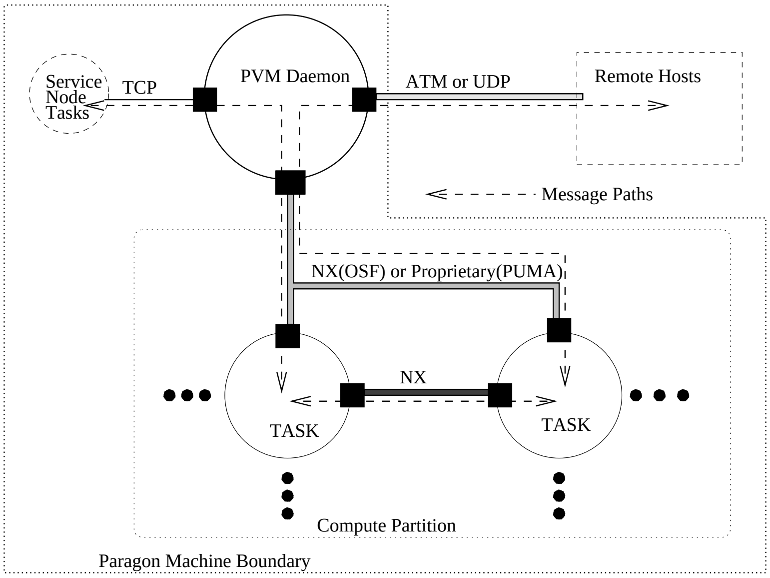

FIG. 2. *Message Pathways and Protocols*

over the fastest available transport.

3.2 Daemon-to-local-task communication

Daemon-to-local-task communication utilizes TCP or Unix domain sockets (sequenced, reliable) in the standard workstation version of PVM. The public Paragon OSF version utilizes NX both for daemon-to-local-task communication and for local task-to-task communication. This means that tasks can use a single protocol to message between the outside world and to local tasks. However, the Puma OS defines a unique collection of calls to enable service node (daemon) to compute node (task) messaging. This required a relatively straightforward change in the daemon to utilize a different message protocol. However, the major impact of this design is that tasks must now multiplex and differentiate between task-to-task communication and task-to-daemon communication. This required a complete rewrite of the low-level message passing logic. Puma daemon-to-task messages must arrive in pre-allocated buffers that are exclusive from the NX buffers. The OSF ports and Puma ports share more than 95% of the machine-specific code and both libraries use NX for local task-to-task communication.

Figure 2 illustrates the different paths a message may take and the different protocols that each node must handle. The message route from task to daemon to daemon to task is termed a 3-way route. These extra message hops add latency and generally impact bandwidth. However, it allows all the compute nodes to multiplex a single high-speed ATM inter-machine interconnect. This routing represents the worst-case scenario in that the pvmd can become a bottleneck. A logical extension is to provide multiple daemons and use the parallelism available in the GigaNet board to get better scalability and sharing of the ATM connection. This will be one avenue of future research.

4 Roadblocks to Wide-Area ATM networking with MPPs

Several issues must be dealt with to enable high-speed networking with MPPs. Once the physical connections are provisioned and intermachine messaging is enabled, the next "problem" to handle is efficient multiplexing of the ATM interface. Since 1000's of nodes are on either side of the single hardware interface, in excess of 1,000,000 "direct" interconnections may be desired by an application. Clearly, true direct task-to-task connection is impractical as the *only* intermachine messaging method. The physical roundtrip latency between Sandia and Oak Ridge is 100ms with the current fiber route. The best attainable roundtrip latency (speed of light in a vacuum) for a straight (2100 KM) connection is approximately 14ms. Per-message connection setup and tear down is also too expensive because this roundtrip latency would be added to every message. Some sort of efficient link-sharing must be instituted so that the illusion of direct connections can be given to the programmer. This allows the programmer to concentrate on how to hide the latency of the inter-machine messages and ignore specific routing details.

Our testbed uses a single task to multiplex the connection and provide reliability. Although simple, this places an unnecessarily constrictive message bottleneck in between the machines. Although it is possible for two compute nodes to talk directly over ATM and bypass the pvmd in communication, reliability would have to be provided either by PVM or by using a sequenced reliable protocol (TCP, e.g.). Since the number of direct connections is a restricted resource, only a limited number of nodes could actually communicate. Another approach is to use multiple "pvmds" to get better parallelism. Instead of funneling all inter-machine messages through a single process, multiple processes could share the load and utilize the parallelism in the GigaNet interface. A hybrid approach is also possible where several message routers have permanent connections, but tasks may set up on-demand direct task-to-task links. Messages would either travel over a direct link or through a pvmd, depending on what was available. For this to be practical, a mechanism would have to be designed so that under-utilized direct links could be torn down and used by other pairs of processes.

While network utilization is getting most of the attention in these early stages, issues such as fault-detection, secure configuration of virtual supercomputers, multi-site scheduling, fair resource management, distributed development environments, multi-platform debugging, high-speed and parallel I/O, and account administration are all essential to building a working production environment. As this project matures, efforts will increase in these areas to build a virtual supercomputer that looks and behaves as a monolithic machine.

5 Results

The usual metric of interest is node-to-node bandwidth. Several interactions must be dealt with to get efficient messaging. Our first attempt at ATM networking yielded bandwidths of only 2MB/sec over a latency-free OC-12 line (about 3% utilization). Our best bandwidth now is approximately 17MB/sec (25% utilization). Clearly there is room for improvement for an absolute bandwidth measurement. Some of the critical items that brought the bandwidth into a usable range were: memory alignment, efficient multiplexing, removing memory copies, and larger windows for intermachine packets.

PVM buffers are usually aligned to 8 byte boundaries. However, the Paragon NX messaging is about 3 times more efficient if messages are aligned to 16 bytes. This simple re-alignment provided a significant increase in bandwidth. Larger windows so that approximately 8 outstanding packets could be "on the wire" between hosts also improved bandwidth. However, the largest surprise was the cost of the system select call when the daemon is multiplexing between sockets and ATM/NX. A "zero" timeout select to effect

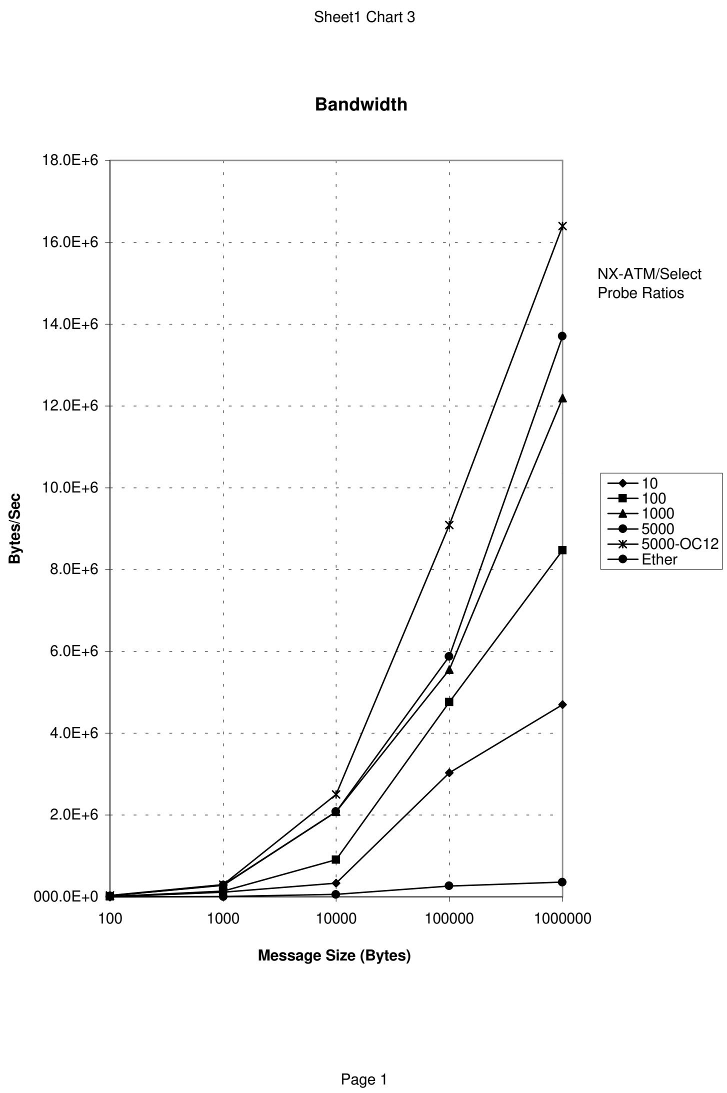

FIG. 3. *Send+Ack Bandwidth with several NX/select probe ratios. Hardware Output is rate-controlled to OC-3 speeds (except where noted)*

a poll of the socket interface consumes at least 10 ms in OSF. Figure 3 shows bandwidth when the ratio of ATM/NX probes to socket select calls is increased. These curves illustrate bandwidth when the OC-12 output is rate controlled to write a maximum of 155 Mbits (OC-3) with a send window size of eight 32KB packets. When the probe ratio is set at 5000 NX/ATM probes for every 1 select (socket) call, the bandwidth is observed at 13.8MB/sec or 80% of the OC-3 theoretical maximum. For the OC-12 curve, the maximum bandwidth is 17 MB/sec or 25% of the OC-12 maximum. Ethernet maximum bandwidth is approximately 375 KB/sec or about 93% of the Paragon's maximum TCP/IP stack speed. Latency was observed to be about 3.5ms roundtrip over ATM and 150ms roundtrip over Ethernet.

The bandwidths illustrated in Figure 3 are measured across a latency-free OC-12 link between the xps35 and xps150 with the time measured as send plus acknowledgment.

ES/Net has instituted policing on their routers to limit bandwidth between Sandia and Oak Ridge to 2 MBytes/sec (about 50% of the shared DS-3 line). Attempts to drive the network faster than this resulted in dropped packets and decreased throughput because of retransmission. The OC-12 interfaces in this case are rate controlled to output a maximum of 38 Mbits/sec (1/16 of OC-12). The network send window is adjusted to 4 outstanding 32 KB packets to limit the overall bandwidth to 2 MB/sec. The network latency is approximately 50 ms in each direction, and this makes it more difficult to sustain high-bandwidths when there are errors on the network. Our preliminary measurements show that PVM is able to saturate the 2 MB/sec link at 1MB message sizes.

6 Conclusions

This paper presents an experimental setup to network several large Paragons using state-of-the-art OC-12 ATM interface cards. Our bandwidth measurements show that high-speed networking can be utilized and usable speeds are achievable. But, room for significant bandwidth improvement remains. Future research will be to build multiple protocol routers that reside on the compute partition. These routers will eliminate costly socket calls and convert only between ATM and NX. Also, they will take advantage of the parallelism available in the ATM board to get better utilization of the available bandwidth when multiple processes are communicating. Since these routers will be specifically tuned to ATM and NX, the usable node-to-node bandwidth should increase. Although we have focussed on PVM in these first stages, the multi-router approach will be not be PVM-only so that plans to modify an MPI implementation for use across networked MPPs can be implemented. This project represents the first essential stages to getting usable networked MPPs with off-the-shelf technology and indicates that the resources can deliver the essential bandwidth.

References

[1] A. Beguelin, J. Dongarra, G. A. Geist, W. Jiang, R. Manchek, V. Sunderam, *PVM: Parallel Virtual Machine, A User's Guide and Tutorial for Networked Parallel Computing*, MIT Press, Cambridge, MA, 1994.

[2] D. E. Comer, *Internetworking with TCP/IP: Volume I, Principles, Protocols and Architecture*, 3rd Edition, Prentice Hall, Englewood Cliffs, NJ, 1995.

[3] D. R. Follett, M. C. Gutierrez, R. F. Prohaska, *A High Performance ATM Protocol Engine for the Intel Paragon*, Available at http://www.cs.sandia.gov/ISUG/html/hspe.html, 1995.

[4] Intel Corporation, *Paragon OSF/1 User's Guide*, Beverton, Oregon, Document number 312489-001, April 1993.

[5] R.J. Riesen, *Puma Web Documentation*, http://www.cs.sandia.gov/˜ rolf/puma/puma.html, April 1996.