

Giga-Plant Scalable Cluster*

David M. Halstead, Brett Bode, Dave Turner, and Vasily Lewis
Scalable Computing Laboratory, Ames Laboratory
Wilhelm Hall, Ames, IA 50011, USA, help@scl.ameslab.gov

Abstract

The Giga-Plant is a next generation compute cluster architecture under construction within the Scalable Computing Laboratory (SCL). This work describes the general cluster design philosophy utilized on this machine and others, and illustrates the performance evaluation process that was exercised in order to make an informed hardware purchasing decision. We present network communication throughput results taken from several hardware platforms using Fast Ethernet and Gigabit Ethernet with and without Jumbo Frames. We show that, despite throughput in excess of 800Mbit/s, communication latency is the critical factor in determining the viability of commodity network hardware in parallel processing applications.

1 Introduction

This work will document the general design philosophy used for clusters constructed within the SCL, a philosophy that extends from the component evaluation, to cluster machine task allocation, to the simplification of management issues and configuration of the user environment. To illustrate the viability of this solution we present results from three clusters currently on-site, each of which differs in scale and target application but adheres to the design goals of the Scalable Cluster Model. The first is a 64-node dual Pentium Pro machine connected via a high-density Fast Ethernet switch. The other two clusters are based on Alpha and Power 3 hardware connected by Gigabit Ethernet. We will also outline collaborative work underway to improve network throughput and a project to port the Maui Scheduler to the PC cluster environment, since these are vital components in making the cluster solution an acceptable, general purpose, compute engine for scientific and business applications. The next section will

introduce the evaluation tools used in the design process.

2. Benchmarking

2.1 HINT: Compute node evaluation

The black art of computer performance benchmarking has long been a sport enjoyed by vendors and endured by users during the purchasing cycle. Users either trust the limited machine performance cross-section probed by traditional benchmarks, or port their most demanding code to each new hardware platform as it becomes available (which can be a substantial task in itself). The HINT benchmark [1] is a second generation hardware performance profiling tool which uses a simple, scalable, numerical integration problem to probe the performance characteristics of the machine as a functions of memory used.

Traditional benchmarks are notorious for probing only one area of the performance profile of a machine. For example, the Dhrystone benchmark is very small, utilizing only a few hundred bytes of memory while the Whetstone benchmark uses a few thousand bytes. Both of these are suited to fast primary cache machines. The SPECint uses ~200 kBytes of memory while the SPECfp uses ~10MBytes [2]. The LINPACK benchmark varies with the size of matrix used, but occupies ~30 kBytes for the 100x100 and ~4 MBytes for the 1000x1000. The key aspect of HINT is that it is able to accurately predict the machine ranking that would be given by each of these other benchmarks in a single run, yet it is small and easy to port to different architectures. The output is a plot of the Quality Improvement Per Second (QUIPS) of the numerical integration as a function of problem size. We present a range of curves in Figure 1 to illustrate the power of

* This work is supported by the Applied Mathematical Sciences Program of the Ames Laboratory- U.S. Department of Energy under contract number W-7405-ENG-82

this tool in comparing popular hardware architectures from several manufacturers. The shape of the curve is similar for each machine, with the initial part of the curve rising in the region where performance is limited by the time taken to load the problem into memory and perform the first iterations. In all cases, maximum performance is achieved in primary cache, where the small size of the integration allows the problem to run at close to processor speed. As the numerical integration is performed with greater resolution, the problem grows into secondary cache (and tertiary cache on some platforms), the throughput performance goes through a series of plateaus until it demands access to main memory. Eventually the performance will fall to an unacceptable level as the machine is forced to swap data to disk. For clarity, the ordinate is plotted on a \log_{10} scale to reveal the full performance profile without it being dominated by the main memory results.

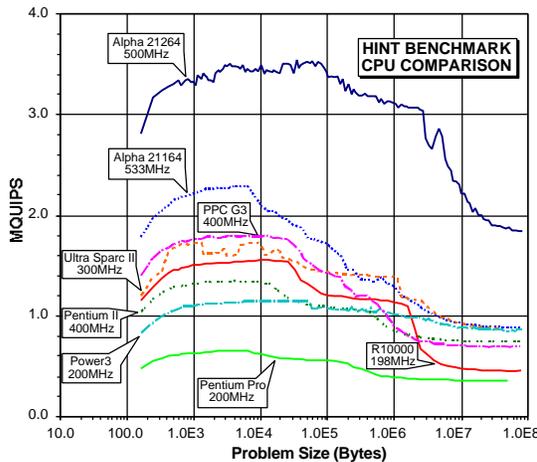


Fig. 1. Compute Node Performance Comparison

The price performance ratio has been the major factor driving the cluster revolution. This may be illustrated with the two Compaq Alpha CPU based hardware platforms tested. It is clear from the graph that the new 21264 architecture out-performs the older 21164 machine by a factor of between 1.5 and 2.7, depending on the problem size. The hardware price for the 21264 is, however, currently a factor of four times as expensive as the 21164 based machine, and requires the use of more expensive memory units to achieve its improved performance. Thus the SCL is currently constructing a cluster of 25 Alpha 21164 based machines using a high speed Gigabit Ethernet network switch.

2.2 NetPIPE: Communications fabric evaluation

The other key component of a cluster is the communication fabric that will facilitate message passing between the compute nodes. The Network Protocol Independent Performance Evaluator (NetPIPE) benchmark is a protocol independent utility, designed to probe the full throughput characteristics between networked computers. It has been used to profile a wide range of high-speed network hardware, including ATM, FDDI, HIPPI, Myrinet, and Gigabit Ethernet [3, 4]. For the evaluation process, two machines are configured as a sender and a receiver. A series of messages, of exponentially increasing size, are sent between them, and the time taken to move the data from memory to memory, and confirm receipt, is measured. The one Byte message (plus header) is used as a latency measurement, and then the test message size is increased until it is in excess of 10 MBytes to give the asymptotic performance for the communication channel. The results are again plotted on a \log_{10} ordinate for clarity.

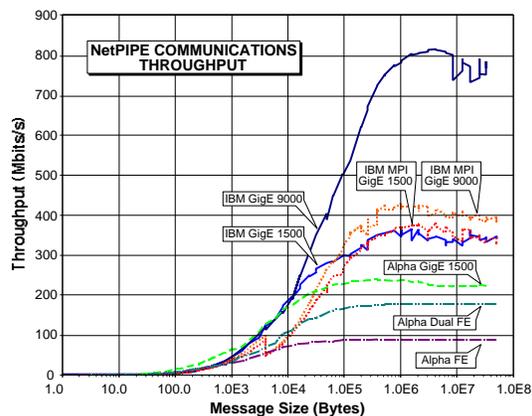


Fig. 2. Network Performance Comparison

Figure 2 shows a selection of NetPIPE performance curves for both Fast Ethernet (100 Mbits/second) and Gigabit Ethernet (1,000 Mbits/second). The Fast Ethernet (Alpha FE) curve was obtained from the Alpha 21164 platform running Linux 2.2.5, but is representative of all machines tested, with the interface realized ~90% utilization for large messages (>1 MByte). We also present results dual NIC data (Alpha Dual FE) that shows a 98% improvement when data is stripped across two interfaces per machine for larger messages (>100 kBytes). The Gigabit Ethernet results for the Alpha 21164 (Alpha GigE 1500) and IBM Power3 (IBM GigE 1500) show the relatively

poor utilization of the interface with only a 23% and 34% peak utilization respectively when using the standard 1,500 Byte Maximum Transmittable Unit (MTU) Ethernet data frames. The IBM hardware has the option of increasing the MTU to 9000 Bytes, and this modification has the dramatic result of improving peak performance to over 800 Mbits/second for large messages. These tests were performed with two machines in a “back-to-back” configuration for simplicity. For this work to be relevant to cluster computing, it is essential that the performance not be impacted to any great extent by the presence of a multi-port network switch, used at the heart of most low cost clusters. We have verified that the presence of a Jumbo Frame switch reduces performance by less than 5% across the board.

3. Ethernet Performance Issues

With the advent of Jumbo Frames, the throughput potential of Gigabit Ethernet can finally be realized for a single stream of data using standard TCP/IP protocols. The next issue that needs to be resolved is the latency-dominated performance observed for small messages. It is clear from Figure 2 that an increasingly severe penalty is paid as the message size decreases below 1 MByte. For messages of less than 100 Bytes, the throughput falls to below 1 MByte/second for all of the curves. At the moment, the measured first packet latency is greater than 100 microseconds for all of the curves. This is due to the complexity of the TCP/IP stack which was designed in the days of 10 Mbit Ethernet to run over heterogeneous, low-reliability, wide-area networks. This overhead may be dramatically reduced in a homogeneous, switched network, cluster environment that utilizes reliable, full duplex connections to every machine. Several projects have shown the potential to dramatically improve the throughput of commodity networks in a cluster setting including U-Net [5], VIA [6] and DART [7] while other efforts have been focused on maximizing the bandwidth utilization of high end communications channels. Examples of this latter class would be Active Messages [8-10] and Fast Messages [11]. In all cases the major benefit is gained from substantially reducing the overhead of multiple copies as data traverses the communications stack on the sending and receiving machines.

Another OS bypass based protocol, call Bobnet, has been developed in a collaboration

between the Ames Scalable Computing Laboratory and Sandia National Laboratory [12]. This protocol has the unique ability to support the full TCP/IP protocol suite, if required for code compatibility, yet exhibits zero copy sends and one copy receives, made possible by the Bobnet driver sitting directly above the hardware layer. This low-level protocol has been shown to improve the one-way latency from ~100 microseconds to under 30 microseconds for the G-NIC Gigabit Ethernet cards running on Intel based machines. It is intended that this work be extended to the Alpha and IBM platforms in the near future.

4. MPI overhead

In order for an applications programmer to utilize the communications throughput made available by Gigabit Ethernet, several fundamental parallel programming libraries must be optimized for this hardware. As an illustration, Figure 2 shows the NetPIPE performance curve obtained from the Jumbo Frames supporting hardware, but utilizing MPI instead of native TCP/IP to pass the series of messages between two IBM cluster nodes. It is clear that, although the 1500 MTU performance is good (IBM GigE 1500 vs. IBM MPI GigE 1500) the substantial communications improvement of the Jumbo Frames is lost when MPI is employed (IBM GigE 9000 vs. IBM MPI GigE 9000). To address this issue, work is underway to port a lightweight MPI implementation, called MP_Lite, to the Jumbo Frames Gigabit Ethernet environment. MP_Lite currently supports only the most fundamental message passing calls, but has been shown to give a substantial performance increase on many parallel architectures [13].

5. The Scalable Cluster Model

A cluster may be defined as “a collection of connected whole computers, used as a single, unified computer resource” [14]. It is also true that, since everyone agrees that there is only one RIGHT way to build a cluster, the next step should be to agree what that way is! For the purpose of this work we will step back from the fray and view the cluster as just another Non-Uniform Memory Access (NUMA) architecture in which the hierarchy of registers, multi-layer cache and main memory is augmented by non-local memory, and that this ordering is defined by access speed. The Beowulf phenomenon [15]

has made parallel computing accessible to the compute hungry masses, yet the Sword of Damocles effect of low budgets and low cost clusters is an important, if initially hidden issue. For any group constructing a 'large' (>16 nodes) cluster computer, the true cost of making it productive is often underestimated in the initial euphoria of being able to maximize the bang for the hardware buck. In order to lower the barrier to cluster computing, the SCL has been tasked with several projects to make the process as easy as possible for both ourselves and other groups. This approach goes beyond the usual documentation process of hardware lessons learned, and addresses the fundamental requirements for building a secure, manageable, scalable cluster. The first of these projects is the Cluster Cookbook [16] that gives a step-by-step guide to constructing a cluster. The second related project is the Scalable Cluster Model SCM which outlines the fundamental philosophy required to keep the management costs sub-scalar as the cluster grows.

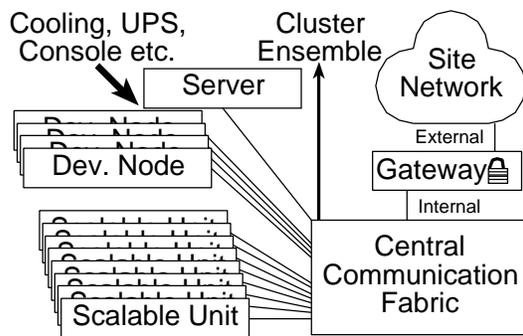


Fig. 3. Scalable Cluster Model Schematic

This model is depicted in Figure 3 and illustrates the seven aspects that define this approach:

Scalable Units: Each of these may be as inexpensive as a low cost single or dual processor PC, or they may be as powerful as the CPlant Scalable Units (SU) [17]. The key aspect for the SCM is the lack of personality on the compute nodes in the SUs, with the only unique information being the MAC address of the default network interface. An alternative would be a collection of machines connected by a Fast Ethernet switch having a Gigabit Ethernet up-link to a central, high capacity, Gigabit Ethernet switch.

Server node: This resource may be multiple machines if required and acts as the central information repository for the cluster, containing

the users permanent file store, the BOOTP server for address allocation, compilers, libraries, cluster monitoring resources and the batch queue master. It also contains the SU system images and is used to push out OS upgrades to the compute nodes. Due to the plethora of services required on this machine, it is the most demanding to configure, and must be protected from external unauthorized access by the gateway node.

Gateway: The gateway node is configured as a bastion host with two interfaces present. The first connects the cluster to the site network to provide a single secure (ssh, scp only) point of access, while the second interface allows access to the internal address space of the cluster.

Communications Fabric: The communications backbone of the cluster is typically a flat switching environment of commodity component Ethernet based hardware, or a tree/grid structure of proprietary interconnects. Typical layer 2 network switches can forward between 100,000 and 4 million 1,500 Byte packets/second.

Development Nodes: In a research environment we have found it advantageous to configure a sub-cluster of four nodes for the purpose of code development and debugging. This allows the compute nodes in the SUs to be restricted to production use only. It also limits the installation of parallel debugging tools, compilers and interactive shells to a few nodes.

Environmental administration: The installation issues of power, cooling, hardware monitoring, rack or shelf mounting etc. depend heavily on the size of the proposed cluster.

Typically a production cluster should be on an uninterruptible power supply, and in an environmentally controlled area. Rack mounting of hardware is an auspicious investment if available, and the ability to monitor the console of a given machine will facilitate hardware failure trouble shooting. If hardware error reports are mirrored to the serial port, then a simple daisy-chained loop can be constructed with null modem cables running from the primary serial port of one machine to the secondary port of its neighbor. Alternatively, terminal servers, or video/keyboard switch boxes can be used.

Systems administration: The minimal configuration expense and interchangeability of each scalable unit node is the main benefit of the SCM. There is also the lightweight monitoring tool, STATMON [18], which allows a simple web interface to query the system and resource status of each machine. We will be using this

sockets based utility to underpin the implementation of the Maui Scheduler [19] to the cluster environment.

6. Future Directions

Bringing the G-plant Alpha-based cluster, utilizing a Gigabit Ethernet communications fabric, into production is the next goal for this project. We will then be improving the Message Passing performance of the commodity network driver to allow Ethernet hardware to compete more effectively with the available higher cost, proprietary, parallel computing interconnect solutions. In addition, we need to address the issue of parallel batch queue submission and machine load balancing. The second cluster based on dual processor IBM Power3 machines will be upgraded to 25 nodes and a large Gigabit Ethernet switch installed which supports Jumbo Frames.

Following on the success of the dual Fast Ethernet NIC investigation, we intend to test dual Gigabit Ethernet NICs in several nodes. It is clear that the network drivers for these machines will have to be optimized for cluster applications, and to efficiently utilize the faster (50MHz and 64 MHz) and wider (64 bit) PCI bus. We will be working towards these goals with several industrial partners including IBM (with whom we have an SUR grant), Packet Engines and Alteon.

Finally, we have been working with the Hebrew University of Jerusalem on the MOSIX parallel operating system. This OS uses process migration to load balance within a cluster environment and has recently been ported to Linux [20]. We will be analyzing the extent to which this method of clustering benefits from a Gigabit Ethernet backbone.

7. Conclusions

The broad experience gained within the SCL has shown clearly that for most research institutions, the promise of cluster based parallel computing has to be application driven. The low cost of the compute nodes, memory, disk and communications fabric make it ideal for departmental level hardware budgets, yet the true expense still rests on the systems administrator and programmer to make the machine productive. Through our research, we have lowered the activation barrier for several groups, and allowed them to design and build clusters

that are optimized for the requirements of their most demanding codes.

7. References

- [1] J. L. Gustafson and Q. O. Snell, *Proceedings of the HICSS-28 Conference*, Wailela, Maui, Hawaii, January 3-6, 1995.
- [2] J.L. Gustafson and R. Todi, *Proceedings of the HICSS-31 Conference*, Kohala Coast, Hawaii, January 1998.
- [3] Q. O. Snell, A. Mikler, and J. L. Gustafson, "NetPIPE: A Network Protocol Independent Performance Evaluator" *IASTED International Conference on Intelligent Information Management and Systems*, June 1996.
- [4] S. Park, J. Lee, and S. Hariri, "Performance Evaluation of ATM and Gigabit Networks" *IEEE Information Technology Conference*, September 1998.
- [5] M. Welsh, A. Basu, and T. von Eicken, "Low-Latency Communication over Fast Ethernet." *EuroPar '96*, Lyon, France, August 1996.
- [6] *VI Architecture*. <http://www.viarch.org>
- [7] R. J. Walsh, "DART: Fast Application-level Networking via Data-copy Avoidance" *IEEE Network*, July/August 1997, 28-38.
- [8] A. Mainwaring, and D. Culler, "Active message applications programming interface and communication subsystem organization." Technical report, UCB. (<http://now.cs.berkeley.edu/Papers/Papers/am-spec.ps>) <http://now.cs.berkeley.edu/Papers/Papers/am-spec.ps>
- [9] S. Lumetta, A. Mainwaring, and D. Culler, "Multi-protocol active message s on a cluster of SMP's." *Proceedings of SC97*, San Jose, CA, November 1997.
- [10] R. Martin, "HPAM: an active message layer for a network of HP workstations." *Proceedings of Hot Interconnects*, 1994. ftp://ftp.cs.berkeley.edu/ucb/CASTLE/Active_Messages/hotpaper.ps
- [11] S. Pakin, V. Karamcheti, and A. Chen, "Fast Messages (FM): Efficient, Portable Communication for Workstation Clusters and Massively-Parallel Processors." *IEEE Concurrency*, 5, April-June 1997, pp. 60-73.
- [12] C. Csanady and P. Wyckoff, "Bobnet: High-Performance Message Passing for Commodity Networking Components" *Proceedings of the 2nd International Conference on Parallel and Distributed Computing and Networks*, Brisbane, Australia, December 1998
- [13] http://cmp.ameslab.gov/cmp/clusters/MP_Lite.html
- [14] Gregory F. Pfister *In Search of Clusters 2nd Edition*, Prentice Hall PTR, (1998).
- [15] <http://www.beowulf.org>
- [16] Cluster Cookbook, Guy Helmer, <http://www.scl.ameslab.gov/Projects/ClusterCookbook/>
- [17] <http://z.ca.sandia.gov/CPlant/index.html>
- [18] <http://www.scl.ameslab.gov/cgi-bin/ALICE/ALICE-Status>
- [19] <http://www.mhpcc.edu/maui/>
- [20] <http://www.cs.huji.ac.il/labs/distrib/index.html>