Performance of Scientific Applications on Linux Clusters

A close look at the performance of several scientific applications running on the Linux clusters using Myrinet network for high-speed communication

Jean-Marc Dressler
Swamy N. Kandadai

November, 2001

Abstract

We present the performance of several scientific applications running on a Linux cluster using a Myrinet network for high-speed communication between nodes. All the applications were run with a message passing interface (MPI). These applications include selections from linear algebra, fluid dynamics, and computational chemistry/life sciences applications. In addition, we also present the performance of basic MPI calls using the Pallas benchmark. This study suggests that Linux clusters with a Myrinet interface can be used for large scale scientific computations.

Introduction

Linux clusters have become the fastest growing choice for a high-performance parallel-computing platform(1). Advances in the microprocessor technologies, coupled with advances in high performance interconnect technologies, have facilitated the deployment of these clusters for high performance computing. Several Linux-based high-performance computing clusters have appeared in the Top 500 supercomputer sites. Performance of parallel applications depend on the cluster interconnect technologies. A low latency and high bandwidth interconnect significantly improves the parallel efficiency of technical applications.

We present the performance of some scientific applications running on a Linux cluster. The applications considered are simple communication performance benchmarks, NAS parallel benchmarks(2), a high performance Linpack benchmark, and three representative applications in the life science area. We have compared the results on the Linux cluster with those from other systems. Most of the measurements were made using Portland Group (STMicro) compilers; however, we also made few measurements using Intel compilers. Comparisons of the performance are presented in the following sections.

Description of the Linux System

Hardware

The nodes of the Linux cluster are a Netfinity® x330 with 1.0 GHz Pentium® III processors and each 330x is dyadic. The PCI bus on these Netfinity nodes are 64-bit wide and run at 33 MHz frequency. The theoretical peak performance of one CPU is 1000 MFLOPS. The Linux cluster is made up of 64 identical nodes.

In addition to the standard IP network connection, the Linux systems are connected through a high-performance Myrinet switch. Myrinet provides a low-latency, high-bandwidth solution to connect the system.

Software

We installed several components of the software for studying parallel performance. We installed the Linux kernel 2.2 as distributed by Red Hat version 6.2 on these nodes. The Red Hat distribution comes with standard GNU tools, such as the C-compiler and the Fortran compiler. For message-passing, we used the MPICH-GM provided by Myricom. This message-passing implementation uses the public domain MPICH, which is based on the library specification for message-passing proposed as a standard by a broadly based committee of vendors and users. MPICH-GM uses the GM library for interfacing with MPICH library to provide high bandwidth

and latency as compared to an IP protocol. The GM system includes a driver and Myrinet interface control program.

The standard compilers for the Linux system are the Portland Group compilers(3) and the Intel compilers(5). Those professional compilers were preferred to the widely used GNU tools due to better performance.

Benchmarks and Results

We selected three different types of benchmarks:

1. Low level message-passing benchmarks to provide the performance of the underlying interconnect technology
2. Simple kernel benchmarks to provide insights into the parallel efficiency
3. Application benchmarks in the life sciences area and in aerodynamic simulations

Low Level Message-Passing Benchmarks

Bandwidth and Latency Measurements

Performance of scientific and technical computing applications depend heavily on the bandwidth and latency for the underlying message passing library. A high bandwidth and a low latency typically enhance the application performance. Table 1 presents the latency and bandwidth for the Myrinet network using the MPICH over GM message passing library.

Table 1. Bandwidth and Latency

Bandwidth (MB/s)	140
Latency (usec)	13

Pallas

Pallas MPI benchmarks (PMB) suite (5) (http://www.pallas.com) is a set of benchmarks targeted at measuring the most important MPI functions used in scientific programming. Its objectives are:

1. Provide a concise set of benchmarks targeted at measuring the most important MPI functions.

2. Set forth a precise benchmark methodology.

The Pallas code is written in C with standard MPI calls. Two sets of benchmarks, PMB-MPI1 and PMB-MPI2, are available and we present next the results for PMB-MPI1 that uses the MPI-1 calls. PMB-MPI2 suite is meant for obtaining performance of one-sided communications and for measuring MPI I/O. Table 2 shows the performance of aggregate communication operations (barrier, broadcast, reduce, gather) using two processors per node for 4 bytes of data.

Table 2. Pallas

Bench type	No. of Processors	Results
Allreduce (4 bytes)	2	27 usec
	4	53 usec
	8	78 usec
	16	128 usec
Reduce (4 bytes)	2	14 usec
	4	26 usec
	8	39 usec
	16	72 usec
Allgather (4 bytes)	2	21 usec
	4	58 usec
	8	129 usec
	16	486 usec
Bcast (4 bytes)	2	12 usec
	4	24 usec
	8	24 usec
	16	39 usec
Barrier	2	18 usec
	4	40 usec
	8	63 usec
	16	109 usec

Stream

Stream (6) is a simple synthetic benchmark program that measures sustainable memory bandwidth (in MB/s) and the corresponding computational rate for simple vector kernels. The benchmark is specifically designed to work with data sets much longer than the available cache on any given system, so that the results are more indicative of very large vector-style applications. The stream benchmark measures the performance of four long vector operations. These operations are representative of the "building blocks" of long vector operations. The stream results for a single CPU are collected below. For comparison, we also present the results obtained on a Compaq Alpha server ES45-6/1000 processor (http://www.cs.virginia.edu/stream).

Table 3. Stream

	Kernel	Per Iteration	1.0 GHz Pentium III	Alpha ES-45-6/1000
		Bytes		
Copy	a(i)=b(i)	16	421.3 MB/s	1940 MB/s
Scale	a(i)=q*b(i)	16	420.4 MB/s	1940 MB/s
Add	a(i)=b(i)+c(i)	24	517.6 MB/s	1978.1 MB/s
Triad	a(i)=b(i)+q*c(i)	24	516.3 MB/s	1978.1 MB/s

Kernel Benchmarks

NPB

NAS (Numerical Aerodynamic Simulation) parallel benchmarks consist of eight programs. The first five (EP, FT, IS, MG and CG) are kernel benchmarks with simple data structure. The simulated application benchmarks which compute the numerical solution to the nonlinear partial differential equations are LU (LU decomposition), SP (Scalar Pentadiagonal), and BT (Block Tridiagonal). Three different classes (A,B, and C) of problems are defined depending on the size of the problem. We present below the results for Class B problem. All the timings reported in the table are elapsed time in seconds. The numbers under speedup are the speedups compared to a 4-way run time which is taken as 1.0. We ran 1 MPI-task on each processor and hence the number of tasks will be the same as the number of processors (CPUs). We used -O3 -tpp6 -Mcache_align options of PGI to compile these programs.

Table 4. NAS

Bench	No. of MPI Tasks	Time in sec.	Speedup
btB	**4**	2,863	1
	9	1,311	2.2
	16	791	3.6
	25	508	5.6
	64	197	14.5
	121	101	28.3
cgB	**4**	661	1
	8	218	3
	16	85	7.8
	32	38	17.4
	64	25	26.4
epB	**4**	123	1
	8	62	2
	16	31	4
	32	15	8.2
	64	7	17.6
ftB	**4**	422	1
	8	223	1.9
	16	125	3.4
	32	63	6.7
	64	33	12.8
	128	18	23.4
isB	**4**	14	1
	8	9	1.6
	16	6	2.3
	32	4	3.5
	64	3	4.7
luB	**4**	1,524	1
	8	785	1.9
	16	691	2.2
	32	221	6.9

	64	117	13
mgB	**4**	107	1
	8	38	2.8
	16	21	5
	32	11	9.7
	64	5	21.4
spB	**4**	2,158	1
	9	984	2.2
	16	552	3.9
	25	357	6
	64	122	17.7

Looking at the results, the scalability going from 4-way to 64-way is quite good and are in the same range as those for other parallel systems. One also notices superlinear performance (see cg.B) and is due to cache effects.

HPL

HPL (High Performance Linpack) is a software package that solves a (random) dense linear system of equations in double precision (64 bits) arithmetic on distributed-memory computers. The performance measured using this program on several computers forms the basis for the Top 500 super computer list. Using ATLAS (Automatically Tuned Linear Algebra Software) for the BLAS library we obtained 74.4 GFLOPS/s for the 64-way (128 processor) system. This translates to around 580 MFLOPS/s on each processor. For the same number of processors, the 375 MHz POWER3 system gave 136 GFLOPS/s (http://www.top500.org). Again, this shows that the Linux clusters shows very good performance for linear systems of equations.

Application Benchmarks

For the application benchmarks, we selected three popular applications in the computational chemistry/life sciences area. These are CHARMM and AMBER which are the two molecular modeling codes and Gausian, which is an ab initio quantum mechanics program.

CHARMM

CHARMM (Chemistry at Harvard Molecular Mechanics) is a program (7) (http://yuri.harvard.edu) for macro molecular simulations, including energy minimization, molecular dynamics and Monte Carlo simulations. We studied the performance of 2 cases to look into the parallel efficiency on the size of the system studied.

1. **DHFR:** This system is a protein (DHFR) surrounded by water molecules. The simulated system consists of 23558 atoms and is simulated for 1000 steps using clusters in cubes method. The results on the Linux cluster are presented in Table 5. In this table the number of MPI-tasks are the same as the number of processors. The timings reported in the table are in seconds. The numbers under the column speedup represent the parallel speedup compared to the performance on single processor which is taken as 1.0. The results indicate that the scalability is quite good upto 16 processors. Also, this study indicates that large simulations that are currently being studied are possible on Linux clusters. We compiled using -fast option of the PGI compiler.

Table 5. CHARMM DHFR

No. of MPI taks	Time in sec.	Speedup
1	3,034	1
2	1,924	1.6
4	1,062	2.9
8	639	4.7
16	390	7.7
32	289	10.5

2. **MbCO:** This system is carboxy myoglobin surrounded by water molecules. An all atom model is used for the simulation. The system consists of 14026 atoms and is simulated for 1000 steps. The results on the Linux cluster are presented in the following Table 6. The timings reported in the table are in seconds. The numbers under the column speedup are the parallel speed up compared to the performance on single processor which is taken as 1.0. As can be seen from the table, the scalability is quite good, 6.8 on an 8-way. These applications normally do not scale beyond 8-16 processors due to the ratio of computation to communication. Applications scale well if this ratio is high compared to the applications where this ratio is small. These results indicate that the performance on Linux clusters for these large simulations are comparable or better than on other parallel systems (http://www.cmm.ki.si/parallel/summary.html).

Table 6. CHARMM mbco

No. of MPI tasks	Time in sec.	Speedup
1	1,651	1
2	846	2
4	445	3.7
8	243	6.8
16	145	11.4

AMBER

The second application we studied is AMBER (Assisted Model Building with Energy Refinement). AMBER(8) (http://www.amber.ucsf.edu/amber/amber.html) is a molecular mechanical force field for the simulation of biomolecues and a package of molecular simulation programs. AMBER consists of six modules: sander, gibbs, roar, nmode, leap and interface.

In the following benchmarks we used the module *sander*. Sander stands for *Simulated annealing with NMR-derived energy restraints*. Sander is the main program for minimization and molecular dynamics simulation. We studied two systems and they are briefly discussed next.

DHFR

The system simulated is a protein surrounded by water molecules (23558 atoms) and is the same that was used earlier for studying the performance of CHARMM. The system was simulated for 1000 steps using Ewald summation technique using the PME option of AMBER. The performance on the Linux clusters are presented below. The parallel performance on AMBER behaves similar to CHARMm parallel performance. We used 1 MPI task on each processor and

the number of MPI-tasks also represent the number of processors. All times reported in this table are in seconds. The numbers under the column Speedup represent parallel speedup and is taken as 1.0 on a single processor. We used -fast -tpp6 -Mnoframe to compile the code.

Table 7. DHFR

No. of MPI tasks	Time in sec.	Speedup
0	2,984	1
2	1,824	1.6
4	1,012	2.9
8	629	4.7
16	422	7

DNA+water

The second system studied is the simulation of DNA fragment in water with counter ions. The system consists of 10232 atoms and was simulated for 12500 steps. The results from the Linux cluster are presented in Table 8. Again, we see fairly good scalability up to 8 processors beyond which the parallel efficiency drops.

Table 8. AMBER

No. of MPI taks	Time in sec.	Speed-up
1	17,096	1
4	6,143	2.8
8	3,495	4.9
16	2,302	7.4
32	1,857	9.2

Gaussian

The third application we studied is Gaussian. Gaussian is an ab initio quantum mechanics package for studying the structure and energies of molecules. The parallelism in Gaussian is implemented via LINDA message passing which uses IP interface. We performed the benchmark using the IP interface of Myrinet in this benchmark. The calculation for this study involved a full MP2 direct SCF (Self Consistent Field) with optimization of monomeric urea using 6-311++G(2df,2pd) basis set and calculating the density at the MP2 level. We present below the performance on 8-processor Linux clusters. All times reported in this table are elapsed time in seconds. These results are compared to a 8-way run on a 375 MHz POWER3 system.

Comparing these results, the Linux cluster performance is comparable to other parallel system platforms.

Table 9. Gaussian

No. of Processors	1.0 GHz Linux Cluster	375 MHz POWER3
8	3,961 sec	3986 sec

Performance Comparisons of Compilers

We studied a small set of preliminary investigation of the performance of compilers on the Linux systems. We used the Portland Group (STMicro) and the Intel compilers for this study. This study by no means is an exhaustive study and used the beta version of the Intel compiler. For this initial study, we looked at the performance of the NAS parallel benchmarks on a 16-processor Linux cluster. The results of running NAS parallel benchmarks are presented in the table below. All times reported are times in seconds. Both the compilers give the same performance for most of the applications. For cg.B and lu.B the performance of PGI is superior to the Intel compiler whereas for bt.B, the Intel compiler gives the best performance. All the NAS codes were compiled with -O3 option using Intel compiler.

Table 10. NAS Parallel Benchmarks

Benchmark	PGI compiler	Intel compiler
bt.B	791	728
cg.B	85	136
ft.B	126	140
lu.B	692	776
mg.B	21	25
sp.B	552	554

In the following table, we compare the performance of running AMBER for the DHFR input using PGI and Intel compilers. The performance using Intel compiler is slightly superior to the PGI compiler.

Table 11. AMBER Benchmarks

No. of MPI tasks	PGI compiler	Intel compiler
1	2,985	2,720
2	1,824	1,658
4	1,012	953
8	629	552
16	422	349

Conclusions

We described above the studies we have undertaken to implement the Linux clusters as viable platform for studying large scale massively parallel applications. We described work we performed on the NAS parallel benchmark as well as on other life sciences and computational chemistry applications. Most of the studies were done on large scale systems currently being investigated. The performance of these applications on the Linux clusters are similar to the traditional systems and provide low cost solution for massively parallel computations.

Acknowledgment

We are thankful to our colleagues for building the Linux clusters and for installing the necessary software and to the management for their support in this study.

References

1. Building Linux Clusters, D. Spector, O'Reilly, Cambridge, MA (2000)
2. D. H. Bailey and J. T. Burton, The NAS Kernel Benchmark Program, NASA TM-86711, Aug 1985.
3. ST Microelectronics (formerly Portland Group Inc (PGI), http://www.pgroup.com)
4. Intel compilers (http://www.intel.com)
5. Pallas Benchmarks (http://www.pallas.com)
6. Stream (http://www.cs.virginia.edu/stream)
7. B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan and M. Karplus, J. Comp. Phys. 4,187 (1983)
8. D. A. Case, D. A. Pearlman, J. W. Caldwell, T. E. Cheatham III, W. S. Ross, C. L.Simmerling, T. A. Darden, K. M. Merz, R. V. Stanton, A. L. Cheng, J. J. Vincent, M.Crowley, D.M.Ferguson, R. J. Radmer, G. L. Seibel, U. C. Singh, P. K. Weiner and P. A. Kollman, AMBER 5, University of California, San Francisco.

Special Notices

IBM products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information provided in this document and information contained on IBM's past and present Year 2000 Internet Web site pages regarding products and services offered by IBM and its subsidiaries are "Year 2000 Readiness Disclosures" under the Year 2000 Information and Readiness Disclosure Act of 1998, a U.S. statute enacted on October 19, 1998. IBM's Year 2000 Internet Web site pages have been and will continue to be our primary mechanism for communicating year 2000 information. Please see the "legal" icon on IBM's Year 2000 Web site (http://www.ibm.com/year2000) for further information regarding this statute and its applicability to IBM.

Any performance data contained in this document was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Trademarks

The following terms are trademarks of International Busines Machines Corporation in the United States, other countries, or both:

IBM, AIX, e(logo), Netfinity

IBM Trademarks information can be found at: http://www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SPEC, SPECjbb, SPECint, SPECfp, SPECweb, and SPECsfs are trademarks of the Standard Performance Evaluation Corporation and information can be found at: http://www.spec.org.

TPC, TPC-R, TPC-H, TPC-C, and TPC-W are trademarks of the Transaction Processing Performance Council. Information can be found at: http://www.tpc.org.

Other trademarks are the property of their respective owners.

Notes on Benchmarks and Values

The benchmarks and values shown here were derived using particular, well configured, development-level computer systems. Unless otherwise indicated for a system, the values were derived using 32-bit applications and external cache, if external cache is supported on the system. All benchmark values are provided "AS IS" and no warranties or guarantees are expressed or implied by IBM. Actual system performance may vary and is dependent upon many factors including system hardware configuration and software design and configuration. Buyers should consult other sources of information to evaluate the performance of systems they are considering buying and should consider conducting application oriented testing. For additional information about the benchmarks, values and systems tested, contact your local IBM office or IBM authorized reseller or access the following on the Web:

TPC	http://www.tpc.org
GPC	http://www.spec.org/gpc
SPEC	http://www.spec.org
Pro/E	http://www.proe.com
Linpack	http://www.netlib.no/netlib/benchmark/performance.ps
Notesbench Mail	http://www.notesbench.org
VolanoMark	http://www.volano.com

Fluent	http://www.fluent.com
Gaussian	http://www.gaussian.com

Unless otherwise indicated for a system, the performance benchmarks were conducted using AIX® V4.2.1 or 4.3, IBM C Set++ for AIX/6000 V4.1.0.1, and AIX XL FORTRAN V5.1.0.0 with optimization where the compilers were used in the benchmark tests. The preprocessors used in the benchmark tests include KAP 3.2 for FORTRAN and KAP/C 1.4.2 from Kuck & Associates and VAST-2 v4.01X8 from Pacific-Sierra Research. The preprocessors were purchased separately from these vendors.

The following SPEC and Linpack benchmarks reflect the performance of the microprocessor, memory architecture, and compiler of the tested system:

- SPECint95 - SPEC component-level benchmark that measures integer performance. Result is the geometric mean of eight tests that comprise the CINT95 benchmark suite. All of these are written in the C language. SPECint_base95 is the result of the same tests as CINT95 with a maximum of four compiler flags that must be used in all eight tests.

- SPECint_rate95 - Geometric average of the eight SPEC rates from the SPEC integer tests (CINT95). SPECint_base_rate95 is the result of the same tests as CINT95 with a maximum of four compiler flags that must be used in all eight tests.

- SPECfp95 - SPEC component-level benchmark that measures floating-point performance. Result is the geometric mean of ten tests, all written in FORTRAN, that are included in the CFP95 benchmark suite. SPECfp_base95 is the result of the same tests as CFP95 with a maximum of four compiler flags that must be used in all ten tests.

- SPECfp_rate95 - Geometric average of the ten SPEC rates from SPEC floating-point tests (CFP95). SPECfp_base_rate95 is the result of the same tests as CFP95 with a maximum of four compiler flags that must be used in all ten tests.

- SPECint2000 - New SPEC component-level benchmark that measures integer performance. Result is the geometric mean of twelve tests that comprise the CINT2000 benchmark suite. All of these are written in C language except for one which is in C++. SPECint_base2000 is the result of the same tests as CINT2000 with a maximum of four compiler options that must be used in all twelve tests.

- SPECint_rate2000 - Geometric average of the twelve SPEC rates from the SPEC integer tests (CINT2000). SPECint_base_rate2000 is the result of the same tests as CINT2000 with a maximum of four compiler options that must be used in all twelve tests.

- SPECfp2000 - New SPEC component-level benchmark that measures floating-point performance. Result is the geometric mean of fourteen tests, all written in FORTRAN and C languages, that are included in the CFP2000 benchmark suite. SPECfp_base2000 is the result of the same tests as CFP2000 with a maximum of four compiler options that must be used in all fourteen tests.

- SPECfp_rate2000 - Geometric average of the fourteen SPEC rates from SPEC floating-point tests (CFP2000). SPEC_base_rate2000 is the result of the same tests as CFP2000 with a maximum of four compiler options that must be used in all fourteen tests.

- SPECweb96 - Maximum number of Hypertext Transfer Protocol (HTTP) operations per second achieved on the SPECweb96 benchmark without significant degradation of response time. The Web server software is ZEUS v.1.1 from Zeus Technology Ltd.

- SPECweb99 - Number of conforming, simultaneous connections the Web server can support using a predefined workload. The SPECweb99 test harnass emulates clients sending the HTTP requests in the workload over slow Internet connections to the Web server. The Web server softwre is Zeus from Zeus Technology Ltd.

- LINPACK DP (Double Precision) - n=100 is the array size. The results are measured in megaflops (MFLOPS).

- LINPACK SP (Single Precision) - n=100 is the array size. The results are measured in MFLOPS.

- LINPACK TPP (Toward Peak Performance) - n=1,000 is the array size. The results are measured in MFLOPS.

- LINPACK HPC (Highly Parallel Computing) - solve largest system of linear equations possible. The results are measured in GFLOPS.

VolanoMark is a 100% Pure Java™ server benchmark characterized by long-lasting network connections and high thread counts. In this context, long-lasting means the connections last several minutes or longer, rather than just a few seconds. The VolanoMark benchmark creates client connections in groups of 20 and measures how long it takes for the clients to take turns broadcasting their messages to the group. At the end of the test, it reports a score as the average number of messages transferred by the server per second. VolanoMark 2.1.2 local performance test measures throughput in messages per second. The final score is the average of the best two out of three results.

The following SPEC benchmark reflects the performance of the microprocessor, memory subsystem, disk subsystem, network subsystem:

- SPECsfs97_R1 - the SPECsfs97_R1 (or SPEC SFS 3.0) benchmark consists of two separate workloads, one for NFS V2 and one for NFS V3, which report two distinct metrics, SPECsfs97_R1.v2 and SPECsfs97_R1.v3, respectively. The metrics consist of a

throughput component and an overall response time measure. The throughput (measured in operations per second) is the primary component used when comparing SFS performance between systems. The overall response time (average response time per operation) is a measure of how quickly the server responds to NFS operation requests over the range of tested throughput loads.

The following Transaction Processing Performance Council (TPC) benchmarks reflect the performance of the microprocessor, memory subsystem, disk subsystem, and some portions of the network:

- tpmC - TPC Benchmark C throughput measured as the average number of transactions processed per minute during a valid TPC-C configuration run of at least twenty minutes.

- $/tpmC - TPC Benchmark C price/performance ratio reflects the estimated five year total cost of ownership for system hardware, software, and maintenance and is determined by dividing such estimated total cost by the tpmC for the system.

- QppH is the power metric of TPC-H and is based on a geometric mean of the 17 TPC-H queries, the insert test, and the delete test. It measures the ability of the system to give a single user the best possible response time by harnessing all available resources. QppH is scaled based on database size from 30GB to 1TB.

- QthH is the throughput metric of TPC-H and is a classical throughput measurement characterizing the ability of the system to support a multiuser workload in a balanced way. A number of query users is chosen, each of which must execute the full set of 17 queries in a different order. In the background, there is an update stream running a series of insert/delete operations. QthH is scaled based on the database size from 30GB to 1TB.

- $/QphH is the price/performance metric for the TPC-H benchmark where QphD is the geometric mean of QppH and QthH. The price is the five-year cost of ownership for the tested configuration and includes maintenance and software support.

The following graphics benchmarks reflect the performance of the microprocessor, memory subsystem, and graphics adapter:

- SPECxpc results - Xmark93 is the weighted geometric mean of 447 tests executed in the x11perf suite and is an indicator of 2D graphics performance in an X environment. Larger values indicate better performance.

- SPECplb results (graPHIGS) - PLBwire93 and PLBsurf93 are geometric means of literal and optimized Picture Level Benchmark (PLB) tests for 3D wireframe and 3D surface tests, respectively. The benchmark and tests were developed by the Graphics Performance Characterization (GPC) Committee. The results shown used the graPHIGS API. Larger values indicate better performance.

- SPECopc results - CDRS-03, CDRS-04, DX-03, DX-04, DX-05, DRV-04, DRV-05, DRV-06, Light-01, Light-02, Light-02, AWadvs-01, AWadvs-02, Awadvs-03, and ProCDRS-02 are weighted geometric means of individual viewset metrics. The viewsets were developed by ISVs (independent software vendors) with the assistance of OPC (OpenGL Performance Characterization) member companies. Larger values indicate better performance.

The following graphics benchmarks reflect the performance of the microprocessor, memory subsystem, graphics adapter, and disk subsystem:

Bench95 and Bench97 Pro/E results - Bench95 and Bench97 Pro/E benchmarks have been developed by Texas Instruments to measure UNIX® and Windows NT ™workstations in a comparable real-world environment. Results shown are in minutes. Lower numbers indicate better performance.

The NotesBench Mail workload simulates users reading and sending mail. A simulated user will execute a prescribed set of functions 4 times per hour and will generate mail traffic about every 90 minutes. Performance metrics are:

- NotesMark - transactions/minute (TPM).
- NotesBench users - number of client (user) sessions being simulated by the NotesBench workload.
- $/NotesMark - ratio of total system cost divided by the NotesMark (TPM) achieved on the Mail workload.
- $/User - ratio of total system cost divided by the number of client sessions successfully simulated for the Mail NotesBench workload measured.

Total system cost is the price of the server under test to the customer, including hardware, operating system, and Domino Server licenses.