High Performance Computing Challenge on small Linux Clusters

Isaac Hailperin
FU Berlin
isaac.h at web.de

September 7, 2005

Abstract

Testing a computer system with a benchmark suite provides data which are suitable for analyzing different aspects of parallel high performance computing. Two clusters with different dual CPU nodes and interconnects are compared in various disciplines. The linpack benchmark is optimized on both systems. Special attention is given to symmetric multiprocessing efficiency.

1 Introduction

Benchmarking a computer system provides data which is more appropriate to compare and assess the performance of a given system than bare hardware details such as CPU frequency and size of the main memory. Application behavior is simulated using algorithms that resemble specific demands. The two main objectives are first to test the operability of the two local installations and second to compare the clusters via measurments of main properties.

1.1 Operability

Operability is tested because high performance clusters consist of various components of specialised cutting edge hard and software. There are various installations around which differ in all kinds of aspects. So for this kind of complex systems it is not a priori clear that the machine works at its full capacity. Precious resources might be wasted because of undetected deficiencies that could be fixed.

1.2 Comparison

Comparing the clusters is important in order to get an idea of how different CPU and network architectures affect the various aspects of performance. Comparison is based on measurments of memory bandwidth, network latency and bandwidth and CPU behavior. First the CPUs are examined with regard to their scalability. For simplicity, I used one, two, four and eight CPUs. Second the CPU's symmetric multiprocessing (SMP) efficiency is analyzed by subsequently using one and two processes per node. SMP enables all CPUs on one node to access a global memory address space and thus faciliates the distribution of multiple processes.

1.3 High Performance Computing Challenge (HPCC)

The High Performance Computing Challenge [1, 2] is a collection of different benchmarks that in part have already been in use as a seperate benchmark. The most prominent one is the High Performance Linpack which is exclusivly used to determine the

ranking of the top500 list of supercomputers [5]. Because of its long history of use, the linpack provides a continous base for evaluating computer clusters. Its shortcome though is that it only measures the ability to solve a linear system of equations. This is overcome by the HPCC which benchmarks a broader range of features.

2 Equipment

2.1 Hardware

Both tested clusters have networks for high performance computing with high bandwidth and low latency. The details are described in table 1.

	"Opteron" cluster	"Xeon" cluster
CPU manufacturer	AMD	Intel
CPU model	Opteron 250	Xeon P4
CPU frequency	2.4 GHz	1.7 GHz
CPU cache	1 MB L2	256 kB
CPU register width	64	32
memory size	4 GB	1 GB
memory model	PC2700 ECC DDR SDRAM	RDRAM
host adapter	Mellanox InfiniBand HA 4X	Myrinet 2000 M3F-PCI64B-2
switch	Mellanox InfiniScale III 2400, 24 ports	M3-E32 5 slot chassis, 2xM3-SW16 line cards
machine type, vendor	Sun Fire V20Z	Megware
nodes	8	16

Table 1: Hardware detail

2.2 Software

For compiling HPCC version 1.0.0 on the Opteron cluster I used gcc 3.2.3 as shipped with Red Hat Linux together with the atlas 3.7.10 library.

For compiling HPCC version 1.0.0 on the Xeon cluster I used gcc 2.95.3 as shipped with SuSE Linux 7.2 together with the PGI 3.3 BLAS library.

3 The Benchmarks

3.1 Network

The basic network parameters are bandwidth and latency. These are measured for two different settings.

3.1.1 Random Ring

For the random ring benchmark, all MPI processes are ordered in a virtual ring. Reported is the geometric mean of ten different randomly chosen orderings. All processes send data to both of their neighbors. Bandwidth per process is defined as the total amount of data that is sent divided by the number of processes and the maximal time needed in all processes. For details see [1].

3.1.2 Ping Pong

For the ping pong benchmark, two exclusive processes exchange data. Several pairs are tested and the maximal latency and minimal bandwidth is reported.

While random ring communication is more likely to occur in a real application, ping pong communication should achieve peak results.

3.2 CPU

3.2.1 High Performance Linpack (HPL)

Of course one way of measuring CPU performace is the HPL. It can achieve nearly peak performance and because of its long history of use it provides consistent data for a wide period of high performance computer construction. Accumulated and per process results are reported.

3.2.2 Fast Fourier Transform (FFT)

A different computational demand is represented by FFT. Star FFT measures double precision complex computation on a single CPU, while MPI FFT uses all available CPUs in parallel. For the HPL as well for FFT floating point operations per second (flop/s) are reported and double precision numbers are used.

3.3 Memory

Memory bandwidth is measured with the STREAM benchmark. It consists of four vector kernels:

$$\begin{aligned} \text{Copy}: &\quad c \leftarrow a \\ \text{Scale}: &\quad b \leftarrow \alpha c \\ \text{Add}: &\quad c \leftarrow a + b \\ \text{Triad}: &\quad a \leftarrow b + \alpha c \end{aligned}$$

For StarSTREAM these are executed on all processes simultaneously without communication. Afterwards the average of all processes is reported.

3.4 Balance

Balance is defined as the random ring bandwidth divided by the HPL per process. It expresses the ratio of communication to computation in byte/flop.

4 Optimizing HPL

The HPCC is driven by an inputfile called `hpccinf.txt` which contains settings for the HPL. Here one can tune the performance. Among the most influential parameters are the size N of the coefficient matrix, the blocking size NB and the process grid P x Q.

N should be large enough to fill the memory, but not too large to avoid swapping. NB is used for the distribution of data. Small values will achieve good load balance but increase communication. P and Q should be chosen according to $P * Q = n$ where n is the total number of processes. Also the ration of $P : Q$ should be something like $1 : k, \quad k \in [1, 2, 3]$. In order to determine the best values for N and NB I tried out several settings. For the Opteron cluster, $N = 12750$ and $NB = 196$ yielded

best results for dual CPU use. On the Xeon cluster $N = 11000$ and $NB = 96$ were used. These values are not optimized for overall cluster peak performance, but showed best results for different machine configurations. This was done to be able to compare SMP efficiency.

Some of the data from the inputfile is also used for other benchmarks such as FFT and STREAM to determine the size of available memory.

5 Results

Table 2 shows results of HPCC on the Opteron cluster. The first number in a box is the mean of ten successive runs. The second number is the standard deviation. Table 3 contains data for the Xeon cluster. Here mean and standard deviation of twelve successive runs are presented.

5.1 Network, InfiniBand versus Myrinet

Comparing the bandwidth of InfiniBand and Myrinet (see figure 1), one can see very similar behavior for SMP mode as well as for the single process mode. The difference is that InfiniBand's bandwidth is larger by a factor of 4. Also for latency both networks show similar behavior, see figure 2. InfiniBand is faster by a factor of 5 and seems to be a little more stable for 1 process per node.

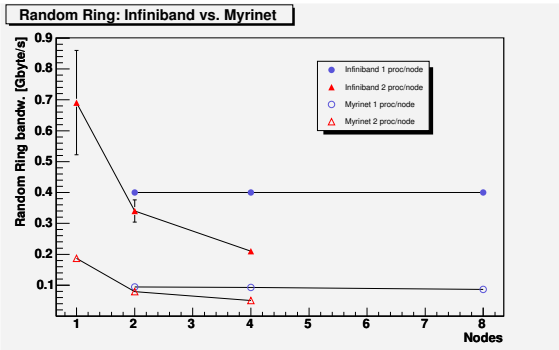

Figure 1: Bandwidth

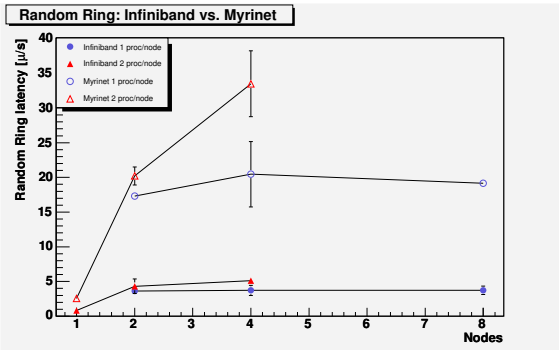

Figure 2: Latency

Results of High Performace Computing Challenge on the Opteron cluster

Nodes * cpu's	Random Ring Bandw. Gbytes/s	Ping Pong Bandw. Gbytes/s	Random Ring Lat. μs	Ping Pong Lat. μs	HPL accu-mu-lated Gflop/s	HPL per process Gflop/s	Balance comm./ comp. byte/ kflop	StarFFT Gflop/s	MPIFFT Gflop/s	Star STREAM Copy Gbytes/s	Star STREAM Scale Gbytes/s	Star STREAM Add Gbytes/s	Star STREAM Triad Gbytes/s
1 x 1	-	-	-	-	7.288 0.027	7.288 0.027	-	0.404 0.010	0.329 0.009	2.053 0.093	2.036 0.084	2.615 0.179	2.551 0.175
1 x 2	0.691 0.169	0.750 0.087	0.800 0.000	0.645 0.000	7.448 0.068	3.724 0.034	185.533 45.171	0.506 0.008	0.615 0.003	2.159 0.019	2.095 0.025	2.502 0.000	2.205 0.026
2 x 1	0.400 0.000	0.800 0.000	3.641 0.234	10.706 1.014	13.976 0.059	6.988 0.030	57.248 0.243	0.506 0.005	0.580 0.010	2.065 0.097	2.043 0.091	2.400 0.135	2.167 0.084
2 x 2	0.340 0.036	0.775 0.020	4.316 1.052	3.653 0.174	12.325 0.182	3.081 0.046	110.276 10.842	0.531 0.004	0.963 0.071	2.374 0.030	2.273 0.056	2.506 0.014	2.502 0.000
4 x 1	0.400 0.000	0.797 0.009	3.719 0.725	4.897 0.000	25.506 0.146	6.376 0.036	62.736 0.359	0.534 0.002	1.039 0.010	2.188 0.075	2.118 0.078	2.474 0.140	2.472 0.134
4 x 2	0.210 0.012	0.782 0.009	5.123 0.391	4.931 0.126	23.092 0.670	2.887 0.084	72.633 3.919	0.560 0.003	1.754 0.070	2.472 0.071	2.303 0.062	2.698 0.029	2.556 0.043
8 x 1	0.400 0.000	0.793 0.011	3.720 0.609	4.915 0.042	47.146 0.222	5.893 0.028	67.879 0.319	0.562 0.004	1.924 0.091	2.359 0.064	2.363 0.093	2.561 0.057	2.503 0.070
8 x 2	0.200 0.000	0.784 0.005	5.068 0.457	5.479 0.081	34.588 1.002	2.162 0.063	92.603 2.687	0.548 0.002	3.107 0.171	2.677 0.028	2.696 0.024	2.699 0.018	2.692 0.021

Table 2

Results of High Performace Computing Challenge on the Xeon cluster

Nodes * cpu's	Random Ring Bandw. Gbytes/s	Ping Pong Bandw. Gbytes/s	Random Ring Lat. μs	Ping Pong Lat. μs	HPL accu-mu-lated Gflop/s	HPL per process Gflop/s	Balance comm./ comp. byte/ kflop	StarFFT Gflop/s	MPIFFT Gflop/s	Star STREAM Copy Gbytes/s	Star STREAM Scale Gbytes/s	Star STREAM Add Gbytes/s	Star STREAM Triad Gbytes/s
1 x 1	-	-	-	-	0.052 0.048	0.052 0.048	-	0.149 0.053	0.134 0.041	1.312 0.432	1.312 0.430	1.464 0.485	1.458 0.484
1 x 2	0.187 0.001	0.384 0.002	2.572 0.016	1.509 0.021	0.494 0.027	0.247 0.013	759.546 42.703	0.185 0.011	0.196 0.004	0.687 0.008	0.683 0.011	0.776 0.009	0.790 0.011
2 x 1	0.094 0.000	0.127 0.008	17.299 0.074	12.299 0.049	0.479 0.117	0.239 0.058	428.126 151.391	0.147 0.045	0.180 0.013	1.220 0.173	1.245 0.172	1.382 0.211	1.329 0.223
2 x 2	0.079 0.008	0.194 0.005	20.222 1.301	8.718 0.018	1.065 0.008	0.266 0.002	297.876 28.289	0.211 0.006	0.284 0.003	0.702 0.004	0.701 0.002	0.794 0.005	0.801 0.004
4 x 1	0.093 0.001	0.114 0.026	20.461 4.704	18.005 12.421	1.042 0.036	0.261 0.009	356.030 13.778	0.215 0.032	0.249 0.062	1.231 0.170	1.241 0.162	1.417 0.180	1.396 0.177
4 x 2	0.050 0.009	0.160 0.014	33.496 4.740	10.545 0.608	1.979 0.301	0.261 0.003	189.943 32.017	0.204 0.005	0.420 0.047	0.703 0.004	0.699 0.003	0.794 0.006	0.815 0.012
8 x 1	0.086 0.012	0.113 0.015	19.153 0.373	21.943 14.097	2.091 0.012	0.261 0.001	329.743 46.821	0.222 0.012	0.539 0.020	1.166 0.229	1.178 0.236	1.326 0.262	1.292 0.251
8 x 2	0.043 0.001	0.123 0.000	37.561 0.076	11.551 0.004	3.974 0.011	0.248 0.001	171.403 2.264	0.187 0.002	0.805 0.006	0.690 0.002	0.689 0.004	0.784 0.005	0.796 0.005
16 x 1	0.091 0.001	0.123 0.001	19.427 0.046	13.991 4.400	4.012 0.026	0.251 0.002	362.214 6.662	0.238 0.020	0.857 0.116	1.329 0.114	1.349 0.059	1.555 0.020	1.546 0.019
16 x 2	0.039 0.001	0.132 0.000	38.359 0.082	11.988 0.002	7.637 0.025	0.239 0.001	164.757 3.542	0.195 0.001	1.539 0.011	0.678 0.002	0.677 0.001	0.775 0.004	0.788 0.003

Table 3

5.2 Other networks

In comparison to other InfiniBand installations [4], the local cluster is quite fast. For random ring bandwidth I measured 0.4 Gbytes/s for 8 single CPUs. One machine(Dell PowerEdge 1850 cluster Intel Xeon EM64T) was reported with 0.144 Gbyte/s random ring bandwidth for single CPUs. Other interconnects are much faster though. The top value was reported for a NEC SX-8/6 SX-8 with a NEC Internode Crossbar Switch with 13.547 Gbyte/s random ring bandwidth.

5.3 CPU

For FFT, not much can be said about SMP efficiency since the errors are too large(see figure 3). The Opteron cluster is about four times faster then the Xeon cluster. This agrees with the higher CPU frequency and doubled register width.

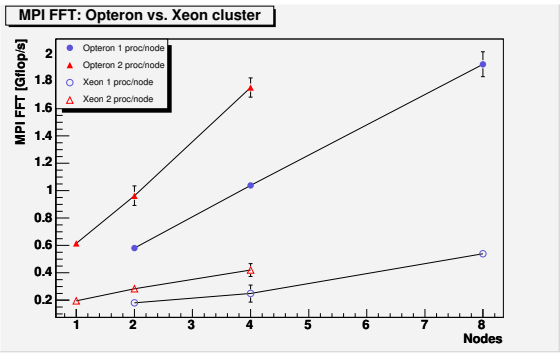

Figure 3: Fast Fourier Transform

The accumulated linpack shows an almost linear increase on both clusters (figure 4). What is strange is that on the Opteron cluster SMP is about 50 % slower than single process per node. One would not expect computation to be slower if an extra CPU is added.

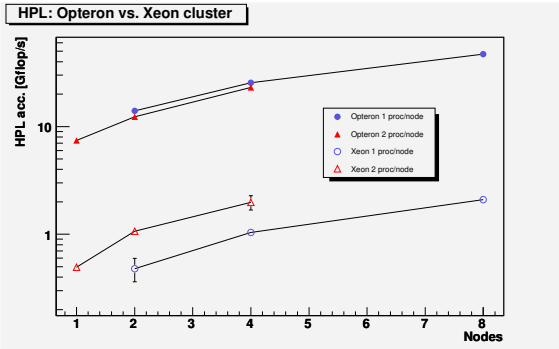

Figure 4: HPL

Some reasons have already been ruled out:

N too large. This would affect the memory usage. If two processes share the same memory, there is less memory per process. To test this I have performed the linpack with small N = 1000. This did not speed up SMP compared to single CPU usage.

SMP communication via network. If two processes on one node would communicate with each other via the network rather then via memory, bandwidth will decrease and latency will increase. The ping pong latency data from table 1 for one and two nodes show a significant difference. Also the following extract from the Pallas benchmark suggests that SMP communication is working:

```
1   Pallas Benchmark, 2 processes
2   Benchmarking PingPong
3
4   1. one node
5   1.1 SMP activated:
6   #---------------------------------------------------
7          #bytes #repetitions      t[usec]   Mbytes/sec
8            1024         1000         5.00       195.37
9            2048         1000         5.00       390.75
10           4096         1000         5.00       781.49
11          65536          640        70.29       889.16
12         131072          320       140.58       889.17
13         262144          160       343.65       727.49
14         524288           80       874.73       571.60
15        1048576           40      1624.51       615.57
16        2097152           20      2999.08       666.87
17
18
19  1.2 SMP deactivated:
20  #---------------------------------------------------
21         #bytes #repetitions      t[usec]   Mbytes/sec
22           1024         1000        10.00        97.67
23           2048         1000        15.00       130.22
24           4096         1000        20.00       195.33
25          65536          640       179.68       347.85
26         131072          320       328.11       380.97
27         262144          160       656.21       380.97
28         524288           80      1312.43       380.97
29        1048576           40      2624.84       380.98
30        2097152           20      5499.67       363.66
31
32
33  2. two nodes:
34  #---------------------------------------------------
35         #bytes #repetitions      t[usec]   Mbytes/sec
36           1024         1000        10.00        97.67
37           2048         1000        15.00       130.22
38           4096         1000        15.00       260.44
39          65536          640       109.36       571.48
40         131072          320       187.48       666.73
41         262144          160       374.96       666.73
42         524288           80       749.93       666.73
43        1048576           40      1499.86       666.73
44        2097152           20      2999.72       666.73
```

I also tried different versions of the atlas BLAS library and the goto BLAS library to rule out a bug in the SMP implementation of the BLAS library. It is remarkable that the Opteron cluster is about 20 times faster in the HPL then the Xeon cluster.

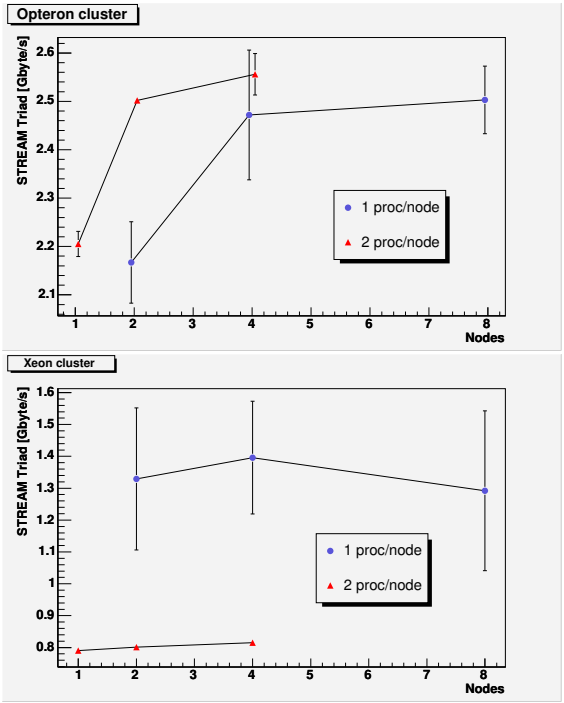

Figure 5: STREAM Triad

5.4 Memory bandwidth

Since all STREAM benchmarks show similar results, I only compare the triad here. As can be seen in figure 5, memory access is nearly independent of the number of processes and nodes on the Opteron cluster, whereas the Xeon cluster exhibits a 40 % drop in bandwidth for dual CPU use. This is due to the particular architecture.

6 Conclusions

For MPI FFT and STREAM, the Opteron cluster shows superior SMP efficiency over the Xeon cluster. In general the overall performance of the former is much enhanced compared to the latter.

The behavior of the HPL on the Opteron cluster is probably due to a bug in the InfiniBand network protocol stack. Currently in use is InfiniBand Gold 1.6.0 [6] which is not the newest version. The HPL results are a verification of known problems with InfiniBand's SMP communication.

The software for the host adapter driver and parallel programming support (MPI [7]) comes in short periods of releases and has shown incorrect behavior in earlier versions. However, the effort of the InfiniBand companies and the open software community gives hope that the problem will be solved soon.

Comparison with data from [4], especially the NEC SX-8/6 SX-8, suggests that once the problem has been sorted out, Opteron's HPL performance might nearly be doubled.

The Xeon cluster shows consistent behavior. Comparing the single CPU performace (figure 4) Opterons supremacy is striking.

7 Acknowledgment

I would like to thank Peter Wegner and Götz Waschk for their guidance, patience and support, and Karlheinz Hiller for organizing Summer school. I am indebted to my grandparents for their proof reading and suggestions. Thanks to all the summer students and the computing department for making my stay at DESY a pleasant one.

References

[1] Luszczek, P., Dongarra, J., Koester, D., Rabenseifner, R., Lucas, B., Kepner, J., McCalpin, J., Bailey, D., Takahashi, D. *Introduction to the HPC Challenge Benchmark Suite*, March, 2005. (`http://icl.cs.utk.edu/hpcc/pubs/index.html`)

[2] Dongarra, J., Luszczek, P. *Introduction to the HPCChallenge Benchmark Suite*, ICL Technical Report, ICL-UT-05-01, (Also appears as CS Dept. Tech Report UT-CS-05-544), 2005. (`http://icl.cs.utk.edu/hpcc/pubs/index.html`)

[3] Rolf Rabenseifner, *Balance of HPC Systems Based on HPCC Benchmark Results*, Proceedings of the Cray Users Group Conference 2005, CUG 2005, May 16-19, Albuquerque, NM, USA, 2005 (`http://www.hlrs.de/people/rabenseifner/publ/publications.html`)

[4] HPCC result page (`http://icl.cs.utk.edu/hpcc/hpcc_results.cgi`)

[5] Top 500 Supercomputer Site (`http://www.top500.org/`)

[6] Mellanox InfiniBand Gold Collection
(`https://docs.mellanox.com/dm/archive/ibgold_1_7_0/ReadMe.html`)

[7] The Message Passing Interface (MPI) standard (`http://www-unix.mcs.anl.gov/mpi/`)