Dual Processor Nodes for HPC Clusters: Additional Data for SMP Nodes

Douglas Eadline, Ph.D.

3rd April 2003

deadline@basement-supercomputing.com

Abstract

Additional results for a PIII and Athlon motherboards are presented. The results indicate a wide range of performance that can be expected from various motherboards and chipsets. Timing data is also supplied. The range of performance indicates that it is possible for a cluster of slower processors, but better SMP performance, may out perform a cluster built from faster processors with lesser SMP performance.

Background

A previous article [1] presented memory contention data for an Intel SE7500WV2 motherboard. In this report, data for a Tyan 2762 (dual Athlon with AMD-760 chipset) and a Supermicro P3TDL3 (Dual PIII with Serverworks LE chipset) are presented.

To measure the effect of two programs running on a dual node we can measure the time it takes for a single copy of a program and then measure the time it takes to run two copies of the same program. If the program scales perfectly then the times should be identical. If the program does not scale perfectly the time should be between one and two times the single program time. A speedup measure can be calculated as follows:

$$speedup = \frac{(CTIME1+CTIME2)*STIME}{(CTIME1*CTIME2)}$$

Where *CTIME1* is the time for running program copy1 concurrently with copy2 of the program, *CTIME2* is the time for running program copy2 concurrently with copy1 one of the program, and *STIME* is the time to run one copy of the program.

From this equations we see that if *CTIME1*=*CTIME2*=*STIME* (perfect speed up) we get a speedup of 2. If (*CTIME1*+*CTIME2*)=2**STIME* (no speedup) we get an answer of 1 which indicates no speed-up.

Test Method and Hardware

In order to test the efficiency of dual nodes we can employ tests from the Beowulf Performance Suite (BPS) (http://www.plogic.com/bps). We used two motherboards:

- Tyan 2762 with dual Athlon MP 1600+ (1.4Ghz) processors, 2 GByte, RAM, running kernel 2.4.20
- Supermicro P3TDL3 with dual Tualatin 1.26 Ghz processors, 2 GByte of RAM, running kernel 2.4.18

Both the Intel V6 (icc and ifc) compilers and the GNU V2.96 (gcc and g77) were used to compile the programs. Information about the test programs and the compile arguments can be found in the NAS suite in the BPS package.

A simple script has been written to run the tests and calculate the speedup for the NAS test suite. The NAS test suite is compiled for one node using test size A. The script was run three times for and the results were averaged. The compiler can be changed by editing the "run_suite" line in the script.

Results

Supermicro P3TDL3

The Supermicro P3TDL3 motherboard was tested first. The speedup for each test is listed below. An average speedup for the whole suite is also given.

Test	GNU2.96	Intel 6.0
CG	.98	.90
BT	1.09	1.10
EP	1.95	1.92
FT	1.32	1.36
LU	1.14	1.04
IS	1.08	1.07
SP	1.02	98
MG	1.03	.97
Average	**1.20**	**1.17**

Speed-up for concurrent NAS programs on a dual SMP node

To provide some background on run times, a single set of time data is presented below for the GNU compilers. Data are in seconds.

Test	STIME	CTIME1	CTIME2
CG	14.64	30.07	30.04
BT	1455.11	2590.23	2588.82
EP	108.48	111.3	111.01
FT	39.22	60.42	60.42
LU	804.20	1394.72	1395.91
IS	5.08	9.48	9.48
SP	949.40	1845.14	1824.05
MG	33.48	63.81	63.80

Example times (in seconds) for NAS programs on a dual SMP node (GNU Compiler)

A single set of time data is presented below for the Intel compilers. Data are in seconds.

Test	STIME	CTIME1	CTIME2
CG	14.11	30.53	30.64
BT	1459.72	2595.35	2594.49
EP	116.91	120.70	120.72
FT	39.45	58.10	58.10
LU	744.18	1401.9	1401.14
IS	5.03	9.51	9.50
SP	905.09	1844.07	1842.95
MG	30.76	64.01	64.00

Example times (in seconds) for NAS programs on a dual SMP node (Intel Compiler)

Tyan 2762

The Tyan 2762 motherboard was also tested. The speedup for each test is listed below. An average speedup for the whole suite is also given.

Test	GNU2.96	Intel 6.0
CG	1.83	1.85
BT	1.58	1.47
EP	2.00	2.00
FT	1.73	1.64
LU	1.60	1.48
IS	1.92	1.88
SP	1.39	1.26
MG	1.59	1.53
Average	**1.71**	**1.64**

To provide some background on run times, a single set of time data is presented below for the GNU compilers. Data are in seconds.

Test	**STIME**	**CTIME1**	**CTIME2**
CG	15.93	17.34	17.26
BT	982.84	1233.99	1235.87
EP	96.54	96.58	96.16
FT	35.27	40.80	40.65
LU	600.66	749.06	742.69
IS	7.07	7.10	7.35
SP	585.39	853.84	853.19
MG	24.43	29.99	30.43

Example times (in seconds) for NAS programs on a dual SMP node (GNU Compiler)

A single set of time data is presented below for the Intel compilers. Data are in seconds.

Test	**STIME**	**CTIME1**	**CTIME2**
CG	15.59	16.83	16.74
BT	864.14	1176.74	1175.82
EP	102.24	102.15	102.09
FT	32.64	39.55	39.03
LU	522.65	729.69	731.30
IS	6.59	6.79	6.94
SP	527.51	820.38	828.80
MG	22.63	29.42	29.87

Example times (in seconds) for NAS programs on a dual SMP node (Intel Compiler)

Intel SE7500WV2

For comparison, the results from the first paper for an Intel SE7500WV2 (2.2Ghz Xeon processors) are provided below. The times listed below were not given in the paper [1].

Test	GNU2.96	Intel 6.0
CG	1.1	1.05
BT	1.45	1.34
EP	1.99	2
FT	1.55	1.55
LU	1.4	1.17
IS	1.78	1.7
SP	1.56	1.2
MG	1.34	1.04
Average	**1.52**	**1.38**

Speed-up for concurrent NAS programs on a dual SMP node

A single set of time data is presented below for the GNU compilers. Data are in seconds.

Test	STIME	CTIME1	CTIME2
CG	4.77	8.45	8.43
BT	675.80	937.11	937.62
EP	109.58	109.47	109.62
FT	24.10	31.19	31.10
LU	387.11	529.24	529.14
IS	3.93	4.32	4.33
SP	595.53	750.67	750.67
MG	14.78	22.30	22.28

Example times (in seconds) for NAS programs on a dual SMP node (GNU Compiler)

A single set of time data is presented below for the Intel compilers. Data are in seconds.

Test	STIME	CTIME1	CTIME2
CG	4.62	8.57	8.59
BT	595.38	896.91	897.44
EP	156.82	156.71	156.83
FT	21.10	31.17	31.28
LU	307.61	512.08	513.79
IS	3.40	4.13	4.05
SP	420.87	679.19	676.19
MG	10.52	20.46	20.51

Example times (in seconds) for NAS programs on a dual SMP node (Intel Compiler)

Discussion

The results indicate a wide range of performance. The AMD 760 chipset shows the best performance, while the older PIII Serverworks LE chipset shows poor performance. Keep in mind the tests are not intended to show which platform is faster and that the Tyan/Athlon and Intel/Xeon do have faster memory. In addtion, the compiler optimizations are identical for all platforms. The tests were designed to show the effect of running two identical programs on the same node. This scenario is common in some cluster environments.

Although the PIII Serverworks performed poorly, this does not discount the use of this motherboard as a dual node. Keep in mind, that these tests are sequential applications running on a single node. In a parallel environment, communication overhead may alleviate the memory contention by spending time with communication issues. For instance, if a Fast Ethernet cluster were to use this motherboard, the SMP memory contention issue may not be as pronounced. If a higher performing low overhead networking is used, then the memory contention may become a bigger performance issue. The Athlon MP 760 chipset shows a clear advantage in memory contention issues. This type of cluster node should show good performance with all types of networking. Indeed, it is possible, given the range of performance, that a dual CPU node cluster of slower processors, but better SMP performance, may out perform a cluster built from faster processors with lesser SMP performance.

Similar to the first paper [1], using the Intel compiler lowers the efficiency. This effect is presumably due to the better efficiency of the code produced by the compiler and thus more dependance on the memory subsystem.

In these tests, we have not considered communication issues or the mix of different programs on the same node. These issues will be addressed in upcoming reports. **Your results may vary depending on they type of application you run on your cluster.**

Test Script

```
#!/bin/bash
PROGS="cg.A.1 bt.A.1 ep.A.1 ft.A.1 lu.A.1 is.A.1 sp.A.1 mg.A.1"
echo "SMP Memory Test" |tee smp-mem.out
echo "`date`" |tee -a smp-mem.out
# generate single cpu codes change -c for different compiler
./run_suite -n 1 -t A -m dummy -c gnu -o
for TEST in $PROGS
do
bin/$TEST>& temp.mem0
bin/$TEST>& temp.mem1 &
bin/$TEST>& temp.mem2
wait
SINGLE=`grep Time temp.mem0 |gawk '{print $5}'`
DOUBLE1=`grep Time temp.mem1 |gawk '{print $5}'`
DOUBLE2=`grep Time temp.mem2 |gawk '{print $5}'`
echo STIME=$SINGLE
echo CTIME1=$DOUBLE1
```

```
echo CTIME2=$DOUBLE1
SPEEDUP=`echo "2 k $DOUBLE1 $DOUBLE2 + $SINGLE * $DOUBLE1 $DOUBLE2 * / p" | dc`
echo "SMP Program Speed-up for $TEST is $SPEEDUP" |tee -a smp-mem.out
done
/bin/rm temp.mem*
echo "`date`" |tee -a smp-mem.out
```

References

[1] Dual Processor Nodes for HPC Clusters: Memory Contention Issues, Douglas Eadline, http://www.hpc-design.com/reports/smp-mem1/index.html,

Acknowledgments

I wish to acknowledge all the authors of the tests suites used in this package.

Copyright