Summer 2004 Physics Internship

Jeremy Pace

This summer I was responsible for the benchmarking of the Beowulf cluster using High-Performance LinPack (HPL), STREAM and NetPerf testing. During this time, many obstacles had to be overcome. The primary obstacle was my lack of knowledge of the Linux operating system. The debate over channel bonding was another. Finally, the lack of comprehensive documentation of benchmarks such as HPL made the implementation process longer and more complicated.

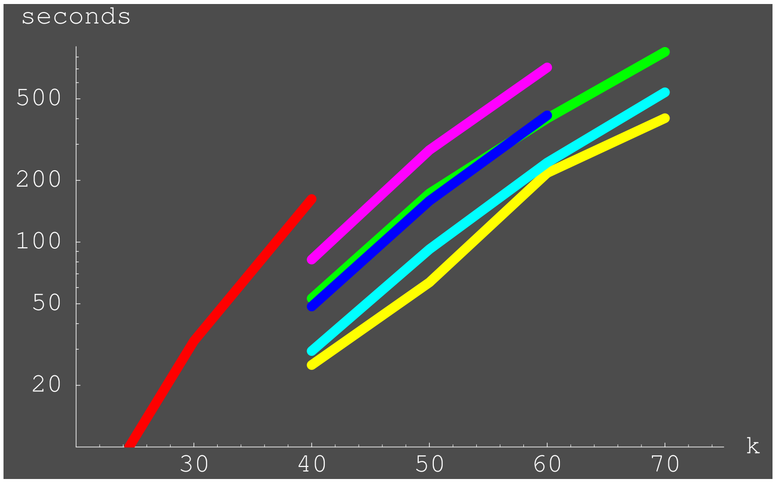
The LinPack test was the first, and most time-consuming test administered. I used three different libraries (math routines): ACML 2.0, ATLAS 3.5.6 and GOTO. One problem that I had was the use of the machinefile tag and the combing of the two separate networks. The nodes in the cluster are connected by two independent networks. These networks are not aware of each other, the only way for the two of them to communicate is through the node hardware.

Our best results came from using GOTO under some finely tuned conditions (each processor was given its own IP address and memory, both networks were utilized, and 80% of the system memory was consumed by the problem size) and impressive 29.09 GFlops was recorded (roughly 66% percent of the 45 GFlop theoretical maximum).

The STREAM test was nothing more than a quick measurement of the transfer rate of the memory bandwidth of each individual processor. The results were somewhat impressive, at roughly 75% of bandwidth. Due to the architecture of the Opteron, one expects this performance to remain as high when both processors are running.

The next step was the NetPerf test. I used this test to measure the TCP send/receive speeds for the two networks. I did four minitests; first to first, first to second, second to second and second to first. The results were somewhat disappointing for this test. The NetPerf test does not differentiate between two IPs on the same machine, so the first network transfer rates are very impressive, but the second network results were tragic and/or disappointing. The primary network results were 95%, unfortunately, the second network results were 5%. Again, this test doesn't really give a good representation of the whole picture, but it does show that the current network settings yielded impressive results.

Using the techniques and information I acquired, I then benchmarked the code for calculating meson states that Dr. van de Sande and Justin Lambright wrote during this summer. The results indicated that for this particular instance of code, it is better to use only one network (with all 16 processors). This came as something of a surprise, but reinforces the notion that the benchmark tests performed are not conclusive, but rather just an idea of the performance capabilities, a guide, not a rule.



Color	Processors	Networks
Red	1	1
Pink	2	1
Blue	4	1
Periwinkle	8	1
Yellow	16	1
Green	16	2