Evaluation of GPU Accelerated NAMD

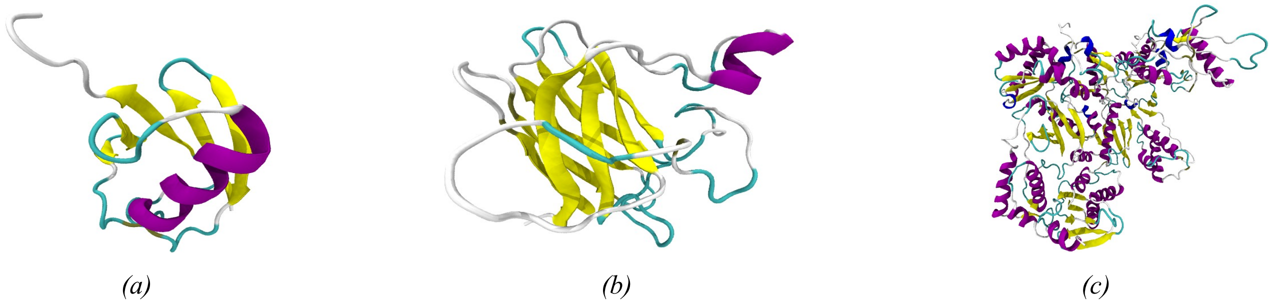

Figure 1: The three benchmark simulations used in this report to compare the cluster, CPU and GPU versions of NAMD. (a) was the small job, an Ubiquitin molecule, (b) is the medium job, SOD1 (superoxide dimutase) and (c) is the large job HIV reverse transcriptase. All molecules were in a solvated system.

Student: Katrina Barnett

Association: Victorian Partnership for Advanced Computing

Supervisor: Mike Kuiper (Research Scientist)

Discipline: Life Sciences

Date: December 2008 – February 2009

Note: Please read the 2008/09 Summer Internship Project summary document "Evaluating the use of GPGPU's in Life Science and Super Computing applications" first, for background information on this case study

Objective

The aim of this component of the 2008/09 VPAC Summer Internship Program[1] was to see if the GPU enabled version of NAMD enabled simulations to be completed quicker than the CPU only version and also test if the accuracy (consistency) of the data was suitable for true scientific simulation purposes.

Introduction

NAMD[2], NAnoscale Molecular Dynamics, a molecular dynamics simulation package which can be used on regular desktop computers and with parallelism on cluster computers. NAMD has recently began producing a version which makes use of GPU technology[3]. While this version is still in development (an official release is due in a few months), we have been completing some comparisons between the traditional cluster computing method and the new GPU enhanced version.

Implementation

Four versions of NAMD were used for the benchmarking, for the cluster computer ('Tango'[4] here at VPAC[5]) where the implementation of NAMD was compiled specifically for the cluster computer. Version 2.6 of NAMD was used, compiled with the gcc compiler and using OpenMPI[6] for parallelism. For the GPU machines, using only CPU technology, a binary version (for 64-bit Linux) and a compiled version of NAMD were used. Both were version 2.6 and were obtained from the NAMD website[7]. the compiled version was compiled using Intel proprietary compilers as suggested[8] by the NAMD developers. the GPU accelerated source code was retrieved from the CVS[9] repository for the currently developing source code on the 17th of December 2008, again, it was retrieved and again, compiled as suggested[10] and run on the NVIDIA GPU machines as the accelerated source code uses the CUDA[11] GPU programming language .

Benchmarking

As said above there were four implementations of NAMD used for comparison. For each of these versions three different benchmarking 'jobs' were run, each of a different size. the small job was of the Ubiquitin molecule and contained 14,463 atoms with box dimensions of 54 angstroms on each side, the medium job, SOD1 (superoxide dimutase) had 34,863 atoms with box dimensions of 72 angstroms and the large job which was the HIV reverse transcriptase, 147,161 atoms and box dimensions of 128x96x128 angstroms. All of the benchmark jobs were of solvated systems. As was discovered as the testing began, to be able to run these jobs using the GPU several restrictions needed to be made in the configuration of NAMD features. these restrictions were included for the tests on all of the implementations. Each job was run on 1, 2 and 4 CPUs for each machine and an extra 8 and 16 CPUs for the cluster computer (t o observe the scale up as the number of processors increased). A confirmation run was done for each test to ensure cohesion of results.

Limitations

It was found that in order to run the jobs on the GPU the *Constant Pressure Control* needed to be 'turned off' (by commenting out the appropriate lines in the

Machine	Cluster Computer ('Tango')					Test Boxes No GPU (Binary)			Test Boxes No GPU (Compiled)			Test Boxes With GPU		
NAMD						NVIDIA (B)			NVIDIA (C)			NVIDIA (GPU)		
#cpus	1	2	4	8	16	1	2	4	1	2	4	1	2	4
Small	18829	10388	5250	2881	1500	15499	8888	5177	16554	8980	4728	1600	2083	2536
	18819	10278	5564	2825	1481	15602	8936	5151	16442	8931	4756	1598	2071	2561
average	***18824***	***10333***	***5407***	***2853***	***1491***	***15551***	***8912***	***5164***	***16498***	***8956***	***4742***	***1599***	***2077***	***2549***
ns/day	0.23	0.42	0.80	1.51	2.90	0.28	0.48	0.84	0.26	0.48	0.91	2.70	2.08	1.70
Medium	47400	26039	13998	7151	4402	38200	20971	11587	39955	21516	11823	4317	3645	3620
	47203	25574	13745	8668	4117	37893	20877	11339	39852	21347	11664	4251	3628	3553
average	***47302***	***25807***	***13872***	***7910***	***4260***	***38047***	***20924***	***11463***	***39904***	***21432***	***11744***	***4284***	***3637***	***3587***
ns/day	0.18	0.33	0.62	1.09	2.03	0.23	0.41	0.75	0.22	0.40	0.74	2.02	2.38	2.41
Large	39916	23315	11032	5977	3268	31545	16936	8949	32947	17097	9451	4172	3074	2586
	39864	20748	10969	6217	3026	31646	16851	9090	33114	17220	9370	4159	3059	2507
average	***39890***	***22032***	***11001***	***6097***	***3147***	***31596***	***16894***	***9020***	***33031***	***17159***	***9411***	***4166***	***3067***	***2547***
ns/day	0.04	0.08	0.16	0.28	0.55	0.05	0.10	0.19	0.05	0.10	0.18	0.41	0.56	0.68

Table 1: Comparison of different implementations of NAMD. Showing total wall time (total execution and computation time) in seconds (black text) and nanoseconds of simulation done per day of wall time (ns/day) in pink text. Depicts small, medium and large simulation runs, two runs of each, for each of the configurations of traditional cluster computer (with pre-optimised installation of NAMD) and the NVIDIA graphics card machine with and without the GPU support. Each test was run on a varied number of CPUs.

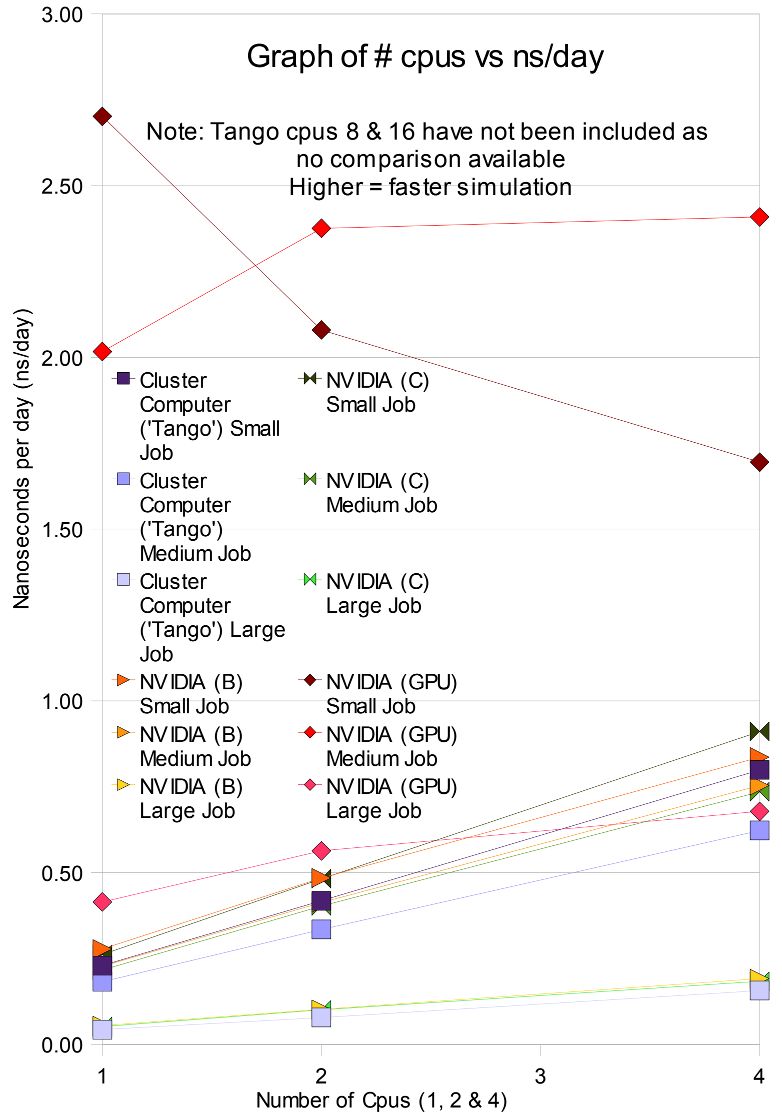

Figure 2: Graph of the benchmarking results (in ns/day) to compare the GPU version of NAMD to the CPU and cluster computing methods. Note that a larger number of ns/day means a faster simulation time.

configuration script) as this caused instability and the simulation would not run to completion. Also, through initial testing it was found that the options *exclude scaled1-4* and *1-4scaling* needed to be removed and the option *exclude 1-3* needed to be included. these options largely aided in maintenance of simulation stability while the option *twoAwayX yes* was included (and the companion *twoAwayY* was excluded) to help with memory management issues.

Larger Simulations

We also attempted to test some very large simulations, the initial test was a fibril simulation with 1,690,271 atoms. While this was not tested extensively, it was found to cause a memory allocation error, this is most likely to be because there was simply not enough memory on the GPU to hold the simulation. Several different configuration options were tested with a similar result.

Results

The results indicate the length of time in seconds for the benchmark to finish. From this the measure of ns/day, the number of nanoseconds of simulation completed per day of simulation running, could be calculated (by using the number of steps completed and the step size, 2 femtoseconds). these results are shown in Table 1 above.

A comparison graph of the results found is shown above (Figure 2). When looking at the graph it is easy to see how much faster the GPU implementation is than the current CPU version.

Another interesting point about the GPU implementation is that the smaller job (shown by the darker colours) actually takes a longer time to simulate than the medium job when the number of CPUs is increased. This perhaps suggests that the time to set up and perform the

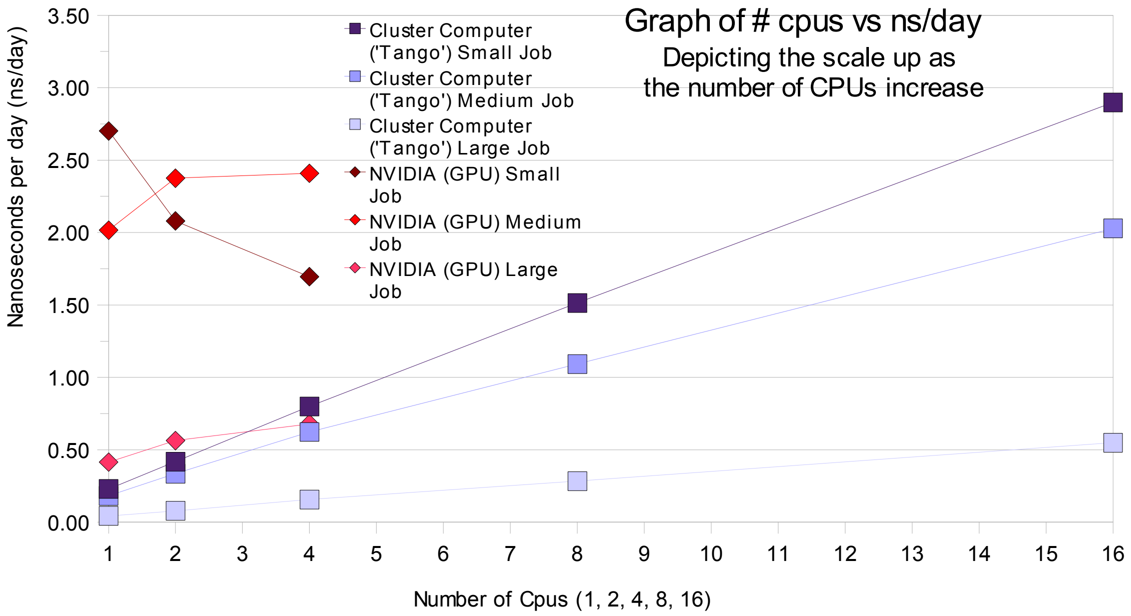

Figure 3: Graph comparing the GPU accelerated NAMD to the cluster computer version. Note how each GPU benchmark for one CPU result is approximately the same as the result for 16 CPUs for the cluster computer. This shows a x16 speed up for using a GPU.

memory transfers between multiple CPUs and the GPU is significant in comparison to the total simulation time.

Figure 3 above shows an extended graph with the results from the cluster computer for eight and 16 CPUs shown along with the GPU enhanced results. From this it can be seen that even using a single CPU and a GPU equates to an average (of the three simulation sizes) of 15.3 CPUs on the cluster computer. This is a substantial speed up for relatively little effort in compiling and running.

As has been said above, the disadvantage of the GPU accelerated NAMD is that there are currently many limitations. While we have attempted to replicate any limitation in the comparison machines, the extra simplifications may give the results found here a more biased slant.

Statistics

In order to validate the results found the output information was examined to determine if using the GPU caused any artefacts or inaccuracies to appear. the metrics that were used to assess this were the temperature, the total energy and the pressure of the systems. the means and standard deviation was calculated for each benchmark run and a graph produced to show any sort of drift or presence of outliers.

It was found that no discernible trend was seen (as in, a drift to higher or lower values), the values stayed within fairly defined boundaries.

Some longer simulations were run (in the order of 1,000,000 steps) to see if there was a drift or variation that could not be seen over a shorter number of steps. No change was observed. A comparison of the CPU version (compiled) and the GPU version of the medium job can be seen in Figure 4 below. Both versions were run using four CPUs to aid in speed.

While the output points fall in a definite region and seem to form a straight line, it can be seen that the range of values is different for all three output values. It was expected that if one phenomena is seen in one of the output statistics it would be replicated in the other two as the three measures are related in the physical system. Both the CPU and GPU versions of this benchmark used the exact same configuration script with the same options enabled and disabled (see the Limitations section above) and started from the same restart file. there is no obvious explanation as to why there is a difference between the two versions, in every respect they are identical except on the hardware and version of NAMD they were run on. This same 'artefact' was seen for each of the different benchmarks.

the explanation that is offered here is simply due to the immaturity of the version of NAMD used for the GPU tests. As it is still in development it is logical to expect that some bugs have not been fixed yet, this is likely to be one of them.

What can be gained from this is the fact that there are no major outliers or drift of values which could be caused by a memory management error when using the GPU. Such an error was quite a real possibility and would render the results obtained from such simulations inaccurate for scientific purposes. This continuity aspect was further studied by forcing the simulation to be deterministic by selecting a seed for the pseudo random number generator (setting this in the NAMD configuration file using the *seed* command) and imposing, as suggested by the NAMD 2.6 User's Guide[12], another restriction of only running the job on one CPU for the CPU version and one CPU plus GPU for the GPU version. This was done and the result for both CPU and GPU version was reproducible. This can be seen by the overlapping points in Figure 5 below (note that this simulation was only run over 100,000 steps, also from a restart file). This proves that (for this simulation and configuration at least) there were no memory errors caused by the GPU version (or the CPU version for that matter) which means the result is reproducible, an essential feature for scientific computing.

Future Possibilities

As has been said several times above, the version of NAMD used to test the GPU implementation is still in development, a beta release has not been completed or released yet. Some of the limitations may be fixed (or better documented) in the future, this would mean more general simulation problems could be completed in the same way that is currently done with the released version (on CPUs only). Features such as the constant pressure control that are currently not supported would be very useful in the future.

Another issue in need for improvement is the lack of support for large, many atom simulations such as the fibril simulation mentioned above. While not all possibilities and configurations were tested, it does seem that jobs that require a large amount of memory cannot run on the GPU. Improvements would need to be made to perhaps break up the job further or control how much GPU memory is requested or allocated at once. This could also be addressed by the addition of larger amounts of memory on future GPU hardware or perhaps using multiple GPU configurations (currently not supported).

Another possibility for the future could be that even if some of these limitations are not resolved in the first few releases this GPU version of NAMD could be used to complete long running simulations

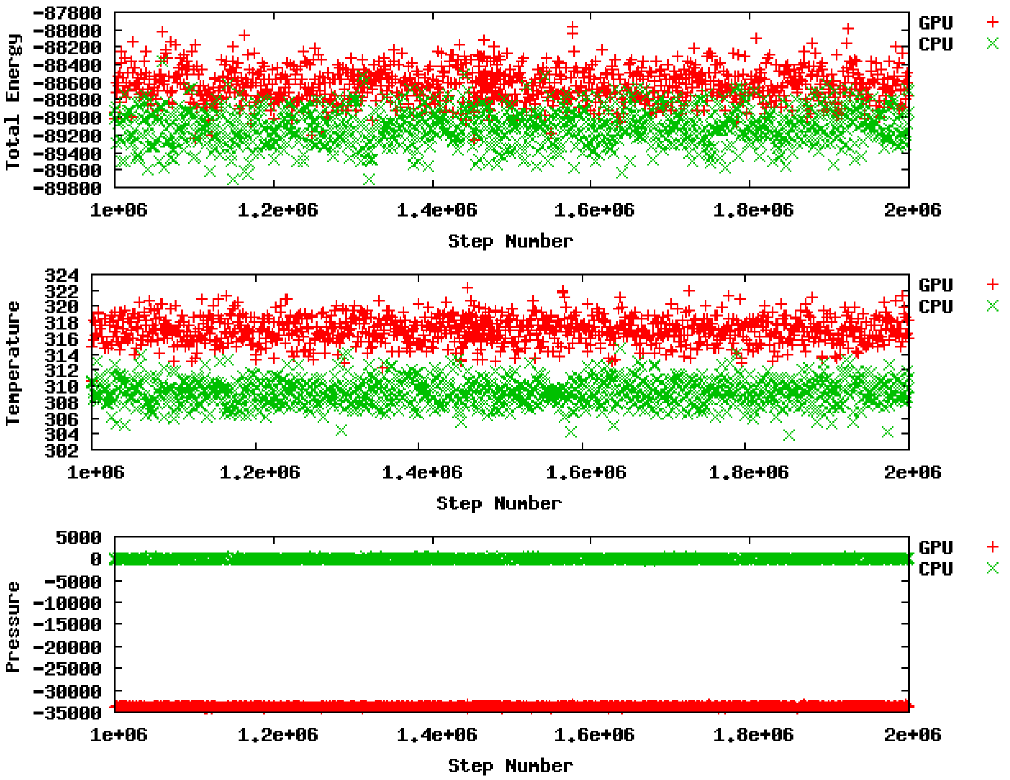

Figure 4: Comparison of the output total energy, temperature and pressure from the medium NAMD simulation with the CPU (compiled) and GPU version of NAMD. Note how both the CPU and the GPU versions remain at steady levels with no trend or major outliers.

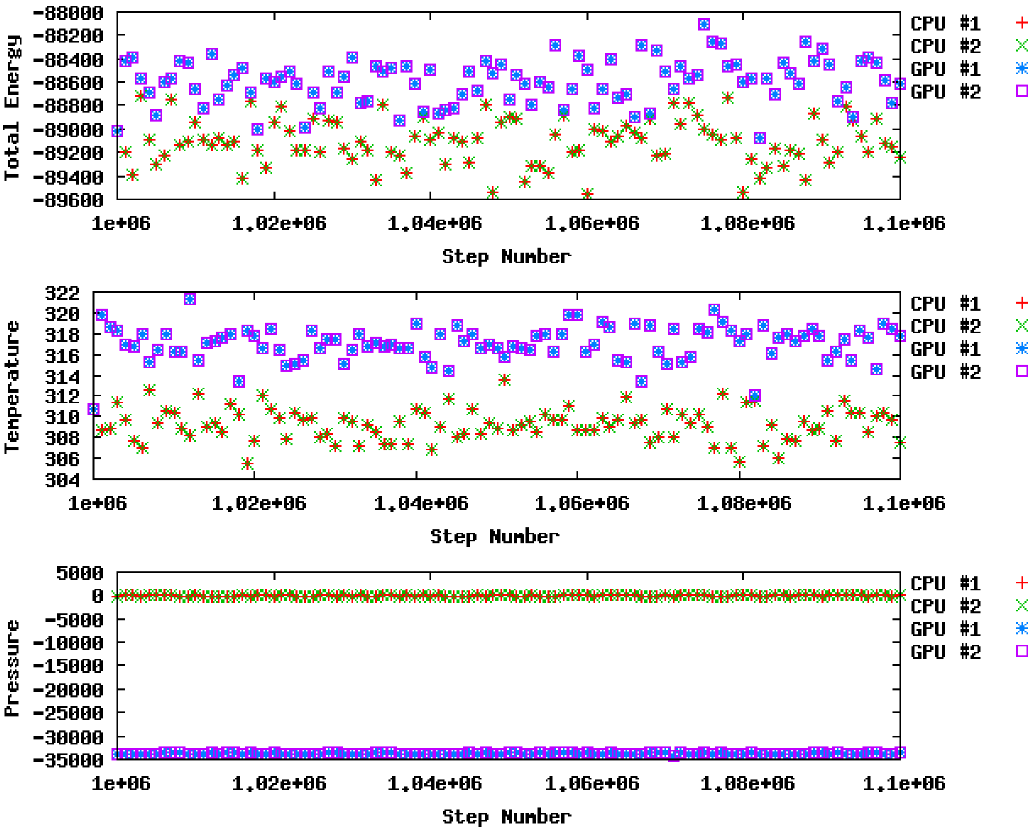

Figure 5: Graphs of the output total energy, temperature and pressure showing deterministic values over two runs (when the same seed value was used). Note that both the GPU and CPU give reproducible results.

quickly which would then allow the researcher to view the output and decide which parts (if any at all) are worth simulating traditionally at a slower, if more accurate, pace.

The problem with these limitations is that once they are resolved it may alter the amount of improvement in simulation time, incorporating the pressure controls or more error checking etc. may decrease the speed performance closer to the current CPU method. But even a slight speed up can, over time, translate into a large amount of compute time that is available for other uses.

Conclusion

Despite many features not yet being available, the speed up in simulation time seen here is impressive. the ability to complete a simulation on a single CPU plus a GPU in the same time that is taken by 16 CPUs in a cluster computer could give much more compute power and time to regular scientific users for comparably little cost outlay.

The compilation instructions and support for all versions of NAMD is useful and a large amount of time is being spent working on this new technology and ensuring that the new, accelerated, version has as much functionality as possible. Features such as reproducible results and favourable comparisons to previous implementations of NAMD will be necessary for the future.

References

1 VPAC Summer Internship Program: http://www.vpac.org/training/internships, Feb 2009.
2 NAMD website: http://www.ks.uiuc.edu/Research/namd/, Feb 2009.
3 Research into GPUs: http://www.ks.uiuc.edu/Research/gpu/#md, Feb 2009.
4 Details on Tango: http://www.vpac.org/services/supercomputers/tango, Feb 2009.
5 VPAC website: http://www.vpac.org/, Feb 2009.
6 OpenMPI: http://www.open-mpi.org/, Feb 2009.
7 NAMD Downloads: http://www.ks.uiuc.edu/Development/Download/download.cgi?PackageName=NAMD, Feb 2009.
8 NAMD Source compiling instructions: http://www.ks.uiuc.edu/Research/namd/development.html
9 CVS website: http://www.nongnu.org/cvs/, Feb 2009.
10 NAMD Source and CVS instructions: http://www.ks.uiuc.edu/Research/namd/development.html, Feb 2009.
11 CUDA website: http://www.nvidia.com/object/cuda_home.html#, Feb 2009.
12 NAMD 2.6 User's Guide: http://biowulf.nih.gov/apps/namd/namd-ug/node26.html, Feb 2009.