Benchmark test of NAMD on Combo

Kevin
CBMSG
(Dated: July 31, 2014)

To more efficiently make use of the computational resources, we ought to do benchmark test for software installed especially for those simulation codes as they usually will occupy the nodes for consecutive weeks. We conclude that due to the aggressive access to CPUs, currently we recommend to use all the 16 CPUs on the node when running NAMD (CUDA). However in the future, we wish to use less number of CPU as it obviously has a better scaling. Comparing NAMD (CUDA) and NAMD (MPI), NAMD (CUDA) is highly preferable as it saves a large number of CPUs and provide acceptable speeds.

INTRODUCTION

To more efficiently make use of the computational resources, we ought to do benchmark test for software installed especially for those simulation codes as they usually will occupy the nodes for consecutive weeks.

METHODOLOGY

Benchmark tests were carried on molecule system provided by GPU tutorial from NAMD website.

RESULTS

Benchmark

TABLE I. This is a table of performance of NAMD (CUDA).

# of CPUs	Benchmark (ns/day)	(ns per day per cpu)
1	0.028	0.028
2	0.048	0.024
4	0.102	0.026
6	0.151	0.025
8	0.200	0.025
12	0.254	0.021
16	0.259	0.016

TABLE II. This is a table of performance of NAMD (multi-core).

# of CPUs	Benchmark (ns/day)	(ns per day per cpu)
1	0.007	0.007
4	0.026	0.006
8	0.052	0.007
16	0.095	0.006

TABLE III. This is a table of amount of speed-ups.

# of CPUs	Speed-ups (X)
1	4.2
4	3.9
8	3.8
16	2.7

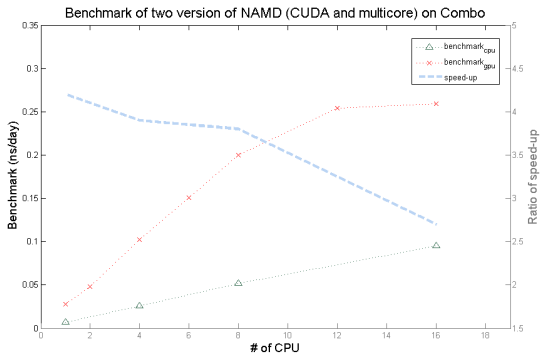

FIG. 1. This figure gives benchmarks of two versions of NAMD (CUDA and multi-core) together with speed-up ratio of CUDA to multi-core version. Ratio of speed-up was calculated by divide benchmark of CUDA version by that of the multi-core version to see by how many times it scales.

FUTURE WORK

We ought to test whether GPU and non-GPU jobs could run on the same node. That means if a job run by NAMD (CUDA) is occupying only 8 of the processors on a node, whether the rest 8 could run another independent non-GPU job, e.g. C/C++ code or even MD and leave the GPU job unaffected. This may reduce the number of processors in use for a single MD with acceptable speed-up.

CONCLUSION

We conclude that due to the aggressive access to CPUs, currently we recommend to use all the 16 CPUs on the node when running NAMD (CUDA). However in the future, we wish to use less number of CPU as it obviously has a better scaling. Comparing NAMD (CUDA) and NAMD (MPI), NAMD (CUDA) is highly preferable as it saves a large number of CPUs and provide acceptable speeds.