Research and Analysis the Performance of NAMD- Molecular Dynamics Simulation Based Multi-core +GPU

Zhang Yang, Chen Wen Bo
Centre of Communication Network,
University of Lanzhou, China
E-mail: zhyang@lzu.edu.cn
chenwb@lzu.edu.cn

Bai Qi Feng ,Li lian
School of Chemistry and Chemical
Engineering, University of Lanzhou, China
School of Information Science and Engineering
University of Lanzhou, China
E-mail: Baiqf11@lzu.edu.cn
lil@lzu.edu.cn

***Abstract*— Nowadays, the GPU can provide about one order of magnitude accelerated over CPU code and even more, so it can be used in large scale science computing applications. The paper based on NVIDIA Tesla C2050 and dual-core CPU server through use NAMD 2.9 simulates three differences molecule number proteins: Beta2- adrenergic receptor, SET9 and Ubiquitin. Through the comparison and analyses the results of the simulations experiment, we can conclusion that the difference systems of molecule will lead to the difference GPU accelerated. The computing times of four GPU is nearly half of the time used by 1 GPU; and this is especially true in the case of macromolecules. Furthermore, from the GPU's memory utilization rate, the larger the protein system is, the higher the memory use of the GPU is. NVIDIA Tesla C2050 is can satisfy an even larger system simulation. The paper provides the best options for application of this software use the GPU and multi-core framework, a reference to building large scale molecule stimulations platforms, and a solution to science application of large molecule stimulation.**

Keywords-GPU;NAMD;CUDA; molecular stimulation; Performance analyses; NVIDIA Tesla C2050

I. INTRODUCTION

GPUs [1] contain hundreds of independent execution units. It can be used in arithmetic-intensive and high degree parallel scientific applications. The floating point arithmetic and memory bandwidth have far exceed fast CPU at present, and provide large scale double precision floating point arithmetic. In the past few years, the GPU floating performance reached the Teraflop, and made computation speed doubled or even one hundred times in ascension, such as simulation of large bimolecular. NAMD [2], as the first GPU realized molecule dynamic software, made stimulation speed improved greatly; the amount of simulating molecules had been increased hundreds of thousands , or even more. At present, the NAMD GPU accelerated performances well in molecular dynamics, molecular surface visualization and molecular orbital display [3]. The latest version of NAMD is faster, having realized the non-bonded force evaluations, energy evaluation and optimization and made large scale molecular stimulation faster. Multi-core [4] technology has reached its maturity. Building multi-core +GPU computing platform can fully play the advantage of both, and provide powerful tools for the faster development of high performance computing science.

II. GPU TECHNOLOGY AND CUDA

The GPU computing or GPGPU is to use graphics processor for general science and engineering calculation [5]. The GPU computing model is to combine the CPU and the GPU in heterogeneous coordination treatment calculation model. The application of serial parts will run in the CPU, while the hard task of computation will be accelerated by the GPU. To users, the application speed is greatly improved due to the performance of the GPU. The GPU is powerful in parallel processing and data transmitting. This parallel includes three levels: instruction level, data level and mission level. The highly efficient transmitting of data is realized by the bandwidth between GPU and video memory and the bandwidth between system memory and video memory. Because most of the transistors in GPU are used for data processing rather than data cache or traffic control, GPU has advantages in intensive computing science applications, and has became a widely used computing platform. Normally the CPU supports two to eight core processors, but the GPU has hundreds of core processors. At present the peak (Flops) and memory bandwidth of the GPU have greatly exceeded those of the fastest CPU; at the same, the GPU can carry out large scale double-precision floating point arithmetic.

In 2007, NVIDIA released the CUDA—Computer Unified Device Architecture [6]. It is a new software and hardware structure. The CUDA changes the programming model and storage model for the GPU; it enables dramatic increases in parallel computing performance for the GPU. Because CUDA treats the GPU as parallel data processing equipment, it therefore contributes to the resolution of scientific and engineering complex computing problems. CUDA enables the execution of the C function or the "kernel" on the GPU device, thus allowing developers to use C++, OpenCL, DirectCompute, and Fortran as high-leveled programming languages and CUDA API to fast develop GPU application programs [7].

Published by Atlantis Press, Paris, France.

III. NAMD

NAMD is a type of molecular dynamics code, capable of quickly simulating large bimolecular systems in paralleled computers. It has been developed by the collaboration of the Theoretical and Computational Biophysics Group and the Parallel Programming Laboratory at the University of Illinois. By means of the numerical, NAMD uses empirical force fields such as AMBER or CHARMM to solve equation of motion for calculation atom trajectory. NAMD is based on Charm++ parallel objects [8], and thus possessing parallel efficiency. It can support hundreds of cores for typical simulations and over 200,000 cores for the largest simulations. NAMD 2.7 has realized GPU accelerated in the non-bonded force evaluation, Energy evaluation is done on the CPU. The latest version 2.9 achieves born implicit solvent (GBIS) model and energy evaluation and minimization; it can also stand by CUDA 4.0. So the computing speed is faster than before.

IV. EXPERICIMENCS DESIGN

Except supercomputers, most GPU servers have one or two GPUs in a single host; the maximum is four. In our experiment, the server model is Dawn W580. It has 4 NVIDIA Tesla C2050 in the host, 32G DDR 1333 memory, two INTEL Xeon 5650 CPU, 6 cores per CPU, 2.66GHz, 24 cores in total. Every GPU includes 448 CUDA cores and its peak reaches 1.02Gflops. So the peak of every single host is 4.12Tflops. Tesla GPU for 20 serial product is based on "Fermi" CUDA architecture. It adds some of the main features that can greatly increase double-precision floating point arithmetic, so it has achieved extremely high precision and scalability, including C++ programming and ECC Memory, and is seven times faster in speed than Tesla GPU for 10 serial product. The NVIDIA Tesla C2050 computing processors can achieve the same super-computing performance as the latest quad-cores CPU does but costs 90% less than the latter and its power consumption is also 95% less than the latter. The NVIDIA Tesla C2050 has 448 CUDA cores; its memory bandwidth is 133GB/s; it can ultimately realize 515Gflops double-precision floating point arithmetic. For molecular simulations [9] [10], the scale of the simulation system can impact the experiment result. The larger the scale is, the more the non-bonded force evaluation is, and thus the accelerated is more evident. Moreover, the VDW evaluation is another important part in molecular simulation.

In NAMD 2.9, the GPU only realized the accelerated of the VDW evaluation, but in the latest version 2.9, the GPU not only realized the accelerated of the VDW evaluation but also the energy evaluation, so the performance is better. The soft environment of this experiment is CUDA 4.0 and NAMD 2.9.

In order to testify the difference between the single and multi GPU in performance, we separately use 1 GPU, 2 GPUs and 4 GPUs to carry out the computing. Furthermore, in order to analyze the impact of cores on GPU efficiency, we add 12 cores and 24 cores to the GPU to testify their proper effects. We select three different protein systems: Bera2、SET9 and Ubiquitin. Beta2-adrenergic receptor (β2AR) belongs to class A G protein-coupled receptors (GPCRs). It can transfer signaling from extra cellular to intracellular across the membrane. It has seven transmembrane domains consisting of seven α helixes. The system contains β2AR and Gs protein with explicit waters and lipids. The total number of the system is 200,000. SET9 is a protein lysine methyltransferase. It can be capable of transferring only one methyl group to target lysine residues. The total number of the system is 56,800. Ubiquitin is a small globular protein, which plays a key role on eukaryotic intracellular protein degradation, and has been found in all tissues of eukaryotic organisms. It consists of 5 α-helixes and 2 β-sheets. The total number of system is 7051.

Regarding timestep, which are used as the parameter, we select the 10ps, 100ps, 1000ps (1ns) so that we can determine the computing speed of the three proteins in different time steps.

V. THE EXPERIMENT RESULTS AND ANALYSES

TABLE I. BETA2 IN 12 CORES

	10ps（m）	100ps	1000ps
1GPU	552.978943	5431.754395	54167.359375
2GPU	332.957397	3165.734619	31745.671875
4GPU	274.061340	2521.300781	24911.230469

TABLE II. BERA2 IN 24 CORES

	10ps	100ps	1000ps
1GPU	582.673401	5635.046387	56488.128906
2GPU	356.944733	3292.193604	32953.097656
4GPU	316.559875	2586.567871	26237.843750

TABLE III. SET9 IN 12 CORES

	10ps	100ps	1000ps
1GPU	104.366135	972.111206	9641.909180
2GPU	85.645981	678.702820	6750.493652
4GPU	82.675430	621.537537l4	5972.527832

TABLE IV. SET9 IN 24 CORES

	10ps	100ps	1000ps
1GPU	117.699104	1057.883179	10648.096680
2GPU	104.715080	766.817444	7621.949219
4GPU	102.096481	737.507874	6790.793457

TABLE V. UBIQUITIN IN 12 CORES

	10ps	100ps	1000ps
1GPU	32.905998	251.452774	2375.378906
2GPU	34.459763	216.567078	2038.278076
4GPU	43.371407	209.281189	1877.912476

TABLE VI. UBIQUITIN IN 24 CORES

	10ps	100ps	1000ps
1GPU	38.498150	626.930664	4951.262207
2GPU	63.900284	667.685486	5097.090332
4GPU	94.444641	777.082886	5936.767578

TABLE VII. 12 CORES,DAY/NS

	Beta2- adrenergic	SET9	Ubiquitin
1GPU	0.627602	0.111238	0.0377698
2GPU	0.364271	0.0779933	0.0318674
4GPU	0.286669	0.0675498	0.0306479

TABLE VIII. 24 CORES, DAY/NS

	Beta2- adrenergic	SET9	Ubiquitin
1GPU	0.649952	0.123102	0.0422625
2GPU	0.37654	0.087601	0.0388901
4GPU	0.296598	0.077689	0.0402171

Table Ⅰ shows the simulation execution times of Beta2 separately in 1 GPU plus 12 cores, 2 GPUs plus 12 cores and 4 GPUs plus 12 cores; table Ⅱ shows the simulation execution times of Beta2 separately in 1 GPU plus 24 cores, 2 GPU plus 24 cores and 4 GPUs plus 24 cores; table Ⅲ, Ⅳand table Ⅴ, Ⅵ separately shows the result of SET9 and Ubiquitin in the same conditions as mentioned above. TableⅦ,Ⅷ shows the Beat2 benchmark time separately in 12 core and 24 cores. In figure 1, 2, 3, it is obvious that with the increase of the numbers the GPU, the computing time is decreasing. When the timestep is set at 1ns, the execution times of the 4 GPUs are the shortest, nearly half of the time used by 1 GPU. For different proteins, the computing time slightly differentiates. For the Ubiquitin, when the timestep is 10ps, the computing time of the 1 GPU is the fastest; however, when the timestep is more than 10ps, the computing time of the 2 GPUs and 4 GPUs is still quicker than that of the 1 GPU. The result is related to the protein itself. When the system of the molecular is too small, the multi-GPU threads are not fully occupied and 1 GPU process threads is enough, so the increase speed at multi-GPU is not clearly. More detail data are described a companion manuscript (Bai et al., manuscript submitted)

Figure 4 shows the day/ns for the three proteins when the timestep is 1ns. We can see that the Bera2 computing efficiency is best. The benchmark time is 0.627602 days/n for the 12 cores plus the 1 GPU, 0.364271 days/ns for the 12 cores plus the 2 GPUs, 0.286669 days/ns for the 12 cores plus the 4 GPUs. The larges the simulation system is, the more evident the effect is. From the number of threads, we can see that the computing time of the 12 cores is quicker than that of the 24 cores, and therefore the most efficient.

In the experiment, through commanding “nvidia-smi –a” we can determine the utilization rate of the GPU and its memory status. When we use the 4 GPUs plus 12 cores, the utilization rate of Beta2- adrenergic reaches to 50 percent at most, and the memory of the GPU has been used by 4 percent; the utilization rate of SET9 reaches to 8 percent at most, and the same amount of the GPU’s memory has been used as in the situation of Beta2- adrenergic. However, when we use the 4 GPUs plus the 24 cores, the utilization rate of Beta2- adrenergic, SET9, and Ubiquitin has respectively decreased to 45 percent, 70 percent, and 30 percent. In the 2 GPUs, the utilization rates of Beta2-adrenergic and SET9 have both reached to 75 percent at most, but the memory use of Beta2-adrenergic is 7 percent, while the other is 2 percent. Best utilization rate of the GPU is found in single GPU, reaching to 88 percent. The above result shows that different protein systems lead to different utilization rates of the GPU; the GPU acceleration on NAMD still has a great room for growth, especially in load balance; and the utilization rate of every GPU on the multi-GPU computing is different. The larger the protein system is, the higher the memory use of the GPU is. As a whole, the memory use of the GPU is not much influenced by the amount of the GPU. NVIDIA C2050 can satisfy an even larger system simulation. In our experiment, we also find that the system memory is the same no matter how many GPU have used. Thus, it is concluded that the memory size is not the main factor in molecule simulation when it reaches at certain degree (experiment use 16G Memory in host), and data exchange mainly takes place in GPU memories.

Although the parallel efficiency of NAMD is determined by the specific combination of hardware, molecular system, and algorithms, the advent of NAMD GPU acceleration makes the research on molecular simulation faster than before. To build large-scaled computing platform costs much and the power consumption is very high. Therefore, the research on the efficiency of NAMD applied on the GPU is helpful for researchers to select more suitable hardware in accordance to specific systems. Hopefully more people can access to the NAMD simulation of large bimolecular systems and make GPU application more efficient and powerful.

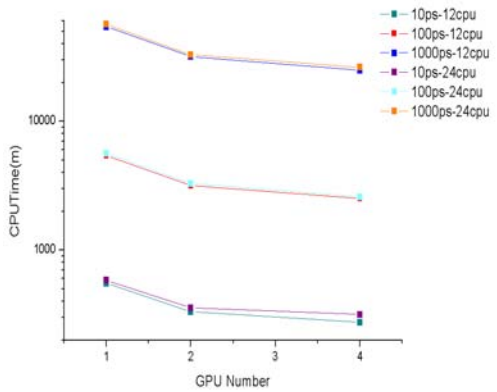

Figure 1 Stimulation Beta2-adrenergic

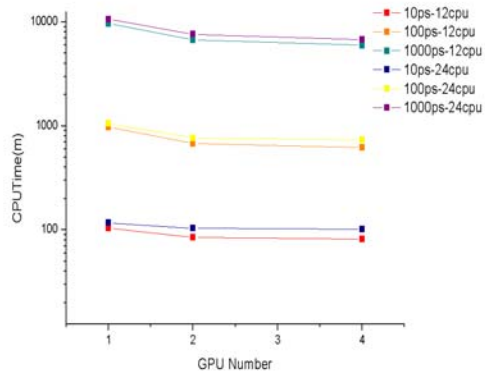

Figure 2 Stimulation SET9

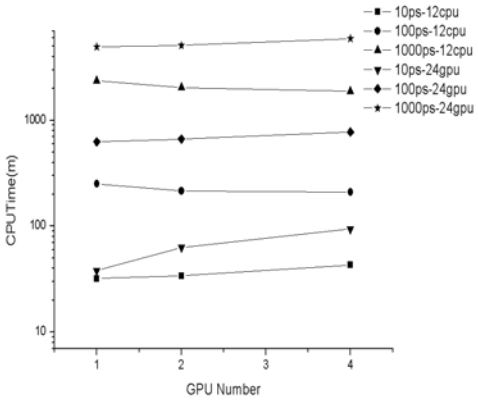

Figure 3 Stimulation Ubiquitin

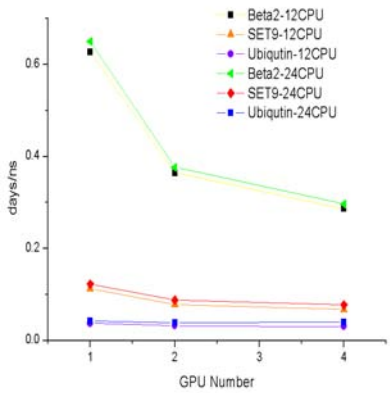

Figure 4 Stimulation in 1ns

REFERENCES

[1] Buck, Ian, Cuda Programming, the International Conference for High Performance Computing, Networking, Storage and Analysis, 2007, http://sc07.supercomputing.org/)

[2] NAMD, http://www.ks.uiuc.edu/Research/gpu

[3] John E. Stone, Jan Saam, David J. Hardy, Kirby L. Vandivort, Wen-mei W. Hwu, and Klaus Schulten. High performance computation and interactive display of molecular orbitals on GPUs and multi-core CPUs. In Proceedings of the 2nd Workshop on General-Purpose Processing on Graphics Processing Units, ACM International Conference Proceeding Series, volume 383, pp. 9-18, New York, NY, USA, 2009. ACM

[4] Volodymyr Kindratenko, Jeremy Enos, Guochun Shi, Michael Showerman, Galen Arnold, John E. Stone, James Phillips, Wen-mei Hwu.GPU Clusters for High Performance Computing. Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on. pp. 1-8, Aug. 2009

[5] John E. Stone, David J. Hardy, Ivan S. Ufimtsev, and Klaus Schulten. GPU-accelerated molecular modeling coming of age. Journal of Molecular Graphics and Modelling, 29:116-125, 2010

[6] NVIDIA,NVIDIA's Next Generation CUDA Compute Architecture:Fermi,whitepaper,NVIDIA,2009,Availableonline,Version1.1,22pp

[7] NVIDIA,NVIDIA CUDA Compute Unified Device Architecture Praogramming Guide,NVIDIA,Santa Clara,CA,USA,2007

[8] J.C. Phillips, R. Braun,W.Wang, J. Gumbart, E. Tajkhorshid, Villa, C. Chipot, R.D. Skeel, L. Kale, K. Schulten, Scalable molecular dynamics with NAMD, J. Comp. Chem. 26 (2005) 1781–1802.

[9] Jeremy Enos, Craig Steffen, Joshi Fullop, Michael Showerman, Guochun Shi, Kenneth Esler, Volodymyr Kindratenko, John E. Stone, and James C. Phillips. Quantifying the Impact of GPUs on Performance and Energy Efficiency in HPC Clusters. International Conference on Green Computing, pp. 317-324, 2010.

[10] John E. Stone, James C. Phillips, Peter L. Freddolino, David J. Hardy, Leonardo G. Trabuco, and Klaus Schulten. Accelerating molecular modeling applications with graphics processors. Journal of Computational Chemistry, 28:2618-2640, 2007 J.C.