Effect of Parallel Processing and Optimization Techniques in Molecular Dynamics

KENJI SATOU[1], KENRI KONNO[2], OSAMU OHTA[3], KAZUNORI MIKAMI[4], KEITA TERANISHI[5], YOICHI YAMADA[1], SHIN-YA OHKI[6]

1 Graduate School of Natural Science and Technology, Kanazawa University
Kakuma-machi, Kanazawa 920-1192, JAPAN

2 Graduate School of Materials Science
3 Graduate School of Information Science
6 Center for Nano Materials and Technology
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, JAPAN

4 Cray Japan Inc.,
1-2-2 Uchisaiwaichou, Chiyoda-ku, Tokyo 100-0011, JAPAN

5 Cray Inc.
Mendota Heights, Minnesota 55120, U.S.A.

Abstract: - Molecular dynamics (MD) has been studied long time due to its attractive function of predicting structure of molecules. Though many studies reported improved algorithms, it is unclear to an end-user of MD software tool to choose appropriate computer architecture, type of parallel processing, and optimization options to compile and execute the tool. In this study, we tested various combinations of parallel processing and optimization options on four different computer architectures, i.e. a vector supercomputer, multi-processor supercomputer with shared and distributed memories, and a PC cluster. Experimental results revealed superiority of PC cluster against other expensive supercomputers.

Key-Words: - Molecular dynamics software, Computer architecture, Parallel processing, Optimization

1 Introduction

As the success of Folding@Home project[1] demonstrates, there is a great demand of biomolecule analysis through molecular dynamics (MD) and efforts have been concentrated on the development of improved algorithm and software [2]. There exist many MD tools: AMBER[3] and CHARMM[4] are the most famous software suites, Tinker[5] and Gromacs[6] are relatively more simple and easy-to-use, myPresto[7] and Peach[8] were developed in Japan, and so on. These software tools are useful for both of commercial development of new pharmaceuticals and academic research in structure and function of biomolecules. Except CHARMm, the tools above are free of charge or distributed at fairly low cost for the purpose of academic research. So, it is popular to personally install one of them and use it also personally or share it in a laboratory. However, even with today's computers dramatically improved in performance, it is still tough computation to solve the structure of large biomolecule like protein with huge amount of water molecules as solvent surrounding it. Therefore, acceleration techniques for MD have been actively studied.

There are many previous works on the acceleration of MD computation. They can be roughly classified into two categories: reduction of computation and parallel computation. In general, application of a technique in the former is limited since there must be a trade-off between reduction and precision of

computation. The latter can be divided into finer categories: 1) parallel computing by vector processor, 2) parallel computing on a computer with multiple CPUs, 3) parallel computing on multiple computers connected with LAN (i.e. PC cluster), and 4) parallel computing on multiple computers connected with WAN (i.e. Grid computing).These are also in the historical order of trends in research and development of MD acceleration techniques. Once a PC was too poor to perform MD, and it was studied to make the best use of a supercomputer with one or a few vector processors for this purpose. After that, a multi-processor machine which has shared or distributed memory and multiple scalar processors connected with high-speed channel and switch became common. As a result, MD acceleration techniques by multiprocessing and/or multithreading were actively studied. Though a programming completely different from vector-parallel processing is required, this approach achieved considerable success by the high-speed communication mechanism and large memory capacity. Utilization of PC cluster can be a natural extension of this approach in significantly lower cost. To hide the latency of LAN, it is popular to use Myrinet instead of Ethernet and high-performance network communication library like SCore.

Though there are various previous works, it is difficult to compare experimental results to each other since they were measured on different computer architectures. In addition, most of the acceleration techniques reported in papers require source-level modification of MD software tools, and unable to be reproduced without deep understanding of source code and parallel programming. Therefore, there is no clear guideline for a biochemist to choose best computer architecture for MD computation. Furthermore, in case of a MD software tool provided as source code (e.g. AMBER), choice of optimization options in compilation of the source code might greatly affect to the speed of MD computation.

Based on the above backgrounds, in this study we measured and compared performances of MD computation with various combinations of machine architectures, parallelization techniques, and optimization options. Except an architecture which definitely requires minimum modification to run the code, the same MD software tool was used without source code modification in the experiment. By avoiding source code modification as much as possible, the experimental results in this paper revealed a guideline for a biochemist to choose the best machine architecture for MD.

2 myPresto and cosgene

In this study, we adopted myPresto Version 3 as MD software tool for performance measurement. myPresto is distributed free of charge for non-commercial use at University. Among programs in myPresto, cosgene performs MD computation. myPresto is provided as source code and executables precompiled in some platforms. To compile cosgene from source code, Fortran 90 is needed. From one source code, an executable for serial computation or an executable for parallel computation via MPI library can be generated depending on configuration parameter. Hereinafter, we call the executables for serial and parallel computations cosgene_serial and cosgene_MPI, respectively.

About vectorization, it was reported that Presto, the predecessor of myPresto, was originally vectorized and achieved high performance on supercomputers like NEC SX series and Fujitsu VP series. However, source code of myPresto is basically independent from Presto and does not include vectorized codes.

3 Computer Platforms

We used the following four computer platforms with different architectures and operating systems.

NEC SX-8

This machine is a descendant of SX-5 which share almost the same vector processors with the Earth Simulator [9]. SX-8 realizes peak vector performance of 16Gflops per CPU (vector processor). In the experiment, we used a model of SX-8 with 8 CPUs and 64GB memory. In case of interactive use, all the 8 CPUs are available, while 6 CPUs in batch processing via a queueing system NQSII. Operating system is SUPER-UX.

SGI Altix 3700

This machine is classified as shared memory multi-processor computer. The model we used contains 32 C-blicks connected with NUMAlink3, where each C-blick has four Itanium2 processors (1.6GHz) and 24GB memory. In total, this model provides 128 CPUs and 768GB shared memory.

Cray XT3

This machine is classified as distributed memory multi-processor computer. The model we used contains 90 nodes connected in 3D torus link, where

each node has four Opteron 150 processors (2.4GHz) and 32GB memory.

Appro HyperBlade Mid-Cluster

This machine is classified as PC cluster. The model we used contains 32 PCs connected with Gigabit Ethernet, where each PC has two Opteron DP Model 250 processors (2.4GHz) and 4GB memory.

4 Parallel Computation Types

We tried the following types of parallel computation for the platforms described in the previous section.

NEC SX-8

- vector-parallel processing through automatic vectorization by compiler with -Chopt option.
- process- or thread-parallel processing through automatic parallelization by compiler with -Pauto option.
- process- or thread-parallel processing conducted by cosgene_MPI.
- combination of these types.

SGI Altix 3700

- process- or thread-parallel processing through automatic parallelization by compiler with –parallel option.
- process- or thread-parallel processing conducted by cosgene_MPI.
- combination of these types.

Cray XT3

- process- or thread-parallel processing conducted by cosgene_MPI (minimum modification is applied to source code of cosgene to run it on XT3).

Appro HyperBlade Mid-Cluster

- process- or thread-parallel processing conducted by cosgene_MPI without compilation (i.e. provided executable was used as is).

5 Compilers and Options

In the configuration of cosgene, we typically specified the following compilers and options for each platform, where FC and FC_MPI denote the name of Fortran 90 compiler for cosgene_serial and cosgene_MPI, respectively, and OPT denotes optimization options passed to compiler. PP=fpp is a special option only for ifort to invoke preprocessor. For more details about options, see the manual of each compiler.

NEC SX-8

- FC = f90
- FC_MPI = mpi90
- OPT = -C debug -D_SMALL_SYSTEM
- combinations of -g (debug), -Chopt (full use of optimization and vectorization upper limits), -Cnoopt (no vectorization and optimization), -Cvsafe (very safe use of optimization and vectorization without side effect), -EP (C preprocessor activation), -pi auto (automatic inline expansion), and -Pauto (automatic parallelization) were tried as additional options.

SGI Altix 3700

- FC = ifort
- FC_MPI = ifort
- OPT = -O2 -static
- PP = -fpp
- -parallel (automatic parallelization) was tried as an additional option.

Cray XT3

- FC = ftn
- FC_MPI = ftn
- OPT = -fast -fastsse -O3 -mcmodel=medium

Appro HyperBlade Mid-Cluster

- FC = pgf95
- FC_MPI = mpif90
- OPT = -fast -fastsse -O3

6 Protein Molecule for MD

For MD computation of biomolecule, we adopted a protein called myosin phosphatase inhibitor CPI-17 with Thr38 replaced with Asp [10]. 1j2m is the PDB code of this protein containing 99 residues (Fig.1). After energy minimization, a new conformation 1j2m_min was prepared and input to cosgene_serial and cosgene_MPI (Fig.2). In MD computation, a force field parameter C99_aa.tpl was adopted, which contains topology information for all amino acid monomers for the AMBER96 force field. 100ps MD simulation was performed in each experiment. Fig.3 shows an example of conformation after 100ps simulation.

performance (4CPUs, cosgene_MPI, additional options allowed). Here we see that acceleration was possible in SX-8, Altix, and XT3, however their best performances were lower than the control point of HBMC.

Table 2. Accerelation ratio

	Control (second)	Best (second)	ratio
SX-8	135823	14803 (-Cvsafe)	9.18
Altix	20452	8301	2.46
XT3	14141	4700	3.01
HBMC	4357	1306	3.34

4 Conclusion

In this study, we tested various combinations of parallel processing and optimization options on four different computer architectures, i.e. a vector supercomputer, multi-processor supercomputer with shared and distributed memories, and a PC cluster. Experimental results revealed superiority of PC cluster against other expensive supercomputers. However, scalability of MPI parallel was not so promising. Similarly, automatic vectorization was not so effective since in comparison with acceleration by -Chopt, around 80% of it can also be achieved by a simple optimization, i.e. inline expansion by -pi auto. It implies that percentage of vectorization by compiler might be low. In other words, though a supercomputer with huge memory is still needed to solve a fine structure of extremely large biomolecules, a common PC with a dual- or quad-core processor and large memory (4GB or more) is one of the competitive alternatives to solve a structure of relatively smaller biomolecules by using a popular MD software tool like myPresto.

References:

[1] S.M. Larson, C.D. Snow, M. Shirts and V.S. Pande., Folding@Home and Genome@Home: Using distributed computing to tackle previously intractible problems in computational biology, Computational Genomics, Horizon Press, 2002.

[2] S.A. Adcock and J.A. McCammon, Molecular Dynamics: Survey of Methods for Simulating the Activity of Proteins, Chem. Rev., Vol.106, No.5, 2006, pp.1589-1615.

[3] D.A. Pearlman, D. A. Case, J. W. Caldwell, W. S. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel and P. Kollman, AMBER, a Package of Computer Programs for Applying Molecular Mechanics, Normal Mode Analysis, Molecular Dynamics and Free Energy Calculations to Simulate the Structural and Energetic Properties of Molecules, *Comp. Phys. Commun.*, Vol.91, 1995, pp.1-41.

[4] B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan and M. Karplus, CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations, *J. Comput. Chem.*, Vol.4, 1983, pp.187-217.

[5] J.W. Ponder and F.M. Richards, An efficient Newton-like method for molecular mechanics energy minimization of large molecules, *J. Comput. Chem.*, Vol.8, 1987, pp.1016–1024.

[6] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation, *J. Chem. Theory Comput.*, Vol.4, No.3, 2008, pp.435-447.

[7] Y. Fukunishi, Y. Mikami, and H. Nakamura, The filling potential method: A method for estimating the free energy surface for protein-ligand docking, *J. Phys. Chem. B.*, Vol.107, 2003, pp.13201-13210.

[8] Y. Komeiji, M. Uebayasi, R. Takata, A. Shimizu, K. Itsukashi, M. Taiji, Fast and accurate molecular dynamics simulation of a protein using a special-purpose computer, *Journal of Computational Chemistry*, Vol.18, Issue 12, 1998, pp.1546-1563.

[9] S. Shingu, H. Takahara, H. Fuchigami, M. Yamada, Y. Tsuda, W. Ohfuchi, Y. Sasaki, K. Kobayashi, T. Hagiwara, S. Habata, M. Yokokawa, H. Itoh, and K. Otsuka, A 26.58 Tflops Global Atmospheric Simulation with the Spectral Transform Method on the Earth Simulator, *Proc. of the ACE/IEEE SC2002 conference*, 2002.

[10] S. Ohki, M. Eto, M. Shimizu, R. Takada, D.L. Brautigan, M. Kainosho, Distinctive Solution Conformation of Phosphatase Inhibitor CPI-17 Substituted with Aspartate at the Phosphorylation-site Threonine Residue, J.*Mol.Biol.*, Vol.326, Issue 5, 2003, pp.1539-1547.