Parallel molecular dynamics simulation for protein sequences on PC-cluster and server

M. BUŢU[a,b*], A. BUŢU[a]
[a]*National Institute for Research and Development of Biological Sciences, Splaiul Independenţei 296, Bucharest, Romania*
[b]*Faculty of Physics, University of Bucharest, Bucharest-Magurele, Romania*

In order to reduce the time of simulation of molecular dynamics production, and to increase the system size, the simulation techniques have been developed to distribute a simulation over a set of processors. For protein sequences simulation, parallel molecular dynamics simulations have been widely used as an important basic technique. Studying molecular dynamics of protein sequences is accomplished by getting “in silico” simulation of the dynamics trajectories. Shortening the time required to obtain these trajectories is possible with parallel computing. The aim of these experiments was to analyze the optimal hardware resources needed for parallel molecular dynamics simulation method of protein sequences. For molecular dynamics simulations there were used two protein sequences - a protein having 127 aminoacids and a peptide having 9 aminoacids, CHARMM software package, PC-cluster and server.

(Received September 23, 2010; accepted November 19, 2010)

Keywords: Parallel simulations, Molecular dynamics simulation, Protein, Peptide, Hardware resources, PC-cluster, Server

1. Introduction

Molecular Dynamics is a technique for computer simulation of complex systems, modeled at the atomic level. Molecular dynamics simulations yield the possibility of describing and understanding the relationships between the structure and the function of biomolecules [1]. This can be a very powerful tool to predict quantities that either cannot be measured directly or accurate experimental data are difficult to obtain. It provides a microscopic view which may serve to explain macroscopic behaviour of a biomolecular system. Molecular dynamics simulations can be performed in a microcanonical ensemble (constant number of molecules, volume, and total energy), a canonical ensemble (constant number of molecules, volume, and temperature), as well as in an isothermal-isobaric ensemble (constant number of molecules, pressure, and temperature). Molecular dynamics can be applied to: sampling configuration space, e.g., simulated annealing to determine or refine structures; obtaining a description of the system at equilibrium, i.e., sampling with appropriate Boltzmann factor (structural and motional properties and values of thermodynamic parameters); obtaining actual dynamics and kinetics, i.e., sampling with appropriate Boltzmann factor and correct representation of the development of the system over time [1]. Molecular dynamics simulations have been widely used to study proteins, peptides and other biomolecules [2-5].

Parallel molecular dynamics simulation has been widely used as an important basic technique for protein sequences simulation. Parallel computing is the simultaneous use of multiple compute resources to solve a computational problem. The problem is broken into discrete parts that can be solved concurrently, each part is further broken down to a series of instructions, instructions from each part execute simultaneously on different CPUs.

Manny papers deal with the possibility of interconnecting the network for parallel molecular dynamics simulation [6-11]**.** There was performed parallel molecular dynamics simulation for hen lysozyme that resulted in the highest trajectory studied - its length was 1 μs. [12-14].

2. Methodology

For molecular dynamics simulations it was used the software package CHARMM (Chemistry at Harvard Macromolecular Mechanics) version 32b1 [15, 16]. Two solvated protein sequences were used. The first was a short protein sequence, a nanopeptide with 140 atoms solvated in a water box with 46×46×46 Å dimensions. This model, is noted SPS1 (Fig. 1) and has 8933 atoms. The second was a long protein sequence, a protein, the human lysozyme, 1YAM [17]. The structure file for 1YAM was taken from PDB [18]. For pdb files preparation it was used MMTSB [19]. The protein had 130 amino acid residues, 2019 atoms and it was added 9 Cl^- ions to get zero electrical charge. The protein was solvated in a cubic water box, and the final model, noted SPS2 (Fig. 2), has 16977 atoms.

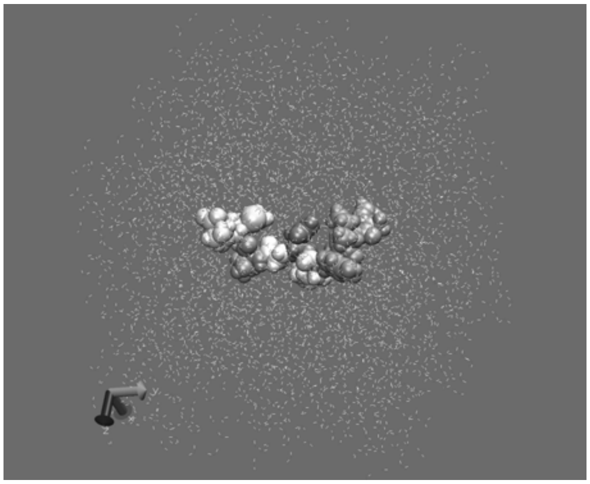

Fig. 1. The solvated protein sequences system SPS1, 8933 atoms.

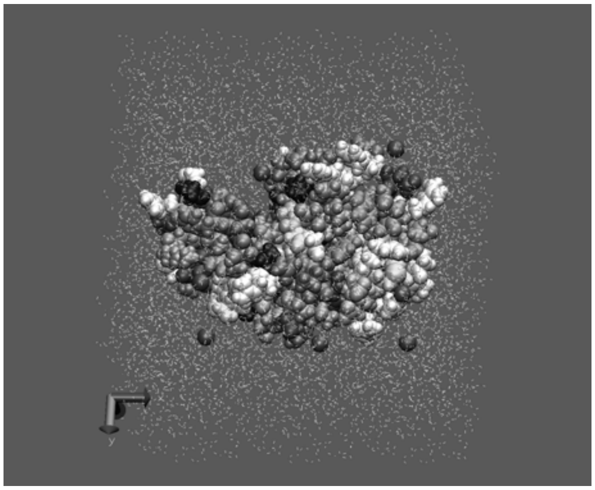

Fig. 2. The solvated protein sequences system SPS2, 16977 atoms.

The simulation of system SPS1 was performed in the NVT ensemble and simulation of system SPS2 in NPT ensemble. The non-bonded cutoff distance was 14 Å. Bonds containing hydrogens were constrained with SHAKE [13], allowing a 1-fs timestep. Both systems were subjected to energy minimization, heating and equilibration. Equilibration of the systems was followed by molecular dynamics simulation production.

A large-scale atomic/molecular massively parallel simulator (LAMMPS) was used [20]. This is a classical molecular dynamics code suitable for modeling large molecular systems. To speed the calculation LAMMPS uses either Ewald or particle particle/particle-mesh (PPPM) techniques [21, 22].

Because we used Ewald summation algorithms, the number of PCs in the clusters and the number of server processors tested was a power of 2 (1, 2, 4, 8, 16). The visualization of the biomolecular system in dynamics was done using VMD [23] and Swiss-PdbViewer [24].

For molecular dynamics simulations experiments were used two types of hardware units: a PC-cluster and a server.

The PC cluster had 8 systems interconnected by a 24 ports gigabit switch and 16 ports rackmount console KVM switch. Seven of the cluster's systems had the following configuration: processor Intel Pentium 4D at 3,2 GHz, motherboard with chipset Intel 865, 512MB DDR2 RAM, HDD 80GB, network board 1GB. One system had the following configuration: processor Intel Pentium 4D at 3,2 GHz, motherboard with chipset Intel 865, 1024MB DDR2 RAM, 3 HDD 80GB/ 500GB/ 500GB, network board 2 x 1GB. In order to avoid the fluctuations in the electricity network were used two UPS APC of 2200 VA which could sustain PC-cluster for about one hour.

The server is IBM X3950 with processor 32 X INTEL XEON MP 3 GHZ, dual core, EM64T, 2 MB Cache L2, 4 MB Cache L3, RAM memory 32 GB PC2-3200 DDR2 400 MHZ registered ECC, memory mirroring, memory hot-swap, HDD 2 x 73GB 2.5" HDD SAS HOTSWAP, 15000 RPM, Combo CD-RW/DVD drive, Controller RAID with 8 ports SAS, 256 MB RAM, battery backup for cache memory, DUAL Gigabit Ethernet 10/100/1000MB/S, Smart UPS 10000VA IBM 10000XHV, operating system MCT0982RN, Red Hat Enterprise Linux Advanced Platform.

3. Results and discussion

When programming the molecular dynamics simulation experiments it is very important to know the performances of the hardware resources that are used. Knowing the real time necessary for the simulation, as a function of the dimensions of the protein system taken into account, determines the choice of the optimal number of nodes on which will be performed the molecular dynamics simulation experiment.

The number of PCs from cluster and the number of processors from server is referred to as number of nodes. The input file for parallel simulation of molecular dynamics production runs on each configuration of the two hardware systems. There were made three readings during the simulation required for obtaining a trajectory length of 1ns (execution of 1000000 steps). The average of the three readings was used for graphical representation of the experiment's results.

In Fig. 3 and 4 it is represented the number of hours necessary to obtain a molecular simulation trajectory with the length 1ns for the two solvated protein sequences systems, SPS1 and SPS2, both on the cluster and on the server, as a function of the number of nods needed for the simulation.

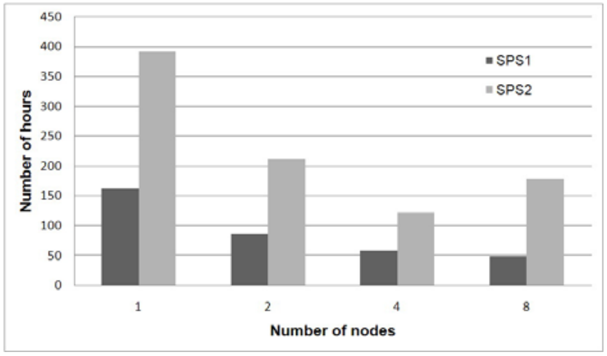

Fig. 3. The number of hours necessary to obtain a trajectory of 1ns as a function of the number of PCs in the cluster for the solvated protein sequences systems SPS1 and SPS2.

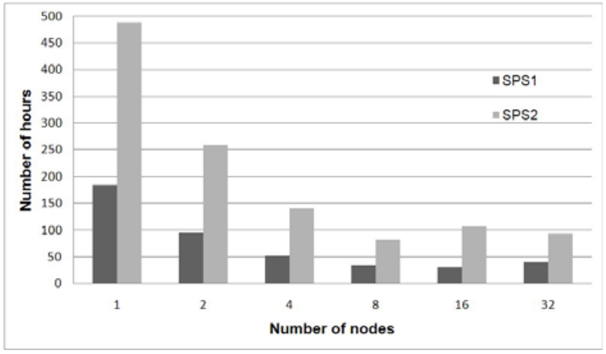

Fig. 4. The number of hours necessary to obtain a trajectory of 1ns as a function of the processor in the server for the systems SPS1 and SPS2.

It is noted that the shortest time obtained for the molecular dynamics simulation of the system SPS1 on the cluster is realized by the simulation on eight nodes, while for the system SPS2 the shortest time is obtained of the simulation on four nodes. When simulating the system SPS2 on eight nodes, the data communication time is longer than the computing time and so it appears a limitation of the performances of the cluster. Therefore the simulation on eight nodes takes longer than the simulation on four nodes.

The molecular dynamics simulation for the system SPS1 obtains the best time when is run on eight nodes, and for the system SPS2 gets the best time when running on sixteen nodes. When running on a number of nodes higher than eight, respectively sixteen, there comes out the limitation of the performances of the server. The cause is the same as the one for the cluster, namely the fact that the data communication time is longer than the computing time.

Calculating the ratio between the average number of hours needed to obtain 1 ns trajectory on a cluster with n PCs and the average number of hours needed to obtained 1 ns trajectory on a single PC we get the speed of obtaining a 1ns trajectory on the clusters against one single PC (Vn) (Fig. 5).

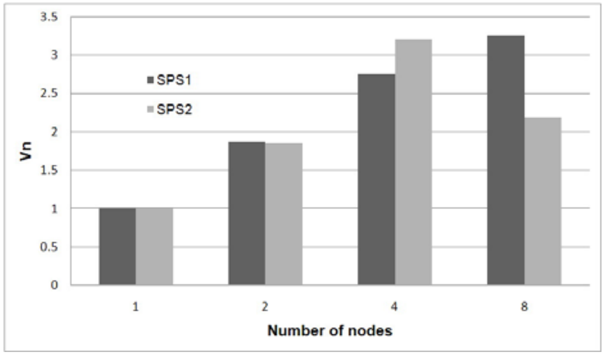

Fig. 5. The speed of obtaining a 1ns trajectory depending on the number of PCs in the cluster for solvated protein sequences systems SPS1 and SPS2.

Thus, we find that, for molecular dynamics simulation for the system SPS1, the running speed of the simulation on four computers in parallel is 2.75 times higher than the running speed on a single computer, and the running speed on eight computers overcomes with 3.25 times the running speed on a single computer.

Running the molecular simulation dynamics for the system SPS2 on four PCs in parallel is done with a speed that is three times higher than the running on a single PC, and the running speed on eight PCs in parallel is over two times higher than running on a single PC.

Using the same reasoning, it was calculated the running speed for molecular dynamics on n processors of the server against the running on a single processor. The results of the simulation for the two protein systems, of the server, was obtained both by calculating the ratio between Vn and number of nodes SPS1 and SPS2, are represented in Fig. 6.

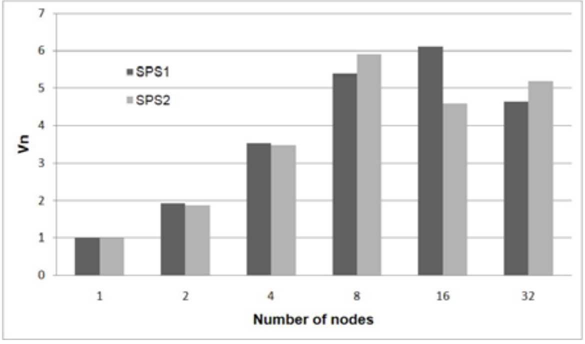

Fig. 6. The speed of obtaining a 1ns trajectory depending on the number processors in the server for solvated protein sequences systems SPS1 and SPS2.

In this case, running the simulation on sixteen processors in parallel is over six times faster than running on a single processor, for the simulation of the system SPS1 and running the simulation on eight processors in parallel is almost six times faster than running on a single processor for the simulation of the system SPS2.

The efficiency of the use of computing capacity of the clusters, respectively the curve presented by these parameter is illustrated in Fig. 7 and Fig. 8, being obvious the decreasing of the efficiency in the same time with the increasing of the number of PCs in the cluster, respectively with number of processors in server.

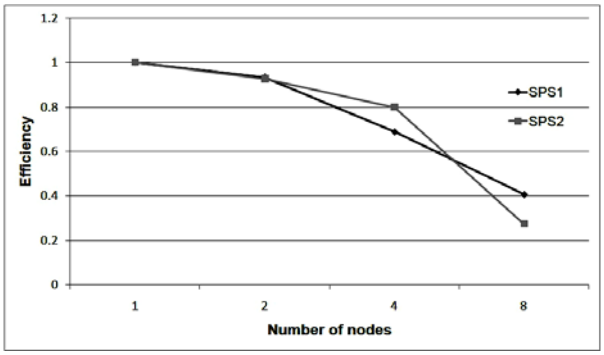

Fig. 7. The efficiency of the molecular dynamics simulation on a PC for solvated protein sequences systems SPS1 and SPS2 as a function of the number of PCs in the cluster.

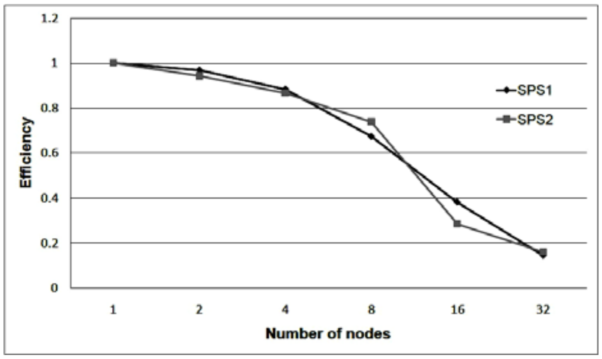

Fig. 8. The efficiency of the molecular dynamics simulation on a processor for solvated protein sequences systems SPS1 and SPS2 as a function of the number of processors in the server.

For the system SPS1, the best performance was obtained with eight nodes in the cluster, respectively sixteen nodes in the server. This uses approximately 40% of the existing computing resources. Using 32 nodes in the server leads to the weakest performance, using 15% of the computing resources. The best performances in molecular dynamics simulation for the system SPS2 are obtained when using four nodes on the cluster and eight nodes in the server, thus using 80% respectively 74% of the computing resources.

4. Conclusions

Performing the parallel molecular dynamics simulation on a PC cluster for the system SPS1 was done in the shortest time (number of hours) when using eight PCs. For the system SPS2, which has a larger number of atoms in both protein structure and the solvent, the simulation takes the least when using four PCs. Thus, to obtain the shortest real time of simulation, the number of PCs used in a simulation depends on the size of the simulated protein system.

Running parallel molecular dynamics simulation on the server for the system SPS1 has the shortest duration when using sixteen processors and respectively eight processors for SPS2. In this case, to obtain the shortest simulation time, the number of processors used is related to the number of atoms in the solvated protein system.

It thus appears that in order to obtain the shortest simulation time, the number of nodes used in parallel molecular dynamics simulation is inversely proportional to the size of the system.

As a function of the hardware resources and the size and the number of the protein systems for which is performed the molecular simulation experiments, can be made a choice regarding the number of nodes on which to make a simulation, thus optimizing the resources utilization and especially winning time.

Acknowledgement

This work was supported by the Program 4 - Partnerships in Priority S&T Areas/ 2nd National Plan for Research, Development & Innovation, research contract no 62-056/2008.

References

[1] Martin Karplus, J. Andrew McCammon, Nature Structural Biology **9**, 646 (2002).

[2] M. Falconi, M.E. Stroppolo, P. Cioni, G. Strambini, A. Sergi, M. Ferrario, A. Desideri Biophysical Journal, **80**, 2556 (2001).

[3] Zhiqiang Wang, James A. Lupo, Soumya Patnaik, Ruth Pachter, Computational and Theoretical Polymer Science, **11,** 375 (2001).

[4] Y. Okamoto, J. Mol. Graph. Model. **22,** 425 (2004).

[5] Richard J. Lawa, Charlotte Capener, Marc Baaden, Peter J. Bond, Jeff Campbell, George Patargias, Yalini Arinaminpathy, Mark S.P. Sansom, J. Mol. Graph. Model. **24,** 157 (2005).

[6] John A. Board Jr., Jeffrey W. Causey, James F. Leathrum Jr., Andreas Windemuth, Klaus Schulten**,** Chemical Physics Letters, **198,** 89 (1992).

[7] B. R. Brooks, M. Hodošček, Chem. Des. Autom. News **7**, 16 (1992).

[8] R. Murty, D. Okunbor, Parallel Comput. **25,** 217 (1999).

[9] James A. Lupo, Zhiqiang Wanga, Alan M. McKenneya, Ruth Pachtera, William Mattsonb, J. Mol. Graph. Model. 89 – 99 (2002).

[10] Jakub Kurzak, B. Montgomery Pettitt, Journal of Parallel and Distributed Computing, **65,** 870 (2005).

[11] Klavdija Kutnar, Urban Borštnik, Dragan Marušič, Dušanka Janežič, Journal of Mathematical Chemistry, **45,** 372 (2008).

[12] Y. Okamoto, J. Mol. Graph. Model. **22,** 425 (2004).

[13] M. Feig, J. Karanicolas, C.L. Brooks, J. Mol. Graph. Model. **22,** 377 (2004).

[14] R. Zhou, M. Eleftheriou, C.-C. Hon, R. S. Germain, A. K. Royyuru, B. J. Berne, Journal of Research and Development, **52,** 19 (2008).

[15] B. R. Brooks, R. E. Bruccoleri, B. D. Olafsen, D. J. States, S. Swaminathan, M. Karplus, J. Comput. Chem. **4,** 187 (1983).

[16] A. D. MacKerell., D. Bashford, M. Bellott, R. L. Dunbrak, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, I. W. E. Reiher, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, M. Karplus, J. Phys. Chem. B. **102**, 3586 (1998.

[17] K. Takano, K. Ogasahara, H. Kaneda, Y. Yamagata, S. Fujii, E. Kanaya, M. Kikuchi, M Oobatake, K Yutani, J Mol Biol. **254,** 62 (1995).

[18] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, P. E. Bourne, The Protein Data Bank, Nucleic Acids Research, **28**, 235 (2000).

[19] J. P. Rykaert, G. Ciccotti, H. J. C. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[20] S. J. Plimpton, B. A. Hendrickson, J. Comput. Chem. **17,** 326 (1996).

[21] S. J. Plimpton, R. Pollock, M. Stevens, Particle-Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations, in Proc. of Eighth SIAM Conf. on Parallel Processing for Scientific Computing, Minneapolis, MN, March 1997.

[22] S. J. Plimpton, J. Comput. Phys. **117**, 1 (1995).

[23] W. Humphrey, A. Dalke, K. Schulten, J. Molec. Graphics. **14**, 33 (1996).

[24] N. Guex, M.C. Peitsch, Electrophoresis **18,** 2714 (1997).

Corresponding author:* marian _butu@yahoo.com