CROATICA CHEMICA ACTA
CCACAA **78** (2) 211–216 (2005)

ISSN-0011-1643
CCA-3000
Original Scientific Paper

Fast Parallel Molecular Simulations*

Urban Borštnik, Milan Hodošček, and Dušanka Janežič**

National Institute of Chemistry, Hajdrihova 19, SI-1000 Ljubljana, Slovenia

RECEIVED FEBRUARY 8, 2005; REVISED FEBRUARY 28, 2005; ACCEPTED MARCH 1, 2005

Keywords
parallel computing
molecular simulations
molecular dynamics

We have developed and built several clusters of Personal Computers (PCs) that we use to perform parallel molecular simulations of chemically, physically, and biologically relevant systems. We describe the distinguishing networking topology of our clusters that enable them to perform classical and quantum mechanical computer simulations faster than standard PC clusters. Several techniques that we have used in parallelizing simulation programs are described. We employed these clusters for simulations of several different molecular systems. Also the computational performance of these simulations on our PC clusters is presented.

INTRODUCTION

Computer simulations are an important tool in studying physical, chemical, and biological molecular systems. Since these molecular simulations are typically large and therefore computationally intensive, improvements in either the computers used to perform simulations or in the simulation algorithms enable enhanced simulation capability.[1] Longer simulations or the treatment of larger systems become possible. Systems of interest are very large or require a long time period of the simulation. Examples include protein folding, receptor recognition, and enzyme catalysis.[2–4]

Single computer processors used for simulations are continually increasing in speed, allowing faster simulations to be performed. A complementary approach to increasing simulation speed is parallel computation, achieved by distributing the computational work on several connected processers so that the processors each perform only a part of the whole simulation and so the work is completed in less time.[5] Clusters of personal computers (PCs) are an effective way to decrease the time needed for molecular simulations. PC clusters were first developed as an affordable and easily attainable alternative to traditional supercomputers.[6] PC clusters have distributed memory, meaning that each processor sees only a part of all the memory in the whole cluster. To transfer data among the processors, message-passing must be used, meaning that a parallel program explicitly sends and receives data over processor interconnections.[7] Since their inception in the mid 1990s, the focus of research in clustering has been on improving the connections among the processors of the cluster with higher data transfer speeds and lower transfer latency.[6,8] The less time that is spent for transferring data, the faster a parallel program may run, therefore fast connections and low latency among processors are critical to achieving good parallel performance. Along with the changing nature of PCs and PC networking, computer simulation programs must be adapted to the new technologies in order to perform optimally on parallel PC clusters.[9,10]

* Dedicated to Dr. Edward C. Kirby in happy celebration of his 70th birthday.
** Author to whom correspondence should be addressed. (E-mail: dusa@cmm.ki.si)

Molecular simulations are very computationally demanding. When using a standard empirical force field, calculating non-bonded interactions among atoms of the molecular system is the most computationally demanding part of the simulation. Any speed improvements in the algorithms for calculating the Hamiltonian of a molecular system therefore enable faster molecular simulations.[11–13]

Molecular dynamics (MD) simulation consists of numerically integrating Newtonian equations of motion. The simulation consists of a number of time steps consisting of two parts. One is the calculation of the Hamiltonian, then in the second part, the forces derived from the Hamiltonian are used in dynamics calculations to compute the new atomic coordinates. In each step of the MD simulation, the Hamiltonian must be calculated.

For optimizing the geometry of a molecular system, the geometry with the lowest energy must be found. A procedure for geometry optimization performs a minimization of the Hamiltonian. Therefore, the energy of many different geometries must be computed before a geometry with the lowest energy found is declared to be the optimal geometry with the minimal energy. For each optimization step the forces acting on the atoms must be calculated. One step of geometry optimization is therefore equivalent to one step of molecular dynamics (MD). In some recent approaches geometry optimization is replaced by MD at very low temperatures.[14]

With classical mechanics, the Hamiltonian encompasses terms describing only the classical interaction among atoms in a molecular system, such as the defined bonds between atoms or electrostatic charges among atoms. Classical mechanics may be augmented with quantum mechanics, which allows for a much greater variety of effects to be studied.[15–17] The principles of geometry optimization and MD simulation remain the same, but with an expanded Hamiltonian that includes a quantum mechanical description of the system or a part of it. In this regard, QM/MM means an improvement of the force field rather than a quantum dynamical treatment of the motion of nuclei.[18]

In this paper we describe the PC clusters that we have developed and assembled and discuss how their unique networking ties in with the improvements in the computer simulation programs. We also present the computer simulation performance of several molecular systems on the PC clusters.

METHODS

Parallel algorithms for molecular simulations were developed to take advantage of the faster calculation speed offered by parallel computers. Since the evaluation of the Hamiltonian of a system is the most computationally intensive part of molecular simulations, it is the focus of parallelization efforts.

The Hamiltonian of a molecular system consists of many contributions that must all be calculated. The most computationally-intensive terms are the terms for non-bonded interactions, which are electrostatic and van der Waals interactions. The computing time required for calculating these interactions is $O\ (n^2)$ for a system of n atoms since the interactions among all pairs of atoms in the system must be calculated. For a system of many molecules, the time required for calculating the non-bonded interactions far outweighs all the others that have a time complexity of at most $O(n)$. Therefore, the focus of improving parallel algorithms is on the non-bonded interactions.

A standard parallel algorithm for molecular dynamics simulation is shown in Figure 1. Each processor performs calculations for only the atoms assigned to it, but it needs the forces and positions of all the other atoms. Therefore in every time step of the simulation, two global data updates among all processors are performed so that all processors have the same data: once to update the forces acting on the atoms and once to update the positions of all the atoms. For efficient parallelization, the computational workload of the processors must be balanced, or else computational time is wasted waiting for an overloaded processor to finish its computations.

The method by which the calculation of non-bonded interactions is distributed among the processors is a significant factor in determining the speed of the subsequent data transfer. A careful distribution, paired with an appropriate algorithm for sharing the resulting forces, enables faster transfers than an *ad hoc* distribution, providing a noticeable speed increase of the simulation.[9] We have used the properties of this type of data transfer in the design of the network for PC clusters to achieve markable speed increases over standard PC cluster networks.[10]

The topology by which the parallel processors are connected determines the method by which the data are distributed. In a ring or 2-dimensional mesh topology, every processor is generally connected to two (ring) or four (mesh) others. A 2-D mesh is depicted in Figure 2. Processors can

```
Set initial coordinates and velocities
while not end of simulation do
    if (atoms have moved more than threshold) then
        determine myatoms, myatompairs
    forall myatoms
        calculate bonded forces
    forall myatompairs
        calculate non-bonded forces
    Distribute forces among all processors
    forall myatoms
        calculate new positions
    Distribute new positions among all processors
```

Figure 1. Overview of a standard parallel molecular dynamics algorithm.

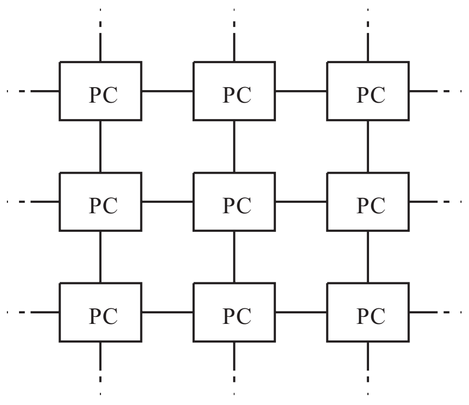

Figure 2. A 2-dimensional mesh topology. Every PC is connected to four others. The connections at the edges of the mesh are usually warped (wrapped around): for example, the three PCs at the right edge would be connected to the three at the left edge, and the three PCs at the top and bottom would also be connected.

communicate only with their immediate neighbors. To transfer data to any of the remaining processors, the data from one processor must be passed along a path of adjacent neighbors to the receiving processor. To distribute all of the data on every processor among all the processers, an order of $O(n)$, where n is the number of processors, steps of transferring data are needed.[19,20] In better connected topologies, such as hypercubes (an example is shown in Figure 3), the number of steps is on the order of $O(\log n)$.[21]

We have paired an algorithm for distributing data on a hypercube to a hypercube network. Because the algorithm transfers different amounts of data during each of the log n steps, we have used connection technologies with different speeds to create the hypercube topology. In this way, the amount of data transfered over the connections is matched to the speed of the connection, resulting in an distribution of data this is balanced according to data size and link speed.[10]

At the National Institute of Chemistry we have built several different CROW (Columns and Rows of Workstations) clusters of PCs. The CROW1 cluster consisted of four

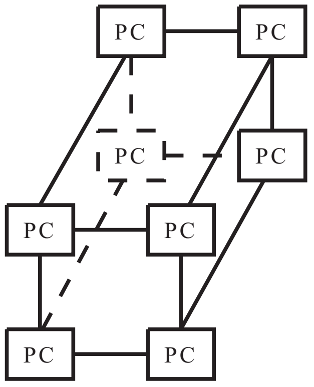

Figure 3. A 3-dimensional hypercube topology for eight PCs. Every PC has as many connections as there are dimensions in the hypercube (three in this example).

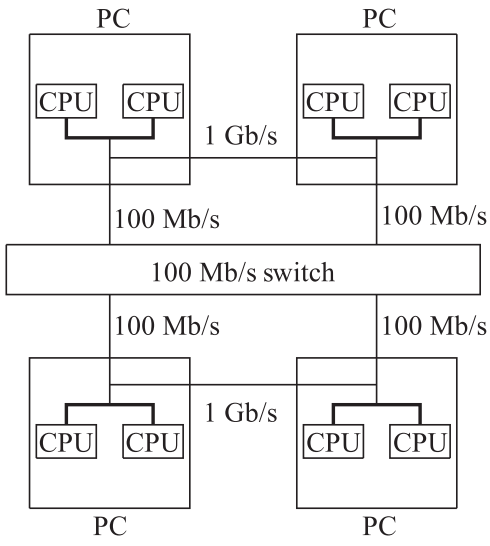

Figure 4. Schematic representation of a group of four PCs (out of 22) in the CROW5 cluster showing the different network speeds used in connecting eight processors. Thicker lines represent faster connections for data transfer. The fastest is the system bus between two processors, a slow 1 Gb/s network connects pairs of computers, and an even slower 100 Mb/s switch is used to connect all the PCs.

dual-processor Intel Pentium II 400 MHz computers connected in a ring topology using 100 Mb/s Ethernet direct connections between the computers. The CROW2 cluster, using 15 single Pentium II 450 MHz computers, was connected in a two-dimensional mesh with 100 Mb/s Ethernet, and the CROW3 cluster, with 32 Intel 466 MHz Celeron single-processor computers, was connected with a torus topology, also with 100 Mb/s Ethernet. The CROW4 cluster consisted of 64 700 MHz Athlon PCs connected in a hypercube topology with a 100 Mb/s Ethernet network. The CROW5 cluster, consisting of 22 AMD MP-1600+ dual processor computers, is connected in a hierarchical hypercube topology with mixed 100 Mb/s and 1 Gb/s Ethernet. Direct connections between pairs of PCs are 1 Gb/s and every PC is connected to a 100 Mb/s switch. The CROW8 cluster, consisting of 56 AMD MP-2200+ dual-processor PCs, is also connected in a hierarchical hypercube topology, but uses only 1 Gb/s Ethernet networking for both the direct connections and the swith. The hierarchical hypercube topology illustrated in Figure 4 is a hypercube (see Figure 3) in which the dimensions of the hypercube are mapped to connections of different speeds. For example, all the data transfer occuring through one dimension is physically transferred through the system bus in a PC, while the data transferred through another dimension is physically transferred through the connections between PCs (the two 1 Gb/s connections in Figure 4).

RESULTS

We have used the PC clusters that we have built to perform computer simulations of three different molecular systems. To demonstrate the improvements in our clusters' topologies and simulation programs, we have measured the computational time for these three different simulations on our clusters using various numbers of processors.

First, a molecular geometry optimization by QM/MM energy minimization was used to study the interaction differences among three different conformers of netropsin binding to the Dickerson-Drew DNA dodecamer[22] as shown in Figure 5. The DNA and the surrounding water was treated with molecular mechanics, while the netropsin conformers were treated quantum mechanically.[23]

Second, a quantum mechanical DFT energy minimization and further analysis of four different type of single-walled carbon nanotubes of varying lengths was used to study their electronic structure.[24] The studied nanotubes were of four types: armchair (3,3) and (7,7) and zigzag (6,0) and (12,0);[25] a zigzag (6,0) nanotube is shown in Figure 6. All types were open-ended with terminating hydrogen atoms. The calculations were performed in vacuo.[26]

Third, molecular dynamics of systems of liquid bulk water were used to study various properties of water, such as the vibrational spectra.[27] The Split Integration Symplectic Method (SISM), which treats high-frequency motion analytically, was used for these simulations.[28,29] A cube of water prior to equilibration is shown in Figure 7.

The speedups of the parallel computer simulations of the studied molecular systems on the CROW5 and CROW8 clusters with different numbers of processors and different configurations are presented in Table I. The speedup of running on multiple processors is defined as the ratio between the simulation time on one processor and the simulation time on parallel processors. A higher speedup is better, the ideal being a linear speedup, when the speedup is equal to the number of processors used. Both of the CROW5 and CROW8 clusters scale similarly.

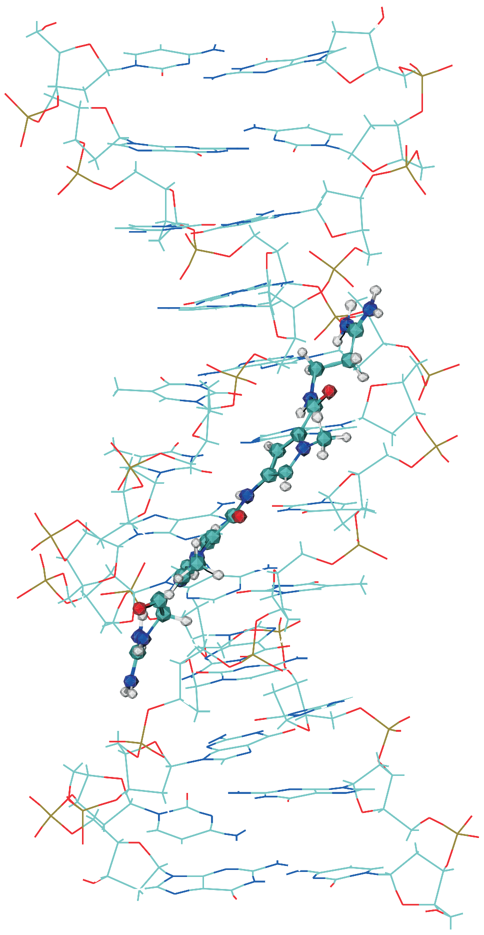
Figure 5. Netropsin bound to the minor groove of DNA, shown with all of the water molecules removed.

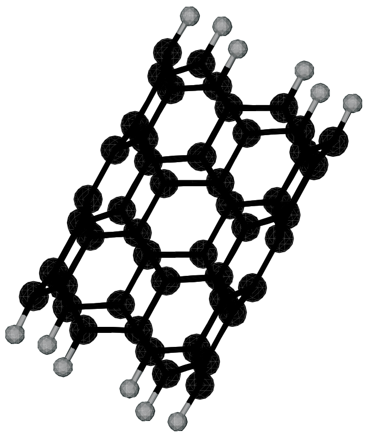
Figure 6. Zig-zag (6,0) single-walled open-ended nanotube with terminating hydrogen atoms. Black circles are carbon atoms and gray circles are hydrogen atoms.

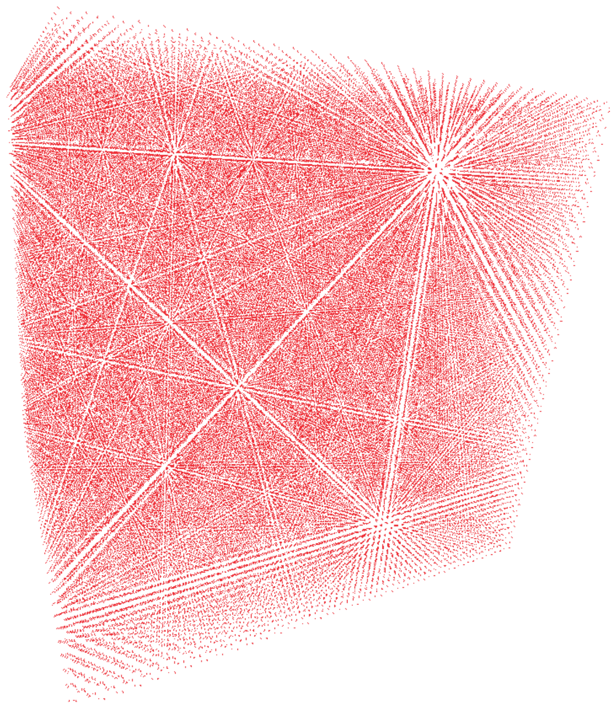
Figure 7. Structure of a periodic box of approximately one million water molecules prior to equilibration.

TABLE I. Speedup of computer simulations on PC clusters with 2, 4, and 8 processors of a QM/MM geometry optimization of a netropsin ligand in the minor DNA groove, a QM-DFT geometry optimization of a carbon nanotube, and an MD simulation of bulk water

CPUs	QM/MM optimization		QM-DFT optimization		MD simulation	
	CROW5	CROW8	CROW5	CROW8	CROW5	CROW8
2 (1x2)	1.87	1.86	1.86	1.86	1.88	1.89
2 (2x1)	1.88	1.88	1.86	1.86	1.90	1.90
4 (2x2)	3.43	3.40	3.32	3.36	3.68	3.72
4 (4x1)	3.50	3.42	3.33	3.38		
8 (4x2)	5.67	5.43	5.71	5.53		

Using single processors communicating over a network rather than a dual-processors configuration is shown to be advatangeous. The speed gained by a faster network switch used in the CROW8 cluster results in a greater speedup.

CONCLUSIONS

In this article we describe the application of various clusters of PCs to the parallel simulation of different molecular systems. Clusters of Personal Computers are an effective way to increase the speed of molecular simulations. The type of network connections used in a PC cluster is crucial in obtaining good performance when many processors are used for parallel simulations. We have shown that it is possible to build clusters that are suitable for a variety of simulation types. Together with novel algorithms, larger and longer molecular simulations will become feasible. In the future we will apply the SISM methodology for solving the time-dependent Schroedinger equation.[30,31]

Acknowledgements. – The financial support of the Ministry of Higher Education, Science, and Technology of Slovenia under grant numbers P1-0002 and J1-6331 is greatly appreciated.

REFERENCES

1. W. F. van Gunsteren and H. J. C. Berendsen, *Angew. Chem., Int. Ed.* **29** (1990) 992–1023.
2. T. Hansson, C. Oostenbrink, and W. F. van Gunsteren, *Curr. Opin. Struct. Bio.* **12** (2002) 190–196.
3. M. Garcia-Viloca, J. Gao, M. Karplus, and D. G. Truhlar, *Science* **303** (2004) 186–195.
4. W. L. Jorgensen, *Science* **303** (2004) 1812–1818.
5. D. W. Heermann and A. N. Burkitt, *Parallel Algorithms in Computational Science*, Springer-Verlag, Berlin, 1991.
6. T. Sterling, D. J. Becker, D. Savarese, J. E. Dorband, U. A. Ranawake, and C. V. Packer, *Proceedings of the 24th International Conference on Parallel Processing*, Oconomowoc, WI (1995), I:11–14.
7. J. L. Hennessy and D. A. Patterson, *Computer Architecture – A Quantitative Approach*, Morgan Kaufmann Publishers, San Mateo, California, 2nd edition, 1990.
8. C. Reschke, T. Sterling, D. Ridge, D. Savarese, D. J. Becker, and P. Merkey, *Proceedings of the Fifth IEEE Symposium on High Performance Distributed Computing*, Syracuse, NY, 1996, pp. 626–636.
9. M. Hodošček, U. Borštnik, and D. Janežič, *Cell. Mol. Bio. Lett*. **7** (2002) 118–119.
10. U. Borštnik, M. Hodošček, and D. Janežič, *J. Chem. Inf. Comput. Sci*. **44** (2004) 359–364.
11. B. R. Brooks, D. Janežič, and M. Karplus, *J. Comp. Chem*. **16** (1995) 1522–1542.
12. D. Janežič and B. R. Brooks, *J. Comp. Chem*. **16** (1995) 1543–1553.
13. D. Janežič, R. M. Venable, and B. R. Brooks, *J. Comp. Chem*. **16** (1995) 1554–1566.
14. F. S. Lee, Z. T. Chu, and A. Warshel, *J. Comp. Chem*. **14** (1993) 161–185.
15. J. Mavri, H. J. C. Berendsen, and W. F. van Gunsteren, *J. Phys. Chem*. **103** (1993) 13469–13476.
16. J. Mavri and H. J. C. Berendsen, *J. Phys. Chem*. **99** (1995) 12711–12717.
17. J. Mavri and J. Grdadolnik, *J. Phys. Chem. A* **105** (2001) 2045–2051.
18. K. P. Eurenius, D. C. Chatfield, B. R. Brooks, and M. Hodošček, *Int. J. Quant. Chem*. **60** (1996) 1189–1200.
19. R. Trobec, M. Šterk, M. Praprotnik, and D. Janežič, *Int. J. Quant. Chem*. **84** (2001) 23–31.
20. R. Trobec, M. Šterk, M. Praprotnik, and D. Janežič, *Int. J. Quant. Chem*. **95** (2004) 530–536.
21. R. A. van de Geijn, *LAPACK Working Note 29 on Global Combine Operations,* Technical Report CS-91-129, University of Tennessee, April 1991.
22. R. Wing, H. Drew, T. Takano, C. Broka, S. Tanaka, and K. Itakura, *Nature* **287** (1980) 755–758.
23. J. Dolenc, U. Borštnik, M. Hodošček, J. Koller, and D. Janežič, *Theochem-J. Mol. Struct*. (2005). In print.
24. I. Lukovits and D. Janežič, *J. Chem. Inf. Comput. Sci*. **44** (2004) 410–414.
25. S. Iijima, *Nature* **354** (1991) 56–58.
26. U. Borštnik, M. Hodošček, D. Janežič, and I. Lukovits, *Electronic Structure of Zig-zag and Armchair Nanotubes Obtained by Density Functional Calculations*. To be published.
27. M. Praprotnik, D. Janežič, and J. Mavri, *J. Phys. Chem. A* **108** (2004) 11056–11062.

28. D. Janežič and M. Praprotnik, *Int. J. Quantum Chem.* **84** (2001) 2–12.
29. D. Janežič and M. Praprotnik, *J. Chem. Inf. Comput. Sci.* **43** (2003) 1922–1927.
30. S. R. Billeter and W. F. van Gunsteren, *Mol. Simul.* **15** (1995) 301–322.
31. M. F. Lensink, J. Mavri, and H. J. C. Berendsen, *J. Comp. Chem.* **17** (1996) 1287–1295.

SAŽETAK

Brze paralelne molekularne simulacije

Urban Borštnik, Milan Hodoščcek i Dušanka Janežič

Razvijeno je i izgrađeno par grozdova osobnih računala (PC) za izvođenje paralelnih molekularnih simulacija raznih kemijskih, fizikalnih i biološki relevantnih sustava. Osebujna topologija umrežavanja ovih grozdova je, u odnosu na standardne PC grozdove, u stanju brže izvoditi klasične i kvantnomehaničke simulacije. Opisano je više tehnika za paraleliziranje simulacijskih programa koji su zatim primjenjeni na niz molekularnih sustava. Diskutirana je također računalna učinkovitost simulacija na predloženim PC grozdovima.