International Journal of Computer Applications (0975 – 8887)
Volume 181 – No. 50, April 2019

Applying Parallel Design Patterns on Molecular Dynamics Simulation

Nilesh Maltare
Government Engineering College
Shamlaji Highway, Modasa
Gujarat, India

Vithal N. Kamat, PhD
Centre for Apparent Energy Research,
Vithal Udyognagar, Gujarat, India

ABSTRACT
Molecular dynamics simulates behavior of atoms and molecules. Molecular dynamics Simulation demonstrates and derives macroscopic properties by atomic interactions.

In this experiment we have redesigned Molecular Dynamics (MD) Simulation by applying different parallel design pattern. Redesigned MD Simulation focuses on adaptability to different architectures and scalability to use with large number of atoms and long duration simulation. The paper demonstrates performance of MD on different architectures.

General Terms
Performance, Design, Parallelization.

Keywords
Parallel Programming, Patterns, Threads.

1. INTRODUCTION
The Molecular dynamics (MD) simulation helps in understanding molecular structure and microscopic interactions between them. With MD simulation we can get benefit like adjustment of length and time. It can be repeated multiple times and predicts multiple properties. MD Simulation has generally been a low throughput method. Recently with the advancement in computing, it is complex molecular interaction efficiently.

The classical MD Simulation is still popular that calculates energy, forces, velocity, acceleration, spatial coordinate of atoms. In this paper, we are experimenting with classical MD Simulation. We are comparing different Parallel implementation of MD Simulation. The objective is to evaluate MD Simulation program applied with parallel design pattern on different parallel environment (MPI, OpenMP) and hybrid (OpenMP-MPI) architectures and to demonstrate pattern based design for flexibility and adaptability.

Molecular dynamics simulation started at 1960s [1], at that time they developed to simulate 100 of atoms. MD Simulation is continuously evolved [2] [3] and experimented. It has wide applications [4] in materials science and nanotechnology.

MD simulation evaluates molecular properties on short time scale for different parameters. At present it is common to simulate systems containing 5000 to 100000 atoms. MD simulations are continuously improving by taking benefit of advancement in High Performance Computing (HPC). It is possible to predict interaction between molecules with certain degree of precision. MD Simulations can be performed for multi-microsecond scale. MD Simulation algorithms are also improved to get benefit of parallelism.

Parallel Programming is also evolving [6][7] and many constructs are available for programmers to achieve parallelism in application, still we face difficulty in parallel programming. The parallel programming is leveraging advancement in software engineering.

2. DESIGN PATTERNS FOR PARALLEL SOFTWARE DEVELOPMENT
Design Patterns for object oriented design of software is proposed in [8]. Design Pattern captures objects, their roles, interaction and collaboration among each other. Design Pattern shares common terminology and raises level of abstraction. Design patterns are proven solutions to commonly occurring problems within a given context in software design. Design Pattern captures well-proven experience in software development and help to promote good design practise. Patterns for software architecture [9] discuss about architectural patterns, design patterns, pattern selection and their use in software.

Every pattern deals with a specific, recurring problem in the design or implementation of a software system. Patterns can be used to construct software architectures with specific properties. Pattern shares common terminology which increase better communication and reduces ambiguity. There exist many literature related to parallel software design [12] [13] [14] [15]. Design patterns applied in parallel programming problem can be solved through generalized reusable solution. Parallel Design patterns identify structural parallelism for application in given context and suggest reusable solution. In [16] Generative design patterns generate parallel framework or skeleton for parallel application writing. CO2P3S generates skeleton. One benefit of skeleton generated with it is implementation independent. Use of Patterns to exploit parallelism is not new. Design Pattern for network and distributed software was proposed by Siu and Singh [10]. Patterns for Parallel Programming demonstrated in book by Matson [12]. Reusable architectural skeleton for parallel applications was also suggested in [13].

In Architecture with multicores, we are finding reduction in communication latency but for scalability we may have to go for cluster of multicores. That is the reason we need to design application to exploit parallelism by use of structural and communication patterns. The pattern based design enhances flexibility, scalability and adaptability. Novice programmers can develop quality application by following best practices captured with design patterns.

3. PARALLEL DESIGN PATTERN FOR MD DESIGN
Parallel Design Patterns categorized for Implementation are:

1. Master Worker

A master process or thread set up a pool of worker processes of threads and a bag of tasks. The workers execute concurrently, with each worker repeatedly removing a tasks from the bag of the tasks. It is suitable for embarrassingly parallel problems where concurrency is easy to model.

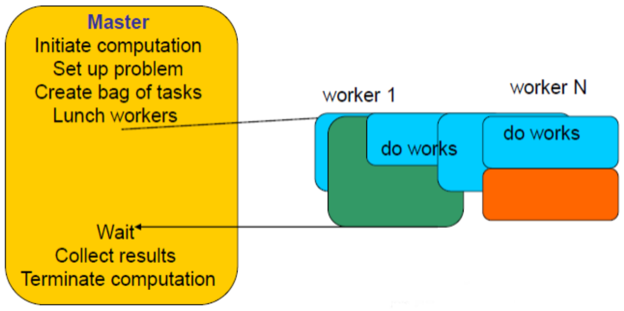

Fig 1: Master Worker

2. SPMD

SPMD (Single Program Multiple Data) is particularly suited to problems in where decomposition is on data decomposition. A key issue in applying this pattern is dividing and mapping original problem's data among threads, this will be handled by considering nature of problem.

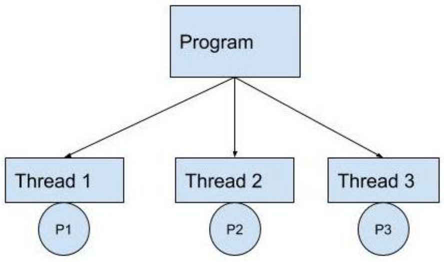

Fig 2: SPMD

Master Worker is suitable for programming in MPI while SPMD can be easily mapped with OpenMP and MPI environment. Benefit of MPI implementation is scalability but it may increase communication time and time in setting up MPI environment. OpenMP is easier in programming and debugging. Serial code can be easily transformed to parallel version with OpenMP. OpenMP can be run only on shared memory computers.

Sequential applications may be transformed into efficient parallel applications by using Parallel Design Patterns. Selection of parallel patterns depends upon problem, dependency, parallel programming environment and architecture. In [19] Automatic parallel pattern detection for algorithmic space is proposed. These types of tools are very limited in their functionality. Task Parallelism is covered and few patterns like Pipeline and Master Worker are detected. In design of parallel application patterns are available for identifying concurrency, communication structure, data structure and implementation mechanism. It is also important to understanding mapping between these. Some patterns are suitable in mapping between concurrency and implementation space [20]

4. MD SIMULATION PARALLEL DESIGN PATTERN BASED SOLUTION

MD is computational intensive application and its complexity grows with number of atoms, range of forces considered, step size and other parameters. MD simulation is parallelized earlier in many ways [17] [18]. MD simulations are performed from small laboratories to world's fastest supercomputers

The MD Simulation uses data structures such as positions and velocities of particles and the computing procedures such as calculations of forces and updates of particle positions are partitioned and allotted to each processor.

Important metrics will be time spent in communication, efficiency, and utilization of processors. Time spent in communication should minimize to increase parallel efficiency. Also, in the case that communications among processors are synchronous, load imbalance, in which one processor with the job done has to wait other processors to finish their jobs, deteriorates the parallel efficiency. There may be different methods of partitioning the like particle and spatial decomposition method which may affect parallelization approaches [11].

In this paper we are considering adaptability of MD Simulation program design. If MD Simulation solution will adopt variety of architecture (shared, distributed, hybrid), It will be useful in different situation and in similar way computation intensive problems can be designed to adopt different architectures. We are also experimented with hybrid programming which utilizes cores within node first to reduce latency and prefer multithreading. While in pure MPI approach scalability is achieved but multi process creation and Inter Process Communication (IPC) can be costly. Hybrid approach takes advantages of both OpenMP and MPI.

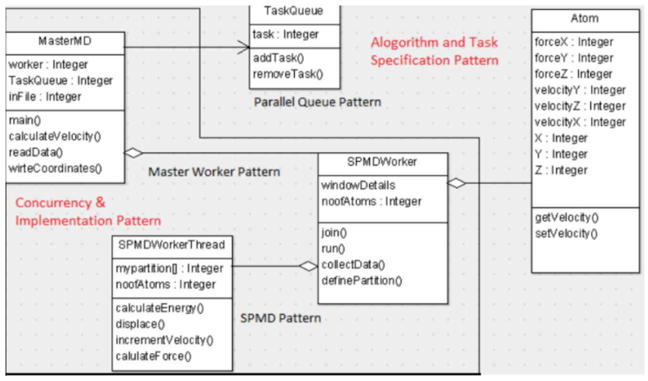

Fig 3: Parallel Design Pattern Based MD Simulation

There are two important objectives for Pattern based approach of MD Simulation:

1. Ease of Programming
2. Adaptability of solution to different architectures

With pattern based solution we will get separation of concern, improved readability and debugging. At the top we separate concurrency aspect with task specification. Task specification should be independent from programming language used. In Concurrency Pattern, we specify concurrency with implementation strategy.

5. RESULTS

In this experiment we are considering forces in three dimensionality of the physical space. The number of atoms in the simulation is considered is 1000 and 16 time steps are evaluated.

We are using following information to compute potential energy and forces:

- Velocity
- Position

- Charge
- Acceleration

To run simulation, we are initializing some parameters with random inputs. We partition the problem the simulation space is divided into rectangular cells and calculate each region in parallel. We are only considering short range forces and if atom goes outside range it will move in respective region. With cell partitioning method, the pair list for an atom in a particular cell is constructed, which contains atom and neighbouring atom within cell. The computational cost for this method becomes also proportional to the number of total atoms.

We compared execution time in seconds in different cases. In Pure OpenMP, we redesign MD simulation by applying SPMD. We have used Master-Worker for writing solution in Pure MPI. In hybrid programming approach we have used composite pattern which uses Master Worker and SPMD to write adaptive solution.

Table1: Execution time in seconds

No. of Process /Threads	Pure OpenMP	Pure MPI	Hybrid MPI-Open MP (Best Case)
1	179.3	218.7	219.53
2	91.9	116.2	109.5
4	51.3	62.8	54.6
8	31.5	32.9	31.9
16	16.3	21.83	16.6
32	16.7	11.33	9.1
64	16.8	5.46	5.1
128	17.1	5.98	3.4

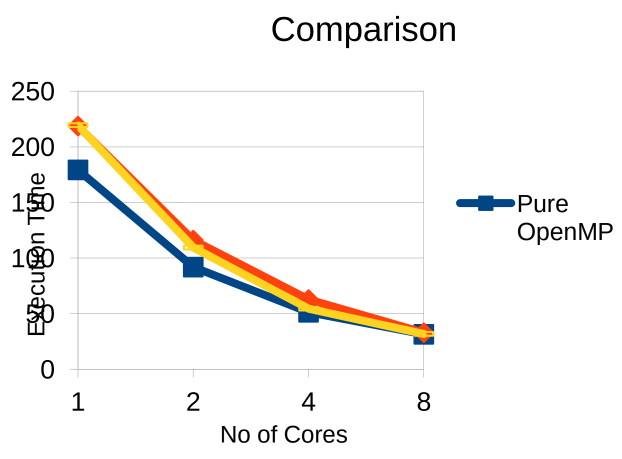

Fig 4: Comparison of Performance up to 8 Cores

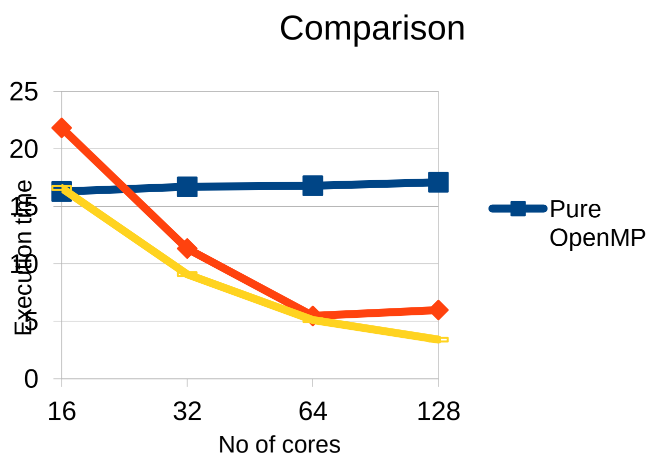

Fig 5: Comparison of Performance up to 128 Cores (Open MP runs on 16 core processor)

6. CONCLUSION

MD simulation is important and wide range of applications. MD Simulations are time consuming and require high end computing facility. The solution for MD simulation should adapt different architectures and performed at different computation size. We can observe that MPI solution is scalable and performs better in large number of processors while Open MP solution is better in when number of process is small that may be due to processor architecture we are using in which we are having only 16 cores per processors. Hybrid programming (Open MP -MPI) combines benefit of both. Hybrid solution can work in different architectures and also scalable.

7. LIMITATIONS AND FUTURE WORK

We have experimented With Shared Memory Address Architecture, Distributed Shared-memory Architectures. The number of cores per processor is limited (up to 16) for experiments. Commonly available shared memory multiprocessors have limited cores. Open MP solution may not be scalable. The hybrid approach needs to be tested for heterogeneous architectures with large number of cores. The parameters calculated are very primitive and step size of simulation is very less.

In future we need to experiment with variety of architectures. We also are looking for adaptive design which changes based on hardware reflections.

8. REFERENCES

[1] A Rahman, "Correlations in the motion of atoms in liquid argon", Physical Review, 1964 – APS.

[2] W. G. Hoover, A. J. C. Ladd, and V. N. Hoover," Historical Development and Recent Applications of Molecular Dynamics Simulation", Advances in Chemistry Series 1983.

[3] Raymond Kapral, Giovanni Ciccotti, "Molecular dynamics: an account of its evolution", Theory and Applications of Computational Chemistry Elsevier 2005

[4] Adam Hospital, Josep Ramon Goni, Modesto Orozco, Josep L Gelpi, "Molecular dynamics simulations: Advances and applications", Advances and Applications in Bioinformatics and Chemistry, November 2015
DOI: 10.2147/AABC.S70333

[5] K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz,N. Morgan, D. Patterson, K. Sen,

J. Wawrzynek, et al. "A view of the parallel computing landscape. Communications of the ACM", 52(10):56–67, 2009.

[6] Culler, D. E., Singh, J. P., & Gupta, A. (1999). Parallel computer architecture: a hardware/software approach. San Francisco, CA: Morgan Kaufman Publishers, Inc

[7] Diaz, J., Muoz-Caro, C., & Nio, A. (2012). A survey of parallel programming models and tools in the multi and many-core era. IEEE Transactions on Parallel and Distributed Systems, 23(8), 1369-1386. doi:10.1109/TPDS.2011.308

[8] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. "Design patterns: Abstraction and reuse of object-oriented design", ECOOP1993 pages: 406-431, Springer-Verlag.

[9] Frank Buschmann et al., "Pattern-Oriented Software Architecture - A System of Patterns" , Wiley, 1996.

[10] S. Siu, A. Singh,"Design patterns for parallel computing using a network of processors" Proceedings. The Sixth IEEE International Symposium on High Performance Distributed Computing, DOI: 10.1109/HPDC.1997.626434

[11] D. M. Beazley, "Parallel Algorithm for Short-Range Molecular Dynamics", World Scientific's Annual Reviews in Computational Physics, 3, 119 (1995).

[12] Timothy G. Mattson,Beverly A. Sanders, Berna L. Massingill. "Patterns for Parallel Programming", 2005 ISBN-10: 0321228111, ISBN-13: 9780321228116, Addison-Wesley Professional.

[13] D. Goswami, A. Singh, and B. Priess. "Architectural skeletons: The reusable building-blocks for parallel applications". In Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applciations (PDPTA'99), pages 1250–1256, 1999..

[14] S. MacDonald, D. Szafron, J. Schaeffer, and S. Bromling. "From patterns to frameworks to parallel programs". Journal of Parallel and Distributed Computing, 2001. .

[15] J.L. Ortega,"Arjona Design Patterns for Communication Components", Proceedings of the 12th European Conference on Pattern Languages of Programming and Computing (EuroPLoP2007), Kloster Irsee, Germany, 2007

[16] J. Anvik, J. Schaeffer, D. Szafron, K. Tan. "Asserting the utility of CO2P3S using the Cowichan Problem Set", Journal of Parallel and Distributed Computing, Volume 65 Issue 12, December 2005.

[17] S. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics," J. Computational Physics, vol. 117, no-1, 1995, pp. 1–19.

[18] K.B. Tarmyshov and F. Muller-Plathe, "Parallelizing a Molecular Dynamics Algorithm on a Multiprocessor Workstation Using OpenMP," J. Chemical Information and Modeling, vol. 45, no. 6, 2005, pp. 1943–1952.

[19] Zia Ul Huda, Rohit Atre, Ali Jannesari and Felix Wolf,"Automatic Parallel Pattern Detection in the Algorithm Structure Design Space", IEEE International Parallel and Distributed Processing Symposium 2016, doi:10.1109/ipdps.2016.60.

[20] Nilesh Maltare, Chetan Chudasama, "Applying parallel design patterns to embarassingly parallel problem", 2016 Symposium on Colossal Data Analysis and Networking (CDAN), 2016.