Performance of LAMMPS on Intel Xeon Phi for Composite Material Analysis

Donthireddy. Prasanthi
M. Tech (High Performance Computing Solutions)
Vel Tech Dr. RR & Dr. SR Technical University
Chennai, Tamil Nadu, India

Dr. Vikas Kumar
CDAC, Pune, India

Pallavi Gavali
CDAC, Pune, India

Abstract

High Performance Computing is the use of parallel processing for running applications reliably, quickly. The composite material analysis is one of the examples of High Performance Computing (HPC). LAMMPS is very useful to run the composite material analysis on many core architectures like Intel Xeon phi to achieve the performance. To accelerate computing, parallelization can be used which can be implemented by using Message Passing Interface (MPI) and on many core architectures like Intel Xeon Phi co-processor. MPI is used to exploit core-level parallelism. Composite material simulation at molecular level is highly needed and recommended.
Keywords: LAMMPS, Composite Material, Intel Xeon Phi, High Performance Computing, MPI, Intel Xeon

I. INTRODUCTION

From past decades, composite material analysis has more computational part. To run the large computational part with a single processor and with a single core, it is very difficult to achieve performance. To achieve performance we are using the high performance computing. In high performance computing, large applications can run with the help of parallel processing. In computers, parallel processing is the processing of instructions by dividing them among multiple processors with the objective of running a program in less time. So composite material simulation at molecular level is highly needed and recommended. LAMMPS is very useful software to run the composite material on many core architectures like Intel Xeon Phi to achieve performance.

Today supercomputer is the computer with good memory and speed. It's having high computational capacity. It is typically used for scientific and HPC applications to run the huge amount of data. Performance of supercomputer is measured in FLOPS (Floating Point Operations per Second).The modern applications of supercomputer are quantum mechanics, weather forecasting, climate research, molecular modeling and seismic data processing. We need to achieve performance with help of parallel programming, GPU programming. HPC has emerged due to increasing demands of processing speed. Composite material is the material of combining one (or) more materials to form a single material that is different from individual material. Each material having its own physical and chemical properties. So many composite materials like con-Crete, ceramic composites, fiber reinforced plastics, metal composites. LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) is the software to run the composite material.

Table – 1
INTEL XEON VS INTEL XEON PHI

Description	Intel Xeon	Intel Xeon Phi
ARCHITECTURE	Intel Xeon CPU E5-2670	5110P
CORES	8 Cores	60 Cores
THREADS PER CORE	1	4 h/w threads

II. PROBLEM FORMULATION AND SIMULATION

The problem was before, running the serial code it takes so much time to achieve performance. serial code is divides the program instructions and execute one after another in serial with one processor. but now-a-days parallel computing is running in present super computers. parallel computing means dividing the program instructions in to multiple processors. All the processor can run the instructions simultaneously. by using parallel programming, we can achieve performance.

The solution to achieve performance for composite material analysis, LAMMPS is the software to run the composite script on Intel Xeon Phi co-processor. Most parallel applications uses SIMD operations for the executing get best performance on Intel Xeon Phi.Co-processor will run the mathematical operations in the computer. Intel Xeon Phi is also known MIC (Many Integrated Core) architecture. Intel Xeon Phi will give better performance rather than normal Intel CPU. It is a highly parallel device provides up to 60 cores, 240 threads and 1.2

tera flops of performance. Each core is designed with vector processing unit. Three modes of execution in Intel Xeon Phi:
1) Offload mode
2) Native mode
3) Symmetric mode
    1) In offload mode, the computation part is their host transfer to the co-processor to run the computational part. But the host and co-processor will not run in parallel.
    2) In native mode, the application runs directly on the co-processor without offload of the host system. The executable is copied from host to co-processor then lo-gin to the MIC card and run the executable.
    3) In symmetric mode, the application processor runs on both host and the co-processor. They communicate with an MPI (Message Passing Interface). Today mainly super computers are using the MIC cards.

III. HPC REQUIREMENTS

I have done implementation in PARAM YUVA II.
1) Model name: Intel Xeon CPU E5-2670 @ 2.60GHz
2) Processors: 16
3) CPU cores: 8
4) Intel Xeon Phi: 5110P each with 60 cores, 4h/w threads
5) Threads per core: 1
6) L1 Cache: 32 KB
7) L2 Cache: 256KB
8) L3 Cache: 20MB
9) Compilers:icc, mpicc (mpich-4.1.0), impi (5.0.3)
10) Operating System: Linux

IV. METHODOLOGY

I used MPI programming.MPI is the message passing interface and it is a distributed memory model. Distributed memory is passing of messages between two systems through MPI.There is no memory contention. MPI is mainly used to run the parallel programs. The method of creating the input script for composite material in LAMMPS is:
1) Initialization
2) Atom definition.
3) Settings.
4) Run as a simulation
    1) In initialization, the atoms are read in a form of file relevant to units, dimensions, boundary, atom style, angle style.
    2) In atom definition, the atoms are created on lattice by using these commands are lattice, create_box, and create_atoms.
    3) 3. In settings, the atoms and molecular topology is defined. We need to set force-field coefficients like angle coefficients and pair coeff etc.
    4) In run as a simulation, whole molecular composite material simulation will run with the steps of 100 or 10,000.This will achieve performance.

V. IMPLEMENTATION AND RESULTS

I implemented on Param YuvaII.In LAMMPS software, USER-INTEL package is there. I checked the performance of normal Intel CPU and Intel Xeon Phi in offload mode.LAMMPS is built with both Intel and OMP (OpenMP) supporting.

A. *Running a job on HPC through PBS Scripting:*

PBS is a job scheduler. To run a job on HPC cluster, we will need to setup a Portable Batch System (PBS) file. In order to submit job on HPC cluster, we need to login to head nodes and submit a PBS job. The qsub is the command to submit a job in to the PBS queue and to request additional resources. The qstat is to check the status of the job.

B. *Creating a PBS Script:*

Fig. 1: PBS Scripting for Intel Xeon

Through qsub command we can submit a PBS job.

Fig. 2: PBS Job Submitting

To check the status of the job in the queue, qstat command

Fig 3. PBS Job Status

C. LAMMPS Results for Intel Xeon and Intel Xeon Phi
I have done execution in Offload mode for Intel Xeon Phi and Intel CPU.I compared the results between Intel Xeon and Intel Xeon Phi.

Problems	Processes	Performance results
in.chain	16	0.6secs and 42.7% utilization of CPU
in.lj	32	0.085secs and 99.9% utilization of CPU
in.eam	32	0.4secs and 99.6% utilization of CPU
in.rhodo	32	1.6 with 1 Co-processor

VI. CONCLUSION

Performance is very critical issue in past decades. Running the large code is very difficult to improve performance. The composite material analysis is highly needed and recommended at molecular level. Running the composite material code in HPC to achieve performance. LAMMPS is very useful to run the composite material analysis code to achieve more performance by using Intel Xeon Phi co-processor.

REFERENCES

[1] Andrew G. Salinger: "Kokkos implementation of Albany: A performance-portable finite element application", Sandia National Labs Technical Report, SAND2013-8430J, 2013.
[2] J.A. Anderson, C.D. Lorenz, and A. Travesset. "General PurposeMolecularDynamics Simulations Fully Implemented on Graphics Processing Units". Journal of Computational Physics, 227(10), 2008.
[3] Changjun Hu: "Efficient Parallel Implementation of Molecular Dynamics with Embedded Atom Method on Multi-core Platforms", 2009 international conference.
[4] Emmanuel N.Millan,Carlos S.Bederian,Eduardo M.Bringa:" Performance analysis of cellular automata HPC implementations",Journal of Computers and Electronic Engineering,Volume 48 Issue C,Novemeber 2015.
[5] Edwards,H.C.; SandiaNat.Labs:Kokkos:Enabling Performance Portability Across Manycore Architectures:IEEE,15-16 Aug.2013
[6] LAMMPS Molecular Dynamics: http://lammps.sandia.gov/