WHITE PAPER

LAMMPS on Advanced SGI® Architectures

Jia Liu*, PhD

Abstract

Molecular dynamics is a computer simulation technique to help in understanding the properties of assemblies of atoms and molecules in terms of their structure and microscopic interaction. LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) is a classical molecular dynamics code with widespread applications in materials research. At the atomic scale, LAMMPS can model soft (biomolecules, polymers) or solid-state (metals, semiconductors) materials. Using a more coarse-grained approach, the program can also model mesoscopic or even macroscopic systems as a collection of interacting particles.

This white paper discusses the LAMMPS (15 Jan 2010 version) performance on three types of SGI systems— the SGI® Rackable™ cluster, the SGI® Altix® ICE cluster and SGI® Altix® UV (Shared Memory Parallel) systems. In addition to presenting the performance results on these three systems, this paper attempts to characterize LAMMPS with profiling tools such as MPInside and hardware event counters. It also investigates the effect of Intel® Hyper-Threading Technology as well as different network topologies performance as it relates to running LAMMPS on advanced SGI® computer hardware systems.

*SGI Applications Engineering

TABLE OF CONTENTS

1.0 Introduction

Molecular dynamics (MD) is a computer simulation technique where the time evolution of a set of interacting atoms is followed by integrating their equations of motion. LAMMPS [1] is a classical molecular dynamics code, which follows the laws of classical mechanics (Newton Laws) to model the motion. Forces on atoms derived from the potential energies, which are represented by the MD force fields, are the most crucial part of the simulation. LAMMPS provides many different potentials for soft materials (biomolecular, polymers), solid-state materials (metals, semiconductors) and coarse-grained or macroscopic system. This makes LAMMPS more general and not biomolecular specific than some of other MD codes.

The MD simulation often involves many thousands or millions of atoms in 3D in a length scale of angstrom ($10^{-10}\,m$), and a timestep scale of femtosecond ($10^{-15}\,s$). Therefore, to simulate a nanosecond ($10^{-9}\,s$) of real time, hundreds of thousands of timesteps are necessary. MD simulation is hence large in the domains of size of atoms and number of timesteps. LAMMPS applies the spatial-decomposition parallel algorithm to speed up the simulation. On p processors, the physical simulation domain is subdivided into p small 3-D boxes, one for each processor. Each processor computes forces and updates the positions and velocities of all atoms within its box at each timestep. It is scalable to a very large number of atoms and processors.

LAMMPS will run on any parallel machine that compiles C++ and supports MPI. This paper compares a set of LAMMPS (15 Jan 2010 version) benchmark performance runs on the various SGI architectures, and attempts to characterize LAMMPS. It is organized as follows: Section 2 describes the three types of SGI systems where benchmarks are run: the SGI® Rackable™ cluster, the SGI® Altix® ICE cluster and SGI® Altix® UV (Shared Memory Parallel) System. Section 3 introduces the benchmark test used. Section 4 presents the results and discussions. Finally, we summarize our findings.

2.0 Benchmark System Details

SGI® Rackable™
1U, 2U, 3U, 4U & XU Rack-mount

SGI® Altix® ICE
HPC Blade Architecture

SGI® Altix™ UV
Shared Memory Architecture

2.1 The SGI® Rackable™ Cluster

The SGI® Rackable™ cluster delivers top value and performance based on the latest Intel® Xeon® X5600 series-based architecture. It combines the flexibility of selecting components for ultimate application right-sizing with the convenience of full factory integration. The below is cluster detail used for benchmarking.

SGI Rackable C2112-4TY14 Intel Xeon® X5670@2.93GHz

- 32 dual-socket nodes, each socket with six-core Intel® Xeon® X5670 (Westmere-EP) 2.93 GHz, 12 MB shared cache per 6 cores (total 384 cores)
- IB Device(s): mlx4_0 FW=2.7.0 Rate=40 Gb/sec (4X QDR)
- Total Mem: 48GB per node, Speed: 1,333 MHz (0.8 ns)
- SUSE® Linux® Enterprise Server 11 SP1
- Kernel Ver: 2.6.32.36-0.5-default
- Hyper-Threading enabled

2.2 The SGI® Altix® ICE Cluster

The SGI® Altix® ICE integrated blade cluster is designed for today's data intensive problems. This innovative new platform from SGI raises the efficiency bar, easily scaling to meet virtually any processing requirements without compromising ease of use, manageability or price/performance. SGI Altix ICE delivers unsurpassed customer value, with breakthrough density, efficiency, reliability and manageability. The following two clusters were used for benchmarking in this paper.

SGI Altix ICE 8400 Intel® Xeon® X5680 @3.33GHz

- 128 dual-socket nodes, each socket has a six-core Intel® Xeon® X5680 (Westmere-EP) 3.33GHz, 12 MB shared cache per 6 cores (total 1,536 cores)
- IB Device(s): mlx4_0 FW=2.7.0 Rate=40 Gb/sec (4X QDR)
- Total Mem: 24GB per node, Speed: 1,333 MHz (0.8 ns)
- SLES 11 SP0 OS, SGI ProPack™ 7 SP1
- Kernel Ver: 2.6.32.13-0.4-default
- SGI Software: SGI Tempo Compute Node 2.1, Build 701r3.sles11-1005252113

SGI Altix ICE 8200 Intel® Xeon® X5570 @2.93GHz

- 128 dual-socket nodes, each socket has a quad-core Intel® Xeon® X5570 (Nehalem-EP) 2.93 GHz, 8 MB shared cache per 4 cores (total 1,024 cores)
- Total Mem: 24GB per node, Speed: 1,333 MHz (0.8 ns)
- IB Device(s): mlx4_0 FW=2.7.0 Rate=20 Gb/sec (4X DDR)
- SLES 10 SP2 OS, SGI ProPack 6 SP5
- Kernel Ver: 2.6.16.60-0.30-smp
- SGI Software: SGI Tempo Compute Node 1.9, Build 605r1.sles10-0909302200

2.3 SGI® Altix® UV Shared Memory System

SGI Altix® UV scales to extraordinary levels-up to 256 sockets (2,048 cores, 4096 threads) with architectural support to 262,144 cores (32,768 sockets). Support for up to 16TB of global shared memory in a single system image, enables SGI Altix UV to remain highly efficient at scale for applications ranging from in-memory databases, to a diverse set of data and compute-intensive HPC applications. With this platform, it is simpler for the user to access huge resources for programming via a familiar OS, without the need for rewriting their software to include complex communication algorithms. The below is the system detail used to run the benchmarks.

SGI Alitx UV 1000 Intel® Xeon® X7542 @2.67GHz

- 32 dual-sockets; each socket has an six-core Intel® Xeon® X7542 (Nehalem-EX) 2.67GHz, 18 MB shared cache per 6 cores (total 384 cores)
- Total Mem: 2TB, Speed: 1,067 MHz (0.9 ns)
- Interconnect: NUMAlink 5, quad-plane routered fat tree
- SUSE Linux Enterprise Server 11 SP1
- Kernel Ver: 2.6.32.24-0.2.1.2230.2.PTF-default

3.0 Benchmarks Descriptions

There are five tests (see the standard distribution of LAMMPS [3]) used to benchmark LAMMPS. As mentioned in the introduction, for a molecular dynamics system, the main ingredient of the simulation is a model for the physical system, which amounts to choose the potential energy of the system when the atoms are arranged in that specific configuration. Below is a summary of the potential used for each benchmark, and they can be further categorized into two groups:

Short range forces modeled with a cut off distance:

- Chain: Polymer chain melt benchmark: Lennard–Jones potential
- Chute: Granular chute flow benchmark: Granular potential
- EAM: EAM metallic solid benchmark: EAM potential
- LJ: Lennard-Jones liquid benchmark: LJ potential

Long range force:

- Rhodopsin: Rhodopsin protein benchmark, it is modeled by CHARMM force field and LJ potential but with long range forces, i.e. Coulombic interactions, each atom interacts with all others. A different algorithm—particle-particle particle-mesh (PPPM) technique [7] is used to compute this long range forces. It is a FFT based algorithm, which will affect the scalability of the test.

All five tests can be run either in **fixed**- or **scaled**- size. The original test sets have 32,000 atoms in each, and the default number of timesteps is 100. It is too small to give meaningful timings on large number of cores, so that the number of steps for the same problem is made to a function of the number of cores as follows:

nCores	Chain	Chute	EAM	LJ	Rhodopsin
1	50,000	100,000	10,000	20,000	1,400
p	50,000 x p	100,000 x p	10,000 x p	20,000 x p	1,400 x p

To better understand how the performance scales as more processors are added, LAMMPS offers options to scale the data. This makes sense, since if the benchmark is unscaled, effects arising from limited parallelism and increases in communication can lead to results that are negative when in reality more processors will be used to solve larger problems.

When run on p cores, the number of atoms in the simulation was scaled as p times of the one-core run, for example, when run on 256 cores; the number of atoms will be 8192,000, if there are 32,000 atoms on one-core. Further, the problem size can be scaled in three dimensions x, y, z as p_x x p_y x p_z and different combinations of p_x, p_y and p_z give different results. In our experiments, we found out that generally scaling in one dimension can obtain better performance than scaling evenly in three dimensions. But at some point, with very larger number of cores, it is not possible to just scale in one dimension.

4.0 Results and Discussions

4.1 Performance Comparison of Fixed-Size Tests

For the fixed-size problems, we measured the main computation loop time (seconds) of benchmarks and convert them into the rate i.e., simulation time/day. The higher the rate of the job, the better the performance was. The unit of simulation time in the Chute, LJ and Chain tests is the reduced unit, tau = t * sqrt (energy/mass/distance^2) see [2]. For the other two tests EAM and Rhodopsin, the simulation time is converted into Nanosecs/Day.

In this section, we first compare the performance of five fixed-size benchmarks on a single core and single node bases, and then present the comparison of parallel scaling performance on a per-core basis.

Single core and single node comparison

In Table 1, on a single core, the SGI Rackable C2112 cluster performs essentially the same as SGI Altix ICE 8200, as the two clusters have the same clock frequency with the exception of the Chute case. On a single node, the SGI Rackable C2112 cluster has 12 cores compared to eight cores of SGI Altix ICE 8200, and it performs on average 1.46x as fast which is close to the ratio of cores (1.5) in the node. Compared to SGI Altix UV 1000, on a single core, the SGI Rackable C2112 cluster performs on average 1.10x as fast, which is close to the ratio of the two processors. As there are the same number of cores in a node on two machines, the single node performance are similar to those of single core.

	SGI Rackable (WSM-EP 2.93GHz) Over SGI Altix ICE 8200 (NHM-EP 2.93GHz)		SGI Rackable (WSM-EP 2.93GHz) Over SGI Altix UV 1000 (NHM-EX 2.66GHz)	
Test	Single Core	Single Node	Single Core	Single Node
Geomean	1.03	1.46	1.10	1.08
Chute_f	1.13	1.45	1.06	1.03
Chain_f	1.00	1.46	1.13	1.08
LJ_f	1.01	1.47	1.11	1.09
EAM_f	1.01	1.48	1.10	1.09
Rhodopsin_f	1.00	1.45	1.12	1.10

Table 1 *Comparison of the SGI Rackable cluster to SGI Altix ICE 8200 and SGI Altix UV 1000*

In Table 2, on a single core, SGI Altix ICE 8400 Xeon® X5680@3.33GHz runs on average 1.12x as fast as SGI Altix ICE 8200 Xeon® X5570@2.93GHz, which is in line with the processor speed ratio 1.14. On a single node, SGI Altix ICE 8400 (12 cores/node) runs on average 1.60x as fast as SGI Altix ICE 8200 (8 cores/node). Compared to SGI Altix UV 1000 Xeon® X7542@2.67GHz, SGI Altix ICE 8400 runs on average 1.23x as fast, which is also close to the processor speed ratio 1.25. The single node performance is similar to those observed on a single core.

	SGI Altix ICE 8400 (WSM-EP 3.33GHz) over SGI Altix ICE 8200 (NHM-EP 2.93GHz)		SGI Altix ICE 8400 (WSM-EP 3.33GHz) over SGI Altix UV 1000 (NHM-EX 2.66GHz)	
Test	Single Core	Single Node	Single Core	Single Node
Geomean	1.12	1.60	1.23	1.20
Chute_f	1.07	1.42	1.17	1.12
Chain_f	1.12	1.63	1.24	1.21
LJ_f	1.14	1.65	1.25	1.22
EAM_f	1.14	1.67	1.24	1.23
Rhodopsin_f	1.14	1.62	1.27	1.22

Table 2 *Comparison of SGI Altix ICE 8400 to SGI Altix ICE 8200 and SGI Altix UV 1000*

Comparison of Parallel Scaling Performance

After considering the single core and single node performance, Figure 1 and Figure 2 (below) present the parallel scaling performance of five fixed-size benchmarks on a per-core basis. The graphs go up to around the number of cores where we still see 50% parallel efficiency. Due to the moderate size (32,000 atoms) of the problem and the nature of force calculation, the most scaled case among five benchmarks is EAM scaled up to 240 cores.

Figure 1 and 2 also show that the scaling performance is in line with the processor core frequency: the higher of the clock frequency, the better the performance is on a per core basis. There are exceptions — SGI Rackable C2112 and SGI Altix ICE 8200. These two clusters have the same clock speed 2.93GHz, but a different infiniband network, former a QDR interconnect versus DDR in the latter. The two clusters show similar performances on low core count, but the SGI Rackable C2112 cluster demonstrates a better parallel scaling on the large core count (96 or larger).

For example, in Figure 1 (a) for Chain case, on 48 cores we still see the similar performance between SGI Rackable C2112 and SGI Altix ICE 8200, but on 96 cores, the former has a speedup 52x (see the green arrow) compared to 40x on SGI Altix ICE 8200 and it runs about 22% faster. Looking into the individual time components of main Loop time, it shows that the "Pair time" (the force calculation time) of Chain case on two machines is similar, but the "Communication time" has been increased about 40% on SGI Altix ICE 8200 compared to SGI Rackable C2112. Similar patterns are observed for the other four tests as highlighted by the green arrows on each graph.

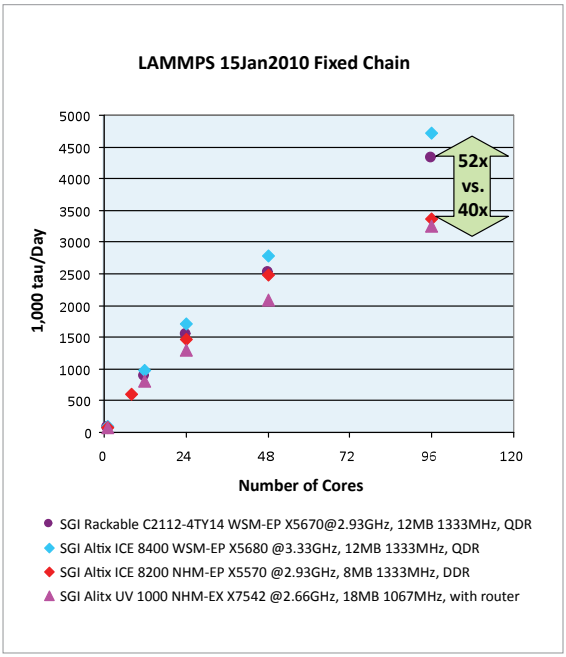

Figure 1 (a)

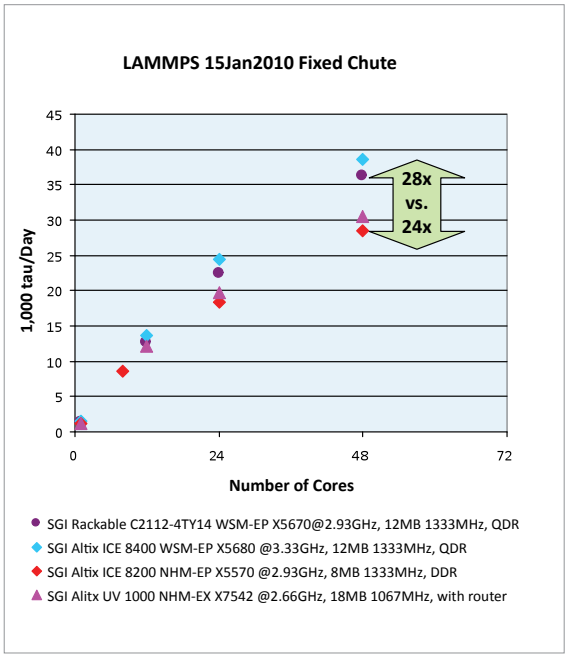

Figure 1 (b)

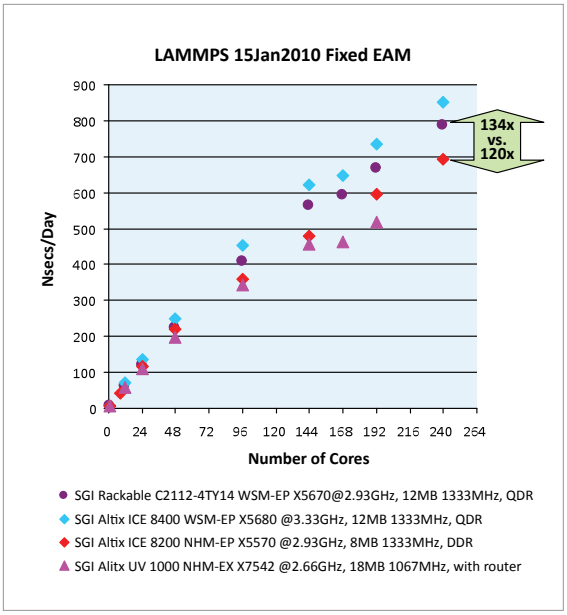

Figure 1 (c)

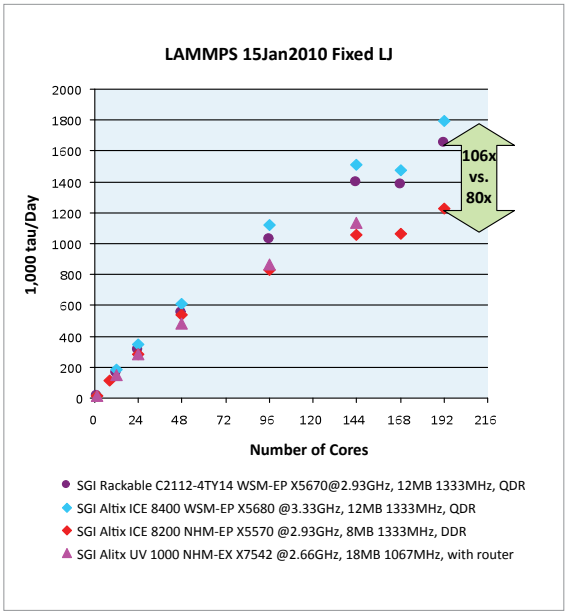

Figure 1 (d)

In Figure 1 (a)-(d), it shows in general that both the SGI Rackable and the SGI Altix ICE clusters demonstrate better parallel scaling than SGI Altix UV 1000 for short-range force tests: Chute, Chain, LJ, and EAM. However, in Figure 2, SGI Altix UV 1000 shows a better parallel scaling (61x) for the Rhodopsin case than the SGI Rackable and the SGI Altix ICE clusters. On 96 cores, even with a lower clock frequency it has similar performance to SGI Rackable C2112 2.93GHz, and faster performance than SGI Altix ICE 8200.

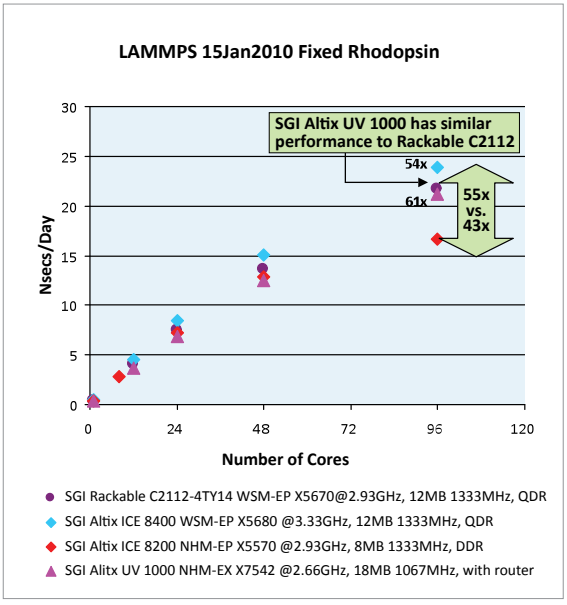

Figure 2

4.2 Discussions

4.2.1 Is LAMMPS CPU Clock Frequency Bound or Memory Bound?

We have already seen in the section 4.1 that LAMMPS scales with the core frequency in general. In Table 1, on a single core, with the same clock and memory speed, SGI Rackable C2112 shows similar performance to SGI Altix ICE 8200 with the exception of Chute case. On SGI Rackable C2112, the Chute test runs 13% faster than that on SGI Altix ICE 8200. Table 2 shows the similar pattern that the performance is in line with the processor core frequency. On a single core, the relative performance ratio of 3.33GHz over 2.93GHz (SGI Altix ICE 8400 vs. SGI Altix ICE 8200) for the tests Chain, EAM, LJ and Rhodopsin are about the same as the processor ratio 1.14. The exception is again the Chute case which is well below the processors ratio.

To investigate the reason of low performance of Chute case on the machine with a higher clock frequency, we have measured Last Level Cache misses (LLC_MISSES) with Performance Counters for Linux (PCL) [5] on SGI Altix ICE 8400. It shows that the ratio of LLC misses is about 10% and 2% for Chute and LJ respectively running on a single core. When the working set is not fit in the cache as the Chute case, the performance can be memory bandwidth bound. This also explains the slow single core performance on SGI Altix ICE 8200 of Chute, as SGI Altix ICE 8200 with NHM-EP X5570 has only 8M cache compared to 12M on WSM-EP X5670.

Furthermore, when running the Chute and LJ in scaled-size as shown in Figure 3 (a) and (b), the performance degradations of a full node run are of about 60% and 14% over a single core run for Chute and LJ respectively on SGI Altix ICE 8400. In Figure 3, the loop time is normalized by the time measured for the single core run. A hypothetical value of bigger than 1.0 will imply that the performance is above expectation and value closer to 1.0 is better. With the performance counter, we measured the ratio of LLC misses is increased to 72% and 39% on 12 cores of a full node respectively for Chute and LJ. When the cache has to be shared among all cores within a socket, more performance is lost.

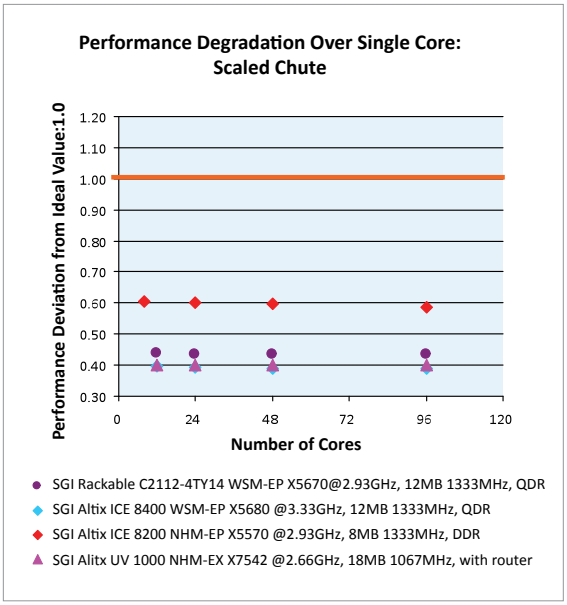

Figure 3 (a)

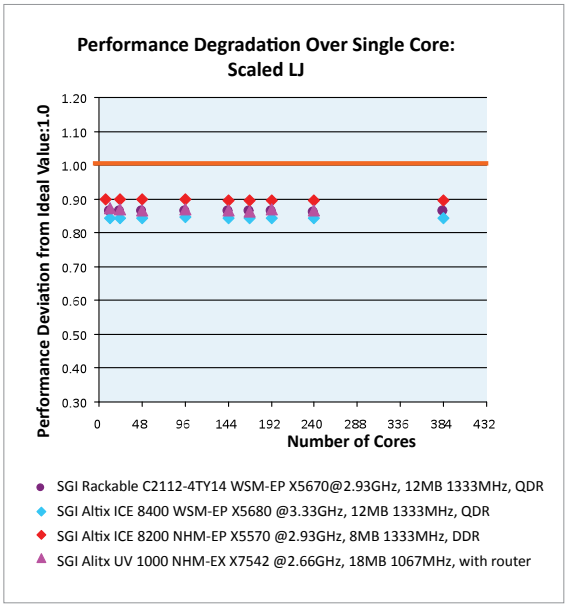

Figure 3 (b)

We also performed experiments with Chute scaled test case using MPInside [6] to profile the LAMMPS and understand the performance degradation on SGI Altix ICE 8400, and the results are shown in Figure 4. The first graph (a) illustrates the benchmark was run on a single node (12 cores), while the second one (b) on 12 nodes with one core per node. The 'Comput' legend is for the computation portion of the code, and the other four legends allred, irecv, send and wait stand for MPI_Allreduce, MPI_Irecv, MPI_Send and MPI_Wait respectively. It can be observed that the communication time accounts for just a fraction of the total Loop time. The Loop time of a full node run (~1500 secs) is more than twice slower than 12 nodes run (~700 secs). Such a time difference indeed results from the computation portion of the code. With one core per node, it has the whole 12M cache and the bandwidth. These are further evidences that Chute is affected by the cache and bandwidth performance.

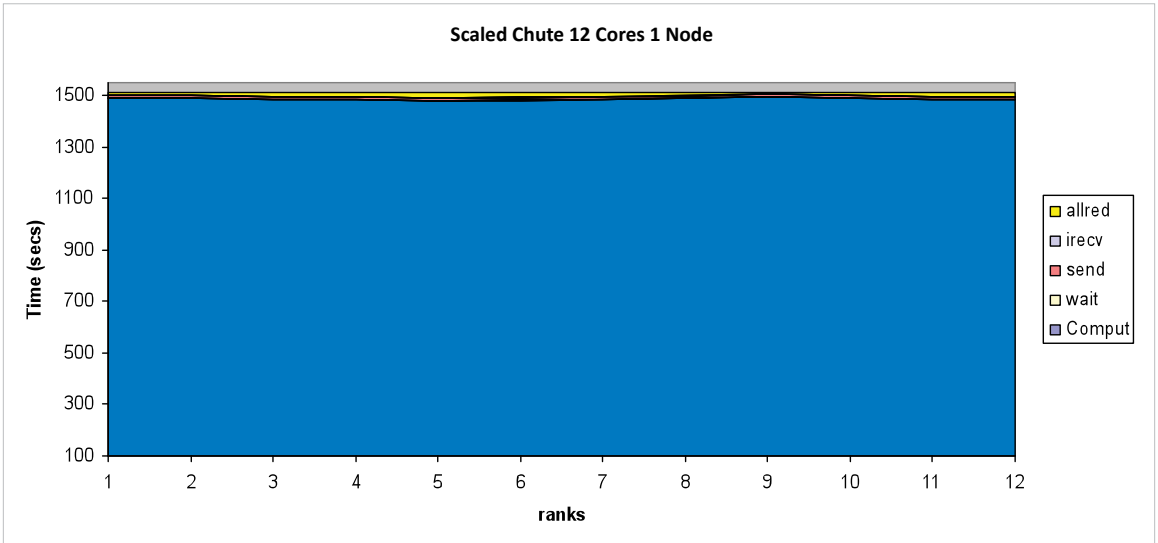

Figure 4 (a) *MPInside profiling results on SGI Altix ICE 8400*

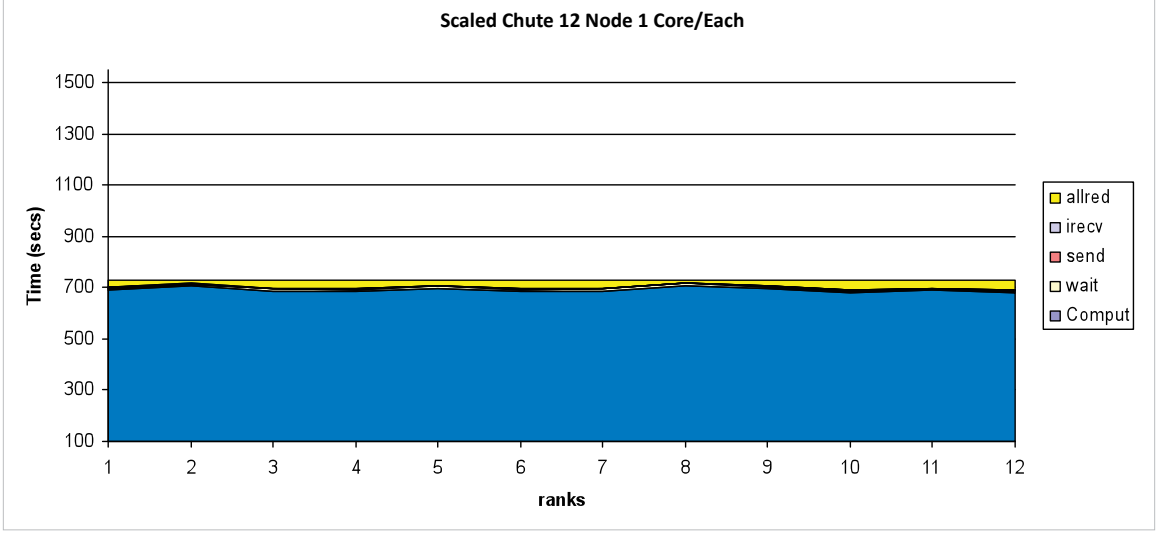

Figure 4 (b) *MPInside profiling results on SGI Altix ICE 8400*

In summary, some potential calculations (the interaction of force fields) are inherently more computation intensive than others, thus, depending on the potential model of a test, the LAMMPS performance can be bounded by both CPU clock and memory performance (bandwidth and latency). Chute is mainly bounded by the cache and memory bandwidth performance, whereas the other four tests are mainly compute bound.

4.2.2 Effect of Hyper-Threading

Another characteristic of LAMMPS code is that it hits random places in the memory to obtain the attributes of atoms for the neighbor search, force calculation etc. When the force calculation is relatively cheap, the CPU is stalled and waiting to access the memory rather than doing computation. The benchmark performance thus may be affected by this memory latency. In [4], it shows that the use of Intel® Hyper-Threading Technology (Intel® HT Technology) is able to reduce the impact of memory latency. To assess the effect of HT for LAMMPS, we first measured the main loop time of fixed size tests on the SGI Rackable C2112 cluster. The results are normalized by the performance of running only on physical cores. Figure 5 compares the results of running on 12 physical cores in one single node to those of running on 24 logical cores of one single node with Intel® Hyper-Threading. It shows that LAMMPS benchmarks indeed benefit from the HT. There are improvements around 2% (Chute_f) to 17% (EAM_f).

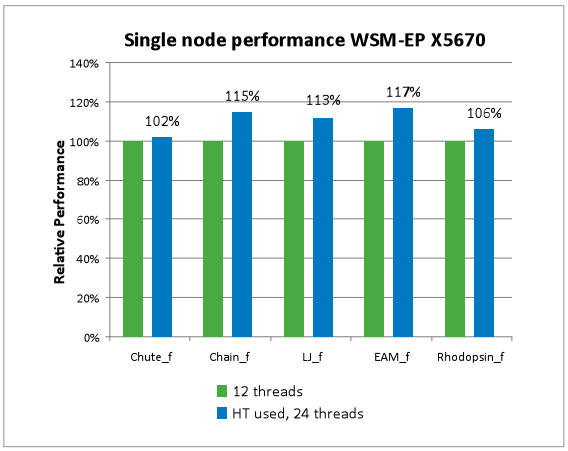

Figure 5

As discussed in section 4.2.1 that Chute is memory bandwidth bounded, which benefits little from HT. Figure 6 further compares both fixed size (a) and scaled size (b) performance on eight nodes: 96 physical cores versus 192 threads. In Figure 6 (a), it shows that HT is having negative effect on Chain and Rhodopsin cases. Chain's force computation is relatively cheaper than the other four tests, with the more number of threads, the communication time diminish any benefit gained over faster force computation. Rhodopsin is limited by the FFT's scalability, with the increase number of threads, FFT time is increased, so the performance gets worse.

While in Figure 6 (b), it shows the similar performance improvement on the scaled size tests on eight nodes as those on a single node in Figure 5 again with the exception of the Rhodopsin.

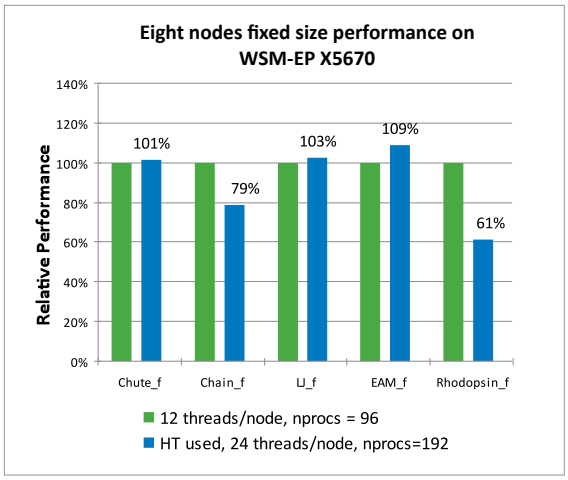

Figure 6 (a)

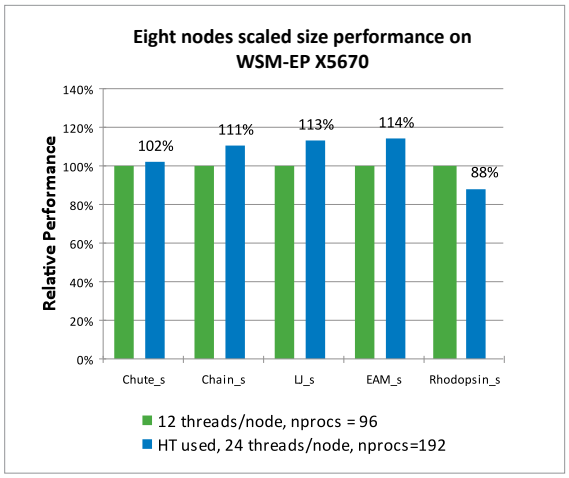

Figure 6 (b)

4.2.3 Effect of Different Network Topologies

The SGI Altix ICE 8400 cluster with its infinaband interconnect has features to offer different network topologies [8]: standard hypercube, enhanced hypercube, fat tree, and all-to-all. In this section, we used only scaled cases to study the topological effect on the pattern of the communication, since the fixed cases are too small to be scaled to very large cores (1536 cores). The below is a summary of different topologies have been configured and the legends being used in Figure 7 (a)-(e).

- SHC 1r: single-rail standard hypercube
- EHC 1r: single-rail enhanced hypercube
- alltoall 1r: single-rail all-to-all
- fat-tree 1r: single-rail fat tree
- SHC 2r: dual-rail standard hypercube
- EHC 2r: dual-rail enhanced hypercube
- alltoall 2r: dual-rail all-to-all

All results at each node count are normalized by the single rail standard hypercube performance at that node count, the higher (> 100%) indicates better performance.

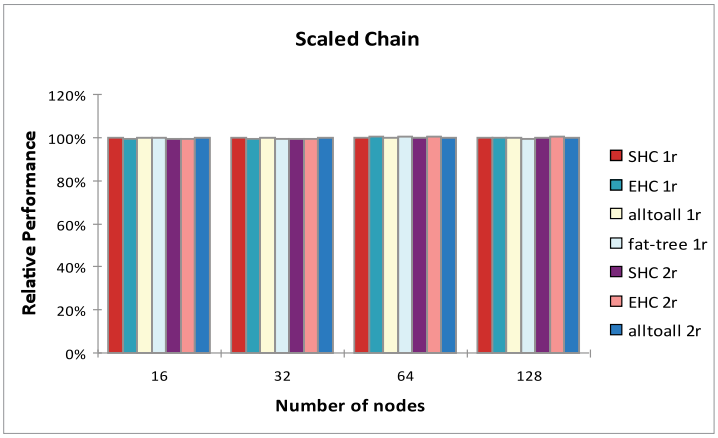

Figure 7 (a): *LAMMPS scaled Chain with different topological configuration on SGI Altix ICE 8400*

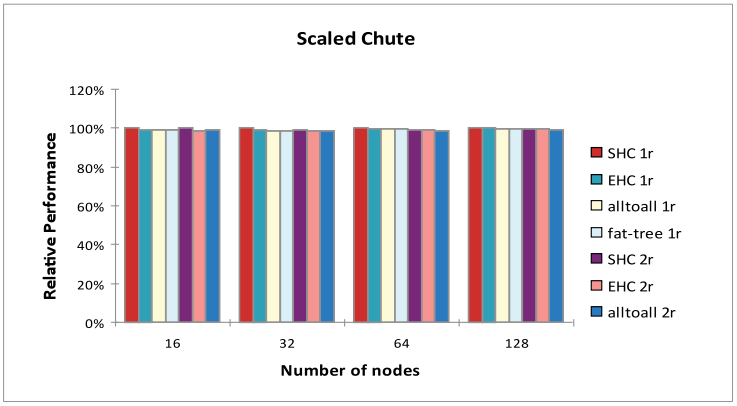

***Figure 7 (b)**: LAMMPS scaled Chute with different topological configuration on SGI Altix ICE 8400*

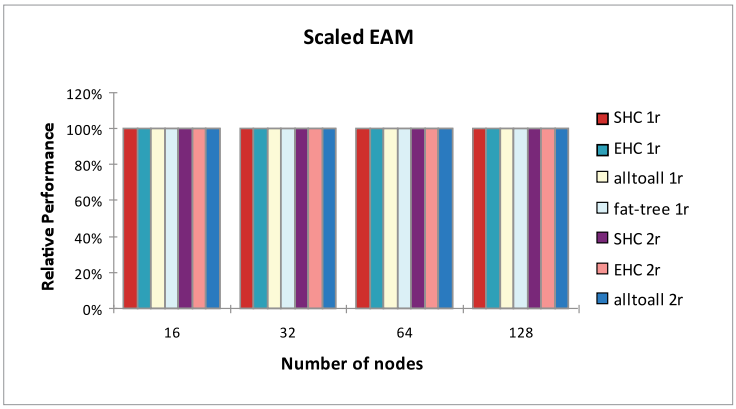

***Figure 7 (c):** LAMMPS scaled EAM with different topological configuration on SGI Altix ICE 8400*

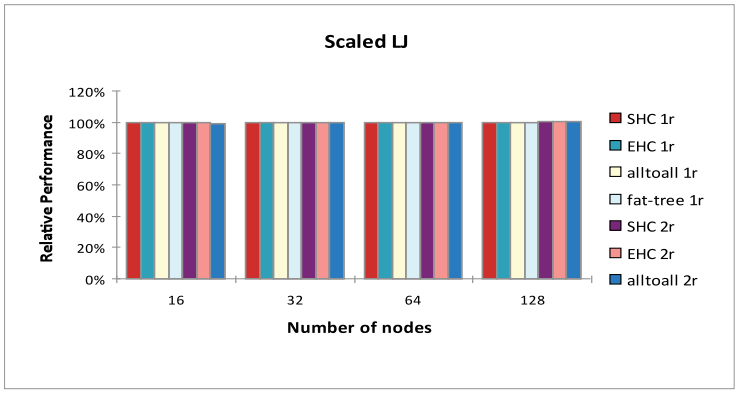

***Figure 7 (d):** LAMMPS scaled LJ with different topological configuration on SGI Altix ICE 8400*

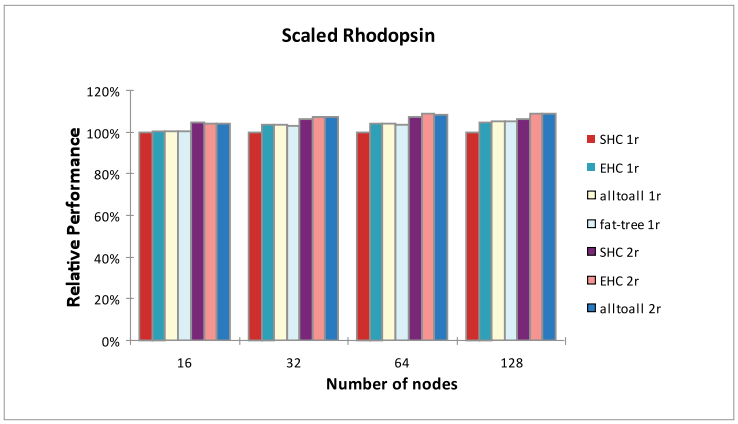

***Figure 7 (e)**: LAMMPS scaled Rhodopsin with different topological configuration on SGI Altix ICE 8400*

In Figure 7 (a)-(d), we can see the message passing is almost not affected (less than 1%) by the different topologies for these four test cases: Chute, Chain, EAM and LJ. While in Figure 7 (e) Rhodopsin case, the difference among the different topologies can be as large as ~9% on 1536 cores. LAMMPS deploys the algorithm to decompose the whole 3D simulation space into a number of boxes. Each processor is assigned to one box, and for short range forces, each processor acquires only the atom positions that are within a cut-off distance, i.e. either in the local processor, or those processors of surrounding neighboring boxes. The communication is hence minimized and local in nature. It will be affected little by different network topologies. However for the Rhodopsin case, it used the PPPM algorithm with a FFT based approach, which is usually limited by an all-to-all communication of data. Dual rail enhanced hybercube and all-to-all topologies give a 9% improvement over single rail standard hypercube on 128 nodes (1536 cores).

5.0 Summary

In this paper, we presented and analyzed the performance of LAMMPS (15 Jan 2010 version) on three types of SGI systems.

- For short range forces:
  - In general, clusters show better parallel scaling than shared memory system – SGI Altix UV 1000, and parallel performance is in line with core frequency;
  - The different network topology on clusters shows little effect on the communication time of this group of tests.
- For long range forces:
  - SGI Altix UV 1000 demonstrates better parallel scaling performance than clusters;
  - The performance difference among different network topology on clusters can be as large as 9%.
- Depending on the potential model of a test, the computation performance can be bounded by clock frequency (e.g. LJ, EAM), or bounded by the cache and memory bandwidth (e.g. Chute).
- For fixed size tests, on large core counts, depending on the ratio of the computation and communication, IB performance makes a big difference (SGI Rackable C2112 vs. SGI Altix ICE 8200).
- Hyper-Threading improves performance (over 10%) for the benchmark (e.g. EAM, Chain and LJ) affected by the memory latency but not bounded by memory bandwidth, however it shows little or negative effect for the benchmark affected by all-to-all communication (Rhodopsin) and memory bandwidth (Chute).

This data can be used as a baseline to guide the efficient use of LAMMPS on advanced SGI® computer hardware systems.

6.0 References

[1] S. Plimpton, *Fast Parallel Algorithms for Short-Range Molecular Dynamics*, J Comp Phys, 117, 1-19 , 1995.

[2] Sandia National Laboratories, *LAMMPS Users Manual.*

[3] LAMMPS software by Sandia National Laboratories: http://lammps.sandia.gov/bench.html

[4] W. Magro, P. Petersen, S. Shah, *Hyper-Threading Technology: Impact on Computer-Intensive Workloads*, Intel Technology Journal Q1, 2002. Vol. 6 Issue 1.

[5] http://perfmon2.sourceforge.net/

[6] D. Thomas, J.P Panziera, J. Baron, *MPInside: a Performance Analysis and Diagnostic Tool for MPI Applications*, Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering, 79-86, 2010.

[7] S. Plimpton, R. Pollock, M. Stevens, *Particle-Mesh Ewald and rRESPA for Parallel Molecular Dynamics Simulations*, Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing, 1997.

[8] SGI Altix ICE 8400 Series System Hardware User's Guide.

Corporate Headquarters
46600 Landing Parkway
Fremont, CA 94538
tel 510.933.8300
fax 408.321.0293
www.sgi.com

Global Sales and Support
North America +1 800.800.7441
Latin America +55 11.5185.2860
Europe +44 118.927.8000
Asia Pacific +61 2.9448.1463