WHITE PAPER

LS-DYNA® Implicit Hybrid Technology on Advanced SGI® Architectures*

Olivier Schreiber, Scott Shaw, Brian Thatch,[†] Bill Tang**

Abstract

LS-DYNA's implicit solver integration with explicit software allows large time steps transient dynamics as well as linear statics and normal modes analysis. Until recently, this capability could only be run on large Shared Memory Parallel (SMP) systems, where the application had access to large memory address space of the model. Distributed Memory Parallel (DMP) implementation of LS-DYNA's implicit solver now allows the factorization of smaller mass and stiffness matrices of the decomposed problem domain by corresponding tasks in less memory. Performance enhancement through SMP processing is moreover also available in the recently introduced 'hybrid' mode. This paper demonstrates how advanced SGI computer systems, ranging from SMP servers addressing large memory space through multi-node clusters can be used to architect and accelerate solutions to meet complex analysis requirements.

** Adapted from [1]*

† SGI Applications Engineering

*** SGI Systems Engineering*

TABLE OF CONTENTS

List of Figures

List of Tables

Introduction

The subject of this paper is to evaluate the use of SGI Octane™ III, Altix® XE, Altix ICE, Altix UV and Altix architectures to Shared Memory Parallel (SMP), Distributed Memory Parallel (DMP) and their combination (hybrid mode) LS-DYNA implicit analyses. The strategies employed by LS-DYNA and the practical importance of such analyses are described in Refs [2] and [3]. Integrated within its explicit framework, LS-DYNA's implicit technology provides the capability to perform transient analyses with larger time steps as well as usual linear statics and modal analyses. How to best use SGI hardware is described in Ref [4].

1. Benchmark Description

The benchmarks used are identical physical problems as in Refs [2] and [3] available in meshes of 100K, 500K, 1M, 2M, 4M, up to 20M nodes. The model represents 6 nested cylinders held together with surface to surface contact, meshed with single elastic material solid elements.

A prescribed motion on the top and a load on the bottom are imposed for one nonlinear implicit time step with two factorizations, two solves and four force computations. Figure 1 illustrates a 921,600 solid elements, 1,014,751 nodes problem leading to a 3,034,944 order linear algebra system.

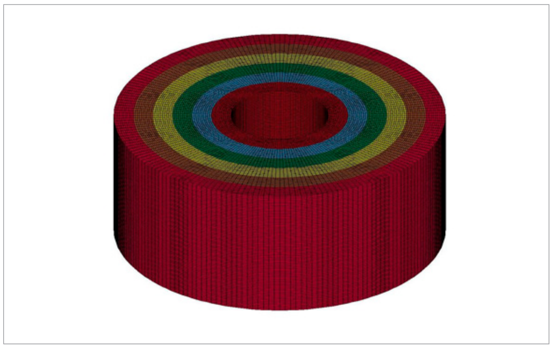


Figure 1: Refs [2] and [3] Cylinder Solid Element Problem Series, 1M nodes

2. Benchmark Systems

Various systems comprised in SGI product line and available through SGI Cyclone™, HPC on-demand Cloud Computing (see section 5) were used to run the benchmark described in section 1.

2.1 SGI Octane III

Scalable deskside multi-node system with GigE or Infiniband interconnects, up to 10 nodes, 120 cores, in two different configurations, with SUSE® Linux® Enterprise Server 10 SP2, SGI ProPack™ 6SP3:

Figure 2: SGI Octane III

2.1.1 SGI Octane III Xeon® X5570

- Dual-socket nodes of 2.93GHz quad-core Xeon X5570, 8 MB cache
- Total Mem: 24 GB Speed: 1333 MHz (0.8 ns)

2.1.2 SGI Octane III Xeon E5540

- Dual-socket nodes of 2.53GHz quad-core Xeon E5540, 8 MB cache
- Total Mem: 24 GB Speed: 1066 MHz (0.9 ns)

2.2 SGI Altix XE 1300 cluster

Highly scalable and configurable rack-mounted multi-node system with GigE and/or Infiniband interconnects.

Figure 3: SGI Altix XE 1300 cluster

- SGI XE270 Administrative/NFS Server node
- SGI XE340 Dual-socket compute nodes of 2.93GHz quad core Xeon X5570 8MB Cache
- 24GB 1333MHz RAM
- SUSE Linux Enterprise Server 11 SP2, SGI ProPack 6SP3
- SGI Foundation Software 1SP5
- Infiniband ConnectX QDR PCIe Host Card Adapters
- Integrated GigE dual port Network Interface Cards

SGI Altix XE 1300 cluster with dual Ethernet and Infiniband switch is illustrated in Figure 4.

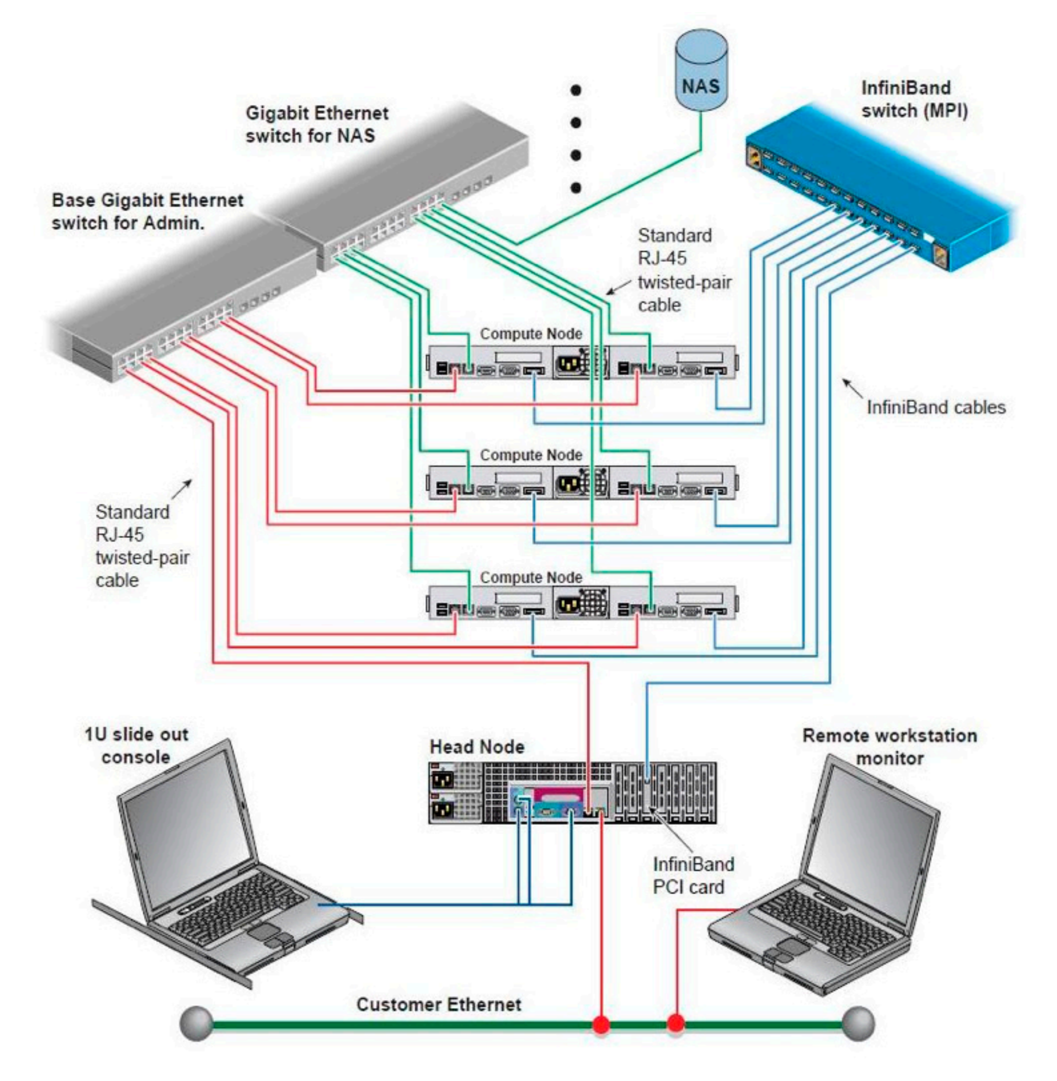

NAS
InfiniBand switch (MPI)
Gigabit Ethernet switch for NAS
Base Gigabit Ethernet switch for Admin.
Standard RJ-45 twisted-pair cable
Compute Node
InfiniBand cables
Compute Node
Standard RJ-45 twisted-pair cable
Compute Node
1U slide out console
Head Node
Remote workstation monitor
InfiniBand PCI card
Customer Ethernet

Figure 4: Dual Ethernet and Infiniband switch cluster configuration example

2.3 SGI Altix ICE 8200 cluster

Highly scalable, diskless, integrated cable-free infiniband interconnect rack mounted multi-node system.

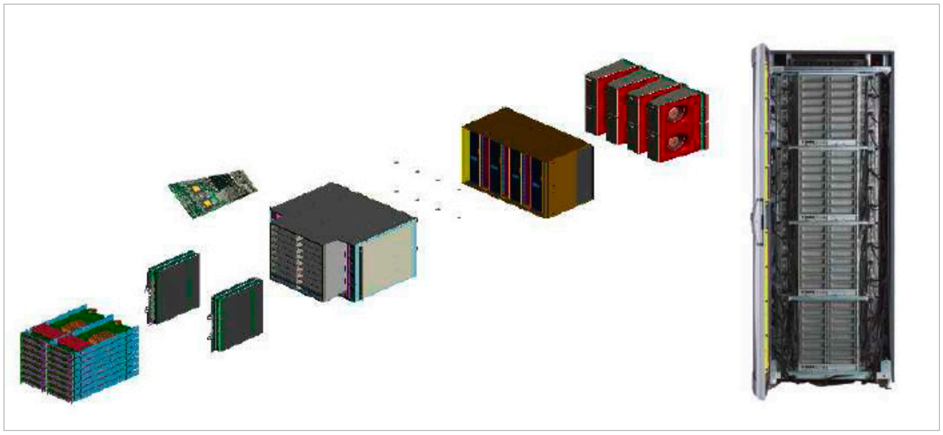


Figure 5: SGI Altix ICE 8200 cluster and IRU

2.3.1 SGI Altix ICE 8200 cluster

SGI Altix ICE 8200EX DDR SLES11 PP6SP6 Tempo v1.10 Dual-socket compute nodes of 2.93GHz quad core Xeon X5570 24GB RAM in 1066MHz DIMMs

2.3.2 SGI Altix ICE 8200 cluster

SGI Altix ICE 8200EX DDR SLES10SP2 PP6SP5 Tempo v1.9 Dual-socket compute nodes of 2.93GHz quad core Xeon X5570 24GB RAM in 1333MHz DIMMs

2.4 SGI Altix 450 and 4700 SMP

Highly scalable Shared Memory Parallel system.

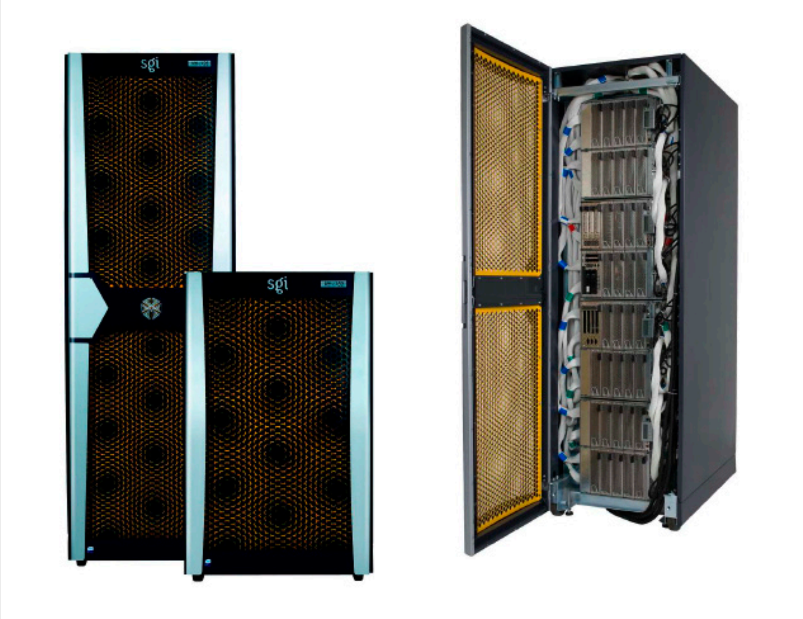


Figure 6: SGI Altix 4700, 450 SMP

- SGI Altix 4700 128 1.669GHz dual core Itanium® 9150M 24MB Cache processors, 512GB RAM NUMAlink® 4

2.5 SGI Altix UV 100, UV 1000 SMP

Highly scalable latest generation x86-based Shared Memory Parallel system.

Figure 7: SGI Altix UV 10, UV 100, UV 1000 SMP

- 12 2.66Ghz 6-core Xeon X7542 (72 cores)
- 192GB RAM
- NUMAlink 5
- SGI Foundation Software, SGI ProPack 7

2.6 Filesystems

Various filesystems may co-exist to use for scratch space:

- Memory-based filesystem /dev/shm
- Root drive
- DAS (Direct Attached Storage)
- NAS (Network Attached Storage)

These filesystem definitions are illustrated in Figure 8

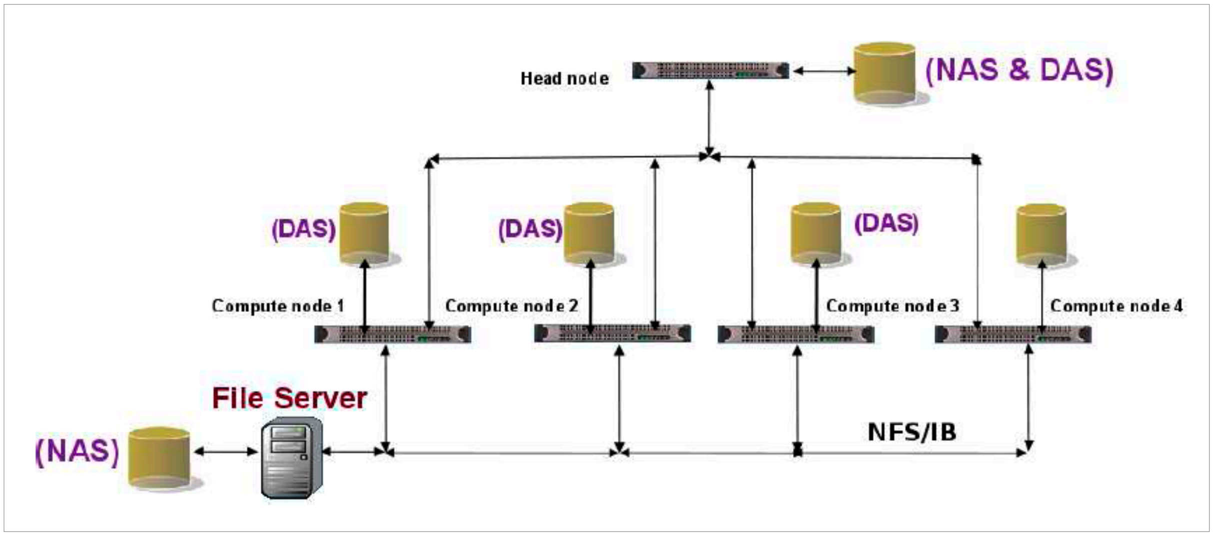


Figure 8: Example filesystems for scratch space

3. LS-DYNA

3.1 Version Used

LS-DYNA/MPP ls971 R5.0 hybrid for SGI Message Passing Toolkit (MPT).

3.2 Parallel Processing Capabilities of LS-DYNA

Parallelism exists in two paradigms:

- Distributed Memory Parallelism (DMP): uses MPI Application Programming Interface focused on physical domain decomposition. The created reduced size geometric partitions have lesser processing resource requirements, resulting in increased efficiency, while the size of the common boundary is kept minimal to decrease inter-process communication.
- Shared Memory Parallelism (SMP): uses OpenMP Application Programming Interface focused on computational loops.

These two paradigms may simultaneously map themselves on two different system hardware levels:

- Inter-node or cluster parallelism (memory local to each node)–DMP only.
- Intra-node or multi-core parallelism (memory shared by all cores of each node)

The hybrid approach provides increased performance yields and additional possibilities of memory utilization by running SMP on intra-node network in combination with DMP on inter-node and/or intra-node network.

3.3 Submittal procedure

Submittal procedure must ensure:

- Placement of processes and threads across nodes and cores within nodes
- Control of process memory allocation to stay within node capacity
- Use of adequate scratch files across nodes or network

Batch schedulers/resource managers dispatch jobs from a front-end login node to be executed on one or more compute nodes. To achieve the best runtime in a batch environment disk access to input and output files should be placed on the high performance filesystem closest to the compute node. The high performance filesystem could be in-memory filesystem (/dev/shm), a Direct (DAS) or Network (NAS) Attached Storage filesystem. In diskless computing environments in-memory filesystem or network attached storage are of course the only options.

Following is the synoptic of a job submission script.

1. Change directory to the local scratch directory on the first compute node allocated bythe batch scheduler
2. Copy all input files over to this directory
3. Create parallel local scratch directories on the other compute nodes allocated by the batch scheduler
4. Launch application on the first compute node. The executable may itself carry out propagation and collection of various files between launch and the other nodes at start and end of the main analysis execution.

3.4 Hybrid mode

3.4.1 MPI tasks and OpenMP thread allocation across nodes and cores

The following keywords are used for the LS-DYNA execution command:

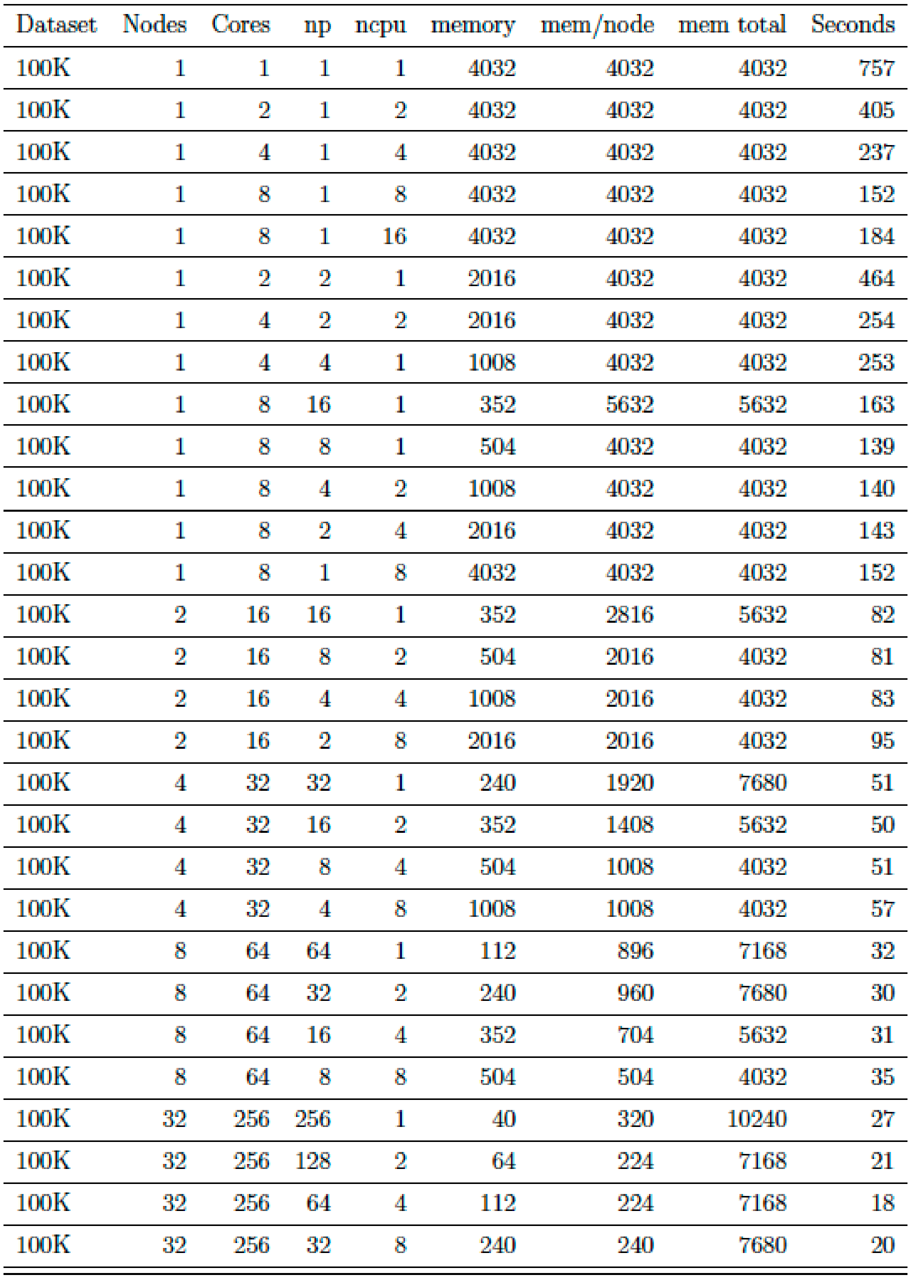
- -np: Total number of MPI processes used in a Distributed Memory Parallel job
- ncpu: number of SMP OpenMP threads
- memory: Size in words of allocated RAM for each MPI process. A word will be 4 and 8 bytes long for single or double precision executables, respectively.

3.4.2 Optimizing turnaround time

Table 1 shows how different combinations of number of MPI tasks (np) and OpenMP threads (ncpu) can be mapped across a given number of nodes and a total number of cores affecting elapsed time (last column). Up to 256 cores, (column 3), were used for these experiments on the SGI Altix ICE 8200EX of subsection 2.3.2. Except the single node runs, all 8 physical cores within each node are used by either an MPI task or an OpenMP thread. The fourth line shows all 8 cores running an SMP thread (ncpu=8) and the fifth line shows the slow down effect for 16 SMP thread oversubscribing the 8 physical cores in a mode called Hyper-Threading. The fastest elapsed times are obtained with a mix of MPI tasks and OpenMP threads for example, for 8 nodes, np=32 with ncpu=2 is faster than np=64 with ncpu=1, for 32 nodes, np=64 with ncpu=4 is faster than np=128 with ncpu=2 or np=256 with ncpu=1. Thus, hybrid mode allows faster turnaround times than with either pure SMP (OpenMP threads) or DMP (MPI tasks) mode of operation as available previously.

The 'memory' keyword value (in MB) used for an MPI task to process factorization and solution of the partitioned linear system in-core is shown in column 6. The next column multiplies this amount by the number of tasks running within each node, i.e. (np/Nodes) to give the memory used per node. In the 100K Nodes dataset case, this value in column 7 is always within the 24GB capacity of the node. The case where it is not is shown in subsection 3.4.3. The next column is the total memory used across all nodes to give an indication of how the run would fit within a Shared Memory Parallel system such as SGI Altix 450, 4700 or SGI Altix UV 10, 100, 1000 of section 2.4 and section 2.5.

This table shows the tradeoffs in applying various number of nodes to problems for maximizing throughput or turnaround time.



Dataset	Nodes	Cores	np	ncpu	memory	mem/node	mem total	Seconds
100K	1	1	1	1	4032	4032	4032	757
100K	1	2	1	2	4032	4032	4032	405
100K	1	4	1	4	4032	4032	4032	237
100K	1	8	1	8	4032	4032	4032	152
100K	1	8	1	16	4032	4032	4032	184
100K	1	2	2	1	2016	4032	4032	464
100K	1	4	2	2	2016	4032	4032	254
100K	1	4	4	1	1008	4032	4032	253
100K	1	8	16	1	352	5632	5632	163
100K	1	8	8	1	504	4032	4032	139
100K	1	8	4	2	1008	4032	4032	140
100K	1	8	2	4	2016	4032	4032	143
100K	1	8	1	8	4032	4032	4032	152
100K	2	16	16	1	352	2816	5632	82
100K	2	16	8	2	504	2016	4032	81
100K	2	16	4	4	1008	2016	4032	83
100K	2	16	2	8	2016	2016	4032	95
100K	4	32	32	1	240	1920	7680	51
100K	4	32	16	2	352	1408	5632	50
100K	4	32	8	4	504	1008	4032	51
100K	4	32	4	8	1008	1008	4032	57
100K	8	64	64	1	112	896	7168	32
100K	8	64	32	2	240	960	7680	30
100K	8	64	16	4	352	704	5632	31
100K	8	64	8	8	504	504	4032	35
100K	32	256	256	1	40	320	10240	27
100K	32	256	128	2	64	224	7168	21
100K	32	256	64	4	112	224	7168	18
100K	32	256	32	8	240	240	7680	20

Table 1: Hybrid mode MPI tasks and Threads for 100K Nodes dataset on SGI Altix ICE 8200, of subsection 2.3.2, (memory is in MB)

Selecting the right combinations of MPI processes and OpenMP threads, good scaling can be achieved as shown by Figure 9

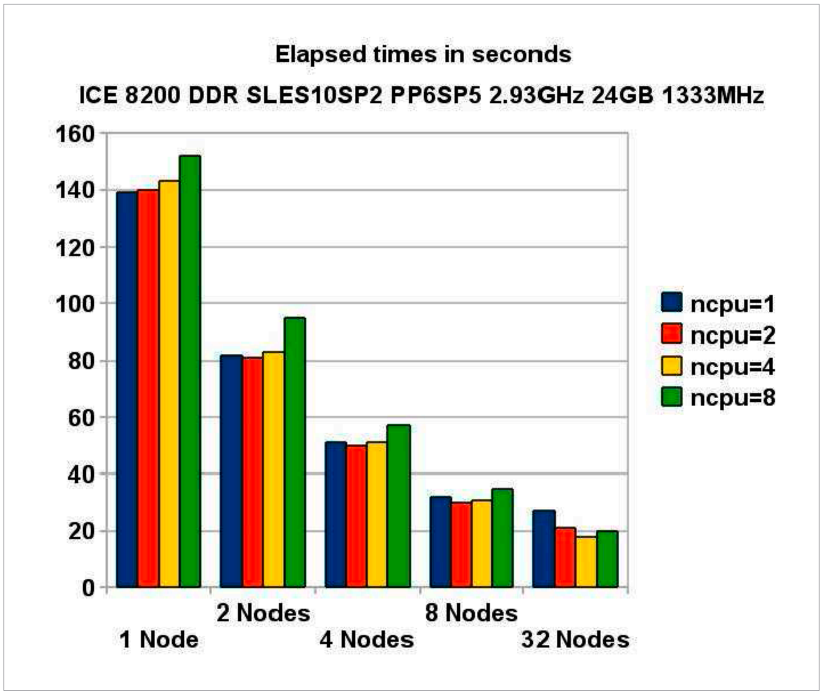


Figure 9: Scaling

3.4.3 Optimizing memory footprint

The larger 500K Nodes case of table 2 shows how 32 MPI tasks lead to a memory used per node (in MB) close to what might be available in some installations (16GB nodes in this example). In that case, applying more nodes and increasing the number of MPI tasks to 64 would allow the solution to run in-core. An alternative would be to use a Shared Memory Parallel architecture as shown in the last 4 lines run on the SGI Altix UV 100 with 12 2.66Ghz 6-core Xeon X7542 (total of 72 cores) of section 2.5. A total of 72GB RAM for this single node would be enough to run in-core.

Dataset	Nodes	Cores	np	ncpu	memory	mem/node	mem total	Seconds
500K	4	32	32	1	1904	15232	60928	967
500K	8	64	64	1	1005	8044	64358	591
500K	9	72	72	1	984	7872	70848	570
500K	9	72	36	2	1364	5456	49104	577
500K	1	72	72	1	984	70848	70848	608
500K	1	72	36	2	1432	51552	51552	577
500K	1	72	24	3	2560	61440	61440	613
500K	1	72	18	4	2872	51696	50832	584

Table 2: Hybrid mode MPI tasks and Threads for 500K Nodes dataset on SGI Altix XE 1300 (section 2.2) and SGI Altix UV 100 (section 2.5), (memory is in MB)

4. Analysis of Benchmark Results

4.1 Effect of Interconnect

The choice of GigE or Infiniband interconnect integrated to a server may affect performance. Figure 10 plots the ratios of elapsed times using GigE over those using Infiniband interconnect run on Altix XE 1300 cluster (section 2.2). For the benchmark series used in this study, factors above 1.2 can be observed for all sizes of datasets and at higher MPI task counts, the performance difference can be over 60%.

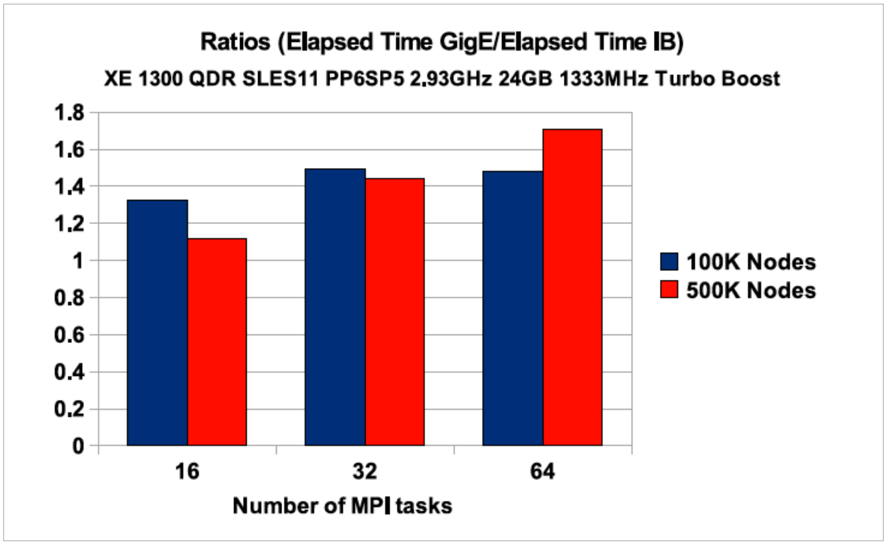


Figure 10: Effect of Interconnect for 100K and 500K Nodes datasets and increasing number of MPI tasks, SGI Altix XE 1300 Cluster (section 2.2)

4.2 Effect of core frequency

The processor core frequency can be expected to affect performance in a non-linear fashion since the problem and the system may be subject to other bottlenecks than pure arithmetic processing. Figure 11 plots the elapsed times normalized by the slowest processor run on Altix XE 1300 cluster (section 2.2). For the benchmark series used in this study, increased core frequencies improve performance linearly for 100K Nodes/1 and 8 core runs but show diminishing returns in the other more complex cases. The ideal performance is represented by the straight line 'Frequencies'.

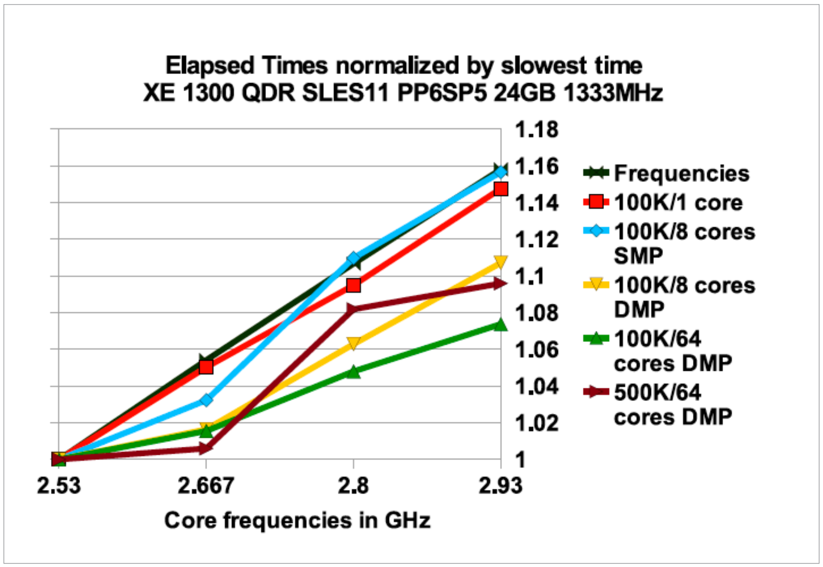


Figure 11: Effect of core frequency on performance for various test cases, SGI Altix XE 1300 Cluster (section 2.2), (Turbo Boost disabled)

4.3 Effect of Turbo Boost

The Turbo Boost is a feature first introduced in the Xeon 5500 series processors, which can increase performance by increasing the core operating frequency within the controlled limits of the processor. For most simulations, utilizing Turbo Boost technology can result in improved runtimes. By design the mode of activation is a function of how many cores are active at a given moment as maybe the case when OpenMP threads or MPI processes are idle under their running parent. Similarly to 4.2, its effects may be mitigated by the presence of other performance bottlenecks than pure arithmetic processing. Figure 12 plots the ratios between elapsed times with Turbo Boost OFF versus ON run on Altix XE 1300 cluster (section 2.2). It shows that for the benchmark series used in this study, Turbo Boost improves performance for 100K Nodes/1 core, 12 cores SMP and 500k 72 cores DMP runs up to the ratio of the maximum frequency over nominal value which is 13% (magnitude represented by first bar). The 100k/12 cores DMP is not helped by Turbo Boost, probably because all the cores are kept busy all the time whereas the 100k/82 cores DMP represent a lightly loaded case for each core which allows Turbo Boost to accelerate all the core frequencies.

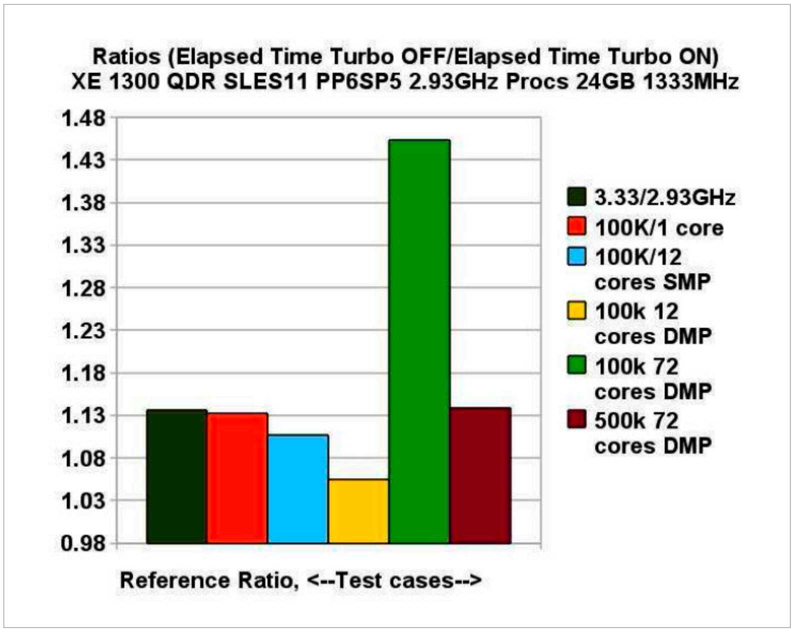


Figure 12: Effect of Turbo Boost on performance for various test cases. SGI Altix XE 1300 Cluster (section 2.2)

4.4 Effect of Hyper-Threading Technology

Hyper-Threading (HT) is a feature which can increase performance for multi-threaded or multi-process applications. It allows a user to run 16 (instead of 8) OpenMP threads or MPI processes on 8 physical cores per node. It is sometimes beneficial for 1 or 2 nodes but at and above 3 nodes communication costs added by the doubling of threads or MPI processes mitigates the benefits. Figure 13 plots the ratios between elapsed times run on SGI Altix ICE 8200 (subsection 2.3.1) without using Hyper-Threading vs using it. Based on the benchmark study no improvement is observed for 100K and 500K Nodes.

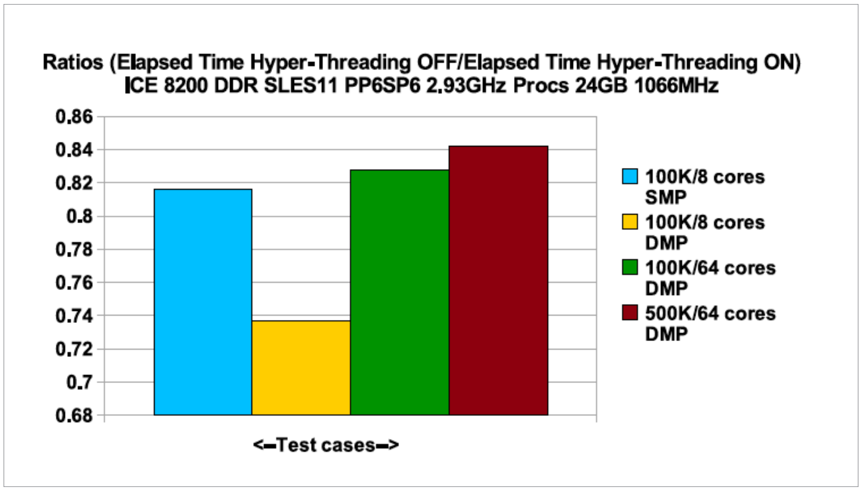


Figure 13: Negative effect of Hyper-Threading on test cases tried. SGI Altix ICE 8200 (subsection 2.3.1)

4.5 Effect of Memory Speed

Memory speed may be important when data motion represents either a bandwidth bottleneck or latency limitation. Figure 14 plots the ratios between elapsed times run on SGI Altix ICE 8200 (section 2.3) using 1066MHz vs 1333MHz DIMMs. It shows that for the benchmark series used in this study, in the two cases run, the improvements are far below the nominal ratio of memory speed which is 25% (magnitude represented by first bar).

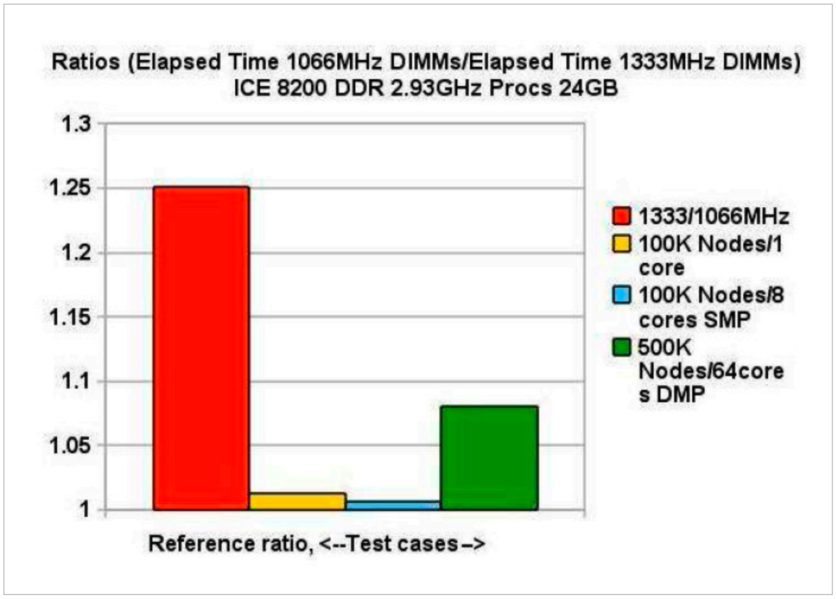


Figure 14: Effect of memory speed is negligible for test cases tried. SGI Altix ICE 8200 (section 2.3)

4.6 Effect of Filesystem

Figure 15 plots the ratios between elapsed times run on SGI Altix XE 1300 (section 2.2) using disk vs memory filesystem. It shows that for the benchmark series used in this study, using /dev/shm when possible can save up to 10% elapsed time.

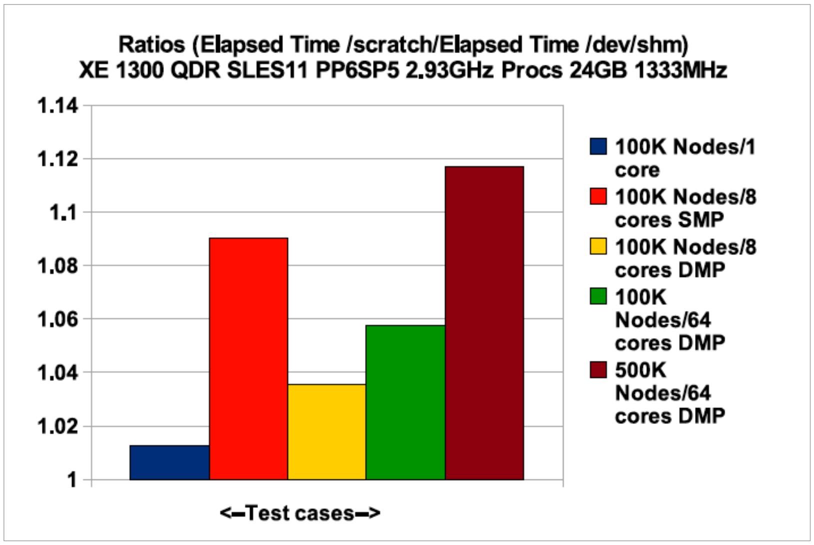


Figure 15: Effect of file system, SGI Altix XE 1300 (section2.2)

5. Access to benchmark systems

SGI offers Cyclone, HPC on-demand computing resources of all SGI advanced architectures aforementioned. There are two service models in Cyclone. Software as a Service (SaaS) and Infrastructure as a Service (IaaS). With SaaS, Cyclone customers can significantly reduce time to results by accessing leading-edge open source applications and best-of- breed commercial software platforms from top Independent Software Vendors (ISVs) such as LS-DYNA from LSTC.

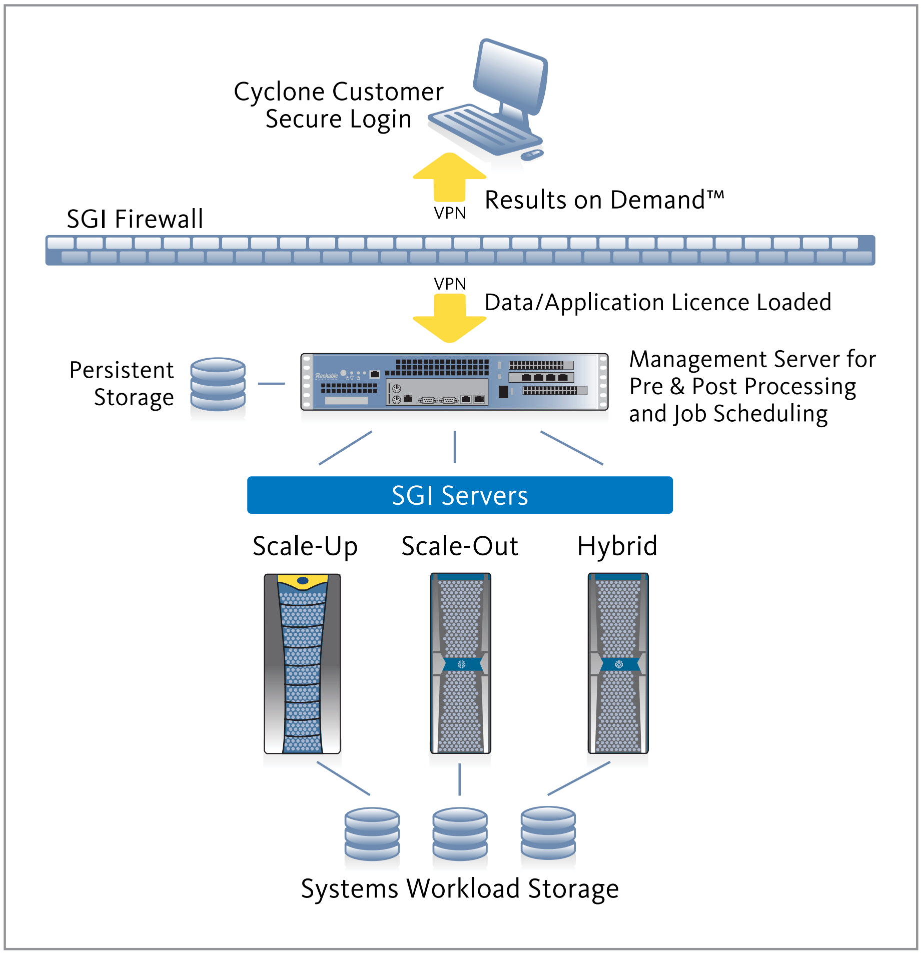


Figure 16: SGI Cyclone – HPC on-demand Cloud Computing

Conclusions

This study showed how interconnect, core frequency, Turbo Boost, Hyper-Threading, memory speed, filesystem effects on implicit LS-DYNA performance can be gauged for various sizes of datasets. At the same time, using the right combination of LS-DYNA hybrid mode parallelism features, most efficient use of either shared or distributed memory systems can be achieved to optimize throughput or turnaround times. Both these insights and optimizations can be used to architect a solution using the full range of systems from SGI to meet complex analysis requirements.

Attributions

LS-DYNA is a registered trademark of Livermore Software Technology Corp. SGI, Octane, Altix, ProPack and Cyclone are registered trademarks or trademarks of Silicon Graphics International Corp. or its subsidiaries in the U.S. or other countries. Xeon and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Linux is a registered trademark of Linus Torvalds in several countries. SUSE is a trademark of SUSE LINUX Products GmbH, a Novell business. All other trademarks mentioned herein are the property of their respective owners.

References

[1] Olivier Schreiber, Scott Shaw, Brian Thatch, and Bill Tang. "LS-DYNA on Advanced SGI Architectures". In Proceedings of 11th International LS-DYNA Users Conference, Dearborn, MI, June 2010.

[2] Dr. C. Cleve Ashcraft, Roger G. Grimes, and Dr. Robert F. Lucas. "A Study of LS-DYNA Implicit Performance in MPP". In Proceedings of 7th European LS-DYNA Conference, Austria, 2009.

[3] Dr. C. Cleve Ashcraft, Roger G. Grimes, and Dr. Robert F. Lucas. "A Study of LS-DYNA Implicit Performance in MPP (Update)". 2009.

[4] SGI. SGI Developer's Guide. Silicon Graphics International, Fremont, California, 2009.

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

North America +1 800.800.7441
Latin America +55 11.5185.2860
Europe +44 118.912.7500
Asia Pacific +61 2.9448.1463