New Features in LS-DYNA® HYBRID Version

Nick Meng[1], Jason Wang[2], Satish Pathy[2]

[1] *Intel Corporation, Software and Services Group*
[2] *Livermore Software Technology Corporation*

Abstract

Numerical noise arising from different MPP core counts compels users to fix the number of cores used by LS-DYNA® MPP e.g. during a vehicle development program. This fixed core count limits job turn-around time and flexibility in managing computing resources. In addition, using a large number of cores for calculations diminishes scalability with pure MPP.

LS-DYNA® HYBRID addresses these issues through the use of both MPI + OpenMP technology. LS-DYNA HYBRID is able to produce consistent numerical results when changing the number of OpenMP threads thereby reducing job turnaround time. In addition, LS-DYNA HYBRID can greatly reduce the number of processors involved in message passing and achieve much better scalability over large number of cores. Furthermore, for the implicit applications LS-DYNA HYBRID not only reduces the memory requirement per node but also decreases IO activity.

Currently, LSTC and Intel® teams are working together with a customer to evaluate LS-DYNA HYBRID code using a custom QA (Quality Assurance) suite. The consistency and performance will be discussed in this paper.

Introduction

Since 2000s, the auto industry has increasingly relied on high-performance computing (HPC) technology and mechanical computer-aided engineering (MCAE) applications to drive innovation in vehicle development. To improve simulation accuracy, the problem size is continually increasing [1] which increases the solution time. Complying to stricter global safety standards combined with shortened development cycle, makes auto OEM's task very challenging in developing simulation models.

Due to the numerical noise [2] arising from different MPP core counts, users often fix the number of cores used in LS-DYNA MPP, throughout their product development. Because of this, it limits their flexibility in managing the computing resources.

With increasing problem size, more computing resource is required. Turnaround time is increased dramatically too. For example, a typical full car crash simulation model [1] with 10M elements takes about 43 hours using 512 cores on a RISC-based cluster. Meanwhile, the parallel computing efficiency is decreased sharply due to a native feature of MPI collective functions cost [3] and MPP contact algorithm.

LS-DYNA HYBRID (MPI + OpenMP) addresses these issues by combining the LS-DYNA® SMP version and the LS-DYNA MPP version.

New Features in LS-DYNA HYBRID Version

LS-DYNA HYBRID was implemented for the X86-64 platform by LSTC and Intel® team. We combined the SMP version and the MPP version together and parallelized all the related subroutines without fundamentally changing the code for compatibility reasons. Three new key features have emerged while implementing LS-DYNA HYBRID version.

- **Consistent numerical results under certain condition**
- **Economic implicit mechanics solution**
- **Super performance on large number of cores**

Consistent numerical results under certain condition

LSTC developed a 'consistent' option in LS-DYNA SMP version per customers' requirement in SMP era. With this option, users will get identical numerical results while changing number of SMP threads. Most LS-DYNA SMP users currently use this option (ncpu= –N) [4] in their simulations. Even though there is a **15%** average performance penalty with this option the advantage of being able to manage their computing resources to full efficiency outweighs the performance penalty. Therefore LS-DYNA SMP users can run their jobs, according to their computing resources and the result remains consistent. LS-DYNA HYBRID has put numerical consistency as the first consideration to allow users to change the number of SMP threads and produce the same numerical results. This feature has been thoroughly and successfully tested with customers' production QA models with latest LS-DYNA HYBRID version.

What is the certain condition? As mentioned in the abstract, customers need to fix number of cores used in LS-DYNA MPP throughout their product development due to the numerical noise arising from different number of MPP core counts. You need to fix number of cores for MPI process at first. Once the number of MPI process is chosen, for example, you decided to use 32 MPI processes in your product development. Then you can run your jobs with 32 cores (32MPI x 1OMP), 64 cores (32MPI x 2OMP), 96 cores (32MPI x 3OMP), 128 cores (32MPI x 4OMP), up to 256 cores 32MPIx 8OMP) and you can get identical numerical results. **So the certain condition is fixing the number of cores for MPI processes**.

Nine production QA models from a customer were selected as the test suite for verifying numerical results and doing performance comparison. The models in the test suite cover different types of vehicle crash simulations and 12 MPI processes were selected for verification. All nine production QA models were tested with LS-DYNA MPP R4.2.1 and LS-DYNA HYBRID R4.2.1 on an Intel® Xeon® cluster, 12 cores, 24 cores, and 48 cores were used for LS-DYNA HYBRID runs. The numerical results and scalability was verified and demonstrated.

➢ Performance of LS-DYNA HYBRID R4.2.1 Version

First, the performance penalty due to the use of consistent option is evaluated on an Intel® Xeon® X5560 cluster. Table 1 shows the performance difference between LS-DYNA MPP and

LS-DYNA HYBRID. There was an average performance penalty of about **8.4%** with the test suite that was used**.** It is safe to say a performance penalty of **8-9%** has to be expected if users migrate from LS-DYNA MPP version to LS-DYNA HYBRID. This penalty is acceptable to most users because of several advantages over pure MPP version.

12 cores	Job1	Job2	Job3	Job4	Job5	Job6	Job7	Job8	Job9	avg
Hybrid	77862s	50983s	48452s	11031s	24345s	10450s	6871s	628s	7898s	
Pure MPP	69763s	47172s	43822s	11172s	22076s	9734s	6237s	596s	7002s	
Ratio	**11.6%**	**8.1%**	**10.5%**	**-1.2%**	**10.3%**	**7.4%**	**10.2%**	**5.37%**	**12.8%**	**8.4%**

Table 1: 12-core elapsed time of nine QA models with LS-DYNA® MPP and LS-DYNA HYBRID and performance difference between LS-DYNA MPP and its LS-DYNA HYBRID

Secondly, all QA models were tested on an Intel® Xeon® X5460 cluster for evaluating scalability at the customer site. Figure 1 and Table 2 below demonstrate the reasonable scalability of LS-DYNA HYBRID version. But, the scalability varies between these models as they involve different load cases. Basically, most typical scalabilities are covered here.

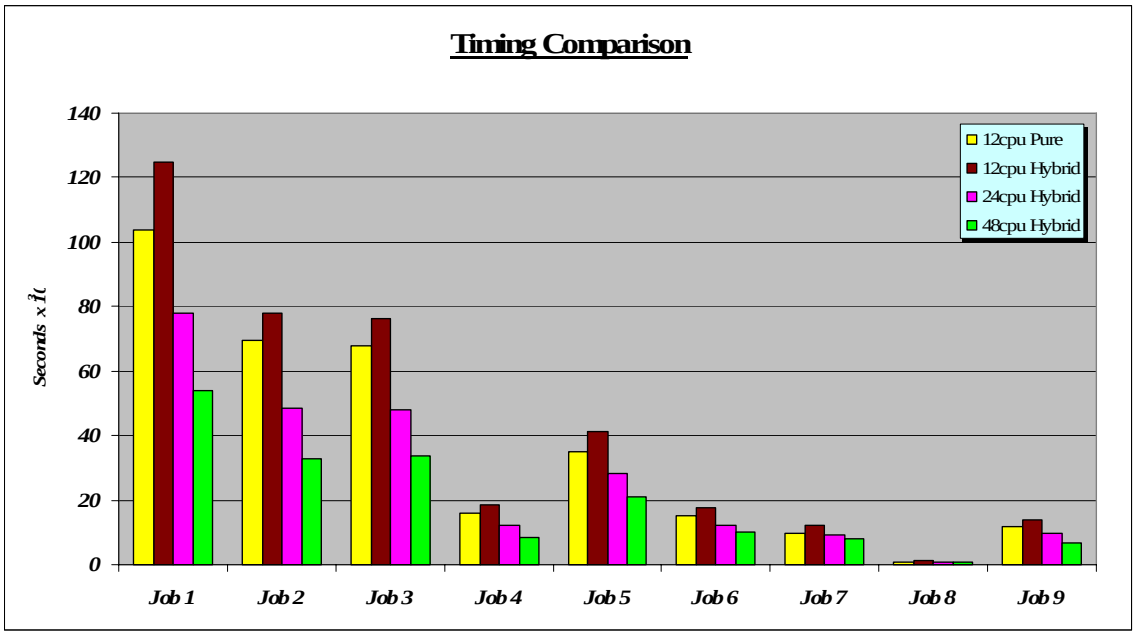

Figure 1: Elapsed time of nine production QA models on 12 cores, 24 cores, and 48 cores. LS-DYNA MPPR4.2.1 and HYBIRD LS-DYNA R4.2.1 both are used. Scalability of HYBRID LS-DYNA is demonstrated.

Speedup	Job1	Job2	Job3	Job4	Job5	Job6	Job7	Job8	Job9
12x-1 job	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
12x-2 job	1.60	1.60	1.59	1.51	1.46	1.45	1.33	1.50	1.43
12x-4 job	2.31	2.37	2.26	2.17	1.97	1.80	1.54	1.14	2.08

Table 2: Speedup of nine production QA models, tested on an Intel® Xeon® X5460 cluster.

- **Identical numerical results of LS-DYNA HYBRID Version**

Identical numerical results are the key feature in LS-DYNA HYBRID version. All ASCII files were verified for each model, ASCII files generated from 12 x-1 jobs, 12 x-2 jobs, and 12 x-4 jobs are identical for each model. Figure 2 shows a snapshot of output of " vimdiff 12x-1/nodout 12x-4/nodout".

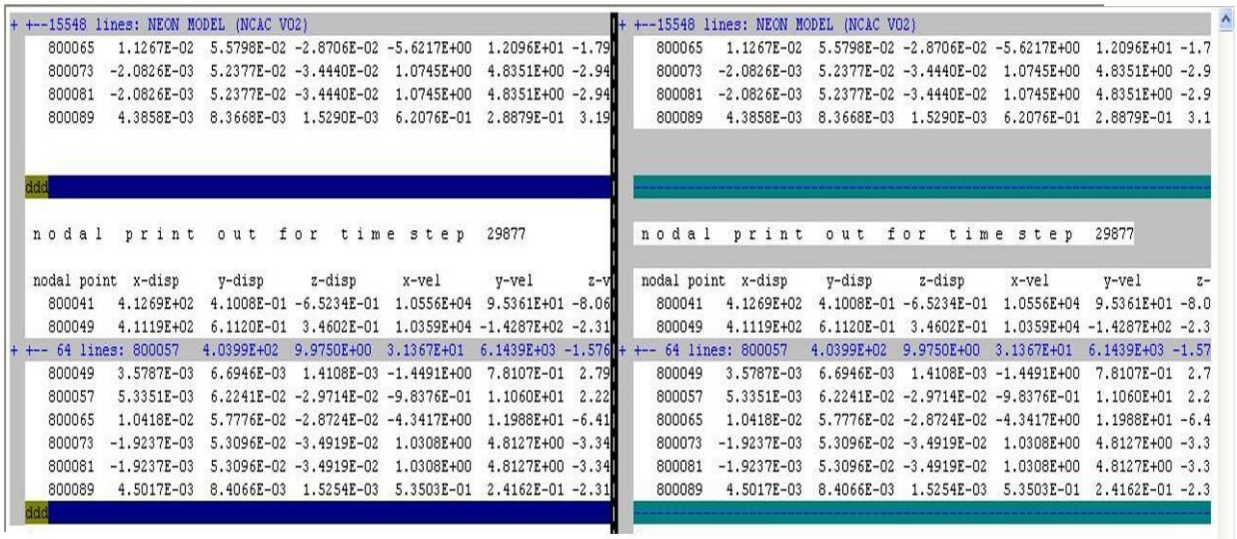

Figure 2: Snapshot of output of " vimdiff 12x-1/nodout 12x-4/nodout"..

The response from four of the nine production QA models was selected to illustrate and publish in this paper. Figure 3-6 shows the vehicle deceleration response curve. All the results match identically as expected.

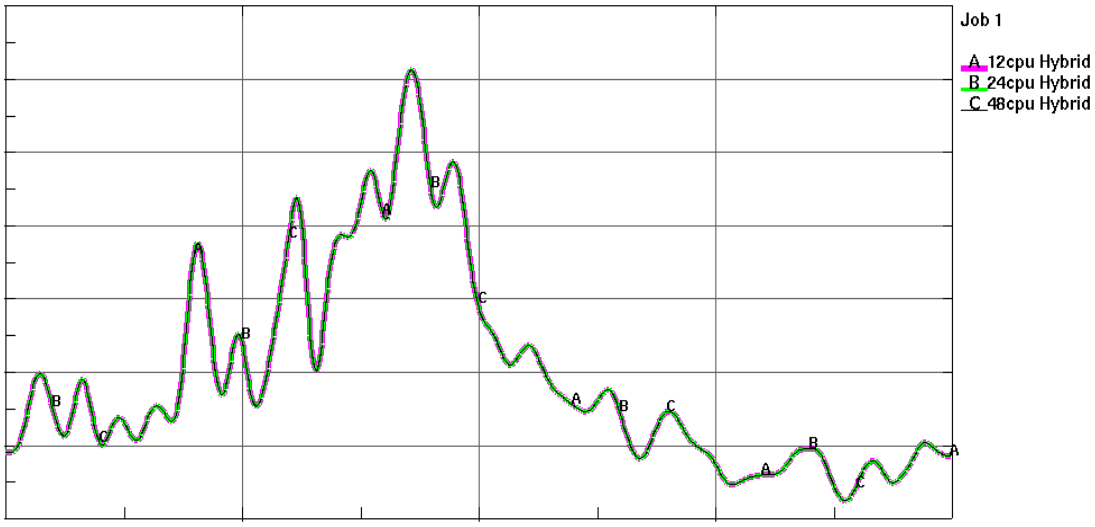

Figure 3: Job1 - vehicle deceleration response as measured at left rear sill.

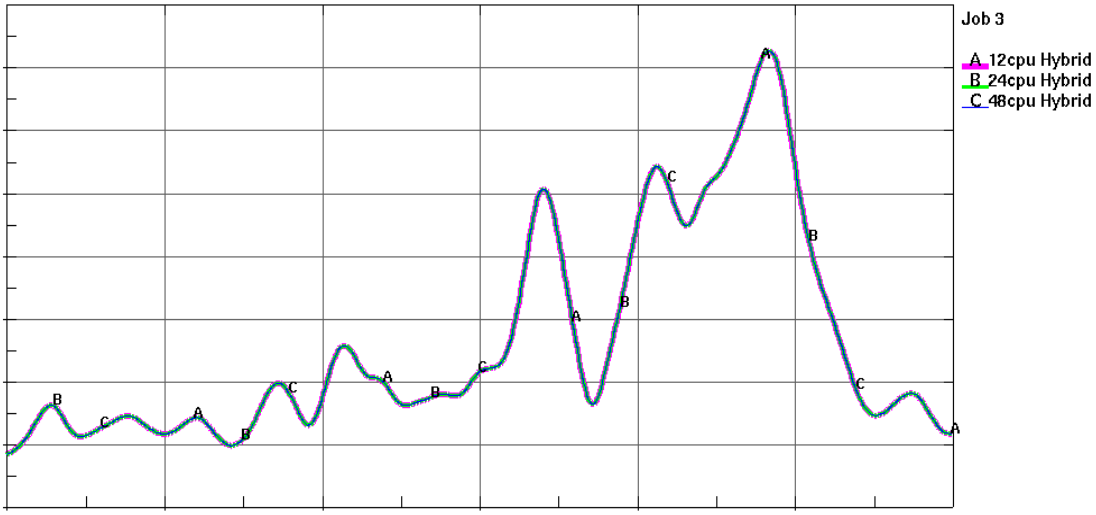

Figure 4: Job3 - vehicle deceleration response as measured at left rear sill.

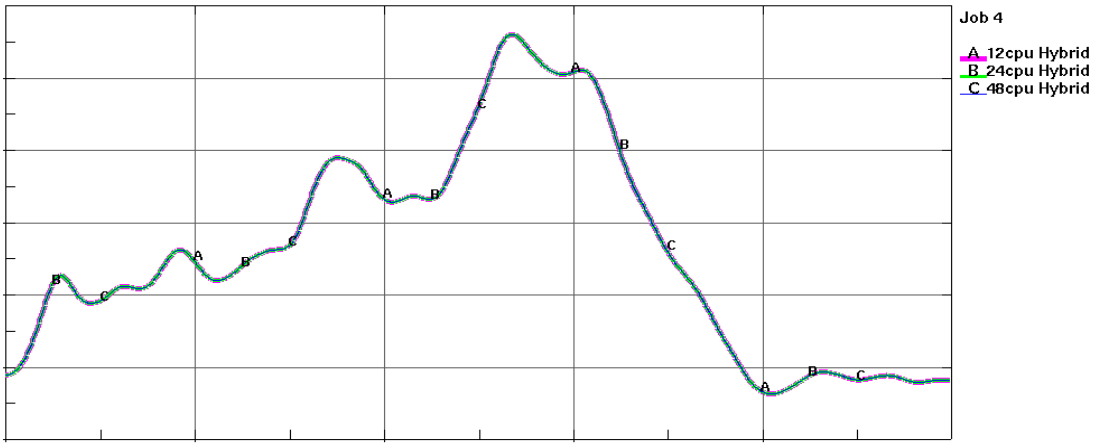

Figure 5: Job4 - vehicle deceleration response as measured at left rear sill.

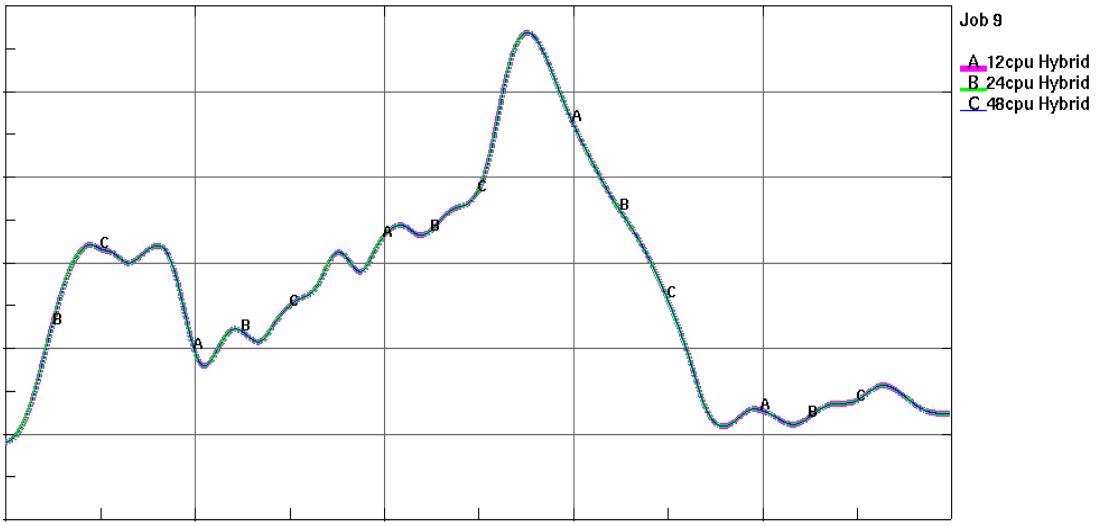

Figure 6: Job9 - vehicle deceleration response as measured at left rear sill.

Economic implicit mechanics solution

Historically, LS-DYNA implicit users have suffered from poor I/O performance, lack of memory space and memory bandwidth in Intel64® -based system when they use MPI version of implicit applications. One of the solutions is to run one MPI process per node with the in-core solver due to limited memory and I/O resource in each node. But then users couldn't use all of the cores for their simulation and hence computing resource is underutilized. LS-DYNA

HYBRID solves this problem very well through SMP technology within each MPI thread. LS-DYNA Implicit users can now solve their problems on existing Intel64® cluster quickly with LS-DYNA HYBRID

The CYL1E6 model, which is an implicit benchmark model developed by Atomic Weapons Establishment (UK), is the first test problem used in this demonstration. The model consists of 6 nested cylinders held together by contact. There are 1 million solid elements (3.27 million degrees of freedom) in the model.

Two systems were selected as benchmark platforms. The first platform is a new 4 socket (8 cores on each socket) Intel® Xeon® X7560 system at 2.26 GHz with 128GB memory for a total of 32 cores in a single node. The other system is an 8-node dual-socket Intel® Xeon® X5560 cluster with 24GB memory. Each X5560 processor socket contains 4 cores and runs at 2.8GHz. The cluster is connected with a QDR (quad-data-rate) IB (Infiniband) network.

In-core solver is a preferred solution as suggested by LSTC, and the in-core solver is a reasonable solution in **existing** X86-64 based system with poor file system. So the in-core solver is used for all tests. During initial test, it was noticed that 31.3GB memory was required for in-core solver in each MPI process for 4 cores of pure MPP job and 7.9GB memory is required for in-core solver in each MPI process for 32 cores of pure MPP job. Since there was only 128GB memory, and a 32 core pure MPP job requires 256GB memory, a single X7560 node could not handle this pure MPP job. However, we can run 4 cores pure MPP job in the single X7500 node, which requires 125GB of memory. This is the minimum choice for LS-DYNA MPP version. LS-DYNA HYBRID overcame this shortcoming. All 32 cores can be used with the same memory requirement.

Unfortunately, the minimum choice for the model on the 8 node X5560 cluster was to use 8 nodes for an 8 core pure MPP job. The reason is there is only 24GB memory per node and each MPI process requires 16GB memory. Table 3 below shows performance of CYL1E6 model.

CYL1E6 Model 921K elements 3.27 M DOFs	Intel® Xeon® X7560 system 32 cores @ 2.26Ghz 128GB memory		Intel® Xeon® X5560 cluster 8 nodes with 8 cores @2.8Ghz 24GB memory	
Cores & nodes used	4 cores 1 node Pure MPP Minimum choice	32 cores 1 node Hybrid Best choice	8 cores 8 nodes Pure MPP Minimum choice	64 cores 8 nodes Hybrid Best choice
Memory requirement	31.2GB per MPI process	31.2GB per MPI process	15.6GB per MPI process	15.6GB per MPI process
Elapsed Time	44013s	7047s	18521s	5541s
Speedup	1.00	6.25	1.00	3.34

Table 3: Elapsed time of minimum run and best run on two benchmark platforms.

The CYL2E6 model, which is also an implicit benchmark model developed by Atomic Weapons Establishment (UK), is the second test problem used in this demonstration. The model

consists of 6 nested cylinders held together by contact. There are 2 million solid elements (6.52 million degrees of freedom) in the model.

The model needs 13GB per MPI process for 32c pure MPP job, but there was only 16GB memory in each node of an Intel®Xeon5460 cluster, so the minimum requirement to run pure MPP on the cluster is 32 nodes. The benchmark was performed using LS-DYNA MPP and LS-DYNA HYBRID for performance comparison on 32 nodes. Figure 7 shows the performance of CYL2E6 model on the 32 nodes Intel® Xeon® X5460 cluster.

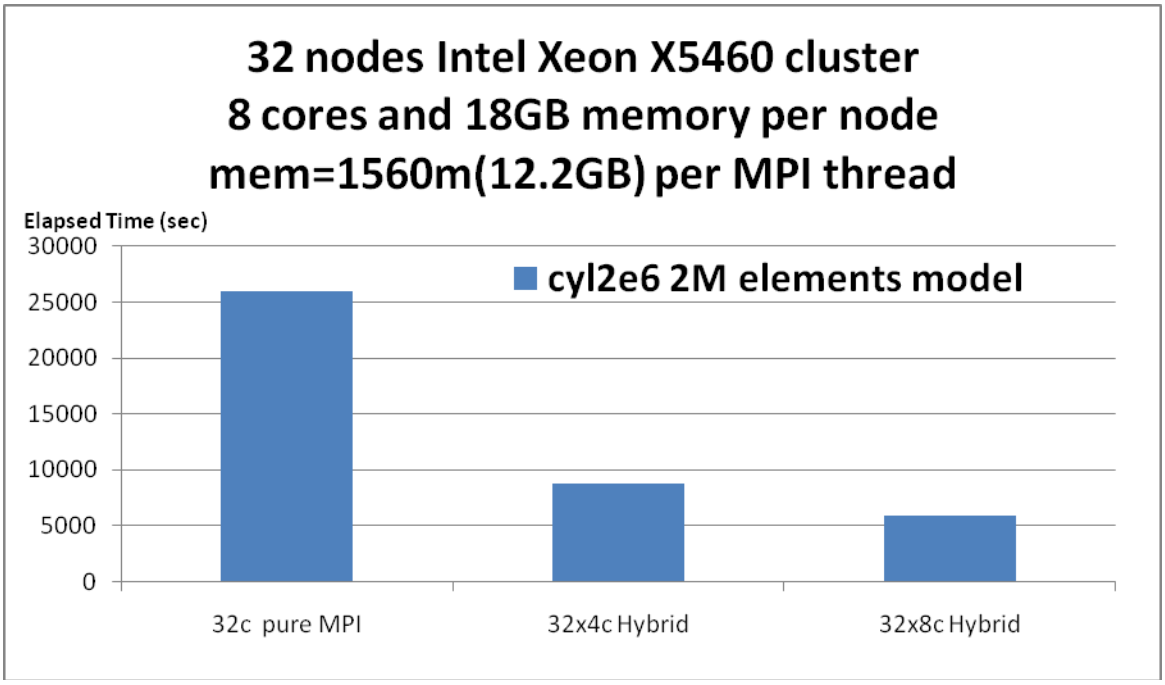

Figure 7: Elapsed Time of CYL2E6 model on 32 nodes Intel® Xeon® X5460 cluster

The solution of the global stiffness linear system for implicit mechanics requires substantial amounts of memory, far more that the explicit solution approach. Running LS-DYNA MPP in a distributed manner using MPI on a compute cluster is the most cost effective to get the required memory for the implicit execution. The usual approach is to use all of the compute cores per computational node on the cluster. But this does not always work for implicit mechanics. The memory requirement per compute core does not decrease linearly as the number of cores increases. There is a minimum amount of memory required per core. The LS-DYNA HYBRID version allows full use of all of the compute cores per computational node using OpenMP threads but keeps the memory in one piece for the minimum memory requirement.

Another key point is I/O. To reduce the substantial amounts of memory for solving the stiffness linear system I/O files are used to store the factorization and for temporary storage. Each MPI process has one file to store its part of the factorization and one file for temporary storage. Using the pure MPI implementation there will be one active I/O file for each compute core. Since there is usually just one I/O device per node on the cluster the overall I/O performance degrades like rush hour traffic on the highway. 8 active I/O files, one for each of the 8 cores, will fight against each other for the same resource. The LS-DYNA HYBRID implementation will only have one

I/O file for the factorization and one for temporary storage per compute node. So there is no I/O degradation.

Super performance on large number of cores

- **Root cause of poor scalability on large number of cores**

The original motivation for developing LS-DYNA HYBRID was to pursue better performance when solving large models on a large cluster. As the number of cores increase, MPI collective functions cost also increases rapidly. Meanwhile, the number of subdomains of the model is increased, for this causes the load imbalance issue and poor parallel efficiency in the contact algorithm both of which are detrimental. The parallel efficiency of a job, run on a large cluster is very important. LS-DYNA HYBRID reduced the number of MPI processes and number of subdomains and hence reduced the MPI collective functions cost.

Figures 8 and 9 show the cost of MPI collective communication of a customer model generated by " Intel® Trace Analyzer and Collector Tool" on 4 nodes and 32 nodes.. Red bars and Brown bars indicate the cost of MPI collective functions. Blue bar and Gray bar show the cost of LS-DYNA MPP. Clearly, the cost of MPI collective functions increases quickly along with increase in number of nodes. LS-DYNA HYBRID is implemented to address this problem through the use of both MPI + OpenMP technology.

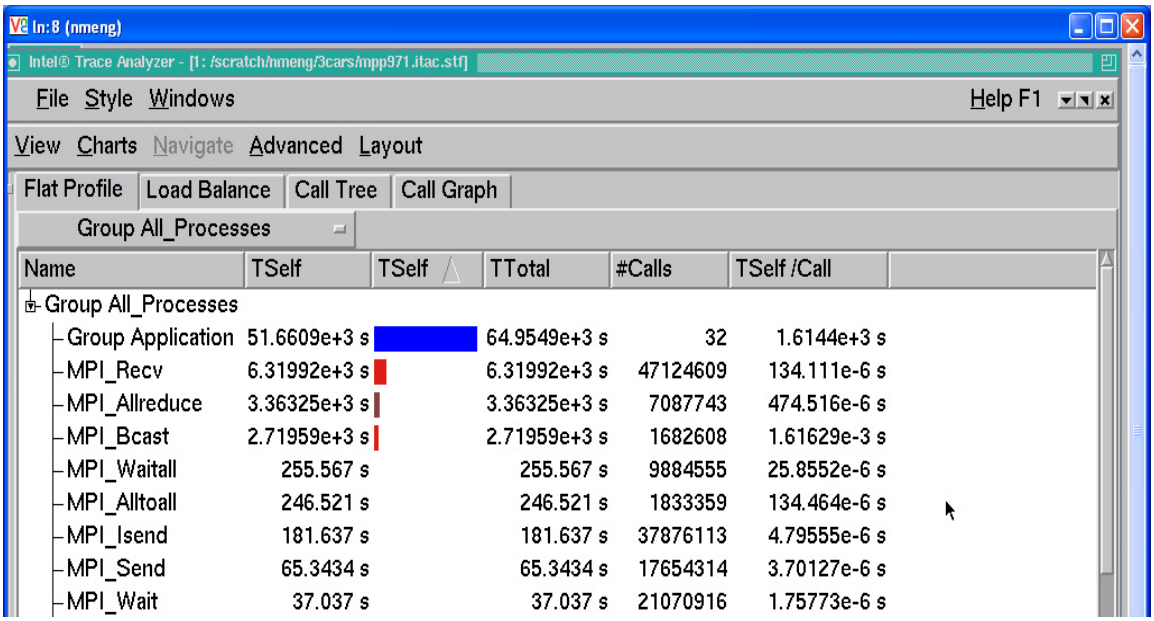

Figure 8: Profile data of a customer model on 4 nodes generated by Intel Trace Analyzer and Collector Tool.

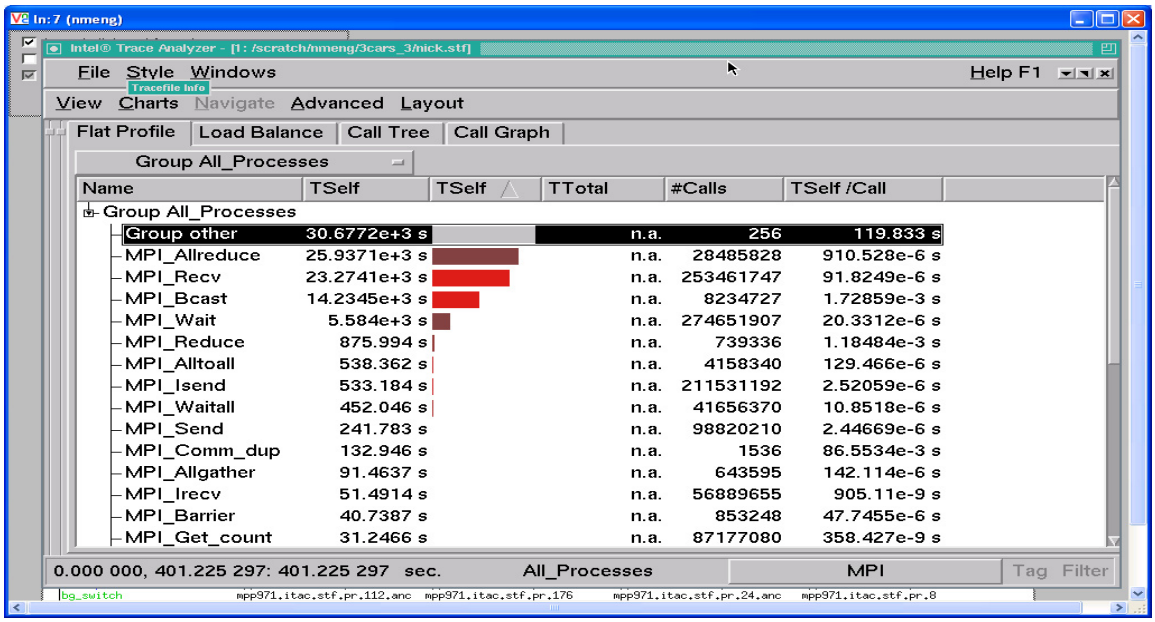

Figure 9: Profile data of a customer model on 32 nodes generated by Intel® Trace Analyzer and Collector Tool.

➢ Performance comparison between LS-DYNA MPP and LS-DYNA HYBRID

The popular 2.4 million elements car2car model, LS-DYNA standard benchmark model, is selected for this next comparison. LS-DYNA MPP Version R4.2.1 and its corresponding version of the LS-DYNA HYBRID both are used for the performance comparison. The benchmark is tested on two Intel® Xeon® clusters. See Table 4 in below.

Node Type	Intel® Xeon® X5560 cluster	Intel® Xeon® X5470 cluster
Processor	2 sockets QC X5560 2.8Ghz	2 sockets QC X5470 3.2Ghz
Memory	24GB memory @ 1333Mhz	16GB memory @1066Mhz
Interconnect	InfiniBand QDR	InfiniBand DDR
OS	RH 5.3U2	RH5.2U2
MPI library	Intel MPI 3.2.1	Intel MPI 3.2.1

Table 4: Benchmark system configuration

The car2car model is tested with LS-DYNA MPP R4.2.1 and LS-DYNA HYBRID R4.2.1 on 32 nodes and 64 nodes on Intel Xeon® X5470 cluster. The numbers in Figure 10 indicate that the performance of 256 MPI processes job is about **15.4%** slower than performance of 64 MPI processes with 4 OpenMP threads job in 32 nodes. An **18.1%** performance gap on 64 nodes was observed. Meanwhile, the same job was tested on 32 nodes in Intel® Xeon® X5560 cluster. The performance gap between LS-DYNA MPP and LS-DYNA HYBRID was about **18.2%.**

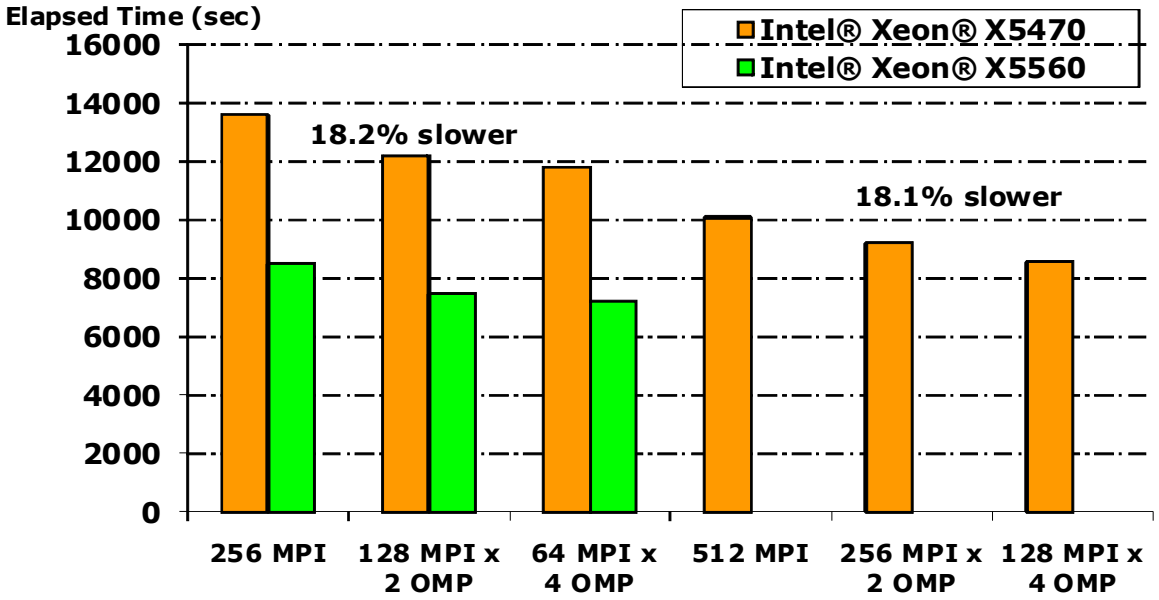

Figure 10: 256-core elapsed times of the pure MPP and the HYBRID versions and the 512-core elapsed time of the pure MPP and the HYBRID on Intel® Xeon® X5470 and X5560 clusters.

Future Work

As described in the paper, the first target of implementation of LS-DYNA HYBRID is achieved. Three new key features are implemented in LS-DYNA HYBRID R4.2.1 and R5. LSTC and Intel® teams will continue to implement more features in next major LS-DYNA HYBRID version. As mentioned in the abstract, the development team worked with a customer to evaluate LS-DYNA HYBRID. The development team would like to invite more customers to help us improve the hybrid code.

References

[1]: Misuhiro Makino, "The Performance of 10-Million Element Car Model by MPP Version of LS-DYNA on Fujitsu PRIMEPOWER", 10th International LS-DYNA® Users Conference, Dearborn, MI, USA June, 2008
[2]: Paul Du Bios and Thomas Frank, "CRASH-SIMULATION OF HAT-SECTIONS RELIABILITY OF THE NUMERICAL MODEL" 3rd European LS-DYNA Conference in Paris, French, June, 2001.
[3]: J. Liu, et al, "Performance Comparison of MPI Implementations over InfiniBand, Myrine and Quadrics", Conference on High Performance Networking and Computing. Proceedings of the 2003 ACM/IEEE conference on Supercomputing.
[4]: LS-DYNA Keyword User's Manual Version 971, Livermore Software Technology Corporation, Livermore, CA 2007.