The Performance of Car Crash Simulation by LS-DYNA® Hybrid Parallel Version on Fujitsu FX1

Kenshiro Kondo
Technical Computing Solution Unit, Fujitsu Limited
9-3 Nakase 1Chome Mihama, Chiba, 261-8588, Japan

Mitsuhiro Makino, PhD
DYNAPOWER Corporation.
6-402 Oyumino2-34-1, Midori, Chiba, 266-0031, Japan

Abstract

The number of elements of car simulation for crash analysis has increased rapidly over recent years in order to achieve better accuracy. Several MPP versions of LS-DYNA using variations of MPI have been widely applied to car crash simulation for better job turnaround. The modern computing hardware has been exploring multi-core CPU technology since the performance of single-core can not meet the current and future demands. The Hybrid Parallelization version of LS-DYNA, taking the advantage of multi-core computing platforms, provides an efficient tool for large- model car crash simulation. This hybrid version combines threads parallelization using OpenMP across multiple cores within a node as well as process parallelization using MPI between nodes to achieve maximum parallelization of the analysis.

In this paper, we revealed performance bottle neck of MPP version for highly parallelization first; then we showed performance and examined accuracy of solution of LS-DYNA Hybrid version. We tested both 2-million and 10-million elements Caravan crash models on both LS-DYNA MPP and Hybrid versions. The superior performance of Hybrid version demonstrated the feasibility of car crash simulation using massively parallel processors.

Introduction

At present, car crash simulation model is using around 2 to 6 million elements, which has become almost double over the last three years. It is believed such trend of employing larger models will continue in order to achieve better accuracy.

On the other hand, the future improvement on the hardware performance of a single CPU system is not going to keep up the same pace as before. The development of multi-core CPUs, packing two or more cores in one CPU, has become a mainstream. Users not only can enjoy lower hardware cost and lower power consumption, but also should observe better computational performance once the application software has been efficiently designed with the advantage of multi-core architecture.

The most common programming paradigm used for car crash simulation at present is MPP (Massively Parallel Processing) version. MPP version uses Message Passing Interface (MPI) language to achieve parallelism. Currently usage of 16 to 32 parallel processes per job has been considered practical for car crash analysis. As simulation models grow bigger, the number of parallel processes used for car crash simulation will increase. In the near future, large-scale analysis over hundreds of cores per parallel job becomes feasible due to the trend in the development of multi-core processor [1]. The shortcomings of MPP version, as described in the following sections, would become critical and need to be addressed. One potential solution is Hybrid Parallelization version which executes threads parallel within a node without the overhead of message passing, and handles process parallelization using MPI between nodes.

Shortcomings of MPP in Large Parallel Environment

There are two shortcomings for MPP in large parallel environment. One of them is communication. For a fixed problem size, as the number of processes increases, the function calls in MPI message communication grows. The ratio of communication to computation would become disproportional and eventually overall performance is dominated by the communication. The other shortcoming is load balance. Since car model is composed by lots of different materials and varieties of elements, the workload distributed on each process would be greatly uneven once the number of processes increases to certain extent. That creates the issue of load unbalance.

Outlines of Fujitsu FX1 and HX600 Systems

There were two systems, Fujitsu FX1 and HX600, used for this evaluation. Both supercomputer systems are located at Nagoya University [2]. Table 1 lists the outlines of both systems.

Machine	Fujitsu FX1	Fujitsu HX600
OS	Open Solaris	Red Hat Enterprise Linux4
Interconnect	InfiniBand DDR(2GB/s) x 1	InfiniBand DDR(2GB/s) x4
Processor	SPARC64VII (2.5GHz) Quadcore	Opteron 8380(2.5GHz) Quadcore
Number of Nodes(Cores)	768Nodes (3072cores)	160Nodes (2560cores)
Total Peak Performance	30.72TFLOPS	25.6TFLOPS
Total Memory size	24TB	10TB
Peak Performance per node	40GFLOPS	160GFLOPS
Memory size per node	32GB	64GB

Table1. Fujitsu FX1 and HX600 systems

Advantage of Hybrid: Communication Amount

The advantage of Hybrid version over the MPP version is the significant reduction in amount of message and less function calls in message communication, meanwhile the Hybrid version still maintains the same level of parallelization. The Caravan 2-million elements car crash model [3] and LS-DYNA Hybrid R5-beta version has been used in this study. Figure 1 shows the total amount of communication of MPP and Hybrid versions respectively. The total amount of communication is the sum of the multiplication of function call and each data length. The amount of message generated in Hybrid version 64procs x 4threads is three times less than that of MPP version 256procs, also the amount of message of Hybrid 256procs x 4threads is twice less than MPP 1024procs. Figure 2 shows the total MPI-function calls occurred in Hybrid and MPP versions. Hybrid version invoked MPI-function calls over 5 times less than those of MPP version. Such difference becomes especially more significant as the number of processors increase.

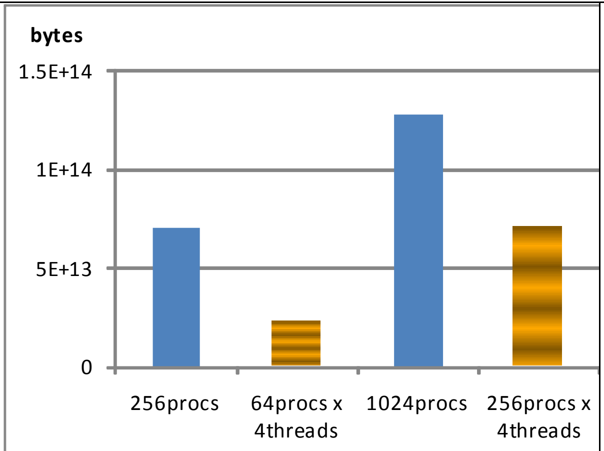

Fig1. Caravan 2M total Comm amount

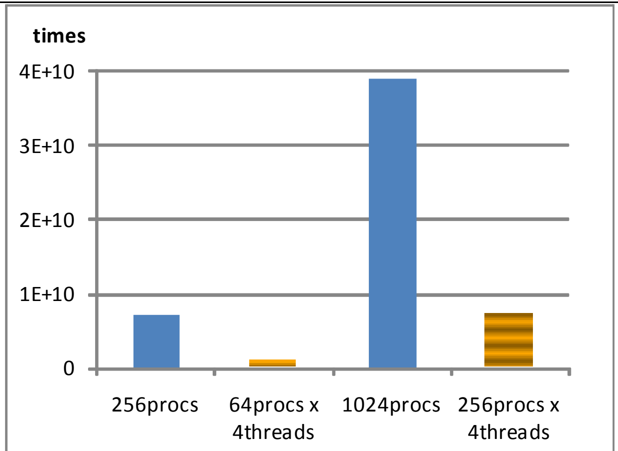

Fig2. Caravan 2M totalComm calls

Advantage of Hybrid: Load Balance

For a system with four cores per CPU, the decomposition parts of Hybrid version is four times less than those of MPP version; hence the calculation amount per partitioned part of Hybrid version is almost four times bigger than that of MPP. With the larger ratio of computation amount to communication, it is easier for LS-DYNA to adjust decomposition to achieve better load balance among the partitions.

Evaluation of Loop Length for Hybrid Version

LS-DYNA has a build-in parameter NLQ as loop length used for analysis, which is designed to improve the performance on different hardware platforms with the different memory architecture, especially, cache size and cache layout. There were four NLQ being tested on FX1 with the Hybrid version, the impact on the performance could be up to 5% as shown in Table 2.

Hybrid	**64procs x 4threads**			
NLQ	**32**	**48**	**80**	**383**
Elapsed(sec)	**5,321**	**5,042**	**5,188**	**5,314**
Ratio	**1.00**	**1.05**	**1.02**	**1.00**

Table 2."Caravan 2M" 20000cycles timing information

Performance of Hybrid Parallelization

LS-DYNA provides detail timing information on several major stages at the end of the log file. For car crash simulation, stages of ‘Element processing’, ‘Contact algorithm’ and ‘Rigid bodies’ in general consume 90% of the total time. In terms of parallelization, stage ‘Element processing’ generally is handled by process-parallel, and stages ‘Contact algorithm’ and ‘Rigid bodies’ are done through threads-parallel. Figure 3 shows total wall clock times of both Hybrid and MPP versions using Caravan 2-million car crash model on Fujitsu HX600. The performance of MPP version got saturated at 512 processes, but Hybrid version still showed parallel efficiency at 1024 cores. Further analysis indicated, as shown in Figure 4, the superior performance of Hybrid version came from the benefit of both stages of ‘Contact algorithm’ and

'Rigid bodies'. The reduction in communication of these two stages, as presented in Figures 1 and 2, gave performance edge to Hybrid version.

The Caravan 2-million elements car crash model has been one of the most popular models used for crash simulation at this moment. As shown in Figure 3, the parallel content of this model has been exhausted once the number of processes exceeded 512. Since the mathematical model will continue to grow for the purpose of obtaining more accurate analytical results, it is justifiable to utilize a bigger model to investigate the feasibility of massively computing and to investigate the potential of its parallel efficiency. We tested both LS-DYNA Hybrid and MPP versions with Caravan 10-million element car crash model [4] on Fujitsu FX1. The analysis was measured till 10 msec. time duration with sy=1000 domain decomposition. Figure 5 shows 120msec simulation elapsed times, which are estimated from the results of 10msec simulation, using both Hybrid and MPP versions. The MPP version is estimated to take 31 hours with 512 processes and 26 hours with 1024 processes to complete the analysis; while the Hybrid version is expected to finish the job with 28 and 23 hours on 512 and 1024 cores respectively. This demonstrates the superiority of Hybrid version over the MPP version when analyzing a humongous car crash model with massively parallel computation. Figure 5 also reveals Hybrid version was able to outperform MPP version even with less than 512 processes on Fujitsu FX1. This put Fujitsu FX1 in a very attractive position for the LS-DYNA Hybrid version.

We will continue our study with more than 2000 cores. One of our potential target machines is Japan's Next-Generation Supercomputer [5] which is using SPARC64™ VIIIfx processor [6].

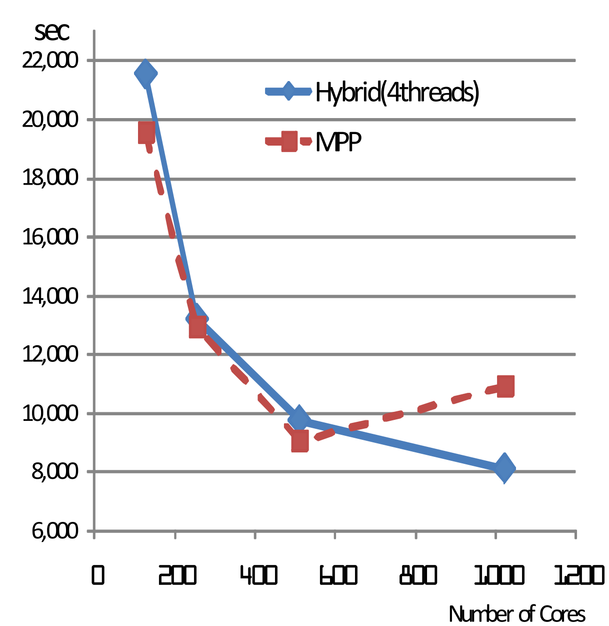

Fig3. Caravan 2m 120msec elapsed time

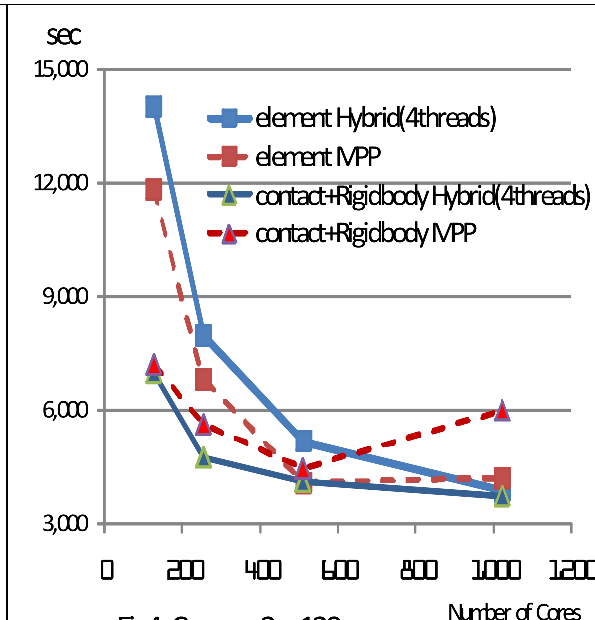

Fig4. Caravan 2m 120msec each phase elapsed time

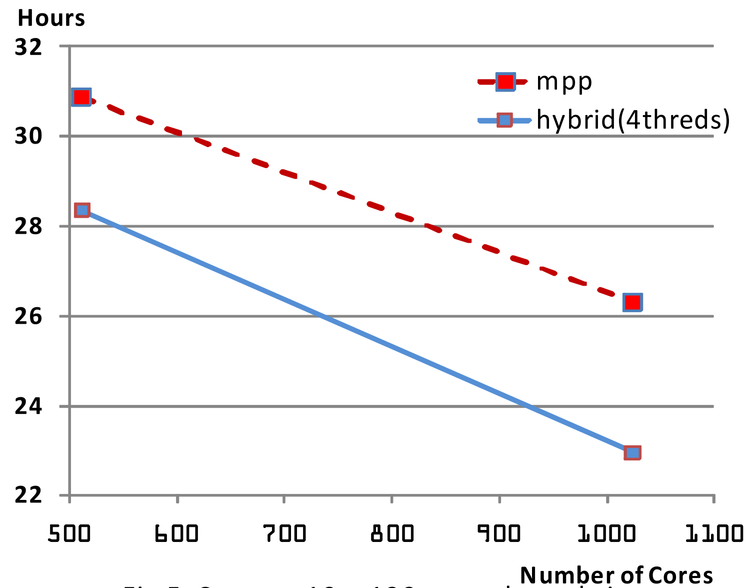

Fig.5 Caravan 10m 120msec elapsed time
(estimated from 10msec simulation)

Conclusions

With the ten-million elements Caravan car model, the Hybrid version of LS-DYNA can outrun MPP version within the number of cores exceeded 512. Executing LS-DYNA Hybrid version on a massively parallel system like Fujitsu FX1 and still obtaining good performance has been proved to be feasible. The LS-DYNA Hybrid version combining with the highly parallel supercomputers will be able to meet the challenge of gigantic car crash models.

Acknowledgements

This work was done by joint research with Professor Ishii of Laboratory of Information Technology Center, Nagoya University. The Authors would like to acknowledge Information Technology Center, Nagoya University for allowing using the FX1 and HX600 of their facility. The authors also would like to extend their appreciation to Drs. Jason Wang of LSTC, Clifford Chen of Fujitsu America Inc. and Mr. Shota Yamada of Fujitsu Limited for their assistance in this Hybrid parallelization evaluation.

References

[1] Fujitsu Scientific and Technical Journal
http://www.fujitsu.com/downloads/MAG/vol44-4/paper09.pdf

[2] Nagoya University
http://www2.itc.nagoya-u.ac.jp/riyou/system2009.htm

[3] http://www.topcrunch.org/

[4] M. Makino, "The performance of 10-Million Element Car Model by MPP Version of LS-DYNA on Fujitsu PRIMEPOWER", 10th International LS-DYNA® Users Conference

[5] Japan's Next-Generation Supercomputer
http://www.nsc.riken.jp/project-eng.html

[6] SPARC64™ VIIIfx Extensions
http://img.jp.fujitsu.com/downloads/jp/jhpc/sparc64viiifx-extensions.pdf