The Performance of 10-Million Element Car Model by MPP Version of LS-DYNA® on Fujitsu PRIMEPOWER

Mitsuhiro Makino
Computational Science and Engineering Solution Center, Fujitsu Limited
9-3 Nakase 1Chome Mihama, Chiba, 261-8588, JAPAN
makino@strad.ssg.fujitsu.com

Abstract

In automotive industries, car crash analysis by finite element methods is a very important tool for reducing the development time and cost. In order to get the accurate results, in addition to the improvement of the finite element technology, such as full-integrated shell elements, smaller size of finite element mesh is used, because finer meshes represent the car geometry more accurately, and reduce the noise of contact force. The batch mesh generator, which is enhanced recently, also needs fine mesh. The use of these fine mesh model increases the computational time.

In this paper, we examine the performance of the fine mesh model. We developed a 10-million elements car model, which is 10 time larger than the current production car model. The performance of large number of CPU by Massively parallel processing(MPP) version of LS-DYNA, is measured on Fujitsu PRIMEPOWER.

Introduction

In order to get the good accuracy in crash analysis by LS-DYNA, many automotive companies use fully integrated shell elements (type16 shell element) for crash analysis. Since the calculation cost of type16 shell element is 3 times higher than type2 shell element, the percentage of type16 shell elements used in car model is different in each company, but I believe all companies start using type16 shell element in critical area of car crash model. In table I, the TOPCRUNCH[1] BMT models are summarized. NEON refined model is using all TRIA element, and TRIA element is not popular in crash analysis. In the 3Cars model, type2 shell element are used. Type2 shell element is very fast and robust, but this element has hourglass mode and the user has to mesh very carefully to minimize the hourglass mode. The Car2car model[2] uses type16 shell element, and is close to the car crash analysis models used in automotive companies.

	NEON refined	3Cars	Car2car	Car2car-10m
Shell element formulation	10 (TRIA)	2 (QUAD)	16 (QUAD)	16 (QUAD)
# of shell elements	532,075	785,018	2,416,244	9,663,272

Table. 1 Comparison of TOPCRUNCH BMT models

In addition to element technology, the demand for using a fine mesh is very strong, because it would

1. represent the car geometry more accurately,
2. reduce the development cost and time by using the batch mesh generator[3],
3. reduce the noise when slave nodes transfer the edge of master segment, and
4. reduce the mesh size effects.

The Car2car is using a 10mm x 10mm mesh size, but customers move to use a finer mesh. We developed a 5mm x 5mm mesh size model and the total number of elements in the model is approximately 10,000,000.

In this paper, the large car model is developed and the performance for a large number of CPUs is reported.

10-Million Element Car Model

The Caravan model was originally developed by NCAC[4]. The number of elements was 0.3 million. For Car2car-10m, one shell element is divided to 16, and the total number of elements becomes 4.8 million in one car. We set up car to car crash analysis. The total number of elements is 9.6 million.

The shell elements are modified to TYPE16 from TYPE2.

The contact definitions are

C-1. single surface contact for each car,
C-2. surface to surface contact between cars with BOX,
C-3. surface to surface contact between engine and surrounding parts,
C-4. tied contact between solid elements and surface shell

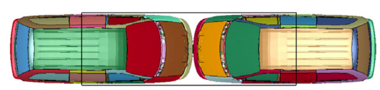

Fig.1 Car1, Car2 and BOX

In the original model, the modeling of the Radiator and Radiator fan were numerically sensitive and the Job was terminated due to a negative volume. We modified the model around the Radiator and the material properties.

The computational time for type16 shell element is 3 times more expensive than default type 2 shell element. With explicit analysis a finer mesh affects the computation time in 2 ways. When the edge size of a shell element reduces to a quarter, the number of elements increases by 16 times, but the computation time is 64 times because the time step is also reduced by one quarter due to the Courant condition.

Fig.2 Engine Surrounding Parts

Table 2 summarizes the timing information for 640k elements, 2400k elements and 10000k elements model. The difference of models is the mesh size, while the element formulation and contact definitions are the same. Each model is executed to 10ms simulation time using 16CPUs on a Fujitsu PRIMEPOWER. The element processing time divided by the number of elements and cycle number is constant (=5.7E-7sec), however, the contact algorithm time divided by the number of elements and cycle time decreases as the number of elements in the model increase.

	640k elements	2400k elements	10000k elements
Element processing	3495	27331	273810
Contact Algorithm	533	3787	31579
# of Shell elements	647674	2416244	9663271
Time step	1.E-6	5E-7	2E-7
Memory(decomp)	82m	302m	1177m
Memory(exec)	37m/14m	149m/46m	545m/181m
Element Processing/ (#of elements*cycles)	5.4E-3	5.7E-3	5.7E-3
Contact algorithm/ (#of elements*cycles)	8.2E-4	7.8E-4	6.5E-4

Table 2 Element processing time and Contact time

In memory (exec), first number is memory used for CPU#0 and second number is for other CPU. The unit of memory is word (4byte/word).

Performance of 10-Million Element Car Model

The performance of the 10-million elements car model is shown in Fig. 3. This result measured until 10ms and the domain decomposition is SY=1000. The element processing is good scalability until 512CPUs, but the contact algorithm does not scale after 256CPUs. The rigid body has bad scalability in all number of CPUs. In small number of CPUs, rigid body spends only small amount of cpu time, but in large number of CPUs, it becomes important because the element processing and contact algorithm become faster.

From this measurement, the estimated time to execute 120ms is 179hours for 64CPU, 54hours for 256CPU and 43hours for 512CPU.

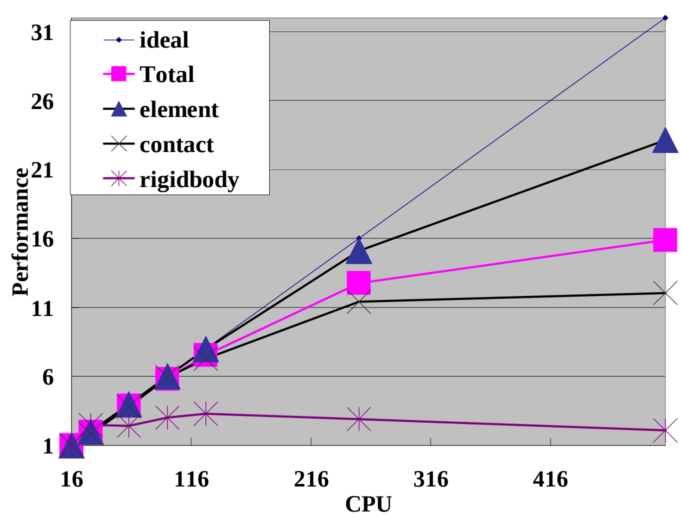

Fig.3 Performance of 10m car2car

Discussion

For executing the large model, the decomposition is a big issue, because it needs a lot of memory. We recommend processing decomposition and execution separately, in order to avoid installing large memory for all execution nodes. The decomposition executes on 1 CPU with large memory and creates decomposition.pre by pfile of Fig.4a, then execute parallel processes in execution nodes with small memory by pfile of Fig.4b.

```
pfile.decomp
   general  { nofull }
   decomposition { numproc NN
                   file decomposition
                   rcblog rcblog }

   mpirun –np 1 mpp-dyna i=input \
   p=pfile.decomp  memory=1800m
```

Fig.4a decomposition

```
pfile.exec
   general  { nofull }
   decomposition {  file decomposition
                    rcblog rcblog }

   mpirun –np NN mpp-dyna i=input p=pfile.exec \
   memory=300m memory2=300m
```

Fig.4b execution

Summary

10million car model with type16 shell element is executed on highly parallel machine. The execution time for 120ms simulation of 10million element model is estimated 43hours by 512CPU of Fujitsu PRIMEPOWER from 10ms execution result.

Acknowledgment

This work was done by joint research with Information Technology Center, Nagoya University. The author expresses his thanks to Information Technology Center, Nagoya University for using their computer system.

References

1) http;//www.topcrunch.org/

2)M. Makino,"The performance of Large Car Model by LS-DYNA on Fujitsu Primepower", 9th International LS-DYNA User Conference,5-1,(2006)

3) Ratzel M., Du Bois P., Fredriksson L. and Schneider D. A study of mesh sensitivity for crash simulations; comparison of manually and batch meshed models, DYNAmore LS-DYNA conference, Bamberg, 2005

4) National Car Crash Center(NCAC) http;//www.ncac.gwu.edu/vml/models.html