LS-DYNA® Performance in Side Impact Simulations with 100M Element Models

Alexander Akkerman, Yijung Chen, Bahij El-Fadl, Omar Faruque and Dennis Lam
Ford Motor Company

Abstract

LS-DYNA has been used for vehicle crash simulations for many years. The models have increased in size over the years but in most cases do not exceed more than a few million elements. However, recently developed material models require much greater levels of refinement resulting in much larger models, perhaps as high as 100M elements. Simulating models of the order of 100M elements in turn requires much higher levels of scalability in order to be feasible in the vehicle development process.
This paper will analyze LS-DYNA performance with a 100M-element sled side impact model running on up to 1,000 and more CPUs with various Intel processors and Infiniband interconnect technologies.

Introduction

Recent adoption of lighter–weight and high strength materials in the automotive industry has led to significant changes in crash simulation modeling. One example is a requirement to characterize fracture/separation in crash simulations of vehicles with such materials. Current state-of-the-art computational methods for fracture characterization generally require meso-scale modeling using 3D solid elements[1]. However, utilizing this methodology leads to models with up to 100M elements and even more. Higher levels of scalability become necessary in order to at least maintain the present levels of time-to -solution to support the vehicle development process.
This paper will discuss challenges with building, running and post-processing a model of this size.

Model Description

The model used in this project is a Side Impact Sled Test shown in Figure 1. The B-pillar is attached to the rocker and to the roof rail. A customized fixture is used to mount the B-pillar subsystem, which is constrained at both ends of the rocker and the roof rail. The B-pillar subsystem is impacted by a sled.

The model was built to capture and predict potential B-pillar spot-weld separation, crack initiation, and crack propagation. Figure 2 shows the B-pillar outer layer in detail. It is modeled in meso-scale with solid element size in the range of 0.2mm. The resultant CAE model has approximately 100M elements including about 500K shell elements used for modeling the rocker, roof rail and B-pillar inner components.

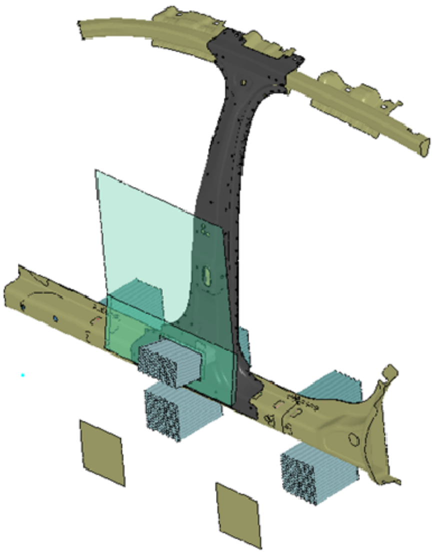

Figure 1.

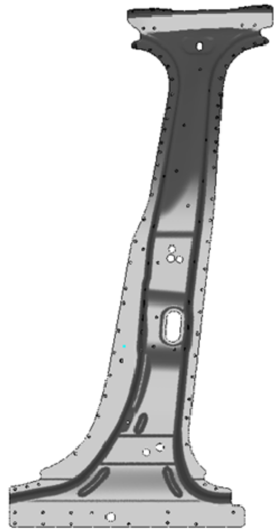

Figure 2.

Figure 3.

The material of the B-pillar outer (Figure 2) is modeled as Piecewise Linear Plasticity with an MIT erosion feature. Shell elements are modeled as basic Elastic Plastic material. There are two contact interfaces – one is a single surface contact including all components and the second is a tie contact used for spotwelds in the model. The final deformed shape is shown in Figure 3. More details about the model and analysis of the engineering results have been documented by Chen et al [1]. Details of the MIT erosion material are described by Bai and Wierzbicki [2].

Pre- and Post-processing

Our initial attempts to build and mesh this model presented significant performance challenges. Even some basic operations such as reading the model using the available tools were not possible. The size of the model was stretching the available software beyond its software verification and testing limits. We engaged LSTC development and after a few updates were able to start working with this model. We then quickly learned that local client hardware was not up to the task handling a model of this size. The interactive manipulation of the model was unacceptably slow. At that point we deployed a high-end compute node with 128GB of memory and GPU capabilities, which allowed us to move the project forward. Still, even after upgrades to the software and hardware and adopting a batch mode for post-processing of the results, the interactive tasks were significantly slower than what we are used to in our daily work. Major improvements in pre- and post-processing technology (hardware and/or software) are required to enable seamless interactive visualization and manipulation of larger models and processing of the results to be practical in daily use.

Performance and Scalability

We used LS-DYNA version 971R7.1 for this project. The base version could not handle a model of this size, however, after a few corrections by LSTC we were able to run the simulation. Due to the model size, domain decomposition and simulation had to be separated into 2 separate jobs, with domain decomposition running in the double precision version of the code on a node with 128GB of memory and the simulation running a regular MPP version in single precision.

We had access to a few variations of Intel Xeon and Infiniband configurations:

- Intel X5672/Infiniband QDR (3.2GHz, 3GB/core)
- Intel E-2670/Infiniband QDR (2.6GHz, 4GB/core)
- Intel E-2670/Infiniband FDR (2.6GHz, 4GB/core)
- Intel E-2650v2/Infiniband FDR (2.6GHz, 4GB/core)

Our scaling tests were performed on 256 to 2,048 CPUs (1,024 on X5672 systems) on all available configurations for the first 20ms of the simulation. We used the default RCB domain decomposition. The results are summarized in Figure 4.

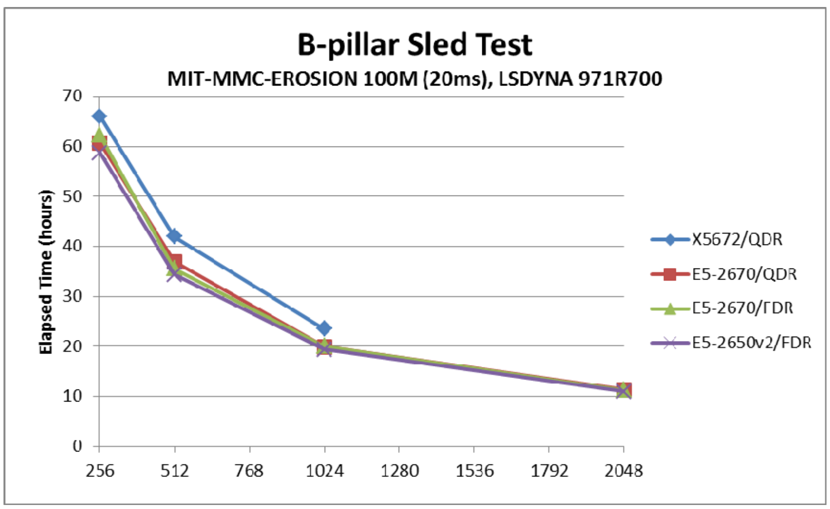

Figure 4.

The results show that the job scales to at least 2,048 processors, although not quite proportionally to the number of processors. Differences in processor or interconnect technologies do not materially affect the performance. The speedup summary is shown in Figure 5.

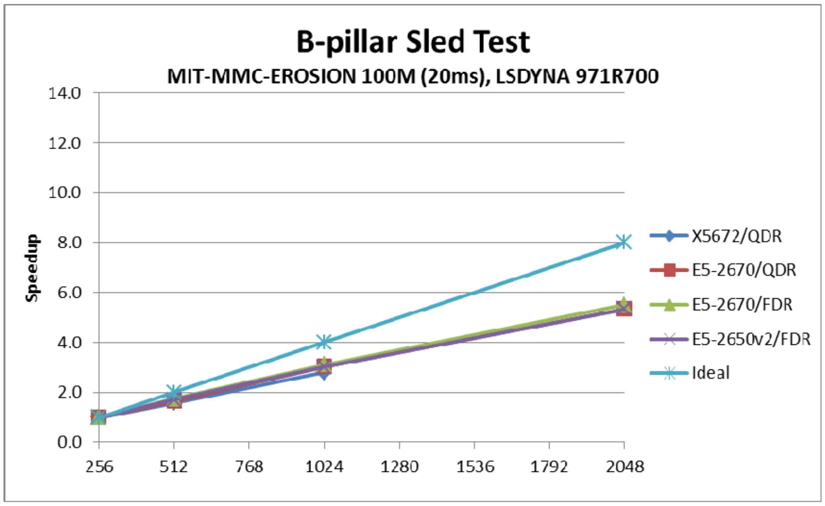

Figure 5.

A full 100ms simulation was completed on 1,024 CPUs in 2.2M time steps. The complete simulation took about 4.5 days of elapsed time on the E5-2670/FDR processors, well short of the goal of 16 hours (overnight). A 2,048-CPU simulation is projected to complete in about 3 days. The 1,024- CPU job progress is summarized in Figure 6.

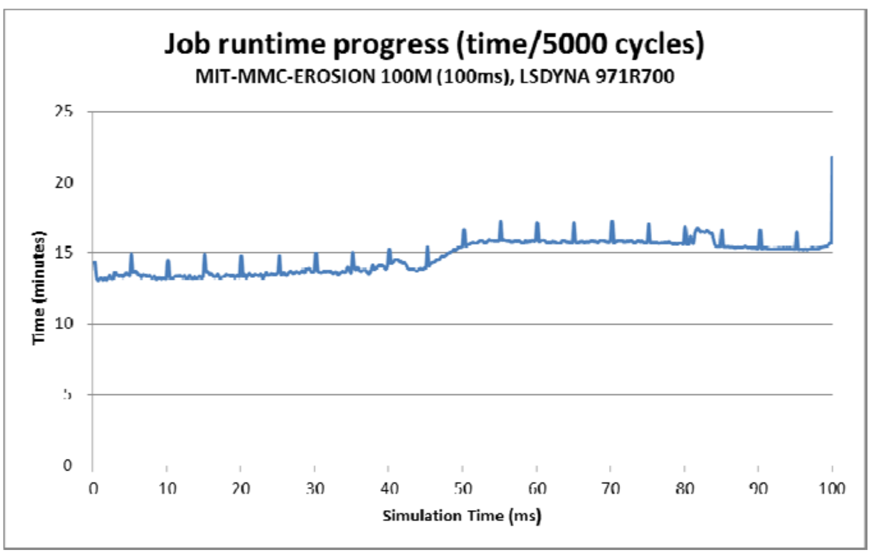

Figure 6.

The chart represents elapsed time of every 5,000 time steps of the simulation. The 2 minute bumps every 5ms are attributed to generation of the d3plot files. We are unsure of the reason for the increase in time starting at about 45ms, however comparing the elapsed time distribution of the 20ms and 100ms simulations on the same hardware (Table 1.), we observed that the element processing contributions remain constant, while rigid body is quite a bit higher in the complete run, thus pointing to the increase in contact time in the complete simulation, most likely resulting from load imbalance in contact calculations.

	20ms	5x20ms	100ms	Increase
Element Processing (sec)	25,000	125,000	125,000	None
Contact (sec)	12,600	63,000	63,000	None
Rigid Body (sec)	12,700	63,500	93,800	48%
Other (sec)	20,700	103,500	110,000	6%
Contact+Rigid Body	25,300	126,500	156,800	24%

Table 1.

Taking a look at the elapsed time distribution from each process (collected from the individual message files), we observed large variations in contact+rigid_body times indicating imbalance in contact calculations on the individual CPUs (Figure 7). These variations are even larger in the 100ms simulation, again, confirming that the contact calculations are the most likely source of lack of greater levels of scalability.

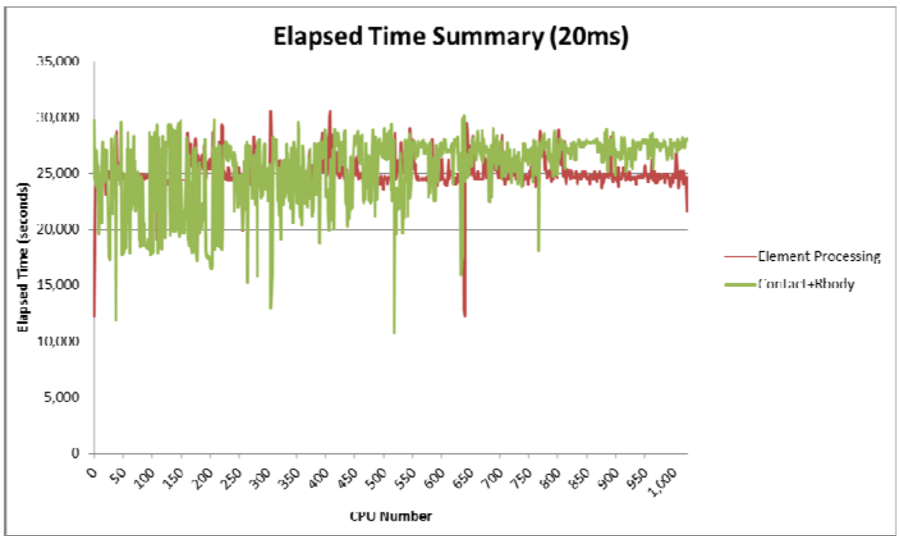

Figure 7.

Conclusions and Next Steps

We demonstrated the ability to handle the model with 100M elements. The required pre- and post-processing tools are now functional and we successfully completed a 100ms simulation on up to 1,024 CPUs, however significant performance challenges remain.

Higher performing graphics capabilities are required to be able to build the model and process the results without lengthy delays. The performance and scalability also need to improve to be able to reduce the time-to-solution to achieve the goal of overnight time to solution. At least 4,096 CPUs and perhaps even 8,192 may be necessary to make it possible.

We intend to continue working on this model with the goal of further improving its performance. Two possibilities are utilizing the LS-DYNA Hybrid version [3] to reduce the number of domains in the model and a more in-depth evaluation of alternative domain decomposition models that would improve the parallel load balance of the simulations and especially the contact calculations. Further studies and understanding of the source of high levels of time spent in the "Other" category also appear to be an opportunity.

Acknowledgements

We would like to thank Jason Wang, Dilip Bhalsod and Phillip Ho of LSTC for their support of this project.

References

1. Yijung Chen, Dennis Lam, Omar Faruque, Cedric Xia, Alex Akkerman, "Meso-Scale FEA modeling to Simulate Crack Initiation and Propagation in Boron", 13th International LS-DYNA Users Conference 2014, Dearborn, Michigan
2. Bai, Y. and Wierzbicki, T., "A new model of metal plasticity and fracture with pressure and Lode dependence," International Journal of Plasticity, 24, 1071–1096, 2008.
3. Yih-Yih Lin and Jason Wang, "Performance of the Hybrid LS-DYNA on Crash Simulation with the Multicore Architecture", 7th European LS-DYNA Conference, 2009, Salzburg, Austria