# LS-DYNA Automatic Re-decomposition

Edouard Yreux[1], Chen Tsay[1], Jason Wang[1]

[1]Livermore Software Technology Corp.

## Abstract

The default decomposition method for LS-DYNA/MPP is RCB which dividing the model based on the initial geometry. If the geometry does not severely distorted during the simulation, this decomposition gives reasonable scaling upto few hundreds cores. LS-DYNA also provides additional “pfile” options which relies on user’s knowledge of deformation to achieve better MPP efficiency.

Unfortunately, there are many problems cannot be easily treated by those options, i.e. bird strike, water wading, FBO, etc. The simulations involve parts with relative motion which are difficult to decompose only once and those jobs are usually suffer from the scaling. Furthermore more cores are used in the simulation, a load unbalancing effects will be amplified and results in poor scalability.

To Achieve better computational load balancing, a new automatic re-decomposition algorithm has been implemented recently. The new method can readjust the load balancing during simulation based on the current geometry. In this study, we will give some typical examples to show how to regain the load balancing and improve the parallel efficiency.

## 1 Introduction

The x86-64 CPU becomes the main computing power for engineering analysis for the past twenty years. In Table 1, we collect few CPU data from Intel Xeon chips. The trend shows there are more cores added on the CPU but the clock rate drops slowly to keep the power consumption in a reasonable range. The growth of the computing power mainly relies on better parallel programming than from single core performance. Application has to divide a workload into many small tasks and computes the task on each core efficiently. LS-DYNA/MPP is developed based on this concept using message passing interface (MPI) and introduced in the mid-1990s.

| Intel Code name | Min. feature sz | Rel. | core/sck | Clockrate | Instruction |
|---|---|---|---|---|---|
| Bloomfield | 45nm | 2008 | 4 | 3.2 GHz | SSE |
| Westmere | 32nm | 2010 | 6 | 2.4 GHz | SSE |
| SandyBridge | 32nm | 2011 | 6 | 3.1 GHz | AVX |
| Haswell | 22nm | 2013 | 18 | 2.5GHz | AVX2 |
| Skylake | 14nm | 2015 | 28 | 2.5GHz | AVX512 |

*Table1: Intel Xeon CPU*

To divide a FEM model, LS-DYNA/MPP first evaluates the total computing cost based on element formulations, material models and features etc. and then decomposes the workload into many sub-domains to keep each has about the same cost. The default method is called Recursive Coordinates Bisection (RCB) which divides the model using length scale from model’s initial geometry as demonstrated in Figure 1.

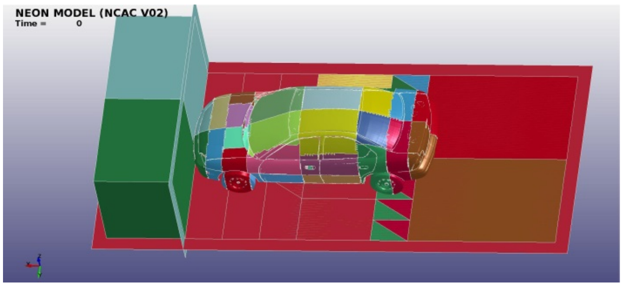


*Fig.1 Standard RCB decomposition by model’s initial geometry*

Each core handles one subdomain of the model and MPI is used to transfer the shared information between domains. To achieve more parallelism under large number of cores, OpenMP thread (LS-

DYNA/HYBRID) can be created from each MPP rank. The speedup of a workload is measured based on the ratio of Elapsed Time to the single processor time. The Elapsed Time can be further divided into several measurable time sections shown in the following equations.

$$Speedup = \frac{\left(T_{cpu} + T_{sys}\right)_{single\ core}}{T_{elapsed}}$$

where

$$T_{elapsed} = T_{cpu} + T_{sys} + \left(T_{mpp} + T_{openmp}\right)_{overhead}$$

For most of the explicit analysis, the system time, $T_{sys}$, is a small fraction of $T_{elapsed}$. Speedup mainly depends on CPU time distribution, $T_{cpu}$ and parallel overhead, $T_{mpp}, T_{openmp}$. It is important to isolate the problem of speedup degradation from uneven CPU load or parallel overhead.

## 2 Detecting of Parallel Deficiency

By default, LS-DYNA/MPP decomposes the model only once at beginning of the run by its initial geometry. However, the load distribution is usually shifted during simulation because of changing of topology. For example, large deformation occurs in one of the sub-domain due to impact and then the material routine for those deformed elements has higher computing cost than other domains. $T_{elapsed}$ can be controlled by this core because all other processors are idling at a synchronization point. This is the most common case of load unbalance. Or, parts in different cores are in contact/FSI and $T_{elapsed}$ increases from contact/FSI searching, penetration calculation and message passing. This will cause both load unbalance and parallel overhead.

User can investigate the problem by checking some LS-DYNA output files. For example, the time between d3plots usually spreads about same interval if time step size remains constant. If the interval increased dramatically as shown in figure 2, which ee should put more attention to find out the cause. Also, the time per zone cycle (NZC) in “glstat” also gives good indicator of the load. NZC gives the average CPU time per element of each cycle for the model. If the value is getting larger, it indicates the code spending more time per cycle due to extra computation or communication, figure 3.

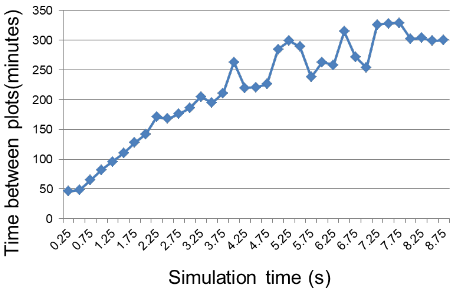


*Fig.2 Time between d3plots getting longer*

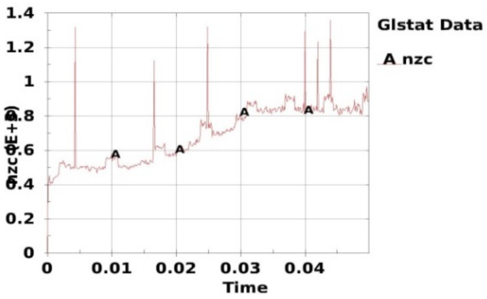


*Fig.3 NZC increases*

To readjust the load distribution or to reduce the communication, a new feature has been developed called “redecomposition” which allows repartitioning the model during simulation using the current geometry.

## 3 Redecomposition

LS-DYNA/MPP has a long supported feature which can restart and repartition the model called fulldeck restart. The feature collects history data for element, material, contact, etc from all processors and output to a master database. The subsequent analysis is started with a new partitions and the history data will be mapped back to the model from this database. Such analysis currently has to operate manually and the decomposition is still based on the undeformed geometry. The redecomposition (REDECOMP) is developed based on this function and will automatically perform the steps but based on the current geometry. This feature can be activated by the following keyword and has few simple options right now.

***CONTROL_MPP_DECOMPOSITION_REDECOMPOSITION**
**FREQ, DEFGEO, CCOST**

**FREQ:** GT.0 time interval between each REDECOMP
LT.0 time interval is divided ENDTIME into abs(FREQ) chunks
**DEFGEO:** decomposition uses current geometry
**CCOST:** Scale factor for element in contact

FREQ controls the REDECOMP frequency. DEFGEO controls the treatment of active/inactive SPH particles during partition and CCOST gives additional weight factor for elements in contact.

To visualize REDECOMP at each step, a new feature is also developed with lsprepost team which is controlled by the keyword.

***CONTROL_MPP_DECOMPOSITION_OUTDECOMP**
IFORM
**IFORM=3:** uses lsprepost format and allows animation

The code outputs a family of files with the name "decomp_parts.lsprepost_s####" after each REDECOMP. The lsprepost 4.7 under "Views>MPP>LOAD" will automatically load the family files to view the partition at each step.

# 4 Examples

## 4.1 Water Wading

LSTC gets more requests for Water Management Analysis recently for example E-coat, wading, etc. The length scale of this type of problem is in the range of 10 to 100 meters and the time scale is around 10 to 100 seconds. To tackle this type of problems, LSTC has developed an implicit SPH solver with excellent FSI capability. The detail information is given in reference [1].

For wading problem as shown in figure 4, order of ten million SPH particles are needed due to the size of the pool and resolution to detect the gaps between structures. While the truck driving through the pool, the location of FSI changes over time. If the model is only decomposed once as before, the FSI between truck and pool will constantly change between processors. Furthermore, water flow created from the motion mixes SPH particles from different domains. Both FSI treatment and SPH particle interaction will generate more MPI messages among CPUs. The simulation becomes slower because extra communication overhead.

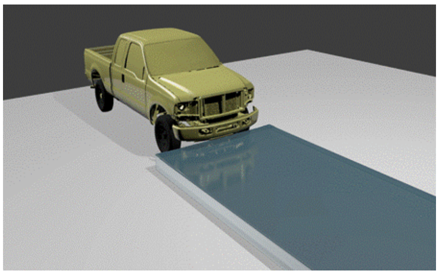
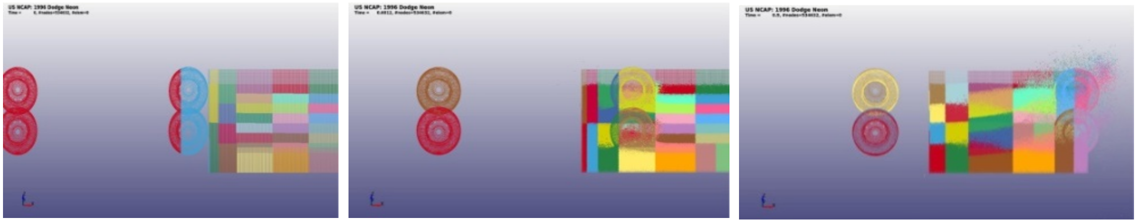


*Fig.4 Water wading, truck moving through the pool*

By REDECOMP, the code was able to put the parts under FSI in the same processor and to rearrange SPH particles in the same doamin as shown in figure 5 a,b,c where each subdomain is represented by a unique color. The plots clearly show the tire and SPH particles under FSI were assigned in the same cores to maintain the data locality. At the same time, it also regrouped the SPH particles for less SPH inter-processor interaction which gives a clear SPH boundary between domains.



(a) time=0. (b) time=1.0 (c) time=4.0

*Fig.5 REDECOMP after each step*

The interval between d3plot became stable over the simulation compared with the default method in Figure 6 and the total runtime was cut by half.

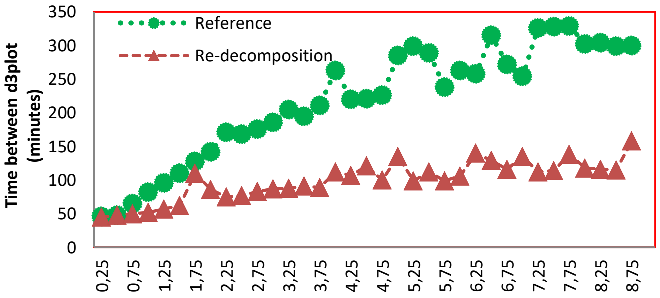
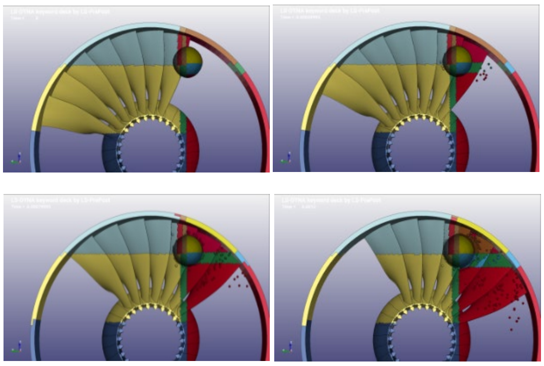


*Fig. 6 REDECOMP gains 2x speed*

### 4.2 Bird strike

It is difficult to maintain good load distribution among processors for bird strike problem by the default method. Bird and blades are usually placed in different cores from their original positions. While blades start to turn, SPH bird is cut by blade edges. Contact force calculation is in several small regions and contact searching is always performed between processors. The mixing SPH of particles also increases the cost of SPH calculation due to communication overhead.

To study this problem, a test model was created from LS-DYNA Aerospace Working Group test suite [2]. For more general overview of LS-DYNA AWG please visit awg.lstc.com. In this example, REDECOMP kept the SPH and FEM elements under contact in the same processor for more efficient calculation. It also regrouped SPH particles to the same processor for less communication. Figure 7 shows the partition changing over time by the new scheme. Figure 8 shows the speedup comparison between default and new scheme. Two times speed up was achieved with higher core count due to less data transferring among networks.



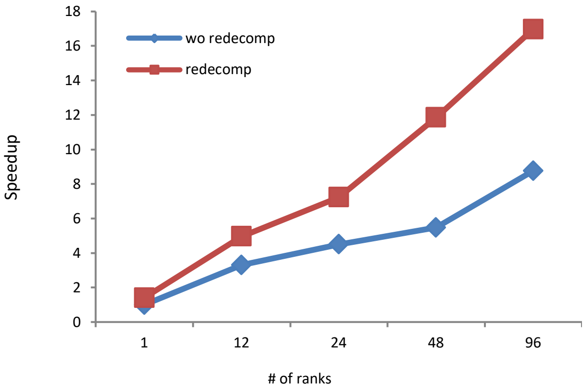
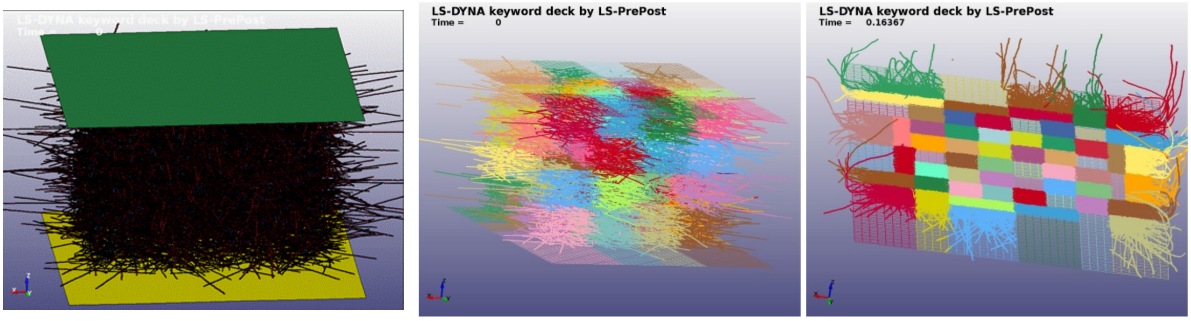


*Fig. 7 REDECOMP to keep N2S contact in local and redistribute particles with less mixing*

*Fig. 8 Scaling with/wo REDECOMP*

### 4.3 Contact Problem

A pure FEM problem is also tested by this method here. A bunch of loose wires with the initial geometry is shown in figure 9 (a). Wires are then compressed by two rigid plates to form a block with designed porosity. In this example, contact algorithm spends most of the CPU time.
By default RCB, the problem is decomposed into many cubical domains as in figure 9 (b). Most of contacts occures inside each domain and relative small amount of the contact happens between adjacent domains required data communication. Therefore, the problem started with a reasonable load balance. When plates moved closer, the original cubical doamins were compressed in z direction. Due to geometry change, contact bucketsort becomes more costly because more potential contact pairs need to be examined and also more processors involved in the searching.
REDECOMP is based on the current geometry. It readjusted the decomposition shape from original cubical domain into 2D in plane with the deformation as shown figure 9 (b), (c). This reduced the number of processors in contact searching and improved the scaling as shown in figure 10.


LS-DYNA keyword deck by LS-PrePost
Time = 0
LS-DYNA keyword deck by LS-PrePost
Time = 0
LS-DYNA keyword deck by LS-PrePost
Time = 0.16367

*(a)* *(b)* *(c)*

*Fig. 9 (a) Initial geometry (b) Initial decomposition (c) Decomposition changes along with changing geometry*

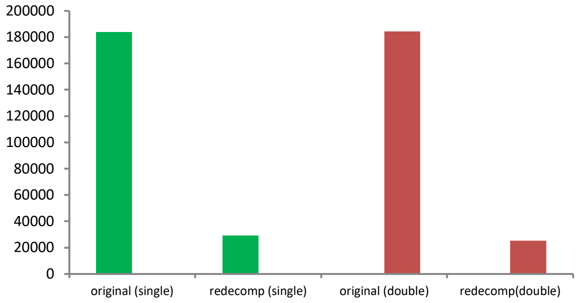


*Fig. 10 Scaling with/wo REDECOMP*

## 5 Summary

REDECOMP is extension of the existing fulldeck restart capability. The new feature mainly reduces the runtime by repartition model for better load balance and also less communication overhead. However, it requires additional IO for collecting and writing history data to a master database. LS-DYNA then decomposes the model using current deformed geometry and maps the data from master database to the new domain and continues the analysis. It creates extra CPU overhead from these process compared with the regular analysis. We should first isolate the source of parallel deficiency based on available information before applying the keyword. The data in glstat, d3plot plot interval, etc. provide good information to identify those problems.

The concept of fulldeck restart is very different from the regular small restart. The former needs carefully restore the data to newly partitioned model and the latter is simply recovered from memory map from the previous step. Some features maybe not properly supported under this feature. It is important to check the results before and after the REDECOMP to see if all history data has been recovered during this process. If the feature is not implemented, please contact us and we will try to work with you and enhance the feature.

## 6 Literature

[1] E. Yreux, "Implicit SPH in LS-DYNA for Automotive Water Wading Simulations", 12th European LS-DYNA Conference, 2019

[2] LS-DYNA Areospace Working Group, “AWG ERIF Test Case Suite Case 2.2 and Case 5“