Dynamic Load Balancing Algorithm for CPM in LS-DYNA

Hailong Teng

Livermore Software Technology Corporation

Abstract

To achieve scalable parallel performance in particle method simulation, we introduce a new algorithm for automatic parallel load balancing and apply it to corpuscular particle method (CPM) in LS-DYNA. Load-balancing is achieved by dynamically using RCB to evenly distribute workload to processors. Several numerical tests demonstrated the efficiency of dynamic load balancing algorithm.

Introduction to CPM

The corpuscular method [1][2] is a coarse-grained muti-scale method developed for gas dynamics simulation. It is based on the kinetic molecular theory, where molecules are viewed as rigid particles obeying Newton's laws of mechanics. In CPM, the gas was not treated as continuum and followed gas molecular dynamics. Instead each particle in the particle method represents a group of gas molecules. The only particle-particle and particle-fabric interactions are perfectly elastic collisions. The method has become state-of-the-art for all airbag applications due to its accuracy and numerical robustness and efficiency.

Dynamic rebalancing for CPM

Parallelization strategies in LS-DYNA usually attempt to divide the large problem domain into a number of smaller sub-domains by domain decomposition. For a good parallel implementation, each processor must be kept busy doing useful work and communication between processors must be kept to a minimum. For typical problems in CPM, as particle usually undergo large displacement, particles are migrating from one sub-domain to another when across domain border, thus creating a workload imbalance.

In this paper, we developed dynamic load-balancing algorithms for CPM following the work of [4]. These are needed to handle millions or tens of millions of particles modeled in large distributed-memory computer systems. The basic idea is to control the domain partitioning to give each process an equal workload during the simulation run. In CPM, each sub-domain is assigned to a single processor in the PC cluster. Particles migrate between processors until an imbalance in the workload exceeds a specified value. Then the re-partitioning is performed. Our method utilizes Recursive Coordinate Bisection (RCB) domain decomposition which recursively halves the model using the longest dimension of the input model. Load-balancing is achieved by dynamically adaptively using RCB to make sure each process has equally workload. *Figure **1*** shows the flowchart of how dynamic rebalancing is achieved in LS-DYNA for CPM.

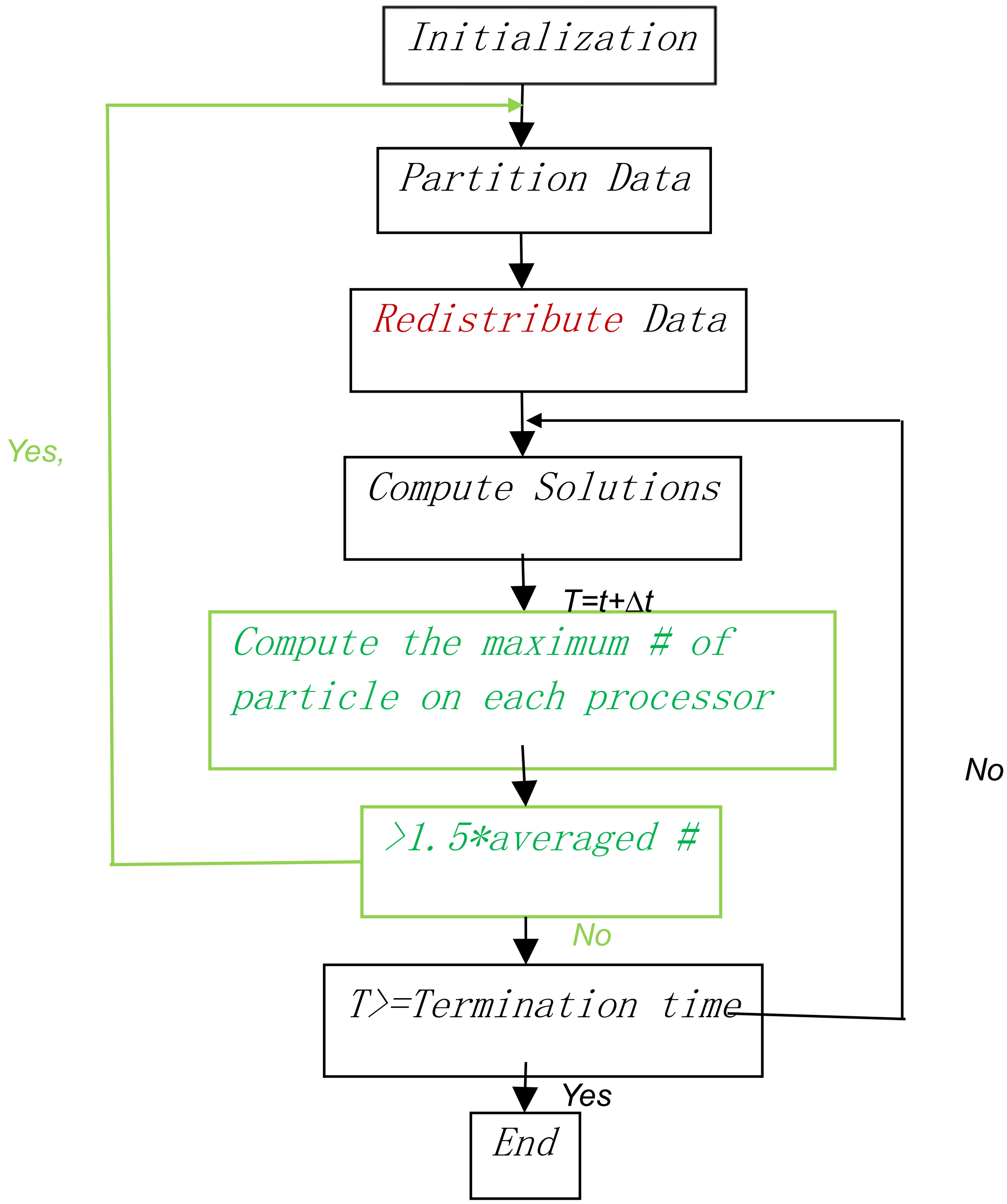

Figure 1 Flowchart of dynamic rebalancing in CPM

RCB algorithm [3] was originally developed for molecular dynamics simulations. The main advantage of this algorithm is a reduced complexity of parallel implementation. Partitioning is performed hierarchically, which means the domain is systematically partitioned at different levels.

For instance, partitioning for 8 processors (*Figure **2***) is performed in three steps. In the first step, the domain is divided in x direction into two columns containing an equal number of particles and, in the second step, each column is again divided in y direction into two rows. Next, each partition is divided in half in x direction, and thus creating 8 sub-domain with each domain contain same number of particles. *Figure **3*** shows the particle distribution at different time-step for a typical curtain airbag model. Here different color represent the sub-domain of different processor.

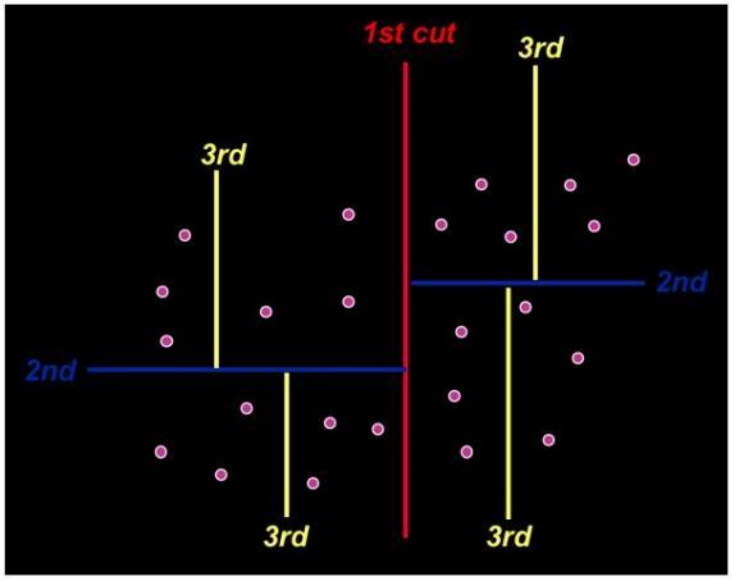

Figure 2 RCB decomposition

Figure 3 Particle distribution at different time after RCB decomposition

Numerical Model

1. Tank test

A tank test is often used to characterize airbag inflator output performance. In a tank test, an inflator is placed inside a closed tank. The inflator is deployed, and a measurement is taken of how much the pressure inside the tank increased. There are two important characteristics of the tank test curve: peak pressure, a measurement of how high the pressure gets; and pressure rise rate, a measurement of how fast the pressure changes from the time deployment begins until the peak pressure is reached. In this study, the tank model was formulated with one inner box and one outer tank. The inflator is placed in the inner rigid box, both of which are located inside an outer tank. There is a venting hole connecting the inner box and outer tank as shown in *Figure **4***.

In simulating the tank test pressure, the proposed CPM with dynamic rebalancing are performed and compared to the R9.2 version which has no rebalancing feature. The air inside tank is modeled by 472,000 air particles and the inflated gas is modeled by 528,000 gas particles. Typical pressure-time curves of inner box are shown in *Figure **4***. It can be seen that the rebalancing code get almost identical results as released R9.2.

*Table **1*** list the total CPU time comparison between R9.2 and current approach that with dynamic rebalancing version (RB). It is evident that dynamic rebalancing strategy plays an essential role in the success of effective parallel implementation for the particles method. When using 12 processes, there is not much difference in term of the total CPU time between R9.2 and RB version, however, the RB version shows much better scalability and scale up to nearly 400 processes. For R9.2 that without rebalancing, total CPU time is reduced from ~17 hours (12 processes) to 5.5 hours (384 processors), while RB is able to further reduce the time to 44 minutes. For R9.2 version, 384 cores results achieves about 3 times speedup with respect to 12 cores as shown in *able **2*** and *Figure **5***, while RB version is able to achieve more than 22 times speedup.

The relative speedup with respect to R9.2 curve is shown in *Figure **6***. It is clearly illustrated the efficiency of dynamic rebalancing strategy. With 384 cores, more than 7.5 times speedup is obtained.

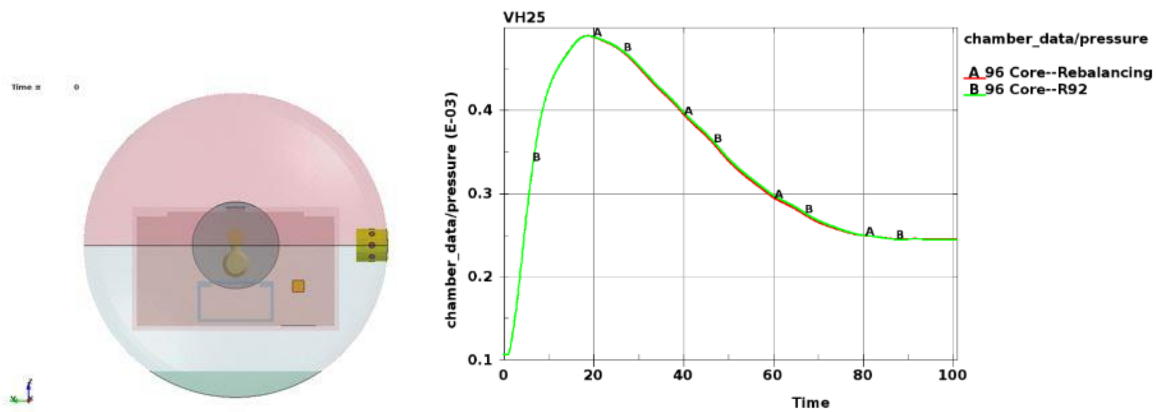

Figure 4 Tank Test model and results

#of cores	Total CPU time(s)		
	R92	RB	Speedup
12	61060 (17hrs)	59850	1.02
24	48891	30759	1.59
48	34214	16869	2.03
96	29861	8845	3.38
192	24031	4467	5.38
384	20052(5.5hrs)	2656 (44min)	7.55

Table 1 Total CPU time comparison

#of cores	Scalability	
	R92	RB
12	1.	1
24	1.2489	1.9458
48	1.7846	3.5479
96	2.0448	6.7665
192	2.5409	13.398
384	3.0451	22.534

able 2 Scalability comparison

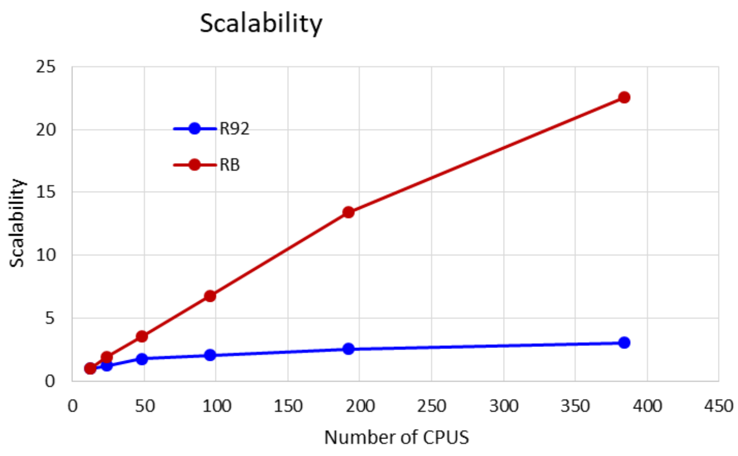

Figure 5 Scalability comparison

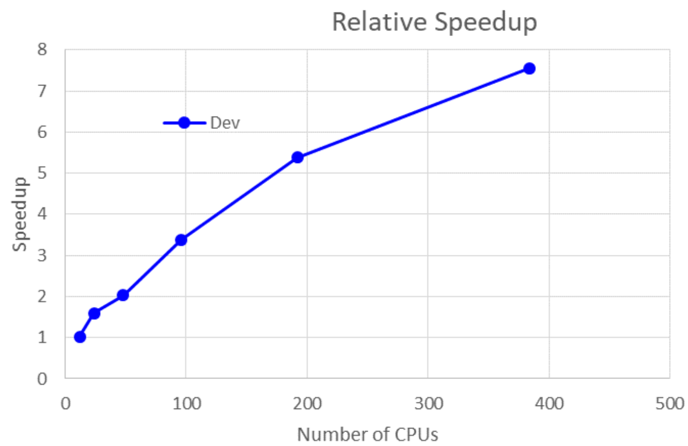

Figure 6 Relative Speedup with respect to R9.2

2. Curtain airbag

Curtain airbag has been widely used in the automotive industry to protect occupants in the events of side impact and rollover. In this paper, a LSTC logo curtain airbag model with 1,000,000 gas particles is studied as shown in *Figure 7*. *Figure **3*** shows the particle distribution at different time-step after re-decomposition is performed.

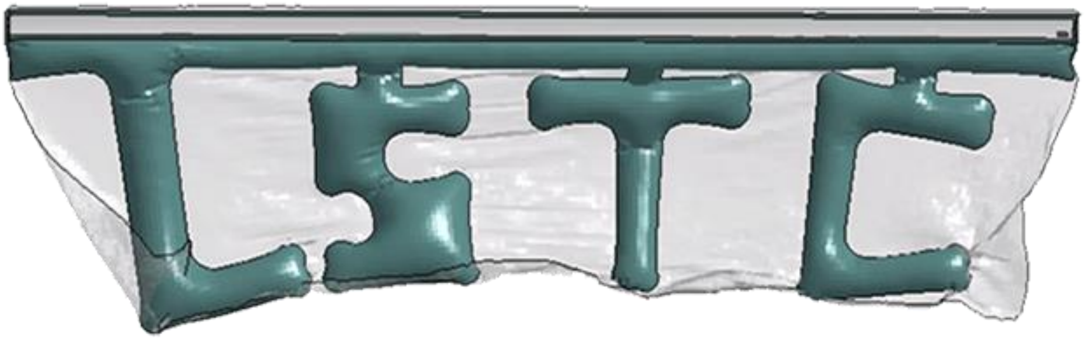

Figure 7 Curtain airbag model

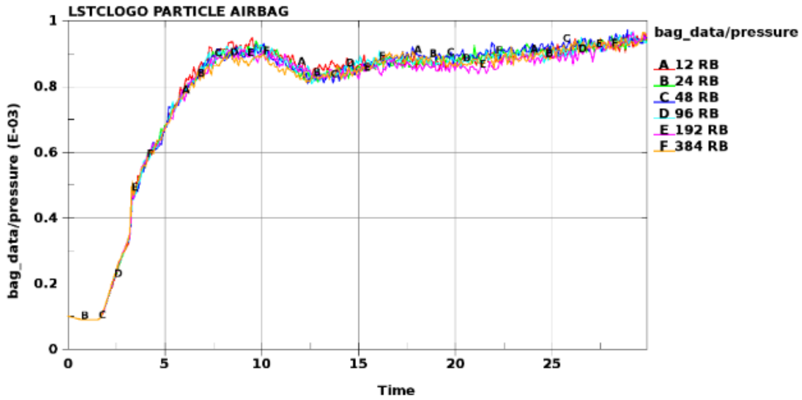

Figure 8 Curtain airbag model

#of cores	Total CPU time(s)		
	R92	RB	Speedup
12	9281	5632	1.64
24	6967	3476	2.00
48	4841	1909	2.53
96	3786	1107	3.42
192	3232	728	4.44
384	3274	632	5.18

Table 3 Total CPU time comparison

#of cores	Scalability	
	R92	RB
12	1.	1
24	1.3321	1.6203
48	1.9172	2.9502
96	2.4514	5.0876
192	2.8716	7.7363
384	2.8348	8.6914

Table 4 Scalability comparison

Curtain airbag represent a much more difficulty problem than tank test. The fabric is soft and initially folded then finally completely deployed, which make particle-fabric interaction simulation time consuming. *Figure **8*** compares the pressure when the number of particles increased from 12 cores to 384 cores and demonstrates that RB version show good consistence.

*Table **3*** list the total CPU time comparison between R9.2 and current approach that with dynamic rebalancing version (RB). With 12 processes, about 1.6 time speedup is achieved by RB version with respect to R9.2. When the number of cores increase to 384, the speedup increase to 5.18 times. With 384 cores, the simulation time is reduced from 54 minutes (3273s) to about 10 minutes (632s).

The scalability study also demonstrate the effectiveness of proposed dynamic rebalancing strategy. For R9.2 version, 384 cores results achieves about 2.8 times speedup with respect to 12 cores as shown in *Table **4*** and *Figure **9***, while RB version is able to achieve more than 8.7 times speedup. The relative speedup with respect to R9.2 curve is shown in *Figure **10***.

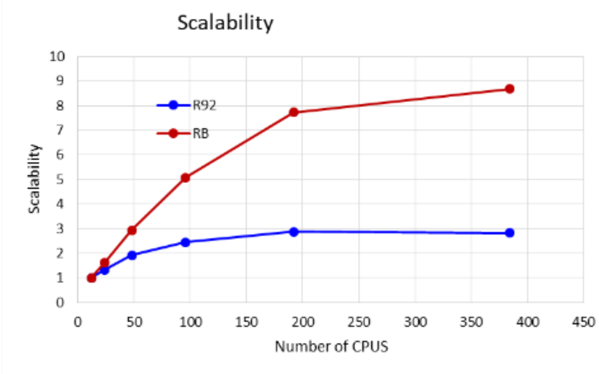

Figure 9 Scalability comparison

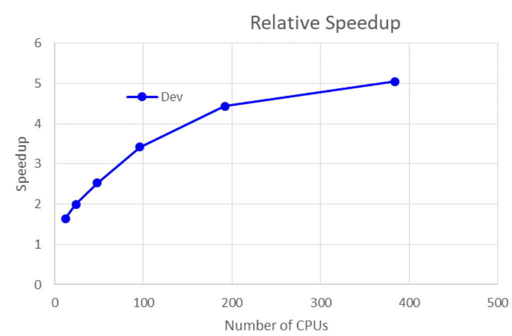

Figure 10 Relative Speedup with respect to R9.2

Summary

We introduce dynamic load balancing approach for particle method and apply it to corpuscular method (CPM). By applying a Recursive Coordinate Bisection (RCB) domain decomposition scheme, a minimization of communication expense and evenly distributed workload can be achieved.

By means of numerical experiment, the performance of the proposed algorithm is validated. It is demonstrated that the dynamic rebalancing that using RCB for dynamic re-partitioning and load imbalance detection and rebalancing scheme can achieve a much better performance in CPM applications. It is worth mentioning that the strategy suggested has been applied to particle blast method [4], and can also be applied to other areas such as DEM, SPH, and other meshless methods. The proposed algorithm was developed in Dev R127362, and partially merged into latest R10, and R11 released version.

References

1. Lars Olovsson ‘Corpuscular method for airbag deployment simulation in LS-DYNA’, ISBN 978-82-997587-0-3, 2007
2. Teng Hailong, et. al ‘The Recent Progress and Potential Applications of CPM Particle in LS-DYNA’, 11th International LS-DYNA Users Conference, 2010
3. S.G. Srinivasan, I. Ashok, H. Jonsson, G. Kalonji, J. Zahorjan ‘Dynamic-domain-decomposition parallel molecular dynamics’ Computer Physics Communication, 102 (1–3) (1997), pp. 44-58
4. Teng Hailong, et. al ‘Scalability study of particle method with dynamic load balancing’, 15th International LS-DYNA Users Conference, 2018