PARALLELIZATION AND PERFORMANCE EVALUATION OF CONTACT-IMPACT SIMULATION

Ji Joong MOON [1], Seung Hoon PAIK[1], Seung Jo KIM [2] and Minhyung Lee[3]

1) *School of Mechanical and Aerospace Engineering, Seoul National University, Seoul 151-742, KOREA*
2) *School of Mechanical and Aerospace Engineering, Flight Vehicle Research Center, Seoul National University, Seoul 151-742, KOREA*
3) *School of Mechanical and Aerospace Engineering, Sejong University, Seoul 143-747, KOREA*

Corresponding Author: Seung Jo KIM, sjkim@snu.ac.kr

ABSTRACT

Parallelization of high velocity contact-impact simulation is very difficult due to many time steps and nonlinearity of materials and complex contact behaviors. In this paper, we describe the parallelization and evaluation the parallel performance of *NET3D* which has capability to simulate brittle material. Parallelization of contact-impact simulation code can be divided by parallelization of internal force calculation and contact treatment. Each of parallelization the evaluation of performance which was carried out on the self-made linux cluster, *PEGASUS* will be discussed.

INTRODUCTION

Among many areas of numerical simulations, high velocity impact simulation requires many computing resources than other applications due to many time steps, large deformations, highly nonlinearity of materials and complex contact behaviors. Moreover, the size of finite element mesh is getting larger due to the increasing requirement of precision of model. Therefore, powerful computing power is required in order to get reliable solutions for a large scale model in a limited time. The parallel computing which utilizes many computing resources at the same time can be one of the efficient answers.

In order to develop the efficient parallel high velocity impact code, there are some difficulties. First of all, a fine serial code which can simulate complicate contact should be prepared and contact parallelization is difficult due to its complexity. Moreover, large scale computing resources are required to evaluate and improve its parallel performance. Due to these reasons, only several researches are reported from limited research groups.

The research about parallel high velocity impact simulation has been studied since early 90's and PRONTO3D which was scalable over thousands of processors in specially designed Operating System has been reported[1]. However, as the eulerian CTH code for impact problems typically runs several days on 16 nodes of SGI Origin 2000[2], the CPU time is still a burden in the real applications. Therefore, parallel high velocity impact simulation has to be studied continuously. From these reasons, *NET3D* code has been developed.

NET3D code is based on Lagrangian scheme and especially focus on numerical simulation of brittle material. In order to simulate brittle material, it has special capabilities, such as element erosion, edge to edge contact and node split scheme.

In this paper, we describe the contact parallel algorithm of *NET3D* code and then, evaluate the parallel performance with several applications such as tayler impact test, oblique metal sphere impact problem. The evaluation of parallel performance was conducted in self-made linux cluster, PEGASUS, which consists of 260 node (520CPUs) and Gigabit Network.

PARALLEL PROCEDURE OF *NET3D* CODE

Parallel procedure of *NET3D* is shown in figure 1. Parallel part has to be considered carefully when it is parallelized. At first, the code collects the basic information from input file such as mesh connectivity data, boundary condition, material model, and so on. And rearrange the data at initialization step. And then, finite element domain has to be partitioned at FE domain partitioning step. FE domain should be carefully divided because it decides the amount of calculation for each processor. Although there are some libraries which can handle, NET3D uses METIS which is based on graph partitioning algorithm which give best partition quality.

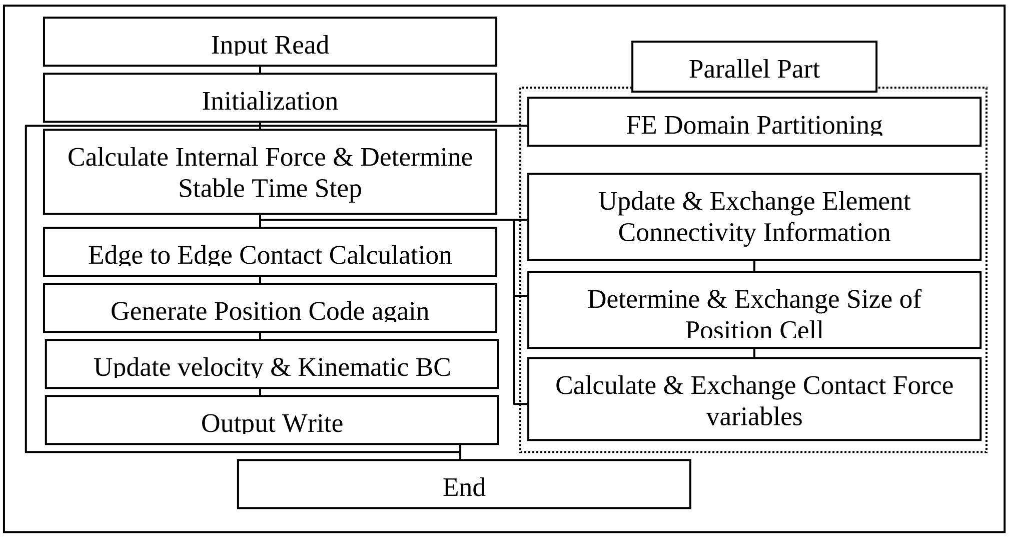

Input Read
Initialization
Calculate Internal Force & Determine Stable Time Step
Edge to Edge Contact Calculation
Generate Position Code again
Update velocity & Kinematic BC
Output Write
Parallel Part
FE Domain Partitioning
Update & Exchange Element Connectivity Information
Determine & Exchange Size of Position Cell
Calculate & Exchange Contact Force variables
End

Figure 1. Parallel Procedure of *NET3D*

Contact-impact simulation procedure can be widely divided by internal force calculation part and contact treatment part. Of course the parallelization can be considered differently and these are discussed in later section. According to this sub domain, each processor calculates the internal force and exchange internal force at boundary node between neighboring sub domain. At this procedure, critical time step which is a function of the size of mesh and material properties is calculated and exchanged and determined minimum value of each sub domain. And then three-dimensional box for contact treatment is decided based on FE sub domain. *NET3D* uses 10% bigger size of FE sub domain and calculate node to surface contact and edge to edge contact force. Finally, single time step is finished with update velocity and kinematic boundary condition and writing output at file. This single time step is repeated until the time reaches the termination time.

PARALLELIZATION OF INTERNAL FORCE CALCULATION

Parallelization of internal force calculation is relatively easier than contact force calculation. The parallelization of internal force calculation is shown in Figure 2 schematically. Several information to communicate between sub domains are required. Those are neighboring sub domain index, shared node list of each neighboring domain which can be decided with mesh connectivity information. This calculation carried out just once at the beginning of code because mesh connectivity is not changed through entire procedure except the element erosion and node split case. The parallel procedure of internal force calculation is followed and is shown in Figure 2 schematically. Domain partition of taylor impact test using METIS is shown in Figure 3.

1) First of all, find out neighboring sub domains (or processor identification number) and make neighboring sub domain list to communicate each other.
2) Each processor calculates internal forces of element and nodes inside of each assigned sub domain.
3) Using neighboring sub domain list, each processor exchanges the internal force at boundary nodes and the internal forces at boundary nodes are summed. At this time, unstructured communication[3] is used to increase the communication performance.

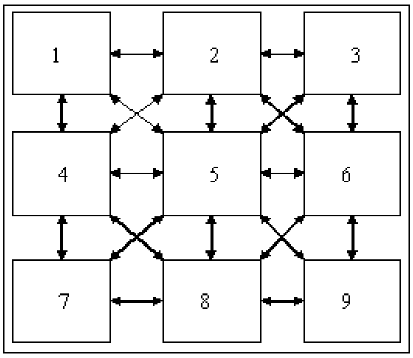


Figure 2. Parallelization of Internal Force Calculation

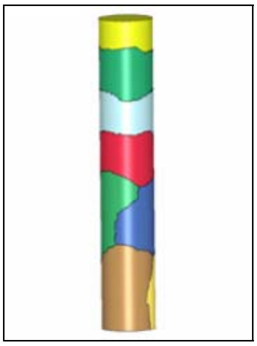


Figure 3. Domain Partitioning Using METIS

PARALLELIZATION OF CONTACT FORCE CALCULATION

The procedure of parallel contact calculation consists of four steps, partitioning of contact search domain, slave nodes and edge information update, contact force calculation and contact force update. The details of each step are followed and Figure 4 shows parallelization of contact force calculation schematically.

STEP 1: Partitioning of Contact Search Domain

First of all, the size of contact search domain should be decided. In case of *NET3D*, three-dimensional box which is 10% bigger than FE domain is used. Each domain stores the minimum and maximum of X,Y,Z direction and shares these information by broadcasting. With these information, each domain distinguishes neighboring contact search domain. And then each domain constructs the simple two-dimensional array which is the index for communication.

STEP 2: Slave Nodes and Edge information update

Each domain sorts out which nodes and edges are included in which contact search domain by using minimum and maximum values of X,Y,Z direction which are shared. And then each processor stores node and edge list according to neighboring contact search domain as two-dimensional array. And each processor exchanges information for contact force calculation.

STEP 3: Contact Force Calculation

Each processor is able to calculate contact force as serial procedure, because information for contact force is already shared.

STEP 4: Contact Force Update

After calculation of contact force at each processor, it has to be exchanged between neighboring contact search domain. Contact force of node at boundary of master surface sends and receives from neighboring domain and contact force of nodes which are received at slave nodes and edge information update step sends back to processors. After each processor receives updated contact force, contact forces of nodes at own domain are updated.

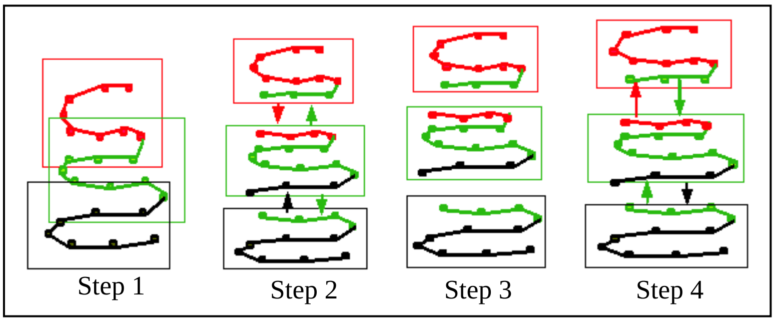
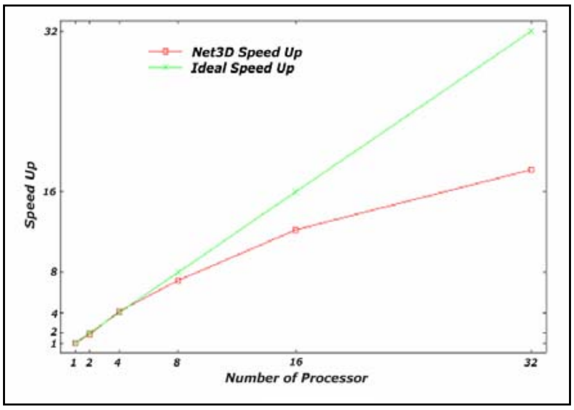


Figure 4. Parallelization of Contact Force Calculation

EVALUATION OF PARALLEL PERFORMANCE

With taylor impact test for internal force calculation and oblique metal sphere impact problem for contact force calculation, scalability test is conducted. Figure 5 and figure 6 show the performance results. 544,885 elements model is used for internal force calculation and 546,489 elements model is used for contact force calculation. *PEGASUS* linux cluster system which is consists of 520 Xeon CPUs and gigabit network system is used for test bed. As shown in figure 5 and 6, the speed up performance of internal force calculation is 12 for 16 CPUs and 5.2 speed up performance obtained for contact force calculation at 8 CPUs. As you notice, the performance decreases after 4 CPUs. That is because contact load of each processor is unbalanced.



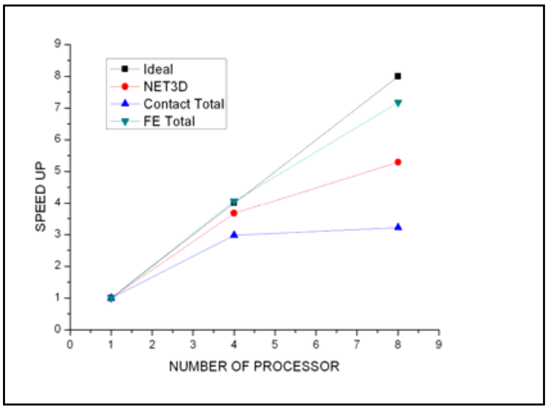


Figure 5. Speed Up Test for Internal Force Calculation (Taylor Impact Test)

Figure 6. Speed Up Test for Contact Force Calculation (Oblique Metal Sphere Impact)

REFERENCES

1. Attaway S.W., Hendrickson B.A, Plimpton S.J, Gardner D.R, Vaughan C.T., "A Parallel contact detection algorithm for transient solid dynamics simulation using PRONTO3D" Computational Mechanics. Vol. 22, pp.143-59, 1998
2. Kevin Brown, Steve Attaway, Steve Plimpton, Bruce Hendrickson, "Parallel Strategies for crash and impact simulations," Comput. Methods Appl. Engng Vol. 184, pp. 375-390, 2000
3. Paik, S.H and Kim, S.J., Parallel performance of Taylor impact test", *Proc. of Conf. on Korea Institute of Military Science and Technology*, Vol 2, pp. 852-855, KAIST, Korea, 2004