

## Parallel Computing of EFG Method on DRBL Cluster

Jiun-Yu Wu<sup>1,2</sup>, Jiun Ting Lan<sup>1</sup>, Kuen Tsann Chen<sup>1</sup>, Yao-Tsung Wang<sup>2</sup>, Steven Shiau<sup>2</sup>,

<sup>1</sup>Department of Applied Mathematics, National Chung Hsing University, Taiwan

<sup>2</sup>National Center for High Performance Computing, Taiwan

e-mail: adherelinux@hotmail.com, lambow@pchome.com.tw, ktchen@amath.nchu.edu.tw, jazz@nchc.org.tw, steven@nchc.org.tw.

**Abstract**—In this paper, we present the application of element-free Galerkin (EFG) method to analyze two-dimensional elastostatics problems, for example, cantilever beam. MPI parallel programming process is exploited to increase the computational efficiency and to mitigate the time consumed in the tremendous calculations using the element-free Galerkin method. We propose a powerful computation efficient architecture for CPU Cluster Using DRBL. The architecture help administrator to quickly deploy and manage CPU Cluster environment [15], it also bring benefit of computational efficiency in scientific computing. We have executed job on DBRL Cluster. The total time, speedup and efficiency have estimated for cantilever beam problem. We execute parallel implementation on DRBL Cluster [10]. For 16 cores, the speedup and efficiency are obtained to be 12.34 and 77.125% in cantilever beam problem.

**Keywords**-EFG; parallel computing; DRBL Cluster; MPI.

### I. Introduction

The partial differentiation equation usually describes a physical phenomenon in engineering science and engineering physics, like Navier-Stokes equation, heat conduction equation, vibration equation, wave equation, and so on. The smoothed particle hydrodynamics method (SPH) was proposed by Lucy [16] and it has been used to solve nonaxisymmetric phenomena in astrophysics. It has been applied to fluid mechanics and structure mechanics, etc. The SPH is represented by a set of particles, which move according to governing equations.

EFG can tackle initial value, boundary problems, linear and nonlinear partial differential problems [7-9]. Meshless methods, computational simulation techniques whose discrete model of the problem domain is described by nodes instead of predefined meshes [1-4]. Belytschko et al. [5, 6] (1994) developed the Element-Free Galerkin (EFG) method which used the moving least-squares (MLS) approximation to construct the shape function and employed Lagrange multipliers to satisfy the essential boundary condition. We use multiprocessor to calculate EFG numerical simulation. MPI parallel programming process is exploited to increase the computational efficiency and to mitigate the time consumed in the tremendous calculations using the element-free Galerkin method.

MPICH2 is a tool of the Message-Passing Interface for CPU [11, 12]. MPICH2 was proposed by Argonne National Lab. MPICH2 is open source which is freely available

license. It support system, including Microsoft Windows, Unix and Linux (ubuntu, centos, Fedora, etc.). The latest of version is 2-1.0 that we can download on official website. MPICH2 is implementation for distributed-memory and shared memory in parallel computing. MPICH2 offer parallel programming library which supports C, C++, Fortran language. The MPICH2 offers us some library which uses very convenience. In this paper, our program has been written in C language using MPI message passing library and execute on DRBL Cluster architecture. We use MPICH2 software easily by DRBL Cluster architecture.

### II. MATHEMATICAL FORMULATION

#### A. Moving Least Squares Approximation

Moving least square (MLS) interpolants is used for the construction of the shape function in EFG method. The approximation  $u^h(x)$  of the field variable  $u(x)$  in the domain  $\Omega$  has the following form:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \quad (1)$$

where

$$\mathbf{p}^T(\mathbf{x}) = [1 \quad x \quad y \quad x^2 \quad xy \quad y^2], \text{ for 2-D} \quad (2)$$

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \quad (3)$$

where

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^n w(\mathbf{x}-\mathbf{x}_i)\mathbf{p}^T(\mathbf{x}_i)\mathbf{p}(\mathbf{x}_i) \quad (4)$$

$$\mathbf{B}(\mathbf{x}) = [w(\mathbf{x}-\mathbf{x}_1)\mathbf{p}(\mathbf{x}_1), \dots, w(\mathbf{x}-\mathbf{x}_n)\mathbf{p}(\mathbf{x}_n)] \quad (5)$$

$$\mathbf{u}^T = [u_1, u_2, \dots, u_n] \quad (6)$$

In the present study, exponential weight function [1] was used as

$$w(d_i^2) = \begin{cases} \frac{e^{-(d_i/c)^2} - e^{-(d_{ml}/c)^2}}{(1 - e^{-(d_{ml}/c)^2})}, & d_i \leq d_{ml} \\ 0, & d_i > d_{ml} \end{cases} \quad (7)$$

Hence, we have

$$\mathbf{u}^h(\mathbf{x}) = \Phi(\mathbf{x})\mathbf{u} \quad (8)$$

### B. EFG Method with Lagrange Multipliers

In the linear elastostatics problem, the variational form with Lagrange multipliers is given by [2]

$$\begin{aligned} & \int_{\Omega} \delta(\mathbf{L}\mathbf{u})^T (\mathbf{C}\mathbf{L}\mathbf{u}) d\Omega - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b} d\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T \bar{\mathbf{t}} d\Gamma \\ & - \int_{\Gamma_u} \delta\lambda^T (\mathbf{u} - \bar{\mathbf{u}}) d\Gamma - \int_{\Gamma_u} \delta\mathbf{u}^T \lambda d\Gamma = 0 \end{aligned} \quad (9)$$

where  $\mathbf{L}$  is a matrix differential operator;  $\mathbf{C}$  is a matrix of material constants;  $\mathbf{b}$  is the vector of external body forces;  $\lambda$  is vector of the Lagrange multipliers.

Hence, the final discrete equation can be written in the following matrix form:

$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ \mathbf{q} \end{Bmatrix} \quad (10)$$

where

$$\mathbf{K}_{ij} = \int_{\Omega} \mathbf{D}_i^T \mathbf{C} \mathbf{D}_j d\Omega \quad (11)$$

$$\mathbf{D}_i = \mathbf{L}\Phi_j = \begin{bmatrix} \Phi_{i,x} & 0 \\ 0 & \Phi_{i,y} \\ \Phi_{i,y} & \Phi_{i,x} \end{bmatrix} \quad (12)$$

$$\mathbf{G}_{ij} = - \int_{\Gamma_u} \mathbf{N}_i^T \Phi_j d\Gamma \quad (13)$$

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad \text{for plane stress} \quad (14)$$

and  $\mathbf{N}_i$  is the Lagrange interpolant for node  $i$  on the essential boundary.

## III. SYSTEM ENVIRONMENT

### A. DRBL

Diskless Remote Boot in Linux (DRBL) is an open source solution to managing the deployment of the GNU/Linux operating system across many clients. DRBL supports lots of popular GNU/Linux distributions, and it is developed based on diskless and systemless environment for client machines. Figure 1 shows DRBL system architecture. DRBL uses PXE/Etherboot, DHCP, TFTP, NFS and NIS to provide services to client machines, so it is not necessary to

install GNU/Linux on the client hard drives individually. Users just prepare a server machine for DRBL to be installed as a DRBL server, and follow the DRBL installation wizard to configure and dispose the environment for client machines step by step. It is really an easy job to deploy a DRBL environment on clustering systems even for a GNU/Linux beginner, hence cross-platform and user-friendly are the key factors that make the DRBL become a superior clustering tool.

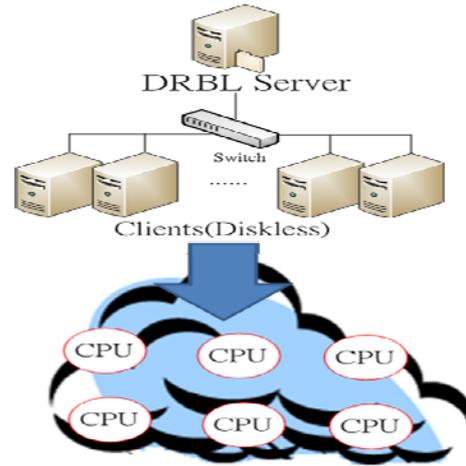


Figure 1. DRBL Cluster architecture.

DRBL can efficiently deploy diskless or diskfull cluster environment, and manage client. It configures these services (TFTP, NIS, DHCP, and NFS) to build a cluster environment. According to this implementation, administrator just needs two steps to deploy cluster environment. (1) Step 1: Installs DRBL packages and generates kernel and initrd for client; (2) Step 2: setup environment parameters, such IP address, and numbers of clients. It also provides cluster management and cluster system transformation (diskfull or diskless system).

The DRBL Cluster uses computers of PC classroom in our research center. One of computers has already been installed software (such as: MPICH2, fort77, g++, gcc, etc.) as DRBL server. It's very flexible to transform between two different modes cluster environment (diskfull and diskless) through DRBL. The cluster has 1 server, 7 clients, The PC are equipped with Intel® Core(TM)2 Quad CPU Q9550 @ 2.83GHz. Table I illustrates Hardware specifications and Software list.

TABLE I. HARDWARE SPECIFICATIONS AND SOFTWARE LIST

Hardware (PC)	Software
Intel <sup>(®)</sup> Core™ 2 Quad CPU Q6600 2.4Hz	Ubuntu 10.04
8 GB RAM	Kernel 2.6.32.21

160GB Hard disk	DRBL 1.94-27
Intel 82571EB Gigabit NIC	gcc, g++, fort77, MPICH2
<b>Hardware(Network switch)</b>	
Linksys SLM2048 48 port 10/100 Gigabit Switch	

IV. NUMERICAL EXAMPLE AND PARALLEL IMPLEMENTATION

Consider a cantilever beam with length  $L=48m$  and height  $W=15 m$  was subjected to a concentrated load  $P=1000 N$  at the free end, as shown in Fig. 2.

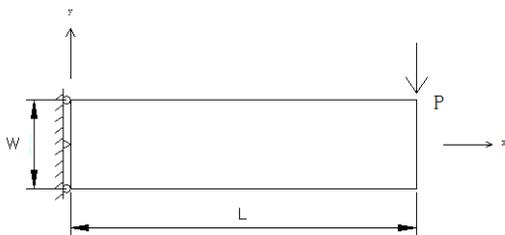


Figure 2. Cantilever Beam

This case was solved for the plane stress with Young's modulus  $E = 3 \times 10^7 N/m$  and Poisson's ratio  $\nu = 0.3$ . A regular arrangement of  $48 \times 15$  nodes and regular integration cells with  $4 \times 4$  Gauss quadrature were used. The normal stress  $\sigma_x$  and the shear stress  $\tau_{xy}$  along the line  $x=L/2$  is shown in Figs. 3 and 4, respectively. The relative error of the  $\sigma_{max}$  and  $\tau_{max}$  of  $x=L/2$  are 0.06% and 0.77%.

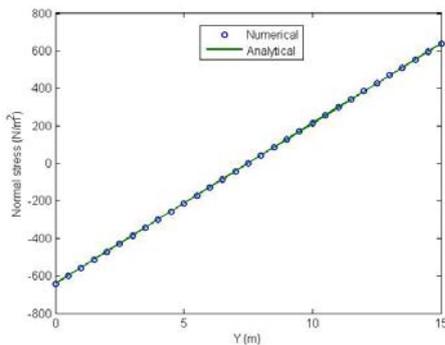


Figure 3. Distribution of normal stress  $\sigma_x$  at  $x=L/2$ .

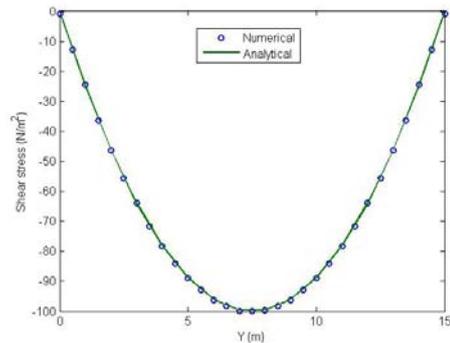


Figure 4. Distribution of Shear stress  $\tau_{xy}$  at  $x=L/2$ .

A. Parallel algorithm

We have divided data decomposition in the parallel algorithm. Whole domain is divided into small subdomains and each processor performs the work on the subdomains. The EFG parallel implementation is consist of three parts in our code.

- First part is the following: Input data, nodal coordinates Gauss points, shape functions and its derivatives. Set up on the essential boundary. Calculation of each cell (subdomains for  $\{k\}$ ).
- Secondary part is the following: Each processor sends subdomains to assembly of the system matrices  $\{K\}$ . We will obtain the formulation :  $AX=B$
- Finally, A and B are known, we can use Gaussian Elimination for solving X unknowns.

We have measured performance between a multiprocessor system and a single processor system. Table II shows performance efficient. The figure 5 shows variation of speedup with number of processors. The figure 6 shows variation of efficiency with number of processors.

TABLE II. PERFORMANCE EFFICIENT

Number of processors	Total time	Speedup	Efficiency (%)
1	13608.5	1	100
4	3428.27	3.97	99.25
8	1759.8	7.73	96.625
16	1102.65	12.34	77.125

**B. Total time**

The total time is parallel computation run time. The total time is measured by MPI\_Wtime which is MPI's library functions.

**C. Speed up**

We obtained more great performance efficient by 16 processors. The speedup formulation is defined as following:

$$Speedup = \frac{ones\ processor(Total\ time)}{multi\ processors(Total\ time)}$$

**D. Parallel efficiency**

We have discussed about parallel efficiency here. The parallel efficiency formulation is defined as following:

$$parallel\ efficiency = \frac{Speedup}{Number\ of\ processors}$$

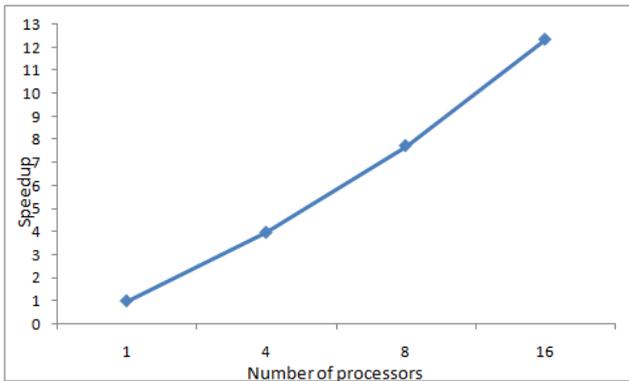


Figure 5. Variation of speedup with number of processors.

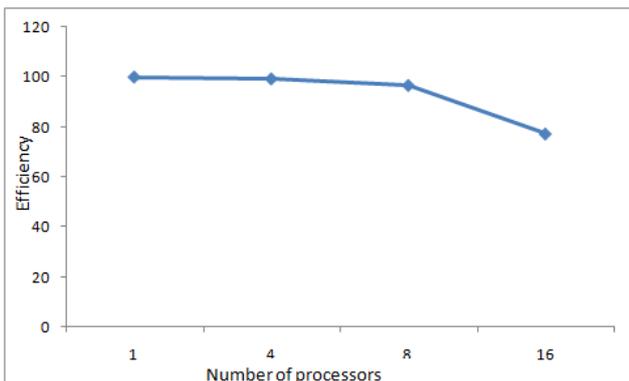


Figure 6. Variation of efficiency with number of processors.

**V. RESULT DISCUSSION**

We have been developed parallel code in C language for cantilever beam problem. We have calculated for total time, speedup and efficiency on DRBL Cluster. The EFG method for cantilever beam problem is accuracy. In the future, we will use EFG method, to solve complex problem in engineering. We have performed parallel programming for EFG method. Our parallel programming has executed on DRBL Cluster system. The DRBL Cluster is very much useful in High performance computing environment.

We consider a cantilever beam that is excited by external force. We use EFG method to solve the normal stress and the shear stress along the line  $x=L/2$ . The relative error of normal stress and the shear stress of  $x=L/2$  are 0.06% and 0.77%. The presented results confirm the theory for cantilever beam applications.

**VI. CONCLUSION AND FUTURE WORKS**

In the paper, we used EFG method to deal with an engineering problem. The finite element needs to build meshes. The EFG is no need to connect these nodes for whole domain. In other words, we don't create and arrange the meshes. Therefore, EFG method just needs to build influence domain. EFG adopts a moving least square approximation function to fit based on nodes to make those field variables are continuous in the domain.

MPI parallel programming process is exploited to increase the computational efficiency and to mitigate the time consumed in the tremendous calculations using the element-free Galerkin method.

In the future, we want to use EFG method to deal properly with kinds of complicate engineering problems like fracture extension, crack growth, kinematic boundary condition. We can employ GPGPU (General-Purpose Computing on Graphics Processing Unit) to deal with large scale problems.

**REFERENCES**

- [1] D. Broek, "Elementary Engineering Fracture Mechanics", Noordhoff International Publishing, pp. 332, 1974.
- [2] G.R. Liu and Y.T. Gu, "An Introduction to Meshfree Methods and Their Programming," Springer, 2005.
- [3] G.R. Liu, "Meshfree Methods Moving beyond the Finite Element Method," CRC Press, USA, 2003.
- [4] P. Lancaster, and K. Salkauskas, "Surfaces by Generated Moving Least Squares Method," Mathematics of Computation, vol. 37, pp. 141-158, 1981.
- [5] T. Belytschko, Y.Y. Lu, and L. Gu, "Element-free Galerkin Method", Int. J. Numer. Method Eng, vol. 37, pp. 229-256, 1994.
- [6] T. Belytschko, Y. Y. Lu, and L. Gu, "Crack Propagation by element Free Galerkin Method," engineering fracture mechanics, vol. 51, no. 2, pp. 295-315, 1995.
- [7] S. Beissel and T. Belytschko, "Nodal integration of the element-free Galerkin method," Comput. Methods. Appl. Mech. Eng, vol. 139, pp. 49-74, 1996.

- [8] L. B. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis," *The Astron. J.*, vol. 8, pp. 1013-1024, 1977.
- [9] B. Nayroles, G. Touzot and P. Villon, "Generalizing the finite element method diffuse approximation and diffuse elements," *Comput. Mech.* vol. 10, pp. 307-318, 1992.
- [10] W. C. Kuo, C. Y. Tu and Y. T. Wang, "Deploy Kerrighed SSI Massively Using DRBL," *HPC ASIA 2009*, 2009.
- [11] DeinoMPI. [Online]. <http://mpi.deino.net/> [accessed; Oct, 2011].
- [12] MPICH2 [Online]. <http://phase.hpcc.jp/mirrors/mpi/mpich2/index.htm>. [accessed; Oct, 2011].
- [13] C. Y. Tu, W. C. Kuo, Y. T. Wang and S. Shiau, "Building Energy Efficient ClassCloud using DRBL", 10th IEEE/ACM International Conference Grid Computing.
- [14] Diskless Remote Boot in Linux (DRBL), NCHC. [Online]. <http://drbl.sourceforge.net/> [accessed; Oct, 2011].
- [15] J. Cope, M. Oberg, H. M. Tufo, and M. Woitaszek, "Shared Parallel Filesystems in Heterogeneous Linux Multi-Cluster Environments," proceedings of the 6<sup>th</sup> LCI International Conference on Linux Clusters: The HPC Revolution, 2005.
- [16] L.B Lucy, "A numerical Approach to the Testing of the Fission Hypothesis", *The Atron. Astronomical Journal*, vol. 82, pp. 1013-1024, 1977.