

Parallel Finite Element Program Generation System

Huai Zhang

Laboratory of Computational Geodynamics, Graduate School of CAS, Chinese Academy of Sciences, Beijing 100039, P.R.C., hzhang@gscas.ac.cn

Guoping Liang

Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, P.R.C.

Abstract

PFEPG (Parallel Finite Element Program Generation system) is a Fortran-based runtime system for the automatic generation of parallel Fortran finite element source code. In this paper, we will document the theoretical basis of this system, both with the efficiency of its automatic loadbalance scheduler, automatic memory allocation scheduler and automatic establishment of communication subroutine library.

The PFEPG system currently running on the Dawning 2000 and IBM SP2 machines, and shows it's a pocketable system with good reliability, scalability and robustness.

1 Introduction

The design of the parallel programs calls the program-designer's knowledge on many aspects, such as the parallel environment and the parallel platform, the parallel algorithm, and the physical model of these problems. So this is very difficult for many people who are familiar with only one field of knowledge. Designing and programming a parallel code becomes time-consuming work. Many scientists and engineers are seeking a system which can automatically generate all the source codes which can run on parallel machines according to the user's model (Partial Difference Equation). The Parallel Finite Element Program Generator System (PFEPG) is based on this conception.

In this paper, the algorithm foundation is given first, then we present the PFEPG' theory basis and its practice.

The establishment of the user level communication, the setup of the parallel communication subroutine library, the automatic management of the communication data's safety, the automatic allocation of the memory, the automatic temporary disk-file and communication-file management are all presented here.

2 The Algorithm Basis

The algorithm basis of the PFEPG system is Lagrange Multiplier non-compatible Domain Decomposition Method (LMDDM). For the sake of simplicity, take the following equation as a model:

$$\begin{cases} -\Delta u + u = f, \Omega \\ u = 0, \partial\Omega \end{cases} \quad (1)$$

Here Ω is the domain of the problem, $\partial\Omega$ is the boundary of domain Ω . First, divide the domain Ω into some sub-domain $\Omega_i (i=1,2,\dots,n)$, Ω_i is quasi-uniform sub-domain with the size of d , then divide each sub-domain into finite element mesh, i.e. divide the sub-domain Ω_i into elements of h_i , let $\Omega^h = \cup\Omega_i^{h_i}$, $\Gamma = \cup\partial\Omega_i$. Then construct the finite element space:

$$S_{h \times H} \subset S_h^0(\Omega) \times S_H(\Gamma) \quad (2)$$

which:

$$S_H(\Gamma) = \left\{ \lambda \mid \lambda|_{\Gamma_{ij}} \in S_H(\Gamma_{ij}) \right\} \quad (3)$$

for h_i and d , take the assumption

$$0 < h_i < d, \quad i=1,2,3,\dots,n \quad \text{and} \quad \lim_{h \rightarrow 0} \frac{n^2 h}{d} = 0, \quad \text{in which}$$

$h = \max\{h_i\}$ 。the weak formation of the non-compatible domain decomposition method based on the Lagrange multiplier is to get $(u, \lambda) \in H(\Omega) \times H(\Gamma)$, satisfy:

$$\begin{cases} \sum_i \left\{ \frac{1}{2} (\nabla u, \nabla v)_{\Omega_i} - \langle v, \lambda \rangle_{\partial\Omega_i} \right\} = \sum_i (f, v)_{\Omega_i} \\ \sum_i \langle u, \mu \rangle_{\partial\Omega_i} = 0 \quad \forall (v, \mu) \in H(\Omega) \times H(\Gamma) \end{cases} \quad (4)$$

To solve this problem, there are two kinds of algorithms. One is PCGM or other iterative methods. The other is LMDDM. It is a kind of task parallelism algorithm. This kind of method has many advantages that the first one does not possess.

$$U_i = A_i^{-1}(f_i - B_i X_i) \quad i = 1, 2, 3, \dots, n \quad (5)$$

take equation (5) into equation (4), get:

$$\sum_i (D_i - B_i^T A_i^{-1} B_i) X_i = \sum_i (g_i - B_i^T A_i^{-1} f_i) \quad (6)$$

then we get the boundary stiffness matrix on boundary $\partial\Omega_i$. After get the boundary variable X_i , back substitute them each sub-domain to get the inner variable U_i .

3 The Designing of the User Level Communication Protocol and the Guarantee of the Data Safety

The target of PFEPG system is to generate a robust, pocketable and scalable parallel Fortran source code for its users. All the work that the users must to do is to offer PFEPG system the information if PDE and the domain information of the problem to be solved. The programs generated by PFEPG system can not only run on the single node but also on a distribute parallel machine. There are mainly seven parts in PFEPG system, they are pre-processing generator, Element subroutine generator, algorithm generator, memory management, the communication subroutine library and the post-processing generator.

For the sake of automatically generate a robust parallel Fortran program, the PFEPG system must contains a user level communication protocol which will guarantee the data safety when the information in transmission. All nodes, without informing each other

beforehand, will know what the next communication step will be, such as sending, receiving, the size of the sending messages and receiving messages. This protocol is based on the Message Passing Interface. As shown below:

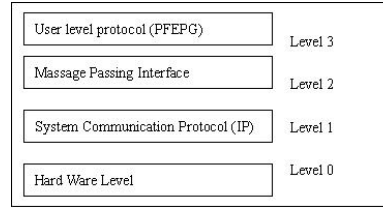


Figure 2 The communication levels

The MPI's message passing library (MPL) supplies rich procedures for different types of users, which contains two communication methodologies: point to point communication and collective communication. Since these communication methodologies have different advantages and disadvantages under different situations, rational selection of these methodologies can improve communication efficiency and the parallel degree of parallel communication programs. For example, by running test on the two parallel machines the Dawning 2000 and IBM SP2, the communication data size is 512k bytes, the efficiency of communication of the buffer mode is at the highest efficiency, as the following graph (figure 3).

So this system supply a automatic selection of communication methods on a different parallel machine for users, and the goal of the selection of communication under different message size supplies a higher efficiency of communication mode for users.

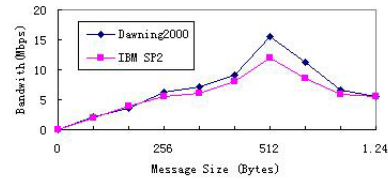


Figure 3. The compassion of buffer communication

4. Automatic Management of Node Assignments and Automatic Implement of Load Balance

When designing parallel algorithms and parallel

computing programs, a main pending problem is that how to implement the load balance of each node. In the designing of PFEPG system, one of the emphases is how to implement the load balance technique of computing assignments without users' direct interference. The arithmetic as follows:

a. Account the amount of Computation in sub-domain $T_i \quad i=1,2,3,\dots,n$; map the amount of computation in each sub-domain to jobs' queue which length is N ;

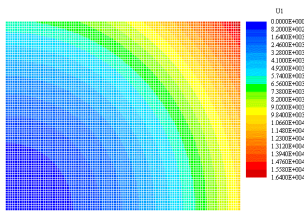
b. On the assumption that there are $M+1$ nodes to be concerned with this parallel process, map N assignments to M nodes, make the No. j node have k jobs, then system know the amount of assignment in each

$$\text{node } Mtask_j = \sum_{i=1}^k T_i ;$$

c. Optimize the task of the node of M , let $Max(Abs(Mtask_i - Mtask_j)) = Min$, here $i=1,2,3,\dots,m \quad j=1,2,3,\dots,m$; map the optimized task sequence to the M nodes $Mtask_i \Rightarrow Rank_i \quad i=1,2,3,\dots,m$;

5. Numerical Sample

In order to validate the reliability and pocketability of PFEPG, we have done a lot of numerical test. The results showed that the Fortran source code generated by PFEPG system can run on the Dawning 2000 and IBM SP2. We compute an equation of Possion, $\Delta u=4x$, equally divide the two dimension rectangle domain into 80 sub-domains 8×10 , divide every sub-domain into 10×10 meshes, the following is the result of the



computation running on Dawning 2000 machine:

Figure 5. the meshes of the sub-domains and the contour graph

6 Conclusion

In this paper, the designing technique of PFEPG system is introduced in detail. From this we can see that the PFEPG is a robust system with good scalability and pocketability. The Fortran Finite Element source codes generated by PFEPG can run both on Dawning 2000 and IBM SP2 without any changes or modification.

Acknowledgement

The authors gratefully acknowledge the inspiration of Prof. Shi Jincai of the National Research Center of Intelligent Computation (NRCIC), Prof. Liu Haichang and graduate student Ba Zhixin of Tsinghua University. Many helpful and constructive suggestions are essential to this project. The authors also sincerely thank NRCIC for the use of Dawning-2000 parallel computer.

References

[1].The Languge multiplier domain decomposition method of hybrid finite element, Guoping Liang, Ping Liang, J. Computation Math., Vol. 3 1998
 [2].MPI: A Message-Passing Interface Standard, Message Passing Interface Forum, 1994,5