8th. World Congress on Computational Mechanics (WCCM8)
5th. European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 – July 5, 2008
Venice, Italy

PERFORMANCE ENHANCEMENT OF EIGENVALUE PROBLEM BASED ON BLOCK LANCZOS ITERATION IN PARALLEL COMPUTING ENVIRONMENT

***Wanil Byun[1], Sung Hwan Cho[2], Seung Jo Kim[3] and Si Hyong Park[4]**

School of Mechanical and Aerospace Engineering, Seoul National University
151-744, Seoul, South Korea
[1]wibyun@aeroguy.snu.ac.kr [2]shcho@aeroguy.snu.ac.kr
[3]sjkim@snu.ac.kr [4]shpark@midasit.com

Key Words: *Block Lanczos Method, Parallel Performance Enhancement, Eigenvalue Problem, Multifrontal Solver, IPSAP.*

ABSTRACT

The IPSAP is a finite element analysis program which has been developed for high parallel performance computing. This program consists of various analysis modules such as stress, vibration and explicit analysis module. In the vibration module, the M orthogonal block Lanczos algorithm with shift-invert transformation [1] is used for solving eigenvalue problems. The multifrontal algorithm [2-3] which is one of the most efficient direct linear equation solvers are applied for factorization and triangular system solving phases in the block of Lanczos iteration routine.

The performance enhancement procedures of the IPSAP are composed of three stages:

1) Effective implementation of mass matrix multiplication by a Lanczos vector distribution technique,

2) Idling time minimization in triangular system solving phase by partial inverse of the frontal matrix and the LCM (least common multiple) concept,

3) Communication volume minimization of the factorization phase by modifying parallel matrix subroutines.

The three proposed stages are associated with the following three major costs in eigenvalue extraction which are factorization of K-σM, maintenance of M orthogonality and solving linear systems in the Lanczos iterator. Reducing these prohibitive costs is the proposition of this study.

Performance tuning of the multifrontal solver is conducted from three points of views.

The first one is to reduce communication occurring in the Cholesky factorization. It is notable that the naming convention of subroutines used in the eigensolver is based on BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra Package). Since the symmetric frontal matrix is composed of three matrix entities, hence three routines are required to factorize the frontal matrix as shown in Figure 1. However the parallel matrix operations used in this research are based on panel communication where the each routine performs panel or block communications which are column/row broadcasting or reducing type. Considering the communication pattern, it is apparent that some parts of broadcasting or reducing blocks are duplicated. For example, column broadcasting of panels or K_{11} in POTRF is also present in TRSM. Hence combining the three routines into one will avoid duplicated communications and reduces the cost.

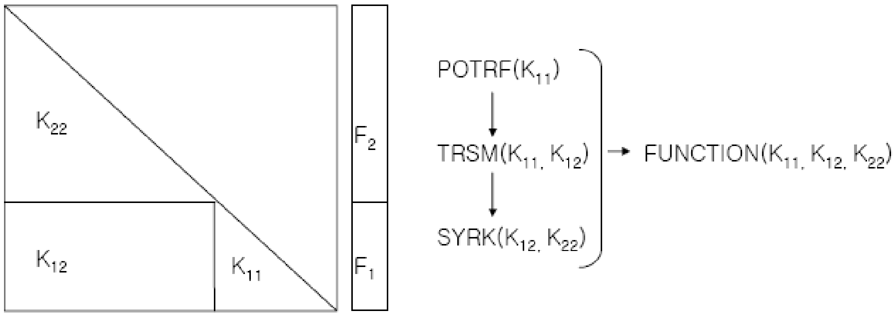

Figure 1. Frontal matrix and factorization routines

The next is to tune the triangular system solving routines. Operations in the triangular system solving are generally composed of TRSM and GEMM with F_1 and F_2. Since TRSM routine has data-dependent algorithmic flow, there may be idle processors between panel broadcasting. One idea to resolve this bottleneck is converting TRSM into TRMM by computing inverse of K_{11} using TRTRI routine. Although TRMM requires the same number of floating point operations and communication volume as TRSM does, the LCM (least common multiple) concept can apparently reduce the idling time. The LCM concept which is originally proposed in the research of GEMM [4] can be applied in case of data-independent communication pattern. In this approach, performance negotiation between computing time for inverse of K_{11} and gains by using TRSM should be taken into account.

The final performance tuning can be conducted by topology control in panel or block communication. We implemented two kinds of topology which are ‘increasing ring’ and ‘split ring’.

REFERENCES

[1] K. Wu and H. Simon, “An Evaluation of the Parallel Shift-and-Invert Lanczos Method”, *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, 2913–2919, (1999).

[2] Jeong Ho Kim and Seung Jo Kim, “Multifrontal Solver Combined with Graph Partitioner”, *AIAA Journal*, Vol. **37(8)**, pp. 964–970, (1999).

[3] Seung Jo Kim, Chang Sung Lee and Jeong Ho Kim, “Large-Scale Structural Analysis by Parallel Multifrontal Solver through Internet Based PCs”, *AIAA Journal*, Vol. **40(2)**, pp. 359–367, (2002).

[4] Choi, J., “A Fast Scalable Universal Matrix Multiplication Algorithm on Distributed-Memory Concurrent Computers”, *Proceedings of the IPPS*, pp. 310–314, (1997)