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

HYBRID PROGRAMMING OF KRYLOV TYPE ITERATIVE SOLVERS ON SMP CLUSTERS

***George Pashos[1], Antony N. Spyropoulos[2], Nikos Anastopoulos[3], Georgios Goumas[4], Nectarios Koziris[5] and Andreas G. Boudouvis[6]**

[1] School of Chemical Engineering, National Technical University of Athens
Athens 15780, Greece
gpashos@chemeng.ntua.gr

[2] School of Chemical Engineering, National Technical University of Athens
Athens 15780, Greece
aspyr@chemeng.ntua.gr

[3] School of Electrical and Computer Engineering, National Technical University of Athens
Athens 15780, Greece
anastop@cslab.ntua.gr

[4] School of Electrical and Computer Engineering, National Technical University of Athens
Athens 15780, Greece
goumas@cslab.ntua.gr

[5] School of Electrical and Computer Engineering, National Technical University of Athens
Athens 15780, Greece
nkoziris@cslab.ntua.gr

[6] School of Chemical Engineering, National Technical University of Athens
Athens 15780, Greece
boudouvi@chemeng.ntua.gr

Key Words: *Cluster, GMRES, Krylov, Message-passing/Shared Memory/Hybrid Parallel Programming Models, MPI, OpenMP.*

ABSTRACT

Modern trends in the system architecture of high-performance parallel platforms focus on the clustering of symmetric multi-processing (SMP) nodes. Message-passing (e.g. with MPI) is necessary for the communication between nodes, but on the other hand it may prove inefficient within an SMP node, where a shared memory model (e.g. with OpenMP or explicit threading), should prove more efficient. A hybrid parallel programming model using message-passing for inter-node communication and shared memory for intra-node communication seems to better adapt to the underlying architecture, since it enables coarser-grain communication (one message per node), more efficient intra-node communication using the shared memory, and more flexible memory-access schemes by all threads executing on a single node. A competent programming framework for these systems can be constructed by the pairing of MPI and OpenMP, for applications in computational mechanics, where there is usually high demand for communication [1].

Large-scale parallel computations favor iterative linear solvers, such as of Krylov type, due to their simple computational kernels (vector-vector products, matrix-vector products) that don't require either complex or excessive communication. The GMRES method is a Krylov-type solver for nonsymmetric linear systems that originate from a wide variety of problems [2]. GMRES contains a modified Gram-Schmidt orthogonalization process that performs m^2 vector-vector products and m matrix-vector products; when the vectors size is 1.000.000, typical values of m are around 100.

Assuming that the matrix is sparse – typically derived from the finite element method – then more than 50% of the GMRES total arithmetic operations come from vector-vector products.

A vector-vector product becomes communication dominated when the vectors are divided among many processing units and the overall parallel speedup is degraded. Hybrid programming MPI/OpenMP performs the communications for each vector-vector product at two stages: at first, the communication of the processors at the same node is performed with OpenMP, which is usually 10-100 times faster than the one with MPI (depending on the node interconnection network) and then follows the communication between nodes. The difference between the hybrid programming and the pure MPI implementation, for the vector-vector product, is that 1 message per node and N messages per node – where N is the number of processors in a node – are sent and received, respectively. The gain, in terms of computational time, lies on the smaller inter-node network traffic and the reduction of the total overhead time for the creation of messages since they are considerably less.

As far as the matrix-vector product is concerned, a memory-aware thread-level parallelization scheme that tackles the memory-bottleneck problem of this kernel is more suitable than creating independent MPI processes. In general, MPI processes are not flexible with respect to the way that data are decomposed and accessed, thus leaving little space for optimization in this case. Results are presented, demonstrating the advantages of hybrid programming in finite element applications on SMP clusters. The computational platform consists of 15 double-processing nodes, interconnected with a high-bandwidth, low-latency Myrinet-2000 network.

REFERENCES

[1] L. Smith and M. Bull, “Development of mixed mode MPI/OpenMP applications”, *Scientific Programming*, Vol. **9**, pp. 83-98, (2001).

[2] Y. Saad, and M.H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”, *SIAM J. Sci. Stat. Comp*. Vol. **7**, pp. 856-869, (1986).