PARALLEL FACTORIZATION OF VERY LARGE SPARSE SPD SYSTEMS ON A NETWORK OF SMP NODES

Pascal Hénon, Pierre Ramet and Jean Roman
LaBRI UMR CNRS 5800, Université Bordeaux 1 & ENSEIRB, F-33405 Talence, France
Action de recherche INRIA ScAlApplix (UR Futurs)

Keywords: High performance computing, parallel sparse Cholesky factorization, static partitioning and scheduling, SMP nodes, memory managment

Abstract

Solving large sparse symmetric positive definite systems $Ax = b$ of linear equations is a crucial and time-consuming step, arising in many scientific and engineering applications. The authors presented in [1, 3] previous works on an efficient static scheduling based on a mixed 1D/2D block distribution for the parallel supernodal version of sparse LL^T factorization with total local aggregation (block contributions are aggregated locally in block structures). Parallel experiments, using MPI communications, were run on an IBM SP2 at CINES on a collection of sparse matrices up to 1 million of unkowns and have shown that our PaStiX software compares very favorably to PSPASES. To solve problems with more than 10 millions of unkowns, which is now a reachable challenge with new SMP supercomputers, we must keep a good scalability and control memory overhead. In this context, we present two major improvements.

The first one consists of taking into account heterogeneous architectures and more particulary those based on SMP nodes like the IBM SP3. Our communication model, used during the static scheduling and mapping step, is extended to manage both data exchanges by shared memory (less costly) and data exchanges by network (more costly). To validate this approach, we have performed experiments [2] on an IBM SP3 at CINES (28 NH2+ SMP nodes, Colony switch) and on a COMPAQ AlphaServer at CEA (192 ES45 SMP nodes, Quadrics switch). For instance, our biggest problem (COUPOLE40000, 2.5D mesh from CEA collection, 26.52 millions of unkowns, 10.84 Tflop for factorization with 14.8 giga non zero coefficients in the factors) can be factorized in 20.23s on 192 ES45 nodes (540 Gflops, about 35% of the peak performance).

The second improvement is an adaptation of our static computation and communication scheduling algorithm to anticipate the sending of partially aggregated blocks in order to free memory dynamicaly. By doing this, we are able to divide the aggregated memory overhead while keeping good performance, and many experiments have shown that aggregated memory reduction is acceptable, up to 50%, in term of time penalty [2]. For instance, thanks to this technique, we can factorize the AUDI matrix (3D mesh from PARASOL collection, 0.95 million of unkowns, 5.2 Tflop for factorization with 1.2 giga non zero coefficients in the factors) in 188s on 64 Power3 processors (28 Gflops). Another natural consequence of this improvement is that we can build accurately the memory access scheme for blocks. So we can prefetch the block access during the computation and communication scheduling with a compatible order. This allow an efficient implementation of an "out-of-core" version of the PaStiX solver. We have achieved very promising results on an IBM Cluster with WH2+ SMP nodes in terms of performance and number of disk I/O.

References

[1] P. Hénon, P. Ramet, and J. Roman. PaStiX: A High-Performance Parallel Direct Solver for Sparse Symmetric Definite Systems. *Parallel Computing*, 28(2):301–321, January 2002.

[2] P. Hénon. *Distribution des Données et Régulation Statique des Calculs et des Communications pour la Résolution de Grands Systèmes Linéaires Creux par Méthode Directe.* PhD thesis, LaBRI, Université Bordeaux I, Talence, France, November 2001.

[3] P. Hénon, P. Ramet, and J. Roman. PaStiX: A Parallel Direct Solver for Sparse SPD Matrices based on Efficient Static Scheduling and Memory Managment. In *Proccedings of Tenth SIAM Conference on Parallel Processing for Scientific Computing*, March 2001.