PaStiX: A HIGH-PERFORMANCE PARALLEL DIRECT SOLVER FOR SPARSE SYMMETRIC POSITIVE DEFINITE SYSTEMS

Pascal Hénon, Pierre Ramet and Jean Roman

LaBRI, Université Bordeaux I & ENSERB, F-33405 Talence, France

Keywords: High performance computing, parallel sparse Cholesky factorization, static partitioning and scheduling.

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. In this work, we focus on the block partitioning and scheduling problem for sparse LDL^T factorization without pivoting on parallel MIMD architectures; we consider a parallel supernodal version of sparse LDL^t factorization with total local aggregation: local block contributions are aggregated locally in block structures.

As usual, three pre-processing phases are required to achieve high performance parallel sparse solving. The *ordering* step computes a symmetric permutation of the initial matrix A such that factorization will exhibit as much concurrency as possible while incurring low fill-in [4]. The *block symbolic factorization* step determines in quasi-linear space and time complexities the block data structure of the factored matrix L. Finally, the *block repartitioning and scheduling* step refines the previous partition by splitting large supernodes in order to exploit concurrency within dense block computations, and maps the resulting blocks onto the processors.

The authors presented in [1] a preliminary version describing a mapping and scheduling algorithm for 1D distribution of blocks. Then, an original algorithm based on a mixed 1D/2D block distribution has been presented in [2]; it computes an efficient static scheduling that fully drives the block computations of the parallel solver. Parallel experiments were run on an IBM SP2 at CINES (Montpellier, France), whose nodes are 120 MHz Power2SC thin nodes. These results show that our PaStiX software compares very favorably to PSPASES [3]: we achieve better solving times in almost all cases, at least for moderately sized parallel machines (up to 64 nodes). In a recent work, we have improved our static computation and communication scheduling algorithm in order to anticipate the sending of partially aggregated blocks in order to free memory and to avoid reaching a critical saturation point. By doing this, we are able to divide the aggregated memory overhead by two while keeping good performance. Moreover, we are currently working on a more efficient criterion to switch between the 1D and 2D distributions of the blocks, to improve the scalability of the parallel solver, and we are also developing a more general version of our strategy to take into account heterogeneous architectures and more particulary those based on SMP nodes like the IBM SP3. For these new developments, our preliminary experiments show that the performance achieved increases significantly.

References

[1] P. Hénon, P. Ramet, and J. Roman. A Mapping and Scheduling Algorithm for Parallel Sparse Fan-In Numerical Factorization. In *EuroPAR'99, LNCS 1685*, pages 1059–1067. Springer Verlag, 1999.

[2] P. Hénon, P. Ramet, and J. Roman. PaStiX: A Parallel Sparse Direct Solver Based on a Static Scheduling for Mixed 1D/2D Block Distributions. In *Accepted at Irregular'2000, Cancun, Mexico, to appear in LNCS*, May 2000.

[3] M. Joshi, G. Karypis, V. Kumar, A. Gupta, and F. Gustavson. PSPASES : Scalable Parallel Direct Solver Library for Sparse Symmetric Positive Definite Linear Systems. Technical report, University of Minnesota and IBM Thomas J. Watson Research Center, May 1999.

[4] F. Pellegrini, J. Roman, and P. Amestoy. Hybridizing nested dissection and halo approximate minimum degree for efficient sparse matrix ordering. In *Proceedings of IRREGULAR'99, Puerto Rico, LNCS 1586*, pages 986–995, April 1999. Extended version to be published in Concurrency: Practice and Experience.