Sparse direct Solvers using Hierarchical Matrices

Inria HiePACS

July 27, 2018

1. Supervisors

Mathieu Faverge (mathieu.faverge@inria.fr) & Pierre Ramet (pierre.ramet@inria.fr), Inria HiePACS

2. Scientific context

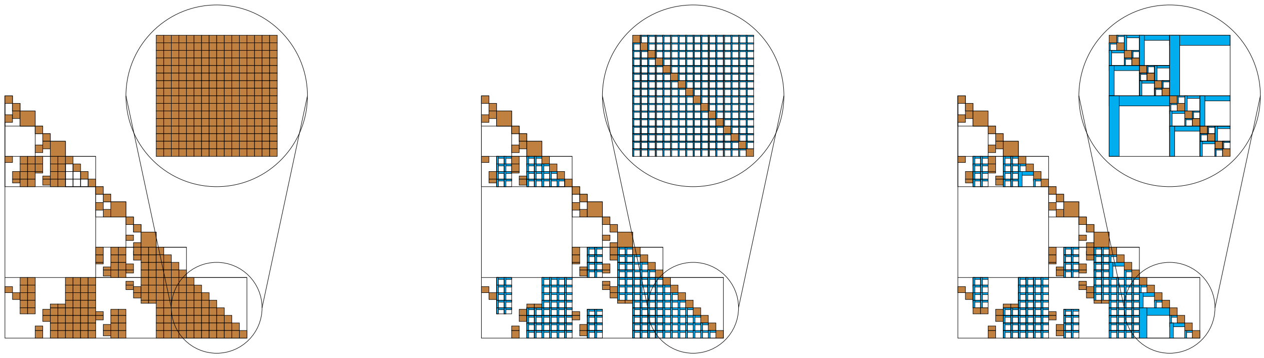
The race to exascale supercomputer systems has led to a large increase in the number of computational resources, and especially in large vectored units present in accelerators like GPUs or Intel KNL. However, this increase in the number of computational units is not followed by a rise of the amount of memory available on those systems. For example, the actual leader of the Top500 list, the system Sunway TaihuLight, is built with 1,3PB of memory for more than 10 million cores, which represents less than 128MB per core, versus an average of 2,5GB per core on systems with less accelerators like Occigen2, the current supercomputer at CINES.

This trend in the **reduction of the amount of memory available** per core forces applications not only to be extremely well parallelized, but also to reduce their memory consumption. In many scientific and engineering applications arises the need to **solve large sparse systems** of linear equations $Ax = b$, which is a crucial and time-consuming step. Many algorithmic solutions exist to solve those systems providing various ranges of numerical precision, robustness, memory consumption, and parallelism. Among them, the use of sparse **direct solvers remains the most robust solution** but their downside is their large memory consumption and time to solution.

In order to tackle very large challenging problems, research needs to be undertaken to improve sparse direct solvers that are strategic to this goal. In the last five years, those solvers have been drastically changed by introducing **compression techniques** to exploit the **low-rank** properties of the **matrices**. Many compression formats such as Block Low-Rank (BLR), $\mathcal{H}$, $\mathcal{H}^2$, HSS, HODLR... are available. They all allow a reduction of the memory requirement and/or the time to solution to solve sparse or dense linear systems with a reduced precision or to be used as a fast preconditioner. Depending on the compression strategy, solvers may require the knowledge of the underlying problem or may do it in a purely algebraic fashion. Some have been shown to be efficient while other are difficult to implement with good scalability. **The SaSHiMi project targets a purely algebraic method** to target the largest spectrum of applications possible, with a **hierarchical low-rank format** that will provide block structures to enable the **use of BLAS 3 kernels** and a **good scalability** of the solution proposed. After detailing the related work on compression techniques for dense and sparse matrices, we present the proposed method and the risks we may encounter. Finally, we discuss ordering techniques that will be investigated to reduce the complexity of using hierarchical formats in sparse direct solvers.

3. Objectives

The objective of the project is to **investigate a genuine hierarchical data sparse format in the context of supernodal sparse direct solvers** to extend the work done within the ongoing PhD thesis of G. Pichon [6]. The envisioned evolution of the solver is presented by Figure 1, with from left to right, the original full-rank storage format, the Block low-rank format from the preliminary work, and the hierarchical format that this project aims at study. The goal of the project with this format, as in the preliminary work, is to provide a **black-box algebraic solver** which does not



(a) Full-rank (b) Block low-rank (c) HODLR format

Figure 1: Objective of evolution of sparse matrix storage format for the SaSHiMi project. On the left, the original full-rank solver, on the middle, the preliminary work on BLR format, and on the right, one of possible the hierarchical format targeted by the project.

require knowledge on the problem's origin to perform the compression. The second objective is to choose a not too complex format to be **able to benefit from high performance BLAS 3** kernels.

The algebraic aspect is important to target the largest spectrum of applications as possible. Even if applications solving PDE will be the main beneficiary of this work, the algebraic implementation will allow for experimentation on any class of problems. The hierarchical formats represent promising and more advanced solutions as opposed to the flat BLR format in terms of computational complexity and memory consumption, however nested bases do not improve asymptotic complexity of the solvers in sparse cases. Thus, a much simpler format like HODLR or general $\mathcal{H}$ will be preferred. A good hierarchical format correctly exploited should provide a breakthrough to solve problems with millions of unknown on a single node, and potentially billions of unknowns on parallel distributed platforms. Indeed, hierarchical formats offer the possibility to expose more data sparsity in the matrices than the flat formats, and thus reduce the time to solution and offers a higher compression ratio than a flat compression format as BLR. Some of them have been investigated in the context of multifrontal solvers (HODLR, HSS) but they have not yet been investigated in a supernodal framework. We believe that doing it in a supernodal framework is the direction to study as it is the most memory-saving strategy for two reasons. First, it avoids the memory extra cost of allocating the fronts inherent to the multifrontal method. Second, the low-rank assembly prevents to form dense blocks before compressing them. This last overhead might be controlled in a flat format, but grows with hierarchical formats due to the recursive clustering. Furthermore, the right-looking supernodal method provides by nature a more important and flexible parallelism, which is essential to a good scalability for the large problems targeted, and for the emerging extreme scale platforms.

Among all those formats, the most advanced formats, such as HSS and $\mathcal{H}^2$, offer the best asymptotic complexities and seem really promising in dense. However, the first one is really efficient only for very large and simple problems as shown in the comparison with the MUMPS solver [5]. And for both, the complexity intrinsic to those format make them really hard to implement efficiently in the dense case. Thus, it seems too complex and too early to exploit them in a supernodal framework, and particularly since their asymptotic complexities are identical in sparse solvers. In this project, we will consider the $\mathcal{H}$ format, and more especially the HODLR subclass. This simpler format with non-nested bases makes it easier to develop efficient kernels based on BLAS 3 operations, while formats with nested bases force to have more memory bound kernels due to their row/column selection.

Reducing the complexity and the memory footprint of the solver will not be enough. An efficient parallel implementation of the solver is mandatory to scale on large distributed systems. The low-rank formats, by nature, do not have the regular pattern of computation of the dense matrix kernels. The use of hierarchical matrices, with irregular sizes like the HODLR format, will break even more this regularity of the computing tasks than with the BLR format. This will also induce more complex task dependencies to take into account the levels of the hierarchical format. That is why all the results of the project will be integrated in the PaStiX solver. This solver has been recently extended to provide alternative solutions to exploit **heterogeneous architectures** thanks to a **task based implementation on top of generic runtime systems** [3] which brings efficient support for accelerators. Prospective work has been made to support distributed systems with those runtimes in [2] and is currently under integration. This project expects to benefit from this flexible implementation to cope with the irregularity and complexity of exploiting hierarchical matrices, and provide an efficient solution on distributed, and potentially heterogeneous, architectures. Thus, the results of the research effort on hierarchical compression formats will not only study the shared memory impact, but also their impact on the distributed aspect of the solver in terms of scheduling and data exchange. In today's architectures, a key aspect in distributed algorithms is the volume of communication. The low-rank compression scheme will naturally reduce the volume of communication, thus improving again the scalability of the solver.

All the results will be developed in the PaStiX solver, and will be publicly available on the repository of the library (https://gitlab.inria.fr/solverstack/pastix). The PaStiX library is available through different interfaces such as Fortran90, Python, PETSc, and Eigen. We expect that moving toward a **hierarchical format** would allow sparse direct solvers and the applications to **break down the memory limit** they encounter. In most cases, this improvement will come for free for applications using the PaStiX library through these interfaces. However, we will also collaborate with external projects which are already using PaStiX to integrate this work in hybrid direct-iterative solvers such as MaPHyS [1] and HORSE [4], and in more complex applications like Jorek and Tokam3X from CEA Cadarache, Alya from BSC, or applications from Cerfacs and Total.

4. Prerequisites

Knowledge in linear algebra, parallelism and C

Basics on numerical stability and graph algorithms will be appreciated.

5. Supervision

This phd will be done in the context of the HiePACS project team at Inria Bordeaux - Sud-Ouest. The candidate will be supervised by Mathieu Faverge and Pierre Ramet (HiePACS team).

6. Research Theme

Domain: Networks, Systems and Services, Distributed Computing

Theme: Distributed and High Performance Computing

7. Duration

36 months with starting date between November 2018 and October 2019.

8. References

[1] E. Agullo, L. Giraud, and L. Poirel. *Robust coarse spaces for Abstract Schwarz preconditioners via generalized eigenproblems.* Research Report RR-8978. INRIA Bordeaux, Nov. 2016. URL: https://hal.inria.fr/hal-01399203.

[2] X. Lacoste. "Scheduling and memory optimizations for sparse direct solver on multi-core/multi-gpu cluster systems". PhD thesis. Talence, France: Bordeaux University, Feb. 2015.

[3] X. Lacoste, M. Faverge, P. Ramet, S. Thibault, and G. Bosilca. "Taking advantage of hybrid systems for sparse direct solvers via task-based runtimes". In: *HCW'2014 workshop of IPDPS.* Phoenix, United States: IEEE, May 2014.

[4] S. L. Ludovic Moya. *HORSE Web page.* http://www-sop.inria.fr/nachos/index.php/Software/HORSE. 2017. URL: http://www-sop.inria.fr/nachos/index.php/Software/HORSE.

[5] T. Mary. "Block Low-Rank multifrontal solvers: complexity, performance, and scalability". Theses. Université Paul Sabatier - Toulouse III, Nov. 2017. URL: https://tel.archives-ouvertes.fr/tel-01708791.

[6] G. Pichon, E. Darve, M. Faverge, P. Ramet, and J. Roman. "Sparse Supernodal Solver Using Block Low-Rank Compression". In: *18th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing (PDSEC 2017).* Orlando, United States, June 2017. URL: https://hal.inria.fr/hal-01502215.