Applied Mathematics and Computation 000 (2017) 1–12

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

The impact of enabling multiple subdomains per MPI process in the TFETI domain decomposition method

Radim Sojka[a,*], David Horák[a,b], Václav Hapla[a], Martin Čermák[a]

[a] IT4Innovations National Supercomputing Center, VŠB - Technical University of Ostrava, 17. listopadu 15/2172, Ostrava, Czech Republic
[b] Department of Applied Mathematics, VŠB - Technical University of Ostrava, 17. listopadu 15/2172, Ostrava, Czech Republic

ARTICLE INFO

Article history:
Available online xxx

Keywords:
TFETI
Domain decomposition method
PETSc
PERMON
Scalability
Oversubscription

ABSTRACT

The paper deals with handling multiple subdomains per computational core in the PERMON toolbox, namely in the PermonFLLOP module, to fully exploit the potential of the Total Finite Element Tearing and Interconnecting (TFETI) domain decomposition method (DDM). Most authors researching FETI methods present weak parallel scalability with one subdomain assigned to each computational core, and call it just parallel scalability. Here we present an extension showing the data of more than one subdomain being held by each MPI process. Numerical experiments demonstrate the theoretically supported fact that for the given problem size and number of processors, the increased number of subdomains leads to better conditioning of the system operator, and hence faster convergence. Moreover, numerical, memory, strong parallel, and weak parallel scalability is reported, and optimal numbers of subdomains per core are examined. Finally, new PETSc matrix types dealing with the aforementioned extension are introduced.

1. Introduction

Finite Element Tearing and Interconnecting (FETI) methods form an important subclass of non-overlapping domain decomposition methods (DDM) [1–6]. They proved to be very successful for the solution of large engineering problems described by elliptic PDEs. These methods solve an original problem by splitting it into smaller subdomain problems that are independent, allowing natural parallelization. The methods blend direct and iterative solvers, taking the best of both. Continuity of the primal solution between subdomains is then enforced using the Lagrange multipliers which are found by the iterative solver. Subdomain problems are solved using the direct solver. The PermonFLLOP library, being a part of the so-called PERMON toolbox [7], implements these methods. This library is based on PETSc [8] and uses the MPI [9] standard for parallelization.

The original FETI-1 method was introduced by Farhat and Roux [1,10]. Farhat, Mandel, and Roux [11] proved so called numerical scalability, i.e. the number of iterations is fixed for the fixed ratio of decomposition and discretization parameters. This established a natural effort to decompose the domain into a large number of subdomains. Yet it is still usual that FETI implementations support only one subdomain per computational core.

Here we consider the benefits of having a large number of subdomains per core. It reduces the condition number of the system operator, number of iterations, computational costs of the stiffness matrix factorization and subsequent triangular solves, and also memory occupied by the factors. This approach is known from [12], where the authors assume distributed

* Corresponding author.
E-mail addresses: radim.sojka@vsb.cz (R. Sojka), david.horak@vsb.cz (D. Horák), vaclav.hapla@vsb.cz (V. Hapla), martin.cermak@vsb.cz (M. Čermák).

http://dx.doi.org/10.1016/j.amc.2017.07.031
0096-3003/

shared memory systems. Their work was extended in [13], dealing with parallel distributed memory system and experiments with up to several millions degrees of freedom (DOFs). A similar idea is used in the multilevel FETI approach [14,15].

Comparing the sizes of benchmarks reported in [16] which have a maximum of $12 \times 12 \times 12$ elements per subdomain with those reported nowadays with e.g. more than $36 \times 36 \times 36$ elements per subdomain on ARCHER [17], we see the enormous progress made in designing supercomputers. See the setting of the first two benchmarks in Section 5.

This paper focuses on details of the PETSc implementation. It presents a new matrix type to efficiently handle data of multiple subdomains on one core. Furthermore, it especially presents strong scalability, which is very rare in papers. Most often, only weak scalability is reported. So the contribution of this paper is presenting systematically both these scalabilities on large-scale problems, and proving that the ideas of [13] are still relevant for today's supercomputers with thousands of computational nodes and tens of cores per node.

The rest of the paper is organized as follows. In Section 2, the TFETI method is explained including spectral properties of the system operator. Section 3 briefly introduces our PERMON toolbox. Section 4 deals with a new PETSc matrix type enabling more subdomains per MPI process. Finally, benefits of the new implementation are demonstrated by numerical experiments in Section 5, where we test four benchmarks for parallel, numerical, and memory scalability up to 700 million DOFs.

2. The TFETI method

FETI-1 [1,10] is DDM based on decomposing the original spatial domain into smaller non-overlapping subdomains. Continuity between these subdomains is enforced by the Lagrange multipliers enforcing fulfillment of specific "gluing" linear constraints. The original FETI-1 method assumes that the boundary subdomains inherit Dirichlet conditions from the original problem where the conditions are embedded into the linear system arising from FEM. This means physically that subdomains whose interfaces intersect the Dirichlet boundary are fixed, while others are kept floating; in the linear algebra speech, the corresponding subdomain stiffness matrices are non-singular and singular, respectively.

The basic idea of the Total-FETI (TFETI) method [4,18] is to keep all the subdomains floating and to enforce the Dirichlet boundary conditions by means of a constraint matrix and Lagrange multipliers, similarly to the gluing conditions along subdomain interfaces. This simplifies implementation of the stiffness matrix generalized inverse. The key point is that nullspaces of the subdomain stiffness matrices are known a priori, have the same dimension, and can be formed without any computation from the mesh data. Furthermore, each local stiffness matrix can be regularized cheaply, and the inverse of the resulting non-singular matrix is at the same time a generalized inverse of the original singular one [19].

Let N_p, N_d, N_n, N_c denote the primal dimension, the dual dimension, the nullspace dimension and the number of processes available for our computation, respectively. The primal dimension means the number of all DOFs including those arising from duplication on the interfaces. The dual dimension is the total number of all equality constraints. Let us consider a partitioning of the global domain Ω into N_S subdomains $\Omega^s, s = 1, \ldots, N_S$ ($N_S \geq N_c$). To each subdomain Ω^s, there corresponds the subdomain stiffness matrix $\boldsymbol{K}^s$, the subdomain nodal load vector $\boldsymbol{f}^s$, the matrix $\boldsymbol{R}^s$ whose columns span the nullspace (kernel) of $\boldsymbol{K}^s$, and the signed boolean matrix $\boldsymbol{B}^s$ defining connectivity of the subdomain s with all its neighboring subdomains. In the case of TFETI, $\boldsymbol{B}^s$ also enforces the Dirichlet boundary conditions. The local objects $\boldsymbol{K}^s$, $\boldsymbol{f}^s$, $\boldsymbol{R}^s$ and $\boldsymbol{B}^s$ constitute global objects

$$\boldsymbol{K} = \operatorname{diag}(\boldsymbol{K}^1, \ldots, \boldsymbol{K}^{N_S}) \in \mathbb{R}^{N_p \times N_p}, \quad \boldsymbol{R} = \operatorname{diag}(\boldsymbol{R}^1, \ldots, \boldsymbol{R}^{N_S}) \in \mathbb{R}^{N_p \times N_n},$$
$$\boldsymbol{B} = [\boldsymbol{B}^1, \ldots, \boldsymbol{B}^{N_S}] \in \mathbb{R}^{N_d \times N_p}, \qquad \boldsymbol{f} = [(\boldsymbol{f}^1)^T, \ldots, (\boldsymbol{f}^{N_S})^T]^T \in \mathbb{R}^{N_p \times 1},$$

where diag means a block-diagonal matrix consisting of the diagonal blocks in brackets. Note that columns of $\boldsymbol{R}$ also span the nullspace of $\boldsymbol{K}$.

Let us mention that the matrix $\boldsymbol{B}$ can be split vertically into two blocks corresponding to the gluing part $\boldsymbol{B}_G$ and the Dirichlet part $\boldsymbol{B}_D$, and at the same time horizontally into blocks $\boldsymbol{B}^s$, $s = 1, \ldots, N_S$, where each block $\boldsymbol{B}^s$ corresponds to subdomain Ω^s and consists of the gluing part $\boldsymbol{B}_G^s$ and the Dirichlet part $\boldsymbol{B}_D^s$. Overall, $\boldsymbol{B}$ possesses the following structure:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_G \\ \boldsymbol{B}_D \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_G^1 & \ldots & \boldsymbol{B}_G^{N_S} \\ \boldsymbol{B}_D^1 & \ldots & \boldsymbol{B}_D^{N_S} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^1 & \ldots & \boldsymbol{B}^{N_S} \end{bmatrix}.$$

The matrix $\boldsymbol{B}_G$ is constructed in such a way that for each pair of connected DOFs, a row with exactly two appropriately placed nonzero values -1 and 1 are added to $\boldsymbol{B}_G$. For numerical and implementation reasons, it is recommended to use fully redundant connections, i.e. all DOFs in the decomposed problem representing the same DOF in the undecomposed problem are connected to each other. Moreover, the ± 1 values of $\boldsymbol{B}_G$ must be scaled by $1/\sqrt{m}$ where m is the multiplicity (number of DOFs in the group). Note that [11] introduces this scaling as a part of the FETI preconditioners. However, it makes sense even if preconditioning is not used, since it improves conditioning on its own.

Similarly, for each Dirichlet boundary DOF, $\boldsymbol{B}_D$ contains a separate row with the appropriately placed value of 1. Notice $\boldsymbol{B}_D^s$ is a zero matrix if the boundary of Ω^s does not intersect with the Dirichlet boundary. For more details, see [20].

For the sake of simplicity, let us assume that the boundary (interface) DOFs are numbered last, and index sets i and b correspond to the internal and boundary DOFs, respectively. In the TFETI approach, index set b includes the DOFs on the

Please cite this article as: R. Sojka et al., The impact of enabling multiple subdomains per MPI process in the TFETI domain decomposition method, Applied Mathematics and Computation (2017), http://dx.doi.org/10.1016/j.amc.2017.07.031