Practical aspects of domain decomposition in Jacobi-Davidson for parallel performance

Menno Genseberger[1]

1 Introduction

Most computational work in Jacobi-Davidson [7], an iterative method for large scale eigenvalue problems, is due to a so-called correction equation. For this, to reduce wall clock time and local memory requirements, [3, 5] proposed a domain decomposition strategy that was further improved in [4] (§ 2 and § 3).

Here we investigate practical aspects for parallel performance of the strategy by scaling experiments on supercomputers (§ 4). This is of interest for large scale eigenvalue problems that need a massively parallel treatment.

2 Domain decomposition

In [3, 5] a domain decomposition preconditioning technique for the (approximate) solution of the correction equation was proposed. This technique is based on a nonoverlapping additive Schwarz method with locally optimized coupling parameters by Tan & Borsboom [8, 9] (belonging to the class of optimized Schwarz methods [2]).
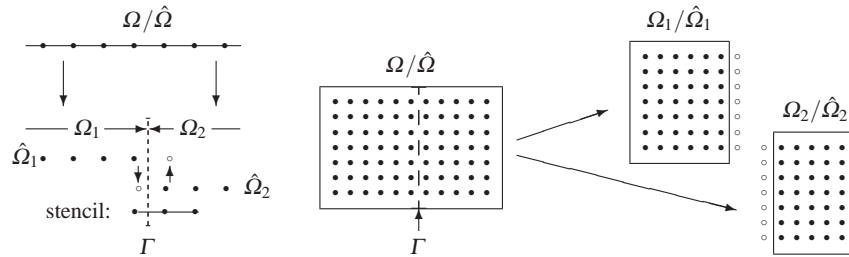

Fig. 1 Decomposition in one (left picture) and two dimensions (right picture).

For some partial differential equation (PDE) defined on a domain Ω with appropriate boundary conditions, Ω is covered by a grid $\hat{\Omega}$ and the PDE is discretized accordingly, with unknowns defined on the grid points, yielding the linear system

$$\mathbf{B}\mathbf{y} = \mathbf{d}. \tag{1}$$

---

[1] Deltares, PO Box 177, 2600 MH Delft, The Netherlands,
e-mail: Menno.Genseberger@deltares.nl

Now, the domain decomposition technique
1. Enhances the linear system (1) into $\mathbf{B}_C \mathbf{y}_\approx = \mathbf{d}_0$ with the following structure

$$
\begin{bmatrix}
\mathbf{B}_{11} & \mathbf{B}_{1\ell} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{B}_{\ell 1} & B_{\ell\ell} & B_{\ell r} & 0 & 0 & \mathbf{0} \\
\mathbf{0} & C_{\ell\ell} & C_{\ell r} & -C_{\ell\ell} & -C_{\ell r} & \mathbf{0} \\
\mathbf{0} & -C_{r\ell} & -C_{rr} & C_{r\ell} & C_{rr} & \mathbf{0} \\
\mathbf{0} & 0 & 0 & B_{r\ell} & B_{rr} & \mathbf{B}_{r2} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{2r} & \mathbf{B}_{22}
\end{bmatrix}
\begin{bmatrix}
\mathbf{y}_1 \\ y_\ell \\ \tilde{y}_r \\ \tilde{y}_\ell \\ y_r \\ \mathbf{y}_2
\end{bmatrix}
=
\begin{bmatrix}
\mathbf{d}_1 \\ d_\ell \\ 0 \\ 0 \\ d_r \\ \mathbf{d}_2
\end{bmatrix}
\tag{2}
$$

in case of a two subdomain decomposition (generalization is straightforward). Here Ω is decomposed in two nonoverlapping subdomains Ω_1 and Ω_2 with interface (or internal boundary) Γ (see Fig. 1). The subdomains are covered by subgrids $\hat{\Omega}_1$ and $\hat{\Omega}_2$ with additional grid points located just outside the subdomain near the interface Γ (the open bullets "∘" in Fig. 1) such that no splitting of the original discretized operator (or stencil) has to be made. For $\mathbf{B}$, the labels $1, 2, \ell,$ and r, respectively, refer to operations on data from/to subdomain Ω_1, Ω_2, and left, right from the interface Γ, respectively. For $\mathbf{y}$ and $\mathbf{d}$, the labels $1, 2, \ell,$ and r, respectively, refer to data in subdomain Ω_1, Ω_2, and left, right from the interface Γ, respectively. Here, subvector y_ℓ (y_r respectively) contains those unknowns on the left (right) from Γ that are coupled by the stencil both with unknowns in Ω_1 (Ω_2) and unknowns on the right (left) from Γ. Subvector $\tilde{y}_r$ ($\tilde{y}_\ell$ respectively) contains the unknowns at the additional grid points of the subgrid for Ω_1 (Ω_2) on the right (left) of Γ. For the unknowns on the additional grid points additional equations are provided with the requirement that the submatrix (the *interface coupling matrix*)

$$
C \equiv \begin{bmatrix} C_{\ell\ell} & C_{\ell r} \\ C_{r\ell} & C_{rr} \end{bmatrix}
\tag{3}
$$

is nonsingular as for nonsingular C the solution $\mathbf{y}_\approx$ of (2) is unique, $\tilde{y}_\ell = y_\ell$ and $\tilde{y}_r = y_r$, and the restriction of $\mathbf{y}_\approx$ to $\mathbf{y}$ is the unique solution of the original linear system (1) ([9, Theorem 1], [8, Theorem 1.2.1]).
2. Splits the matrix $\mathbf{B}_C = \mathbf{M}_C - \mathbf{N}_C$ in a part $\mathbf{M}_C$, the boxed parts in (2) that do not map elements from one subgrid to the other subgrid and a remaining part $\mathbf{N}_C$ that couples the subgrids via the discretized interface with a relatively small number of nonzero elements. (Therefore matrix vector multiplication with $\mathbf{B}_C$ can be implemented efficiently on distributed memory computers.)
3. Tunes the interface coupling matrix C defined in (3) such that error components due to domain decomposition are damped in the Richardson iteration

$$
\mathbf{y}_\approx^{(i+1)} = \mathbf{y}_\approx^{(i)} + \mathbf{M}_C^{-1}(\mathbf{d}_0 - \mathbf{B}_C \mathbf{y}_\approx^{(i)}).
\tag{4}
$$

Note $\mathbf{M}_C^{-1}\mathbf{B}_C = \mathbf{I} - \mathbf{M}_C^{-1}\mathbf{N}_C$, therefore error components are propagated by $\mathbf{M}_C^{-1}\mathbf{N}_C$.
4. Computes a solution of the enhanced linear system from (4) or with a more general Krylov method like GMRES [6] with $\mathscr{K}_m(\mathbf{M}_C^{-1}\mathbf{B}_C, \mathbf{M}_C^{-1}\mathbf{d}_0) \equiv$ span$(\mathbf{M}_C^{-1}\mathbf{d}_0, \mathbf{M}_C^{-1}\mathbf{B}_C \mathbf{M}_C^{-1}\mathbf{d}_0, \ldots, (\mathbf{M}_C^{-1}\mathbf{B}_C)^{m-1}\mathbf{M}_C^{-1}\mathbf{d}_0)$.

The key idea is to use the degrees of freedom, that we have created by the introduction of additional unknowns near the interface, for damping the error components. For this purpose, the spectral properties of $\mathbf{M}_C^{-1}\mathbf{N}_C$ for the specific underlying PDE are analyzed. With results of this analysis, optimal coupling parameters can be estimated, i.e. the interface coupling matrix C defined in (3) can be tuned. In this way error components due to the splitting are damped "as much as possible", optimal choices result in a coupling that annihilates the outflow from one domain to another: absorbing boundary conditions. This leads effectively to almost uncoupled subproblems at subdomains. As a consequence, the number of iterations required for convergence is minimal with minimal communication overhead (due to the explicit step with $\mathbf{N}_C$) between subdomains: an ideal situation for implementation on parallel computers and/or distributed memory.

3 Jacobi-Davidson

For a standard eigenvalue problem $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ each iteration Jacobi-Davidson [7]
1. Extracts an approximate eigenpair $(\theta, \mathbf{u}) \approx (\lambda, \mathbf{x})$ from a search subspace $\mathbf{V}$:
construct $H \equiv \mathbf{V}^*\mathbf{A}\mathbf{V}$, solve $Hs = \theta s$, compute $\mathbf{u} = \mathbf{V}s$.
2. Corrects the approximate eigenvector $\mathbf{u}$ with a correction $\mathbf{t} \perp \mathbf{u}$ that is computed from the *correction* equation:

$$\mathbf{P}\mathbf{B}\mathbf{P}\mathbf{t} = \mathbf{r} \quad \text{where} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}\mathbf{u}^*}{\mathbf{u}^*\mathbf{u}}, \mathbf{B} \equiv \mathbf{A} - \theta\mathbf{I}, \quad \text{and} \quad \mathbf{r} \equiv -\mathbf{B}\mathbf{u}. \tag{5}$$

3. Expands the search subspace with the correction $\mathbf{t}$: $\mathbf{V}_{new} = [\mathbf{V} \,|\, \mathbf{t}^\perp]$ where $\mathbf{t}^\perp \perp \mathbf{V}$.

The linear system described by the correction equation (5) may be highly indefinite and is given in an unusual manner so that the application of the domain decomposition technique needed further development and special attention.

Similar to the enhancements (1) of the linear system (2) in § 2, the following components of the correction equation are enhanced: the matrix $\mathbf{B} \equiv \mathbf{A} - \theta\mathbf{I}$ to $\mathbf{B}_C$, the correction vector $\mathbf{t}$ to $\mathbf{t}_\approx$ and the vectors $\mathbf{u}$ and $\mathbf{r}$ to $\mathbf{u}_0$ and $\mathbf{r}_0$. With these enhancements, a correction $\mathbf{t}_\approx \perp \mathbf{u}_0$ is computed from the following enhanced correction equation [3, §3.3.2]:

$$\mathbf{P}\mathbf{B}_C\mathbf{P}\mathbf{t}_\approx = \mathbf{r}_0 \quad \text{with} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}_0\mathbf{u}_0^*}{\mathbf{u}_0^*\mathbf{u}_0}. \tag{6}$$

The preconditioner $\mathbf{M}_C$ for $\mathbf{B}_C$ is constructed in the same way as the ordinary linear system case shown by the boxed parts in (2). However, because of the indefiniteness, for the correction equation the matrices $\mathbf{B}_C$ and $\mathbf{M}_C$ are accompanied by projections. Both for left and right preconditioning the projection is as follows:

$$\mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}_C^{-1}\mathbf{u}_0\mathbf{u}_0^*}{\mathbf{u}_0^*\mathbf{M}_C^{-1}\mathbf{u}_0}. \tag{7}$$

In case of left preconditioning (for right preconditioning see [3, §3.3.3]) we compute approximate solutions to the correction equation from

$$\mathbf{P}' \mathbf{M}_C^{-1} \mathbf{B}_C \mathbf{P}' \mathbf{t}_\approx = \mathbf{P}' \mathbf{M}_C^{-1} \mathbf{r}_0. \tag{8}$$

However, there is more to gain. For approximate solutions of the correction equation with a preconditioned Krylov method, the Jacobi-Davidson method is an accelerated inexact Newton method that consists of two nested iterative solvers. In the innerloop of Jacobi-Davidson a search subspace for the (approximate) solution of the correction equation is built up by powers of $\mathbf{M}_C^{-1}(\mathbf{A} - \theta \mathbf{I})$ for fixed θ. In the outerloop a search subspace for the (approximate) solution of the eigenvalue problem is built up by powers of $\mathbf{M}_C^{-1}(\mathbf{A} - \theta \mathbf{I})$ for variable θ. As θ varies slightly in succeeding outer iterations, one may take advantage of the nesting by applying the domain decomposition technique to the outer loop as was the subject of [4]. This effectively leaded to two different processes:

- Jacobi-Davidson with *enhanced inner loop*, enhancement at intermediate level with enhanced correction equation (6) and
- Jacobi-Davidson with *enhanced outer loop*, enhancement at highest level with a slightly different correction equation

$$\mathbf{P} \mathbf{B}_C \mathbf{P} \mathbf{t}_\approx = \mathbf{r}_\approx \quad \text{with} \quad \mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}_0 \mathbf{u}_0^*}{\mathbf{u}_0^* \mathbf{u}_0}. \tag{9}$$

The amount of work for both processes per outer iteration is almost the same. However, Jacobi-Davidson with enhanced outer loop turned out to be faster as it damps remaining error components from the previous outer iteration in the next one.

4 Scaling experiments

For the two processes, in [4, § 5.1] different eigenvalue problems have been considered including variable coefficients and large jumps. Here, to investigate practical aspects for parallel performance, we consider the eigenvalue problem for the Laplace operator as results for different numbers of subdomains show more regular behavior (see for instance Fig. 3 in [4]). Except for the first experiment about different decompositions, in all experiments we take for the domain Ω the unit square, decompose Ω in p square subdomains, and cover each subdomain by a 256×256 subgrid. Jacobi-Davidson is started with a parabolic shaped vector $x(1-x)y(1-y)$ for $0 \le x \le 1$ and $0 \le y \le 1$ (see also [3, § 3.5.1]) to compute the most global eigenvector (for which the corresponding eigenvalue is the closest one to zero) of the two-dimensional Laplace operator on Ω until the residual norm of the approximate eigenpair is less than 10^{-9}. We apply right preconditioning in the enhanced correction equation for exact solves with the preconditioner (i.e. exact subdomain solves) to enable a Schur complement approach. The preconditioner $\mathbf{M}_C$ is constructed only once, at the first Jacobi-Davidson outer iteration. The remaining linear system is solved with GMRES [6].
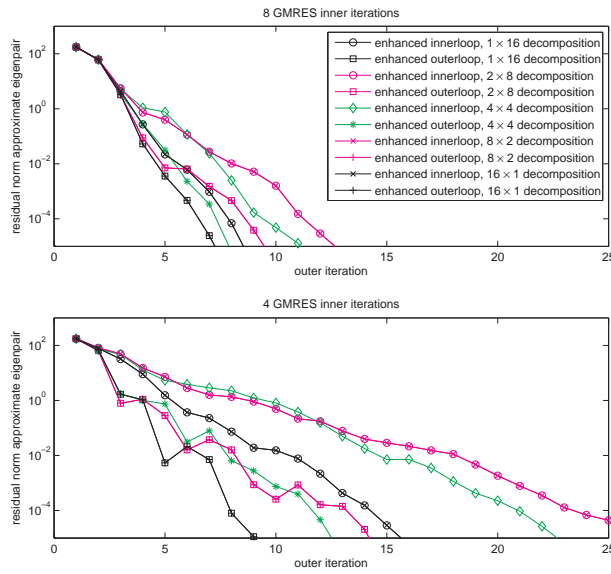
Fig. 2 Residual norm of the approximate eigenpair as a function of the Jacobi-Davidson outer iteration for the different decompositions with GMRES(8) (top) and GMRES(4) (bottom).

Implementation is in Fortran77 with calls to BLAS, LAPACK, and MPI. Note, however, that Fortran compiler, BLAS, LAPACK, and MPI versions differ on the specific hardware which is of influence on the (parallel) performance. Scaling experiments are performed on the following hardware:

- Curie linux-cluster (2 Intel eight 2.7 GHz core E5-2680 node, InfiniBand QDR, Intel Fortran 12, BLAS/LAPACK from MKL, Bull X MPI),
- H4+ linux-cluster (1 Intel quad 3.4 GHz core i7-2600 node, 1 GB/s Gigabit Ethernet, Intel Fortran 11, MPICH2),
- IBM POWER5+ system Huygens (16 IBM single 1.9 GHz core Power5+ node, 1.2 GB/s InfiniBand, XL Fortran 10, BLAS from ESSL, MPI from IBM PE),
- IBM POWER6 system Huygens (16 IBM dual 4.7 GHz core Power6 node, 160 GB/s InfiniBand, XL Fortran 12, BLAS from ESSL, MPI from IBM PE),
- Lisa 2008 linux-cluster (1 Intel Xeon 3.4E GHz core node, 800 MB/s InfiniBand, GFortran, MPICH2),
- Lisa 2012 linux-cluster (2 Intel eight 1.8 GHz core Xeon E5-2650L node, Intel Fortran 12, BLAS/LAPACK from MKL, OpenMPI),

On the H4+ and Lisa 2008 linux-clusters one subdomain is assigned to one node. On the other hardware one subdomain is assigned to one core. Results presented here are averages of three measured wall-clock times.

First we study different decompositions for a fixed number of subdomains for the same (discretized) eigenvalue problem. We keep the overall grid fixed to a size of 1024×1024 gridpoints and consider configurations with a 1×16, 2×8, 4×4, 8
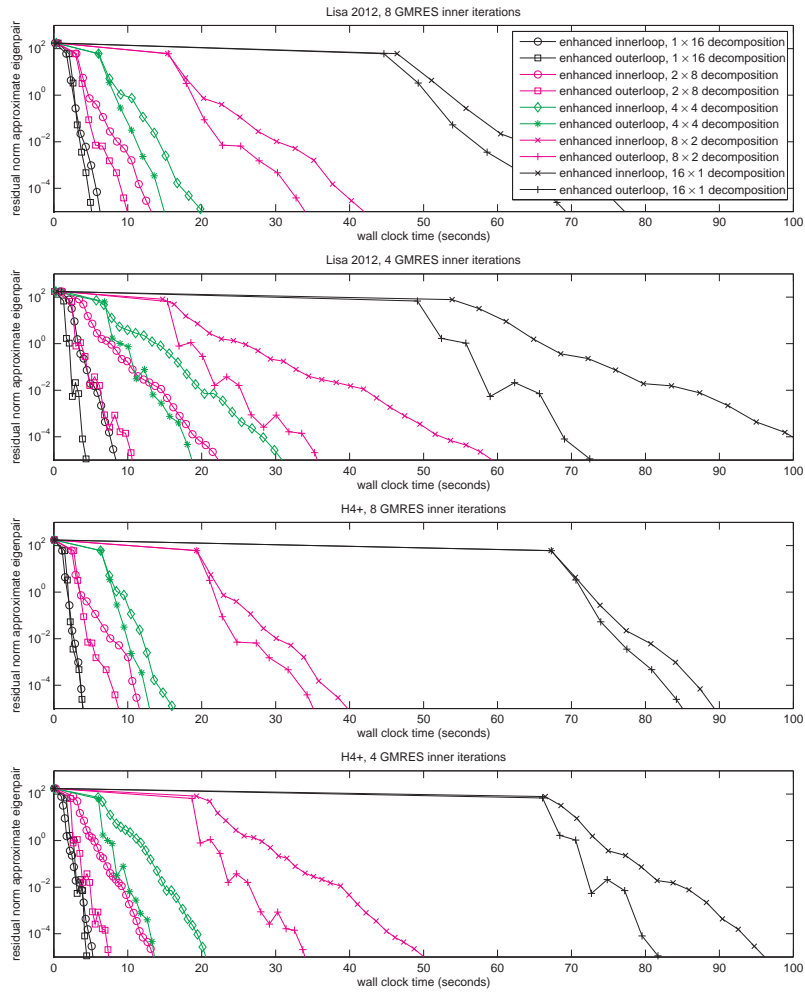
Fig. 3 Residual norm of the approximate eigenpair as a function of the wall clock time for the different decompositions. Shown are both enhanced innerloop and enhanced outerloop for the Lisa 2012 and H4+ linux-cluster and a fixed number of 8 and 4 inner iterations with GMRES.

$\times$ 2, and 16 $\times$ 1 decomposition, respectively (resulting in subgrids of size 1024 $\times$ 64, 512 $\times$ 128, 256 $\times$ 256, 128 $\times$ 512, and 64 $\times$ 1024, respectively). So the number of subdomains is 16 with 65536 unknowns per subdomain in all configurations, but the subdomains differ in shape. Fig. 2 shows the residual norm of the approximate eigenpair as a function of the Jacobi-Davidson outer iteration for the different decompositions. Shown are both enhanced innerloop and enhanced outerloop for a fixed number of 8 (top) and 4 (bottom) inner iterations with GMRES. As expected, the convergence histories for configurations which are mirrored (for instance 2 $\times$ 8 and 8 $\times$ 2) coincide. Decomposition in only one direction needs the least number

of outer iterations for convergence. For the tuning of the coupling between the sub-domains we only took into consideration the one dimensional character of the error modes. For decompositions in two directions error modes will have a two dimensional character and are therefore harder to damp. Fig. 3 shows the residual norm of the approximate eigenpair as a function of wall clock time for the different decompositions. Shown are both enhanced innerloop and enhanced outerloop for the Lisa 2012 and H4+ linux-cluster and a fixed number of 8 and 4 inner iterations with GMRES. By comparing the mirrored configurations it can be observed that the grid ordening may significantly lower the performance. This is mainly in the construction of the preconditioner with LAPACK (initial horizontal lines in the figure). Although processors of the H4+ linux-cluster are faster, use of the MKL implementation of LAPACK resulted in a faster construction of the preconditioner at the Lisa 2012 linux-cluster. After the construction of the preconditioner, the process at the H4+ linux-cluster goes faster than the Lisa 2012 linux-cluster. At the H4+ linux-cluster communication is between 16 nodes over a relatively slow network, at the Lisa 2012 linux-cluster communication is fast inside a 16 core node with shared memory. So, we may conclude that the process is dominated by computational work. This confirms the remarks at the end of § 2 about the minimal communication overhead.

For the massively parallel behavior, we first extend Fig. 6 from [4] with results from (weak) scaling experiments on more recent hardware (IBM POWER6 system Huygens, Curie, and H4+). In Fig. 4 it can be observed that the trend holds, but now for lower wall clock times as processor speed has increased further for the more recent hardware.

To further investigate the weak scaling we start with a decomposition in 16 sub-domains (on 1 node with 16 cores) on the Curie linux-cluster and increase everytime the number of subdomains in both directions with a factor 2. From 16, 64, 256, 1024, 4096 to 16384 subdomains (cores), resulting in up to more than 10^9 unknowns. For an efficient overall method, we will now use (see [1, §4])

$$\|\mathbf{r}^{(i)}\|_2 < 2^{-j}\|\mathbf{r}^{(0)}\|_2 \qquad (10)$$

as a stopping criterion for the inner iterations (GMRES) at the jth Jacobi-Davidson outer iteration. Here $\mathbf{r}^{(0)}$ is the residual at the start of the inner iterations and $\mathbf{r}^{(i)}$ the residual at the ith inner iteration. Fig. 5 shows the results for Jacobi-Davidson with enhanced outerloop. Note that in this figure we choose the scaling of the x-axis to be quadratic to have a better impression. The figure indicates that for a large number of subdomains the wall clock doubles when the number of subdomains increases in both directions with a factor 2. This can be explained from the local behavior of the error modes due to domain decomposition: mainly one dimensional near the interface. The additional work to damp these error modes effectively depends on this local behavior.
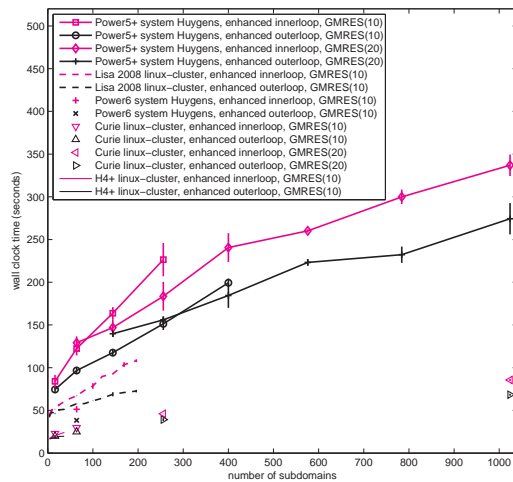
Fig. 4 Massively parallel behavior on different hardware.
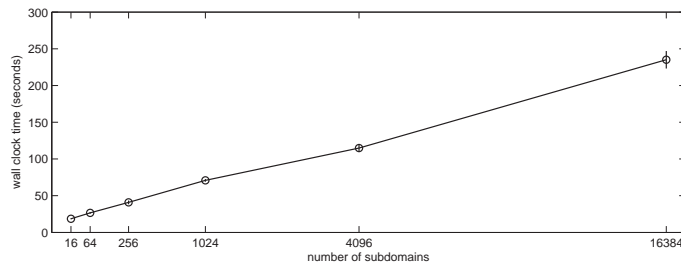

Fig. 5 Massively parallel behavior on the Curie linux-cluster (quadratic scaling of the x-axis).

References

1. Fokkema, D.R., Sleijpen, G.L.G., van der Vorst, H.A.: Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. SIAM J. Sci. Comput. **20**, 94–125 (1998)
2. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**, 699–731 (2006)
3. Genseberger, M.: Domain decomposition in the Jacobi-Davidson method for eigenproblems. Ph.D. thesis, Utrecht University (2001)
4. Genseberger, M.: Improving the parallel performance of a domain decomposition preconditioning technique in the Jacobi-Davidson method for large scale eigenvalue problems. Applied Numerical Mathematics **60**, 1083–1099 (2010)
5. Genseberger, M., Sleijpen, G.L.G., van der Vorst, H.A.: An optimized Schwarz method in the Jacobi-Davidson method for eigenvalue problems. Domain Decomposition Methods in Science and Engineering, pp. 289–296. UNAM (2003)
6. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM J. Sci. Stat. Comp. **7**, 856–869 (1986)
7. Sleijpen, G.L.G., van der Vorst, H.A.: A Jacobi-Davidson iteration method for linear eigenvalue problems. SIAM J. Matrix Anal. Appl. **17**, 401–425 (1996)
8. Tan, K.H.: Local coupling in domain decomposition. Ph.D. thesis, Utrecht University (1995)
9. Tan, K.H., Borsboom, M.J.A.: On generalized Schwarz coupling applied to advection- dominated problems. Domain Decomposition Methods in Scientific and Engineering Computing, pp. 125–130. Amer. Math. Soc. (1994)