Schur complements and direct solvers applied on a simple domain decomposition model: trend and behavior of a monolevel strategy

Boufflet Jean-paul[1], Lefrançois Emmanuel [2], Vayssade Michel [2]
Université de Technologie de Compiègne, CNRS
[1] UMR 6599 Heudiasyc & [2] UMR 6066 Roberval
F-60205 Compiègne cedex - France
{jean-paul.boufflet, emmanuel.lefrancois, michel.vayssade}@utc.fr

We investigate trend and behavior of parts of numerical treatment chains used in numerical mechanics [1, 2]. The framework of the study concerns non overlapping domain decompositions and the use of Schur complement method and direct solvers [3]. In this context, the feasability and the computing performance depend on the choices made on:

1. the decomposition of the initial domain (number of subdomains, number of nodes of a subdomain, size of interface problems);
2. the gathering of the basic computing tasks into parallel macro tasks;
3. the generation of the tasks graph involved by the decomposition;
4. the scheduling of this tasks graph on a parallel computer architecture.

Unfortunately the overall behavior sustains significative variations due to the initial domain. It is difficult to weight the pros and cons between implementations of points 2, 3, 4 and to exhibit trend and behavior. A simple decomposition model for a standard mechanical problem has been proposed in [4]. This is a plate structure meshed with quadrilaterals as shown in figure 1. Our objective is to control the size of the initial domain and its decomposition.

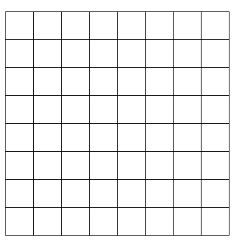

Figure 1: The initial domain: a plate structure of $(2^h+1)\times(2^h+1)$ nodes ($h=3$).

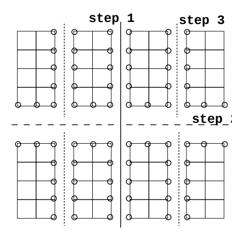

Figure 2: The initial domain split into eight subdomains by applying successive vertical and horizontal cutting lines ($d=3$).

The parameter h fixes the size of the problem and the decomposition step is d. At each step, each subdomain is split into two. That corresponds to apply vertical or horizontal cutting lines as shown in figure 2. For a given h, we can evaluate: the number of subdomains $N_s(d)$, the number of nodes of a subdomain $n_{SD}(d)$, an upperbound for the size of the biggest interface problem $n_{big\_intf}(d)$, the number of nodes of the unique interface problem $n_{unique\_interf}(d)$, and an upperbound for the semibandwidth of a subdomain $b_{max}(d)$.

We study here the implementation of figure 3. Four types of macro tasks are defined gathering basic computing tasks. We assume a unique interface problem to be built. Under these assumptions on the gathering of the computations into parallel macro tasks and on the generation of the tasks graph and using the defined quantities, the volumes of data and the volumes of computations can be estimated under the dense or band assumptions. We can build the estimate of macro tasks of type A, B, C and D, respectively named $\mathcal{E}_A(d)$, $\mathcal{E}_B(d)$, $\mathcal{E}_C(d)$ and $\mathcal{E}_D(d)$. Figure 4 shows the trend of $\mathcal{E}_{global}(d) = \mathcal{E}_A(d) + \mathcal{E}_B(d) + \mathcal{E}_C(d) + \mathcal{E}_D(d)$, the estimation of computations along a path from a type A task to the type D task as $N_s(d)$ the number of subdomains. Estimations for the volumes of data can also be built.

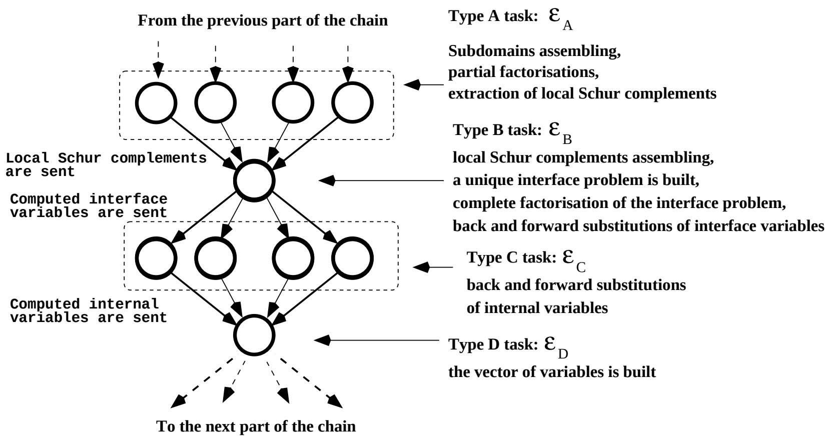

Figure 3: The computing graph of a monolevel strategy. In this example the decomposition factor is $d = 2$: there are four sudomains. We show the tasks performed on subdomains issued from the gathering of matrix computations and basic tasks of the Schur complement method.

Moreover, scheduling strategies for the tasks graph can be studied varying the decomposition factor d and the size factor h. As an example, figure 5 shows the scheme where two type A tasks and two type C tasks are treated by each processor. Models can be built to observe the evolution of volumes of data and computations according to the scheduling strategy.

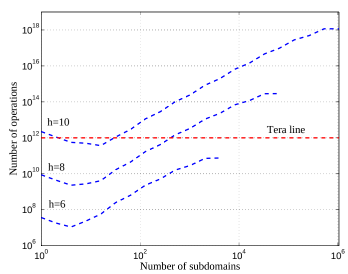

Figure 4: Evolution of $\mathcal{E}_{global}(d)$ as $N_s(d)$ the number of subdomains for the band case. The parameter h varies from 6 to 10 stepped by 2.

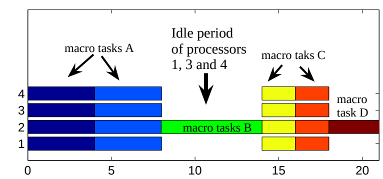

Figure 5: The scheme of the scheduling of macro tasks issued from the decomposition as for figure 2 for $d = 3$ using 4 processors.

This paper presents the trend and the behavior of the described monolevel strategy on the regular problem we proposed, exploring the impacts on estimated volumes of data and computations. The real problems are obviously more complicated. The models can give some insights for real applications.

References

[1] O.C. Zienkiewicz, Robert L. Taylor, The Finite Element Method 1,2, 3 (Butterworth-Heinemann, 2000).

[2] G. Dhatt, G. Touzot et E. Lefrançois. – Méthodes des éléments finis – Hermès, Lavoisier, 2005.

[3] J.P. Boufflet and E. Lefrançois, Finite Element Method direct solver from sequential to parallel computation: an initiation using a domain decomposition approach http://www.hds.utc.fr/~boufflet/siam/su_siam.pdf, 2006.

[4] J.P. Boufflet and E. Lefrançois, A regular decomposition model for studying direct parallel solvers issues Université de Technologie de Compiègne, Department of Computer Science, http://www.hds.utc.fr/~boufflet/plate_model.pdf, 2005.

[5] C. Denis, J-P. Boufflet and P. Breitkopf, A Load Balancing Method for a Parallel Application Based on a Domain Decomposition, International Parallel and Distributed Processing Symposium (2005)