POLITECNICO DI TORINO
Repository ISTITUZIONALE

Low-cost High Performance Grid Infrastructure for Domain-Decomposition Methods in Computational Electromagnetics

Original
Low-cost High Performance Grid Infrastructure for Domain-Decomposition Methods in Computational Electromagnetics / Terzo, O., Ruiu, P., Mossucca, L., Caragnano, G., Francavilla, M.A., Vipiana, F.. - ELETTRONICO. - (2011), pp. 94-99. (Sixth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing(3PGCIC2011) Barcellona, Spain October 26-28, 2011) [10.1109/3PGCIC.2011.24].

Availability:
This version is available at: 11583/2460925 since: 2021-04-26T11:49:03Z

Publisher:
IEEE

Published
DOI:10.1109/3PGCIC.2011.24

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

12 September 2026

Low-cost High Performance Grid Infrastructure for Domain-Decomposition Methods in Computational Electromagnetics

O. Terzo, P. Ruiu, L. Mossucca
Advanced Research on Computing Architecture and Security (ARCAS)
Istituto Superiore Mario Boella (ISMB)
Via Pier Carlo Boggio 61, Torino 10138, Italy
{terzo,ruiu,mossucca}@ismb.it

M. A. Francavilla, F. Vipiana
Antenna and EMC Lab (LACE)
Istituto Superiore Mario Boella (ISMB)
Via Pier Carlo Boggio 61, Torino 10138, Italy
{francavilla,vipiana}@ismb.it

***Abstract*—Computational Electromagnetics (CEM) is scientific discipline aimed at simulating the interaction of electromagnetics fields with structures in a particular environment. The geometries to analyze are typically very complex and they could be simulated starting from a CAD model, possibly including fine levels of detail. This is reflected in a high computational effort that must be faced by high performances infrastructures. This paper addresses the potential benefit of adopting a grid computing infrastructure to solve CEM problems. In particular here we focus on a Domain Decomposition (DD) technique to reduce the complexity of a MoM analysis of large and complex structures. A low-cost grid infrastructure has been built, composed of heterogeneous recovery machines, both physical and virtual, where CEM numerical experiments have been carried out without the need to modify the source codes of the employed algorithms. Virtualization enables the optimization of resources and simplifies infrastructure management. Some performance tests have been carried out in order to verify the time gain and the overhead introduced by the grid infrastructure.**

Keywords-CEM; grid computing; virtualization; high performance; Domain Decomposition; Method of Moments.

I. INTRODUCTION

The accurate and efficient solution of Maxwell's equation is the problem addressed by the scientific discipline known as Computational Electromagnetics (CEM). Many macroscopic phenomena in a vast number of fields are governed by this set of differential equations: electronic, computer, geophysical, medical and biomedical technologies, virtual EM prototyping, besides the obvious antennas and propagation applications, only to cite some of the possible involved fields. As a consequence, many efforts are dedicated to the development of new and more efficient electromagnetic tools. Besides, the advent of powerful computers is a key factor in the renewed interest in the field of CEM: many practical problems, unsolvable a few years ago, can be nowadays easily addressed with the power of present technologies. This leads to the possibility of addressing larger and more complex geometries.

Boundary Equation Methods (BEM) are well suited for solving time harmonic electromagnetic problems: the problem is discretized and resorted into a linear system of equations, expressed in matrix form. Their appeal is mainly due to the fact that they require discretization of the discontinuity surface only, rather than the whole volume of interest as in Finite Element Methods (FEM) for instance, thereby resulting in smaller system matrices. The drawback is the cost in computing the elements of the matrix, and the denseness of the matrix itself. For large problems these drawbacks are usually counterfeited though, and BEMs usually represent the most efficient (if not the only possible) choice

In many applications one would like to simulate the electromagnetic behavior, in terms of an input impedance and a radiation pattern in antenna problems for example, or in terms of a Radar Cross Section (RCS) for typical scattering applications. A full-wave analysis is often required when a high accuracy is demanded: this is often the case in what is called *virtual prototyping*, where the objective is to have reliable simulations in order to minimize the number of measurements, and as a consequence their cost. The same idea applies to the certification phase: the final objective would be to certify the product from an EM point of view through simulations, strongly reducing the cost of Electromagnetic Compatibility (EMC) measurements. Moreover, one usually wants to analyze a complete structure, including all the details, possibly simulating directly the CAD model: this is chased in order to save human "cleaning" time for removing useless details (which would require some EM expertise, which the final user may not be supposed to possess), while maintaining at the same time the original geometric complexity to take into account all the geometrical details of the model. Unfortunately this is reflected in a higher computational effort, due to the increased number of unknowns, and in the worsening of the spectral properties of the linear system when a complex geometry is analyzed. Furthermore, the linear discretization of the surface is proportional to the wavelength of the

electromagnetic field at the desired frequency: as a consequence the number of unknowns scales as the square of the frequency.

In this work we present a Domain Decomposition (DD) technique, which allows the subdivision of the original complex problem into a set of smaller subproblems. Due to the organization in blocks of the problem, a Grid Computing (GC) approach will be used in order to process the subproblems in a parallel fashion. Preliminary results will be shown to prove the effectiveness of the combination of the DD approach with a GC technique.

II. FORMULATION

A complete discussion about the EM modeling and the mathematical aspects of the formulation goes beyond the scope of this paper; only a short summary of the problem will be presented in the following. In order to simplify the mathematical model, in the following the analysis of the electromagnetic behavior of Perfectly Electric Conducting (PEC) objects in a free space environment will be briefly introduced. Nevertheless, the authors would like to point out that the described approach is not limited to PEC objects in free space, in fact it is applicable to different formulations as well (dielectric objects or layered media problems for instance): in other terms it is a *kernel free* method.

The Electric Field Integral Equation (EFIE) is a very versatile approach to the full-wave analysis of complex electromagnetic problems: for PEC objects the EFIE can be written by enforcing the boundary condition on the surface of the object, i.e. the tangential component of the electric field vanishes on the surface:

$$\hat{n} \times \underline{E}(\underline{r}) = 0 \qquad \text{for } \underline{r} \in \Sigma \tag{1}$$

where $\hat{n}$ is the versor normal to the PEC surface Σ, $\underline{E}$ is the electric field, and $\underline{r}$ represents the position vector. After invoking Love's surface equivalence theorem [1], the PEC can be removed and substituted with a set of equivalent electric surface current densities $\underline{J}$, defined on the surface Σ, and radiating, together with the external sources, the same fields in the region exterior to Σ. The EFIE can then be written in terms of the introduced surface current densities $\underline{J}$ as:

$$j\omega\mu\int_{\Sigma}\frac{e^{-jk|\underline{r}-\underline{r}'|}}{4\pi|\underline{r}-\underline{r}'|}\underline{J}(\underline{r}')d\underline{r}' - \frac{j}{\omega\varepsilon}\nabla_s\int_{\Sigma}\frac{e^{-jk|\underline{r}-\underline{r}'|}}{4\pi|\underline{r}-\underline{r}'|}\nabla_s\cdot\underline{J}(\underline{r}')d\underline{r}' = -\underline{E}^{inc}_{\tan}(\underline{r}) \tag{2}$$

where ω is the angular (or radian) frequency, ε and μ are, respectively, the electric permittivity and the magnetic permeability of free space, $k = \omega\sqrt{\varepsilon\mu}$ is the wavenumber, and $\nabla_s = \nabla - \hat{n}\frac{\partial}{\partial n}$ the surface gradient operator.

To solve eq. (2) the Method of Moments (MoM) is applied, which discretizes the problem in a two-step procedure. Firstly, the unknown surface current density is expanded as a linear combination of N known basis functions f_n, with unknown expansion coefficients J_n:

$$\underline{J}(\underline{r}) = \sum_{n=1}^{N} J_n \underline{f}_n(\underline{r}) \tag{3}$$

Then, after substituting eq. (3) into eq. (2), the discretized EFIE is tested onto a set of weighting functions w_m, to obtain the linear system compactly written as:

$$[Z][J] = [V] \tag{4}$$

where *[J]* is a column vector collecting the unknown expansion coefficients J_n, and

$$V_m = -\left\langle \underline{w}_m(\underline{r}), \underline{E}^{inc}_{\tan}(\underline{r}) \right\rangle \tag{5}$$

$$Z_{mn} = \left\langle \underline{w}_m(\underline{r}),\ j\omega\mu\int_{\Sigma}\frac{e^{-jk|\underline{r}-\underline{r}'|}}{4\pi|\underline{r}-\underline{r}'|}\underline{f}_n(\underline{r}')d\underline{r}' - \frac{j}{\omega\varepsilon}\nabla_s\int_{\Sigma}\frac{e^{-jk|\underline{r}-\underline{r}'|}}{4\pi|\underline{r}-\underline{r}'|}\nabla_s\cdot\underline{f}_n(\underline{r}')d\underline{r}' \right\rangle \tag{6}$$

In the following the Galerkin testing will be assumed (the sets of testing and basis functions coincide, i.e. $w_n = f_n$), and the Rao-Wilton-Glisson functions, introduced in [2] and defined on triangular facets, will be used.

Due to the dense impedance matrix resulting from the MoM discretization of the EFIE, it is well known that the storage, the impedance matrix fill-in time, and the matrix-vector operations are of $O(N^2)$ complexity, where N is the number of unknowns. As a consequence, standard techniques are severely limited by the matrix size and condition number involved in the problems of interest. Two basic strategies are usually adopted in order to overcome this problem: the first is based on the so called fast methods, aiming at reducing the cost and the memory occupation at each step of an iterative algorithm; the second class of approaches is based on a compression of the number of unknowns of the system.

Here we focus on a Domain Decomposition (DD) technique to reduce the complexity of a MoM analysis of large and complex structures. The DD method employed here is based on [3], and basically consists in dividing the overall structure to be analyzed into blocks, recognizing that the degrees of freedom of the field coupling between the solutions on these portions are limited, and building entire domain basis functions on each block. In particular, we want to stress that this approach is well-suited for applications requiring the solution of the system (4) with multiple Right Hand Sides (RHS), i.e. more than one forcing vector *[V]* (different vectors *[V]* represent different excitations, for instance different plane incident fields). A typical application is, for instance, the computation of the RCS, where the problem has to be solved for many directions of the incident field. For this class of problems, one would like to invert the system matrix *[Z]*, e.g. using the LU decomposition, and use the inverse to compute the solution for many RHS. The cost of the solution is virtually independent on the number of RHS; unfortunately, the complexity of a matrix inversion scales as $O(N^3)$. On the other hand, fast solvers do not allow a system inversion: the system has to be solved within an iterative scheme, and the cost of the solution grows linearly with the number of RHS.

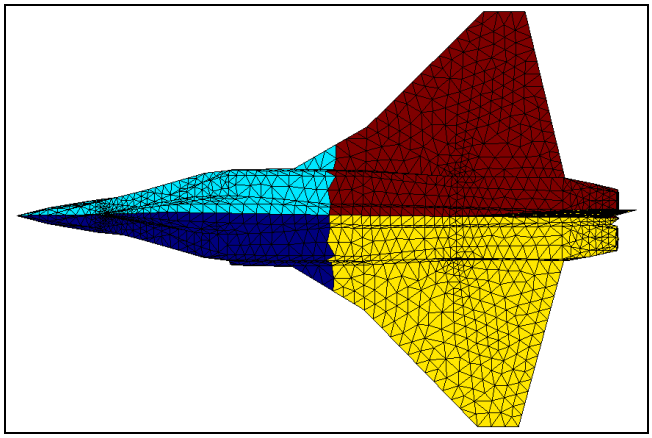

Figure 1. A fighter discretized with 4774 triangles is subdivided into 4 blocks, shown with different colours.

The DD approach however performs a compression of the original system: the number of unknowns in the new basis is strongly reduced, since only a smaller number of entire domain basis functions is required in order to represent the degrees of freedom of each block: with this in mind, a direct solution of the compressed system becomes more appealing.

The macro basis functions generation scheme starts with the separation of the overall geometry into blocks, exploiting the multi-level cell grouping algorithm described in [4] and [5] for the generation of the Multi-Resolution basis functions. Then, on each block taken in isolation, the entire domain functions are chosen in the set of responses to the required incident field, and to other sources placed at the borders of the block and around it, to make up the space of all (rigorous) solutions restricted to that block. Once the set of the solutions to all these sources is computed, the minimum number of necessary responses has to be determined. This is done through a combination of a Singular Value Decomposition (SVD) and a Gram-Schmidt (GS) procedure, applied to the matrix that collects the responses from all the sources: the functions are then selected with a proper thresholding on the associated singular value.

III. HYBRIDIZATION WITH A GRID COMPUTING TECHNIQUE

Due to the decomposition of the original problems into a larger number of subproblems, the scheme is well suited to a parallelization approach, since the subproblems (which will be referred to as blocks in the following) are disjoint, and the elaboration of the blocks is intrinsically a parallelizable operation. Since the generation of the entire domain basis functions on each block is independent from the other blocks, a high scalability is expected. As an example, Figure 1. shows a fighter subdivided into 4 different blocks, identified by different colors. The four blocks can be processed in parallel by four different processes, in order to generate the basis functions describing each block.

Parallelization can be achieved using two different techniques: parallel programming or grid computing. Parallel programming is the technique by which have been obtained the best results in this field (500 billion of unknowns) [7]. There are important barriers to the broader adoption of these methodologies though: first, the algorithms must be modified using parallel programming API like MPI and OpenMP, in order to run jobs on selected cores.

The second aspect to consider is the hardware: to obtain results of interest supercomputers with thousands of cores and thousands of GB of RAM are required. Typically these machines are made by institutions to meet specific experiments and do not always allow public access. Grid computing is a rather more easily applicable model for those who do not fulfill the requirements listed above. For its adoption the only requirement is to have at disposal computers to use within the grid. The machines may be heterogeneous, both in terms of the types (workstations, servers) and in terms of hardware resources of each machine (RAM, CPU, HD); besides they can also be recovery machines. With the Virtualization technology the number of processes running on each machine (for multicore ones) can be optimized by instantiating multiple virtual nodes on the same physical machine. The real advantage of this model, however, is that researchers do not have to worry about rewriting code to make their application parallelizable. One of the drawbacks in the adoption of this model is the addition of overhead introduced by the grid (i.e. inputs time transfer) and by the hypervisor that manages the virtual machines. It has been shown, however, that the loss of performance due to the hypervisor is not particularly high, usually not exceeding 5% [8].

For the above reasons, in our case it was decided to build a low-cost grid infrastructure, composed of heterogeneous recovery machines, both physical and virtual (Figure 2.), on which to run scientific applications without the need to modify the source codes of the used algorithms. The ultimate goal is the reduction of execution times of CEM applications.

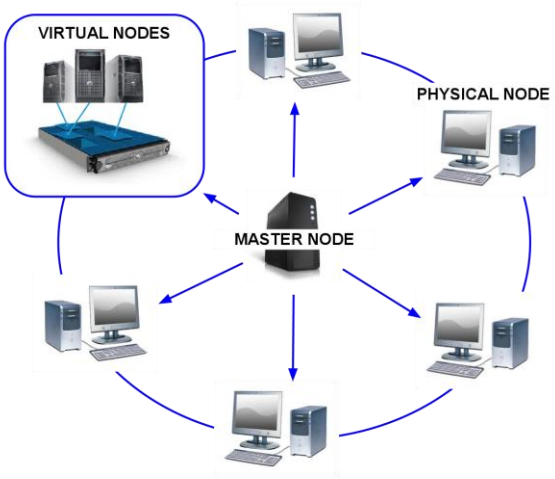

Figure 2. Grid Architecture with virtualized nodes.

TABLE I. GRID NODE SPECIFICATIONS

Node	Node type	CPU model	Virtual CPU	RAM [GB]	OS installed	Storage [GB]
master	physical	Intel Core 2 Duo E7300 @ 2.66GHz	-	4	Ubuntu 10.04 x86_64	140
wn1	physical	Intel Pentium 4 @ 3.20GHz	-	3.5	Ubuntu 10.04 x86_64	39
wn2	physical	Intel Core 2 6420 @ 2.13GHz	-	2	Ubuntu 10.04 x86_64	140
wn3	virtualize	Intel Xeon E5440 @ 2.83GHz	2	4	Ubuntu 10.04 x86_64	22
wn4	virtualized	Intel Xeon E5440 @ 2.83GHz	2	4	Ubuntu 10.04 x86_64	22
wnextra	virtualized	Intel Xeon X3470 @ 2.93GHz	2	4	Ubuntu 10.04 x86_64	10

IV. THE GRID ARCHITECTURE

A grid infrastructure is composed by physical machines, called nodes, on which softwares that allow its use in the grid are installed: a middleware and tools for the local execution of jobs. The middleware used in this project to build the grid infrastructure is the Globus Toolkit 5.0.3 that provides grid services for managing security, for the transfer of files and for the remote execution of jobs.

Thanks to the Virtualization it has been possible to create multiple virtual nodes on the same physical machine, optimizing the hardware resources of nodes. In this way it was possible to increase the available nodes of the grid, i.e. the parallelization of the system, improving the overall performance.

The virtualized systems also help to improve infrastructure management, allowing to use a template of a virtual node (previously configured with all the necessary software) to create virtual nodes in a short time, speeding up the integration of new nodes on the grid and therefore improving the scalability of the infrastructure. For the same reasons, the availability of nodes is improved since in case of damage of a virtual node the system will be able to quickly restore it, reducing the downtime due to his recovery. As hypervisor the open source KVM has been used, which allows creating *fully-virtualized v*irtual machines (VM), running an unmodified kernel. The basic requirement for the installation of this hypervisor is that the processor of the machine has virtualization support (Intel VT or AMD-V). The kernel component of KVM is included in mainline Linux. The realized infrastructure consists of two types of nodes: a master node and multiple worker nodes. The master node that manages the security (it is the Certification Authority) is responsible for the scheduling of jobs and resources, and performs the splitting of the input data. In this first phase of study we decided to equally distribute the inputs on all nodes in order to have comparable workloads in terms of execution times. On the other hand the worker node is in charge of the execution of a part of the whole process. On each worker node a Local Scheduler has been installed, to verify the correct receipt of inputs in a specific directory, and to run the process execution. If the execution finished successfully, the output is sent to the master node, that will reassemble it with the outputs received from other nodes

V. PERFORMANCE TESTS AND RESULTS

The grid used for performance testing consists of machines not purchased specifically for this purpose, rather they were already available to researchers of ISMB. For this reason, its composition is heterogeneous in terms of machines types (workstations, servers) and in terms of hardware resources of each machine (RAM, CPU, HD).

Since the code was not optimized for parallel execution on multiple cores, it was decided to virtualize some nodes in order to run multiple processes on the same physical machine.

The requirements that the machines must have to apply this technique are two: a number of processors greater than two, and a processor supporting virtualization (Intel VT or AMD-V). Within the pool of available machines, two machines met these criteria: on these machines three VMs have been created. To run the tests six nodes with the following configuration have been used: a master node, two physical worker nodes (wn1, wn2) and three virtualized worker nodes (wn3, wn4, wnextra). The details of the machines used for the experiment are shown in TABLE I.
The test conducted was useful to verify the gain in terms of time, comparing the execution of a CEM application on the grid infrastructure to the sequential execution on a single computer. The experiment allowed verifying the practical usefulness of the adoption of distributed infrastructures in this kind of applications.

TABLE II. NODES EXECUTION TIME

Node	Executed blocks	Number of executed blocks	Time [h:min:s]
master	-	0	0:02:55
wn1	5 10 15 20 25 30 35 40 45 50 55 60 65	13	1:42:19
wn2	1 6 11 16 21 26 31 36 41 46 51 56 61 66	14	1:22:33
wn3	2 7 12 17 22 27 32 37 42 47 52 57 62 67	14	1:24:50
wn4	3 8 13 18 23 28 33 38 43 48 53 58 63	13	1:21:38
wnextra	4 9 14 19 24 29 34 39 44 49 54 59 64	13	0:56:22
wn2	All	67	6:56:51

As a test-case a jet fighter aircraft (shown in Figure 1.) has been discretized with a linear mesh density around 5cm. The input file has a size of 17MB and consists of about 156K unknowns. The aircraft is illuminated with a plane wave at the frequency of 600 MHz The geometry has been subdivided into 67 blocks, each one described in a mesh file of about 7.5 MB.

The first test performed was the execution of the entire process on a single physical machine. The machine chosen for this test was the worker node 2 (wn2: despite the smaller amount of RAM, the processing time of this machine is close to the average processing time of the other nodes. The machine split the main input into smaller chunks and ran the sequential execution of individual blocks; the total processing time for 67 blocks on wn2 was equal to 6h 56min 51s.

In the second test the splitting was delegated to the master node (master), which has also been responsible for the distribution of the single blocks to different nodes. The execution time of the build process of the blocks and the file transfer is 2min 55s, and therefore it is negligible when compared to the total execution time. The files have been transferred equitably by the scheduler in order to balance the execution on each node. The execution times of different nodes are very similar, the only exception being the virtual machine wnextra which showed better performances, due to the higher performance processor. The total execution time of the grid is equal to the maximum execution time of individual nodes, i.e. 1h 42min 19s of wn1.

TABLE II. summarizes the execution times on different nodes, both on the grid and on sequential execution. The total time reduction is about 75%, corresponding to a speed-up factor of 4x.

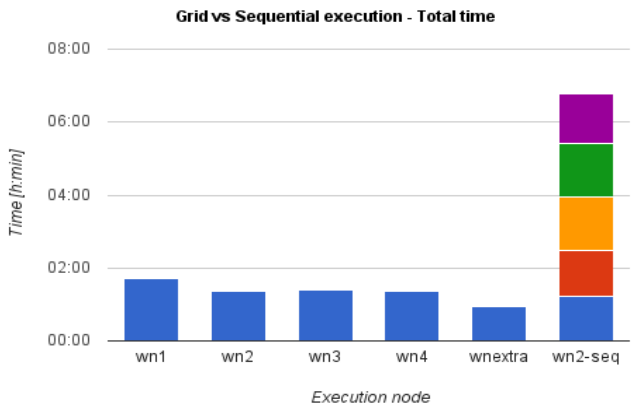

Figure 3. Total execution time comparison between grid enviroment and sequential execution.

Figure 3. shows the comparison between the processing times of individual nodes on the grid and the sequential execution: through this comparison it can be appreciate the performance gain of the grid.

Finally, in Figure 4. the comparison between the execution of individual blocks on the grid and in sequential mode is represented. The graph is useful for assessing the overhead introduced by the grid and by the virtualization of the machines. From the comparison between the execution of a group of blocks on a given node within the grid infrastructure, and the sequential execution on the same machine (i.e. the second pair of columns) it can be deduced that the grid introduces an overhead (equal to 2min 38s in this case), primarily due to file transfers, which is negligible though when compared to the total execution time.

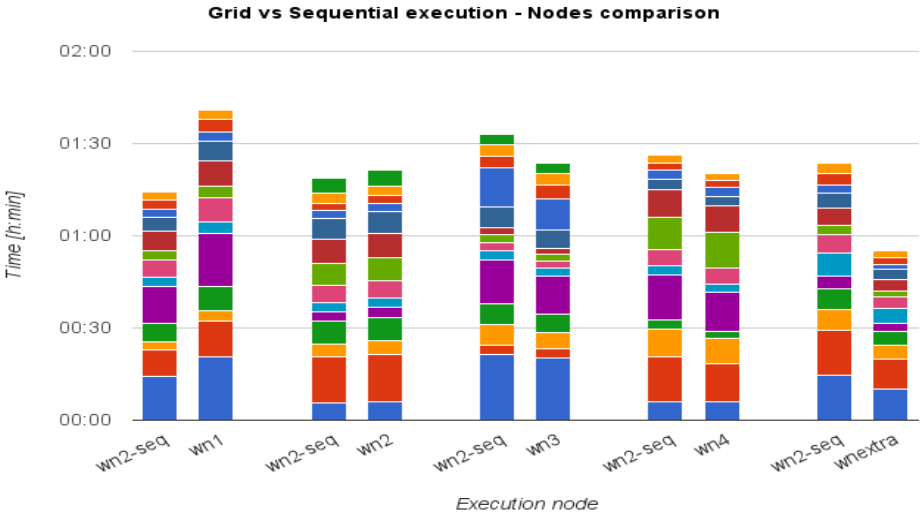

Figure 4. Block execution time of each node compared to the same block executed in sequential mode (wn2-seq).

VI. CONCLUSIONS & FUTURE WORKS

The high computational power needed to solve CEM problems requires the use of high performance computing infrastructures in order to reduce the simulation time. However, it is not always possible to use supercomputers and parallelize the code of the algorithms used for the calculation, and a good solution can be represented by grid infrastructures. For this project a grid of 6 nodes (both physical and virtual) has been realized, consisting of computers already available to ISMB Researchers.

The results showed that the adoption of this infrastructure has reduced the execution time of 75% with respect to the sequential execution on a single machine. It was also noted that the overhead introduced by the grid is negligible when compared to the total execution time.

The Virtualization has allowed optimizing the hardware resources of the machines, letting to run multiple blocks in parallel on the same physical machine, and did not introduce a significant overhead compared to the overall system performance.

Studies are underway to improve the implementation of the scheduler, ensuring that blocks will be distributed more efficiently to the nodes. In particular, a system to take into account the weights of the jobs and the weights of the nodes available will be developed, assigning jobs to specific nodes on the basis of predetermined criteria (size of files, hardware resources, availability nodes, the average time of execution, etc.). The Scheduler will also be able to turn on and off the virtual nodes according to the needs of the system, considering the load level of the grid. Furthermore the virtual nodes will be sized dynamically as needed to perform specific jobs (requirements for RAM, CPU, storage).

This first phase of the project was aimed at reducing the generation of the entire domain basis functions on each sub block of the original geometry. Subject of ongoing research is the solution of the obtained linear system within the grid infrastructure: this topic will be part of future contributes.

A further improvement will be the integration of the system with Public Cloud Platforms (e.g. Amazon EC2) in order to increase the system scalability, asking for virtual nodes usage only when necessary.

REFERENCES

[1] A.E.H. Love, "The integration of the equations of propagation of electric waves", Phil. Trans. Roy. Soc. London, Ser. A, vol. 197, pp. 1-45, 1901.

[2] S. M. Rao, D. R. Wilton, and A. W. Glisson, “Electromagnetic scattering by surfaces of arbitrary shape,” IEEE Trans. on Antennas and Propagation, vol. 30, pp. 409–418, May 1982.

[3] L. Matekovits, V. A. Laza, and G. Vecchi, “Analysis of Large Complex Structures With the Synthetic-Functions Approach”, IEEE Trans on Antennas and Propagation, Vol. 55, No. 9, Sept. 2007, pp. 2509-2521

[4] F.P. Andriulli, F. Vipiana, and G. Vecchi, “Hierarchical bases for non-hierarchic 3D triangular meshes”, IEEE Trans. on Antennas and Propagation, Vol. 56, No. 8, Part 1, Aug. 2008, pp. 2288-2297.

[5] F. Vipiana, F.P. Andriulli, and G. Vecchi, "Two-tier non-simplex grid hierarchic basis for general 3D meshes", Waves in Random and Complex Media, Vol. 19, No. 1, Feb. 2009, pp. 126-146.

[6] A. Esposito, L. Tarricone, L. Catarinucci, B. Di Chiara, and M. Strappini, “New perspectives for computational electromagnetics with grid computing”, Proc. of 2004 URSI Int. Symp. On Electromagnetic Theory (EMT-S), 2004.

[7] J C Mouriño, A Gómez, J M Taboada, L Landesa, J M Bértolo, F Obelleiro and J L Rodríguez “High scalability multipole method. Solving half billion of unknowns”, 2009

[8] Chierici A., Verald R., A quantitative comparison between xen and kvm, 17th International Conference on Computing in High Energy and Nuclear Physics (CHEP09), IOP Publishing Journal of Physics, 2010