New Perspectives for Computational Electromagnetics with Grid Computing

A. ESPOSITO, L. TARRICONE, L. CATARINUCCI, B. DI CHIARA, M. STRAPPINI*
Dept. Innovation Eng., University of Lecce, Via Monteroni, 73100 Lecce – Italy
* Dept. Inf. Eng., University of Perugia, Italy
tel: +39-0832-297226, fax: +39-0832-325004, email: luciano.tarricone@unile.it

Abstract
The demand for computing power in computational electromagnetics (CEM) is continuously increasing. Meanwhile, cooperative engineering is becoming more and more present in daily research and development workflows. Projects are often developed by teams which interact remotely, and need tighter and tighter connectivity. Grid Computing seems a promising way to satisfy both the need of high-performance computing platforms, and the requirements for effective cooperative computing. This is proved on two real applications, the former on antenna array design, the latter on FDTD analysis of human-antenna interaction.

I. Introduction
The Computational Electromagnetics (CEM) community has demonstrated an adequate capability for exploiting large-scale computing to attack its new challenges for over a decade [1,2], and continues to attend to such themes, as proved by recent papers [3,4] (of course, the cited papers and journal issues are far from representing an exhaustive bibliography of the subject).

A recently emerging trend, due to the improvement of distributed information technologies (IT), has been an acceleration of research and development processes towards concurrent and cooperative engineering, accompanied by the so-called “internet-revolution”, and the booming of web applications. The extraordinary perspectives opened by the web have reinforced the momentum towards process integration and cooperative computing. Consequently, joining together supercomputing facilities and the world of web-based tools seems to be the key feature to open new perspectives in industrial and scientific processes, and a recent technology is proposing itself as the most natural way to pursue such a goal: grid computing (GC).

The technology of GC has led to the possibility of using networks of computers as a single, unified computing resource, clustering or coupling a wide variety of resources, including supercomputers, storage systems, data sources, and special classes of devices distributed geographically, and using them as a single unifying resource (computational grid).

In such a scenario, the CEM community is still playing a minor role. Even though several EM problems could draw substantial advantage from GC, the theme of GC has been almost entirely absent until now in technical books and scientific journals dealing with CEM. On the contrary, several practical applications can immediately take advantage from the afore-mentioned technology.

In this paper, we shortly describe GC, and propose some possible applications, demonstrating the relevance of this new technology for CEM.

II. Grid Computing General Concepts
A computational “grid” consists of software and hardware resources, for instance computational resources, storage systems, catalogs, network resources and sensors. A resource may also be a logical entity, such as a distributed file system, computer cluster or distributed computer pool. Grid computing software (the so-called “grid middleware”, GM) pools resources together into a unique virtual system and makes it possible for anyone on the network to access it (or its facilities). As shown in Fig. 1, a grid can span domains of different dimensions, starting from local grids, made up of nodes connected by fast LANs, up to global grids, made up of nodes owned by different organizations and connected by the Internet.

In all these scenarios, grids permit “coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations” [5]. This means that a wide spectrum of possibilities is opened by GC, ranging from the exploitation of idle CPU to solve computationally intensive problems, to the building up of the so-called “meta applications”, applications that derive from the coupling of dispersed multi-disciplinary components.

In the most general case, grid resources are supposed to be geographically distributed and to be owned by different organizations, each with proprietary policies regarding security, resource allocation, platform maintenance and so on. Such an environment depends strongly upon the construction of a robust infrastructure of fundamental services able to smooth out mismatches between different machines, security policies, scheduling policies, operating systems, platforms, filesystems and so on. Besides, resource sharing must be highly controlled, with resource providers and consumers defining clearly what is shared, who is
allowed to share and the conditions under which sharing occurs. Furthermore, access to resources has to be carefully scheduled in order to extract the maximum performance from the available resources, and applications should have the possibility of tailoring their behaviour dynamically in order to cope with resource failure, a highly probable event in such a variegated context.

All these requirements can be summarized by the need to allow transparent access to resources as if they were belonging to a single unified “metacomputer”. There are many grid projects worldwide aiming at achieving this ambitious goal, Globus Toolkit (GT) being one of the most promising: it is rapidly becoming the GM de facto standard [6], and has been chosen in our experimentation. Details will be given in the oral presentation, describing how to set-up a computational grid.

III. Applications and Results

III.1 Computer-Aided Engineering (CAE) of Rectangular Aperture Array Antennas
The CAE of rectangular-aperture arrays is a complex problem. In a previous paper [7] it was demonstrated that it can be fruitfully partitioned into four main sub-problems, namely Analysis of the Feeding Sections (AFS), Analysis of the Mutual Coupling among apertures (AMC), Evaluation of the whole Scattering Matrix of the circuit (ESM), Evaluation of the Radiation Pattern of the array (ERP). Each sub-problem can correspond to a single, independent module. Each module can in principle be executed independently from the others, provided that suitable input data are available.

Let us assume, for instance, that the four modules previously enumerated (AFS, AMC, ESM, ERP) have each been developed by separate and independent working groups, adopting different software and hardware technologies. Let us also suppose that the only constraint for each team is represented by fixed standards for data input and output, so that the four modules can cooperate together via files, or via equivalent data communication (sockets, ftp, etc.). We assume that the groups are interested in sharing their applications so that the global task can be performed (CAE of the whole array) but require that their own module remains a proprietary application, resident on their own platforms, with all the guarantees of security of data and applications. We finally suppose that every team can in principle be interested in “offering” its module for external use at a certain cost.

In conclusion, we have four different modules, distributed geographically through the web, and must guarantee an efficient cooperation among them, with high reliability and security requirements. An environment supporting high-performance computing is also needed, as well as the capability of acting as a broker, regulating even aspects such as CPU or application costs, commercial transactions, etc.

GC, when suitably used, allows to fulfill all the enumerated requirements, with no-costs for software (all open source) and hardware (provided that an internet connection is available). Thanks to this, different research teams can develop, in an absolutely independent fashion, portions of the whole application, with very loose constraints, limited to input and output data, delegating to the grid environment the complex task of providing the hardware and software interface to interconnect such autonomous partitions. It is also worth mentioning that the four partitions of the application (AFS, AMC, ESM, ERP) can in turn be partitioned into subtasks, and developed in a distributed and cooperative fashion, so that several levels of distribution and parallelism can be sustained.

Moreover, being the grid itself a distributed-memory parallel platform, it allows a zero-cost access to supercomputing facilities. Of course, performance depends strongly on the available connectivity, this being not a peculiar limitation of grids, rather than of any distributed systems. Finally, as discussed into details in the oral presentation, the same infrastructure is suitable for a full management of economic models of

![Diagram of grids]

Fig. 1: Examples of grids. A Local Area Network – LAN- can host a local grid, and its local grid can itself be part of a wider grid (for instance at Wide Area Network – WAN- level).
cooperation, governing transactions among users, and monitoring cooperative jobs, up to the creation and hosting of suitable databases with the history of all contacts among users.

III.2 Parallel FDTD with GC

FDTD is one of the most well-known full-wave methods, used for a large variety of CEM applications. It is quite CPU-intensive, and amenable to parallel implementations, requiring simple domain-decomposition policies. A recent paper [3] presented a clear overview on the method and its parallelisation using Message Passing Interface (MPI), as well as an adequate bibliography, and we address the interested reader to it for such aspects. Another recent paper [8] described in detail the specific FDTD implementation we adopt in this paper, as well as a parallel implementation on Single-Instruction Multiple-Data architectures.

We have developed an FDTD code, using MPI, and compatible with GC. This is straightforward, thanks to MPICH-G2, a grid-enabled implementation of the MPI standard. The results attained by using a computational grid are resumed in Tab. I, where the speed-ups achieved for grids of different size are reported. Data refer to the analysis of a half-wave dipole radiating in the vicinity of a homogeneous dielectric sphere (see fig. 2), at 900 MHz. Two domain sizes are considered, so that for a $\lambda/10$ space step two different numbers of cells are attained (a cube with a 256-cell edge, and a cube with 400-cell edge). The adopted computational grid is an interdepartmental grid, with a Giga-Ethernet, in a low-load condition. It can be observed that the performance achieved are quite similar to the one related to costly parallel platform, though grids are cheaper, and guarantee a huge amount of additional advantages.

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Speed-up on a 256x256x256 case</th>
<th>Speed-up on a 400x400x400 case</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.3</td>
<td>1.6</td>
</tr>
<tr>
<td>4</td>
<td>2.7</td>
<td>2.8</td>
</tr>
<tr>
<td>8</td>
<td>4.1</td>
<td>4.6</td>
</tr>
</tbody>
</table>

Tab. I: Speed-ups achieved in an inter-departmental grid.

The GC implementation of an MPI version of FDTD demonstrates that GC allows a no-cost access to a potentially unlimited set of computational resources, retaining all the features peculiar to a parallel distributed system. Any application developed using a programming paradigm based on message-passing standard MPI can be straightforwardly migrated, with no limitations with respect to the programming language used. A research team is not required to arrange its own supercomputing platform, neither to set up dedicated connectivity. The proposed alternative strategy is to invest in GC know-how.

IV. Conclusions

In this paper we have introduced the concepts of GC to researchers involved in CEM. Applications to CAE of aperture array antennas, and to FDTD with parallel computing have been proposed. The results obtained have demonstrated that GC is an effective way to produce low-cost, secure, controlled and flexible cooperative and distributed engineering, with apparent and immediate achievements. It has also been proved that GC is a viable approach to parallel computing.

GC can open a new perspective in the approach to large and complex CEM problems, and is a new challenge that the CEM community should definitely face in order to take full advantage from the continuous and fascinating enhancements of IT.

REFERENCES