Electromagnetic Simulation of MEMS-Controlled Reflectarrays based on SCT in Grid Environment

F. Khalil*[1,2], H. Aubert [1,2], F. Coccetti[1] , R. Plana[1,3] ,
Y. Deunneulin[4], B. Miegemolle[1], T. Monteil[1], H. Legay[5]
(1) LAAS – CNRS Laboratory, 7 Avenue du Colonel Roche, Toulouse Cedex 4, 31077, France, http://www.laas.fr
(2) INPT - ENSEEIHT, 2 Rue Charles Camichel, Toulouse Cedex 7, 31071, France, http://www.enseeiht.fr
(3) UPS, 118 Route de Narbonne, Toulouse Cedex 9, 31062, France, http://www.ups-tlse.fr
(4) ID – IMAG Laboratory, ZIRST 51, Avenue Jean Kuntzmann, 38330, Montbonnot Saint Marin, France, http://www-id.imag.fr
(5) Alcatel Alenia Space, 26 Avenue JF Champollion, 31037 Toulouse, France, http://www.alcatelaleniaspace.fr

Introduction

For the electromagnetic modeling of reflectarrays controlled by Radio-Frequency Micro-Electromechanical Switches (RF MEMS) a new original approach, named Scale Changing Technique (SCT), was developed and tested [1]. Based on the partition of the reflectarray surface in planar sub-domains with various scale levels, this technique derives the phase-shift dynamics from the simple cascade of networks, each network describing the electromagnetic coupling between two scale levels. Figure 1 displays the partition for one MEMS-controlled phase-shifter cell of the reflectarray. Higher-order modes allow the accurate representation of the electromagnetic field local variation while lower-order modes are used for coupling the various scales.

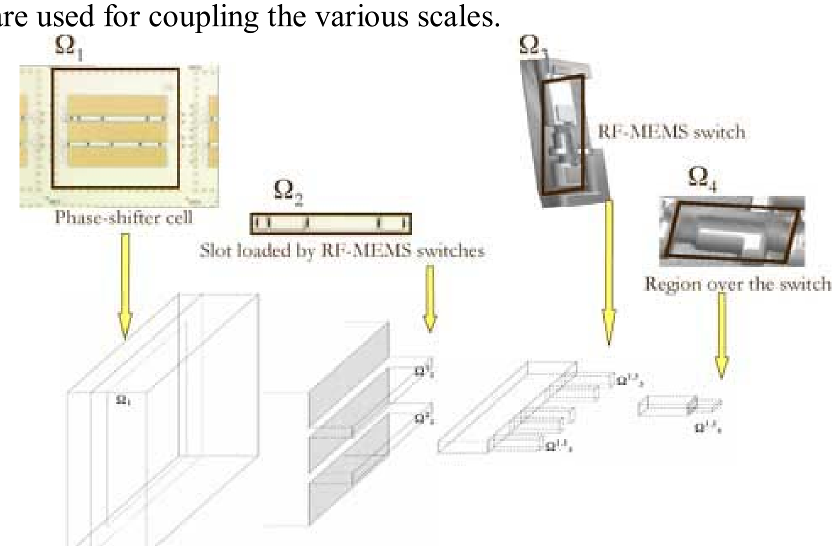

Fig. 1. Multi-scale view of planar phase-shifter used in MEMS-controlled reflectarrays.

The SCT-based simulation tool consists of a MATLAB code developed to run on a single machine in order to predict the phase-shift for all possible MEMS configurations. The 1024 configurations of a phase-shifter cell are obtained by the UP/DOWN positions of the 10 RF MEMS switches inserted in the slit of the 3 metallic patches [2]. This problem represent a very computational demanding task on stand alone machines.

In this work we report how parallel distributed computing in grid environment offers a very suitable tool for this specific problem. The multi-port networks are computed on GRID5000 nodes [3] and it has yielded a 90.5% of simulation time reduction while keeping the same accuracy for the several frequency values we have tested.

Parallelization of the SCT

The code developed for the SCT is one of many serial codes that are limited by the total CPU time that they require to run. Often the individual tasks are actually independent of one another and therefore can be potentially be run simultaneously (in parallel) on different processors. This approach can greatly reduce the actual time required to obtain a scientific result. In our work, the use of the grid allow us to distribute processes on one or many clusters which may belong to different geographically distributed sites. The individual tasks are relatively small in terms of code size and execution time. This fact means that we have fine-grained, or "tightly coupled, parallelism". The data are transferred among processors frequently in amounts of few memory data. The smaller the granularity, the greater the potential for parallelism and hence speed-up, but the greater the overheads of synchronization and communication. In fact, a distributed-memory system introduces the problem of how to distribute a computational task to multiple processors with distinct memory spaces and gather the results from each processor into one solution [4]. Consequently, program units must have the ability to communicate with another in order to cooperatively complete a task. Any interaction among processes is achieved through an explicit exchange of messages. Each node is connected to the central node with a fast interconnection network. The nodes share data by exchanging messages through this interconnection node (scatter/gather data from all members to all members of a group).

Another important consideration which can be controlled is load balancing. This means that if tasks take different amounts of time then execution time will be governed by the slowest process. In general grid's clusters are homogeneous. So static load balancing yields good performance. We decided a priori in order to assign a fixed amount of work to each node. Moreover the access to the computational nodes has been carefully scheduled in order to ensure correctness, extract the maximum performance from them, and avoid race conditions. Due to the grid software tools facilities, we can track the behavior and states of allocated resources at every moment (interactive reservation mode). Nodes reservation is a further grid advantage since no unexpected processes will be running in the background. The scalability, or the ability to dedicate an increasing number of processors for the computing task, is another critical feature to take in consideration.

The theory of parallel computational follows fundamental laws that limits the benefits at a given extend. To study the scalability, the first factor to consider is the speedup. This is defined as the time it takes for a program to execute in serial fashion (with one processor) divided by the time it takes to execute in parallel (with several processors). The law governing the speedup by using parallel processors versus the case of only one serial processor is known as Amdahl's Law. The best way to use the information from calculated speedup and efficiency in order to make sense of Amdahl's Law is the Speedup Curve given in Figure 2 for the present case.

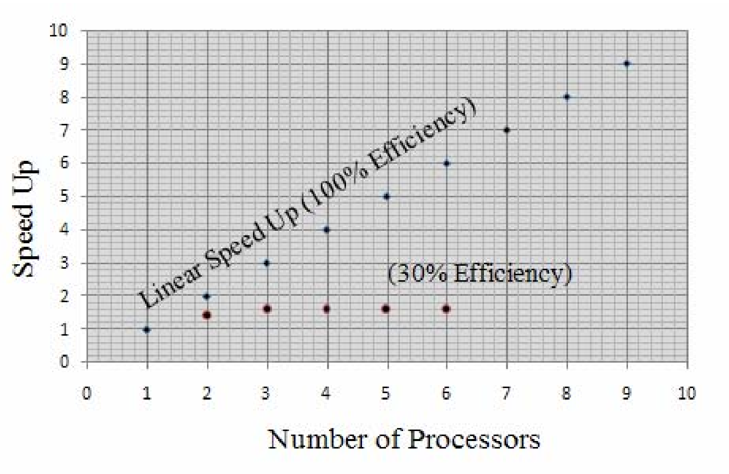

Fig. 2. Scalability analysis

The best speed up we could expect would have yielded a 45 degree curve. In our case, the results settle down to a constant logarithmic value which remains below the ideal limit of the 45 degree line. This law shows that it is indeed the algorithm and not the number of processors which limits the speedup. Also note that as the curve begins to flatten out, the efficiency is drastically reduced. Must be mentioned that Amdahl's Law ignores many of the actual implementation limit as the finite communication time between the machines

Performance Analysis

The performance of the new enhanced code was tested in grid environment with a parametric study of frequency. Figure 3 shows the results of the fully frequency analysis for the 1024 configurations obtained by switching the 10 MEMS component between the two working states (UP/DOWN). The analyzed frequency range is between 11.7GHz and 12.5GHz. Using a single 2.0GHz AMD Opteron

246 computer with 2GB RAM, the full analysis of the different configurations has required 11 minutes.

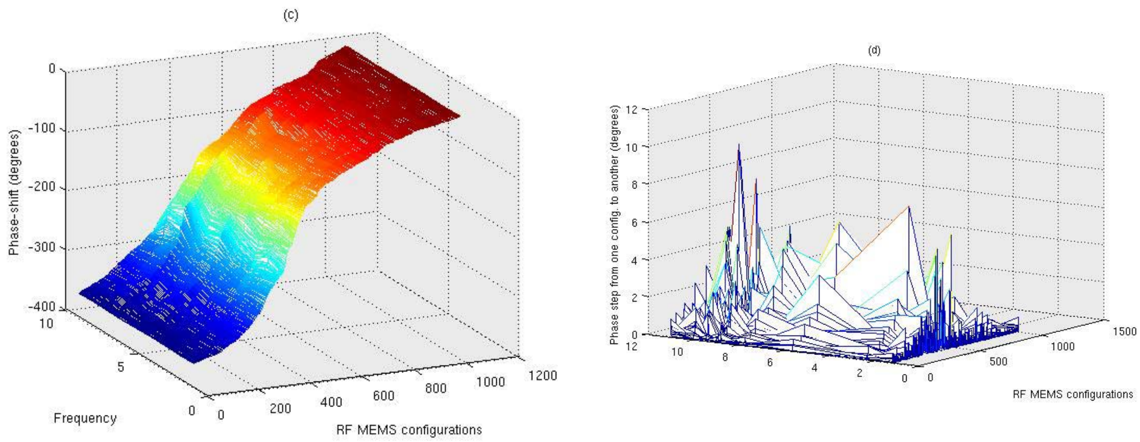

Fig. 3. Phase-shifts and losses of 1024 available UP/DOWN MEMS configurations in the given frequency range

Grid analysis running on a 18 nodes system on Rennes site clusters based on the same exact computer specifications required just 0.6 minutes which means a 90.5% of elapse time reduction. The same method used here can be used for the convergence study where many parameters are varied. This operation is typically to time consuming to be carried out. The time speedup demonstrated in this work is quite promising results, especially in the case of reflectarray containing 20 x 20 cells similar to the one analyzed here. In this case the approach of distribute computing is believed to provide the most advantageous and time efficient solution.

Acknowledgment

The authors wish to acknowledge the National Research Agency (ANR) for support of MEG Project and also the contribution of the PhD student Bernard Miegemolle.

References:

[1] E. Perret, H. Aubert, and H. Legay, "Scale Changing Technique for the Electromagnetic Modeling of MEMS-Controlled Planar Phase-Shifters", IEEE Trans. On Microwave Theory and Tech., Vol. 54, Issue 9, pp. 3594-3601, September 2006.

[2] H. Legay, "MEMS technology for terminals and satellite antennas," ESTEC Final Report, Contract N° 17269/03/NL/AG, September 2005.

[3] The official website of GRID5000. https://www.grid5000.fr

[4] L. Tarricone, and A. Esposito, *Grid Computing for Electromagnetics*, Artech House, 2004.