Fractal Element Antenna Genetic Optimization Using a PC Cluster

Nuno Alves, William Crespo-Miranda, Felipe Fernandes, Fabio Santana

College of Engineering, Boston University, Boston, MA
nca@bu.edu, wcrespo@bu.edu, felipe81@bu.edu, fabiosan@bu.edu

Robert G. Hohlfeld and Nathan Cohen

Boston University and Fractal Antenna Systems, Inc.
bob@fractenna.com, chip@fractenna.com

ABSTRACT

Antenna optimization by genetic techniques is computationally intensive and so the motivation for development of effective computational tools is high. Genetic algorithms and genetic optimization possess a natural computational parallelism in that electromagnetic simulations evaluating the figure of merit of antennas in the population being optimized can be carried out independently in separate processors. We show the design of a prototype PC cluster implemented at Fractal Antenna Systems, Inc. that exploits this natural computational parallelism. Techniques for optimizing the overall performance of a PC cluster for this problem are discussed as well as scaling of performance with the number of processors. Because interprocessor communication loads are very low, performance is expected to show near-linear speedup, even up to ~100 processors, consistent with Amdahl's law.

INTRODUCTION

Mathematically antenna optimization is characterized as a "stiff" optimization problem because antenna performance, by whatever figure of merit is appropriate for the design problem at hand, usually depends sensitively on the geometry of the antenna.

Furthermore, antenna optimization is also usually characterized by having several distinct maxima in the figure of merit function as viewed in the space of antenna geometric parameters. For these reasons, conventional optimization techniques such as gradient-based methods (Press *et al* 1992) are prone to fail unless used with care and with substantial physical insight into the properties of the antenna system. A gradient-based optimization algorithm will often be "captured" by a secondary maximum in the figure of merit function and the optimization process will terminate before the true physical maximum of the figure of merit function is located.

In response to these difficulties, many researchers have begun investigation of genetic algorithms (GA) for antenna optimization (see *e.g.* Haupt 1995) A well-designed genome coding directly for the antenna geometry in space, or a fractal genome as used in Genetic Optimization (GO, Cohen 1997a, 1997b, Hohlfeld, Moschella and Cohen 2002) will sample the parameter space relevant to an antenna design problem densely enough that the true global maximum of the figure of merit function will be found, provided the population of individuals in the GA/GO is large enough. Evolution of the population will allow individuals to be found closer and closer to the true performance maximum.

Offsetting these advantages is the high computational cost of GAs/GOs. Each candidate antenna in the population being studied requires an electromagnetic simulation sufficiently detailed to evaluate the figure of merit for that antenna. The process must be carried on in successive generations because the properties of successive generations of antennas changes as they evolve, and so the number of individual electromagnetic simulations required to solve a realistically complex design problem can run to many thousands.

A PC cluster computer (Ridge *et al.* 1997; Hargrove, Hoffman, and Sterling 2001) has been seen to be an effective means of approaching a wide variety of computational problems. We now discuss the properties of the GA/GO algorithm that make it a natural match for PC cluster computation.

Parallelism for GAs/GOs

This basic description of computational costs suggests a strategy for speeding up operation of a GA/GO. Each electromagnetic simulation in the GA/GO is independent of all other electromagnetic simulations used to evaluate the population of antennas. It is natural to match the problem granularity to the granularity of the parallel computer system by assigning an individual processor to a single electromagnetic evaluation. Parallelism is thus achieved across the population in any given generation of the GA/GO. Since all individuals in a given generation must be assigned figures of merit before individuals are eliminated by Darwinian selection and new individuals are generated by genetic processes, an inherent serial feature of the GA/GO remains because generations must be treated serially. We term this property of parallelizing GA/GOs in this fashion "individual parallel/generation serial". It is this intrinsic parallelism of the GA/GO problem that allows for a significant speedup in GA/GO computations.

Performance of parallel computations usually depends sensitively on the properties of interprocessor communication in the computation (Flynn 1995, Leighton 1992). Processor network topology, total communication load, communication intermittency, and degree of regularity of interprocessor communication are usually the most relevant issues by which interprocessor communication affects overall parallel computational performance. For concreteness, assume that the GA/GO is optimizing wire antennas and that the electromagnetic simulations are being performed by NEC. In such a situation the communication to a processor running NEC can simply be a wire table, typically a few kilobytes or at most a few tens of kilobytes for very large problems. The result of a NEC simulation can be reduced to a figure of merit in the same processor carrying out the NEC calculation; thus the communication back from a processor to a central processor running the genetic components of the GA/GO need only be a single floating point number (plus irreducible communications handshaking). Since NEC runs of this size typically use at least a few minutes of run time, even on high-end commodity PCs, the communication load for a reasonably sized system can be quite modest and easily accommodated by conventional Ethernet interconnections between processors. For almost all this time during an optimization run, the Ethernet will be essentially unloaded.

In the simplest embodiment of such a system, one processor will be reserved for running the genetic components of the GA/GO and distributing wire tables to the other processors in the PC cluster. Typically this processor will be very lightly loaded compared to the processors running NEC.

An Implementation of a PC Cluster for GA/GO Antenna Optimization

We describe here a PC cluster system being constructed at Fractal Antenna Systems, Inc. (FAS) that embodies the principles described above. The initial system contains 5 nodes running NEC and a single node running the genetic components of the FAS GO program. The system is designed to increase performance by the addition of processors with linear speedup through the range of tens of processors (see the discussion of projected system performance below).

Each of the 5 nodes running NEC is a 1.2 GHz Athlon PC with 128 MB of random access memory and a 20 GB hard drive (see Figure 1). All nodes are connected through a 16 port 10/100 Mbps Ethernet switch. Use of a 16 port Ethernet switch allows for a reasonable level of system expansion beyond the initial system described here. The Server/Host Node is a dual-processor Pentium III with a clock speed of 1 GHz with 256 MB of random access memory per processor and a 30 GB hard drive. Generation and evolution of the antenna populations is carried out on the Server/Host node, wire tables are transmitted over the Ethernet connection to the processor nodes, which carry out NEC simulations, evaluate the prescribed figure of merit functions, which are then transmitted back to the host. An uninterruptible power supply and connectivity to the FAS intranet is also included.

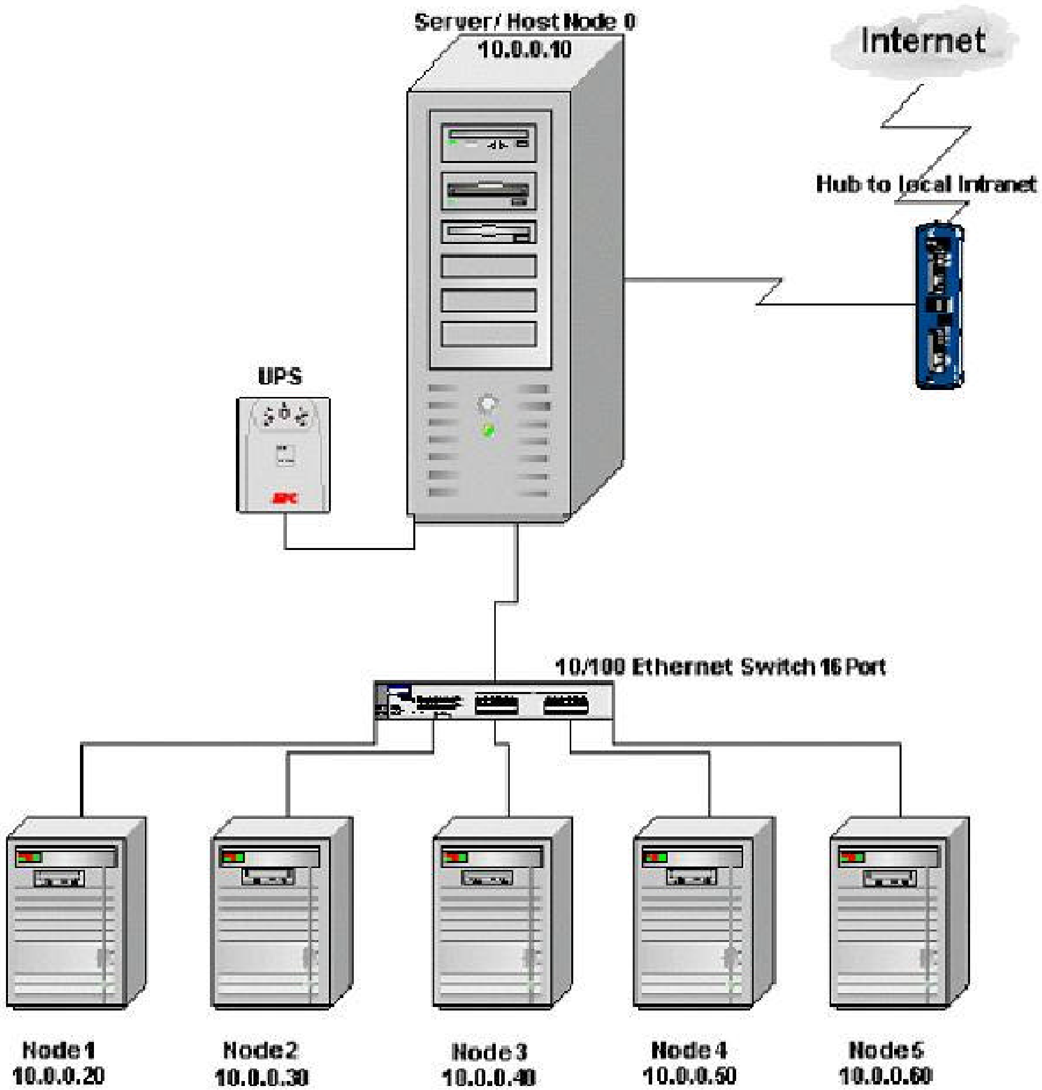

Figure 2: Schematic layout of the PC cluster being implemented at Fractal Antenna Systems, Inc. for GA/GO antenna optimization. Nodes 1 through 5 run NEC and evaluate the figure of merit for antennas in the population being optimized. They are connected through an Ethernet switch to a host computer that executes the genetic portions of the code, distributes wire tables to nodes, and collects figure of merit information for application of natural selection.

All nodes use the Linux operating system. Interprocessor communication and management of NEC execution and figure of merit evaluation are controlled by Linux shell scripts.

Load Balancing and Scaling of Performance

Computational load balancing is crucial to effective operation of any parallel computational system. We note here one crucial issue germane to the GA/GO we are implementing. NEC execution time is $O(N^3)$, where N is the number of entries in the wire table. When the number of processors in the system in the system is less than the number of antennas in the GA/GO population it is necessary to sort the wire tables by size and distribute the largest wire tables in any given generation out to NEC nodes first. If, on the other hand, one were to distribute wire tables in arbitrary order, the situation could arise in which a very large wire table was executing in one NEC node and all other NEC nodes in the system were idle, leading to low overall efficiency. The necessity of sorting wire tables in each generation originates from the individual parallel/generation serial character of this optimization problem.

Lastly, we consider the levels of computational efficiency expected for our system and the computational speedup expected for our system as more processors are added. The discussion of the individual parallel/generation serial character shows that the generation of genomes in a new generation, the generation of wire tables from the genomes, and sorting of those wire tables, make up the intrinsically serial portion of our GA/GO. Timing of our GA/GO algorithm on realistic problems running on a single processor show that the intrinsically serial portion of the algorithm makes up < 1% of the total run time, the remainder due to NEC execution and evaluation of the figure of merit functions for individuals in the population. On the basis of Amdahl's law (Flynn 1995), we can predict the speedup on our system as a function of the number of processors:

$$S_p = \frac{1}{s(1-1/p)+1/p} \ .$$

Here S_p is the speedup obtained running on p processors and s is the fraction of serial code. Since $s \le 0.01$, and communication costs will be negligible, we can expect very nearly linear speedup with p in our system, provided that all NEC nodes complete the last (*i.e.* the smallest) NEC evaluations of a given generation at nearly the same time.

Summary and Conclusions

Computational costs of GA/GO algorithms for antenna optimization motivate the development of parallel systems for their implementation. These costs are primarily driven by the cost of electromagnetic simulations for each individual in the genetic population being optimized. The most natural description of the parallelism of GA/GO algorithms is "individual parallel/generation serial" because all individual electromagnetic simulations may be carried out independently, but all individuals in a given generation must be evaluated and figures of merit assigned before proceeding to the next generation of the GA/GO. This type of algorithmic parallelism is naturally accommodated in a PC cluster. Such a system is expected to be very efficient because the costs of interprocessor communication can be very low and the amount of intrinsically serial execution time is small. Thus, a PC cluster running a GA/GO algorithm provides a very efficient and cost-effective means of carrying out antenna optimization.

References

Cohen, N., "Genetic Antenna Optimization with Fractal Chromosomes", Proceedings of ACES97, pp. 305-308, (1997a).

Cohen, N., "Fractal Coding in Genetic Algorithm (GA) Antenna Optimization", *Proceedings of IEEE-APS/URSI pp. 1013-1017* (1997b).

Flynn, M. J., **Computer Architecture: Pipelined and Parallel Processor Design**, Jones and Bartlett Publishers, Boston and London, (1995).

Hargrove, W. H., Hoffman, F. M., and Sterling, T., "The Do-It-Yourself Supercomputer", *Sci. Am.*, pp. 72-79, (August 2001).

Haupt, R. L., "An introduction to Genetic Algorithms for Electromagnetics", *IEEE Antennas Propagat. Mag.*, **37**(2), pp. 7-15 (1995).

Hohlfeld, R. G., Moschella, D., and Cohen, N., "Prospects for Fractal Coded Genetic Optimization in Electromagnetics", Proceedings of ACES Conference (2002).

Leighton, F. T., **Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes**, Morgan Kaufmann Publishers, Inc., San Mateo, CA (1992).

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., **Numerical Recipes in C: The Art of Scientific Computing (2nd Ed.)**, Cambridge University Press, Cambridge (1992).

Ridge, D., Becker, D., Merkey, P., and Sterling, T., "Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs", Proceedings, IEEE Aerospace (1997).