Computational Parallelization Strategy Applied in Full 3D Ray-Tracing Wireless Channel Modeling

André M. Cavalcante, Marco J. de Sousa, Claudomiro de S. Sales Jr, João Crisóstomo W. A. Costa, Gervásio P. S. Cavalcante and Carlos Renato L. Francês
Federal University of Pará, Applied Electromagnetism Laboratory, Belém, PA, 66.075-900, Brazil
Email: jweyl@ufpa.br, Phone/Fax: +55 91 3183-1634

***Abstract*— This paper presents a computational parallelization strategy applied in propagation models based on 3D ray-tracing techniques. This approach considers that the rays are independent from each other, what allows a uniform division of the tasks by equal and random distribution of rays among the parallel computer nodes. The strategy efficiency is proved by simulation where the results are discussed.**

***Index Terms*— Parallel computing, cluster of PC's, 3D Ray-tracing.**

I. Introduction

The great growth in mobile communications needs fast and accurate prediction of radio wave propagation for system deployment. Such predictions can represent an important role in determining network parameters including coverage, transmitted-data rates, optimal base station locations, and antenna patterns. In this context, ray-tracing based radio propagation prediction models have shown promise, mainly in modern radio wave propagation environments [1]-[6]. Although ray-tracing approaches are very useful in the design, analysis, and deployment of wireless networks, it has been recognized that these models are computationally very expensive and require a considerable amount of processing time to attain reasonable accurate prediction results [2],[5].

A number of approaches has been proposed to shorten the computation time for ray-tracing prediction models. In [2], the complexity of the building databases was reduced by simplifying footprints. Data filtering and cleansing techniques have been proposed in [6]. In order to address the same problem some procedure approximation methods are also employed in [7]. All these approaches have a common trade-off: they trade prediction accuracy for processing time. A natural way to overcome the above trade-off is to use the parallel and/or distributed computation techniques to speed up computations, while keeping the accuracy intact [5]. More specifically, the usage of a cluster of PC's (sometimes referred as a class of COW's - Cluster of Workstations) is particularly attractive as such computer system configurations are readily available at this time.

Recently, some parallel computation strategies have been proposed in order to reduce the required computational time without affecting the prediction accuracy requirements [5],[8]. In [5], the strategy of parallelization proposed is very complex and difficult to implement. This approach was applied in a 3D ray-tracing model with some restrictions in the diffraction mechanism (Vertical Plane model) [2] and is very dependent on the SBR algorithm implementation. The parallelization strategy proposed in this paper (proposal initially idealized in [8]) is very simple to implement computationally and can be applied easily in full 3D ray-tracing channel model without any restrictions if desired. This new approach allows to reduce or even eliminate many restrictions early imposed in ray-tracing models by practice reasons (high computational cost), favoring to improve the accuracy and a possibility of incorporating new propagation mechanisms, such as diffuse scattering [5]-[9] and propagation in forest environment [10]. Additionally, it allows analyzing more complex structures (scenes).

Making use to such parallel/distributed computation methods, the main objective of this paper is both to distribute and/or parallelize some components of a 3D ray-tracing prediction model among multiple nodes of a COW in such a way that the processing time will decrease proportionally to the number of nodes involved. Although the design and analysis of a COW-based prediction model involving the examination of a complex set of interrelated issues such as computational concurrency, computing unit (task) granularity, task allocation and scheduling, communication and synchronization, as well as workload-balancing [11], the main performance indexes adopted in this work are the load balancing and the speedup ones, because the overall design goal is to generate predictions as quickly as possible and with approximately the same workload in each node of the COW.

This paper is organized as follows: Section 2 discusses the baseline ray-tracing based radio propagation prediction algorithms. Section 3 outlines the computational parallelization strategy proposed in this paper. Section 4 presents some simulation in order to validate the parallel ray-tracing model. Conclusions are made in Section 5.

II. Ray-Tracing Techniques

There are basically two approaches for tracing of rays: the first one is based on Image Theory (IT) [12]. Such approach is strongly dependent on size and complexity of

the environment; it has been more used in small and simple environments involving only reflections [12]. Some authors related the possibility of incorporating diffraction and transmission points searching algorithms in such approach, but even so, its intrinsic limitation early mentioned remain. On the other hand, the shoot-and-bouncing-ray (SBR) method (referred sometimes as ”Brute-Force”) is the ray-tracing approach more suitable for large and complex environments, involving any combination of basic interactions (reflection, transmission and diffraction). The core of the SBR method is also the main source for computational intensity [5]. In this method, rays are launched with an angular separation from source points, which are either transmitters or diffraction corners acting as secondary sources. Each raypath may encounter reflections, diffractions and transmissions. In order to achieve reasonable prediction accuracy, the angular separation needs to be very small and usually less than 0.6^{o} [5]. Consequently, the number of raypaths between transmitters and receivers may be explosive and extremely long CPU processing times are required to examine all the raypaths in question. As the coverage of a wireless system increases, and the corresponding network environment becomes more complex, the interactions between raypaths and geometric objects including various types of buildings, terrain, and vegetation make matters worse calling for even more dramatic increases in computation [5].

The first natural effort in order to reduce the processing time in such model is optimizing the ray-face intersection tests (or shadowing tests). There are several approaches related to that optimization, such as BSP (Binary Space Partition), SVP (Space Volumetric Partition), Angular Z-buffer algorithm, BV (Bounding volumes) and so on [12]. Additionally, efforts have been made to parallelize the code that implements the ray-tracing algorithm [5],[8]. In IT, the parallelization of the code is not trivial, since the data structure used in this technique (tree of images) is totally concatenated, hindering the splitting of the tasks and the load balancing among the processors nodes. On the other hand, the SBR technique is already is intrinsically parallel, because the rays that are launched by transmitting antenna are independent each other, allowing that the SBR code is directly applicable in the parallel programming paradigm. Therefore, the code parallelization strategy was developed over a SBR approach.

III. Parallelization Strategy

For the SBR ray-tracing algorithm, the fundamental and active element is the ray. A ray consists in a possible propagation path (exact or approximate) inside a simulated propagation environment. Each ray is created on an independent way and also interacts independently with the scene obstacles (i.e. walls, floor, building, etc) until its associated energy become small enough or the number of interactions allowed is reached such that the ray can be discarded. Thereby, all computational effort is associated to the object ray, and as the rays are independent from each other, it may be structured a computational parallelization strategy where the rays are distributed among the processors nodes which constitute parallel computer (cluster). Only the scene, transmitting and receiving antennas would be shared, however, these objects could be replied and their copies maintained physically in the main memory in each node of the cluster.

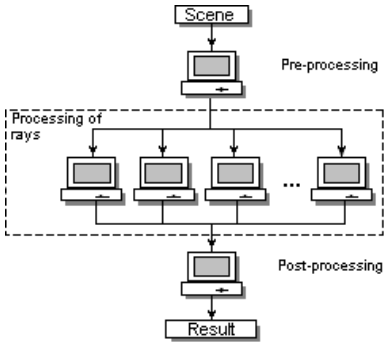

Fig. 1. Parallelization proposal for SBR algorithm with the stage of intensive processing (processing of rays) constituted by isolated nodes.

An important issue about the proposed strategy parallelization is that, as the rays are independent from each other, there is no necessity of communication among the processors nodes of the cluster during the rays launching simulation stage (processing of rays in Fig.1), which practically concentrate all the computational cost of the SBR algorithm. Without the necessity of communication, the parallelization efficiency would be theoretically always 100%, independent of the quantity of processors used. However, some problems related to particularities in the ray simulation process can commit the load balancing among the processors nodes, reducing efficiency.

According to classical SBR algorithm, the processing of rays stage consists in the repetition of interception tests of the ray under process with all faces that compose the scene. Besides, this task demands some additional computational cost necessary to generate new rays, as reflected, transmitted and diffracted ones.

As the parallelization strategy prescribes the division of the rays among the processors nodes, and as each ray can generate a different number of descendant rays depending on the scene complexity to which they are driven, the natural consequence would be not load-balancing among the nodes, damaging the parallelization. In order to overcome this problem, it would be necessary a previous acknowledgement of the behavior of each ray generated by a transmitting antenna. Unfortunately, this behavior only can be determined through the simulation of rays, which is just the problem to be solved. However, it would be possible

to achieve an adaptive load-balancing dividing continually the rays among the processors nodes as the rays are processed, however, this approach is very complex in the algorithm implementation sense. Certainly, the appropriate solution to the problem is discovering a way to distribute correctly the initial rays among the cluster processors. A simplistic solution would be a spatial division of the initial rays, being each node responsible for the rays processing designed to its domain (spatial region). However, it may occur that a domain is gone back for a spatial region with many obstacles, while another one is gone back for an absolutely empty region, damaging the parallelization strategy. The proposal of this paper is to demonstrate a method which the set of rays to each node is formed through a random choice performed over all the rays that compose the rays source. This way, each spatial region would be constituted of rays which processing would be performed by different processor nodes. A random distribution of rays would be more efficient as larger the total number of emitted rays is, exactly the case that most justifies the use of parallel computation. Through this technique, the processing load of a homogeneous cluster would be balanced through the distribution of the equal number of rays (randomly chosen) for each node. For a heterogeneous cluster, the rays number of each node must be proportional to its processing capacity. Evidently, discovering the processing capacity of computers may be done previously, it being even possible to estimate it based on characteristics of hardware and software.

Specifically, the pre-processing shown in Fig.1 would consist of the initial rays separation to each cluster node. This is the unique difference among the processes of each node, characterizing the solution proposed in this paper as SPMD (Single Program Multiple Data) paradigm [11]. The different listings of rays would be distributed through the network together with the scene (equal for all cluster nodes). This initial communication could be implemented, for instance, using MPI (Message Passing Interface) standard communication library [13]. However, a simplest strategy would be the construction of a customized input file to each process (node), distributed through a network file system, as the NFS (Network File System), used in UNIX systems [14]. When the rays simulation is over, each isolated process can send their results through the network using MPI, or make available in the form of local file shared through the NFS. The processing result would be a report of all the rays launched or just a report of all the rays that reached some regions of the scene previously determined (reception points). Particularly the record of all launched rays could be a large quantity of data, which storage in main memory would be unviable. Therefore, if the recording of results is undesirable in local files, it will be necessary to send them continually through the network (for example, with the aid of MPI), while the simulation elapses. The reception and organization of results consist in the post-processing stage, as schematized in Fig. 1.

IV. Results

In order to validate the parallelization strategy, it was considered as study of case a simple outdoor geometry consisting of four building with a rectilinear street pattern, as shown in Fig.2. Each building consists of a lossy dielectric (ϵ_r = 4, σ = 0.05 S/m). A perfectly conducting ground plane is assumed. To avoid any symmetry in the propagation of the rays and creating a suitable configuration in order to test the proposed strategy, the transmitter was located at x=37.5, y=6.26 m at a height of 3 m and the field points were placed along the line AB at a height of 2m (Fig.2). As ray-launching algorithm it was considered the classical full 3D SBR model together with the UTD (Uniform Theory of Diffraction) described in [8],[12]. The fields were calculated at a frequency of 900 MHz with raypaths with up to 8 reflections and 2 diffractions. The effects of paths which diffract over the rooftops were neglected due to the transmitting and receiving antennas were located right below the building heights, and in these situations, such paths are usually of negligible power compared to other paths which propagate among the buildings.

The simulations were carried out under a COW consisting of eight (08) nodes. All nodes are constituted of an Athlon XP 1800+ processor and main memory of 1.5 GB.

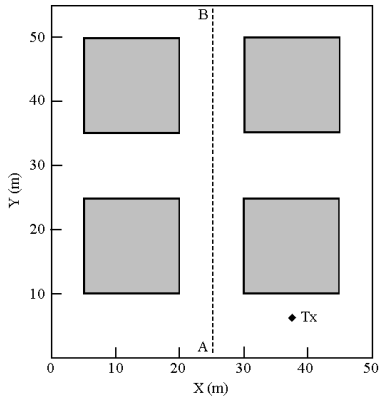

Fig. 2. Four building geometry used for the numerical results

The parallelization strategy performance was basically evaluated by speed-up and load balancing factors. According to Fig.3, the speed-up factor obtained in case with a large number of rays launched by source (655.362 rays) was almost linear (ideal), just where the ray-tracing model is more accurate. It shows that the random distribution of the initial rays and the work performed by each node over a small data quantity are more efficient to these situations. Table I shows the number of rays processed

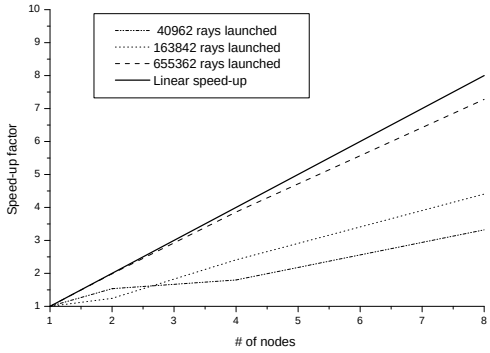

Fig. 3. Speed-up factors to distinct number of launched rays

(load balancing) by each node in several COW setups for 655.362 rays launched by source, presenting a maximum processed rays difference related to average value around 10%. For a small number of rays launched (where the ray-tracing model is less accurate), the strategy lost efficiency due to random distribution fails to a small number of samples (rays). However, it is evident the necessity of parallelization of the SBR code, since the largest serial runtime obtained in our model was close to 15 hours.

TABLE I

PROCESSED RAYS (LOAD BALANCING)

Node	1-node COW	2-nodes COW	4-nodes COW	8-nodes COW
1	138268864	69574698	35030073	16043524
2	-	68694166	33450856	17491353
3	-	-	34018621	16881723
4	-	-	35769314	19158096
5	-	-	-	17772988
6	-	-	-	17910682
7	-	-	-	16457421
8	-	-	-	16553077

V. CONCLUSION

Utilization of parallel processing has been able to improve the efficiency of most scientific models used in Engineering. In the field of telecommunications, several computationally intensive problems are addressed, as radio-propagation ones. In this paper, a computational parallelization strategy to speed up computations of the SBR algorithm for ray-tracing radio propagation prediction models was proposed. In such approach, the initial rays launched by source are simply distributed in a random way among processors nodes that compose the parallel computer (cluster of PC's - COW). Due to independence of the rays, no communications among process are necessary in the ray simulation stage (main stage of the SBR algorithm), resulting therefore in a parallelization model almost idealized. Simulation results showed that the parallelization strategy efficiency increases with the increase of the number of rays launched by source, it has been able to reach speed-ups close to the linear case. The load-balancing analyses were also performed with good results, showing a maximum load difference related to average value around 10%. The strategy scalability is naturally guaranteed due to independence of the rays, allowing the overall runtime to be reduced by a factor as large as possible to incorporate new processors nodes in the COW. Applying other ray-tracing acceleration techniques together with the parallelization strategy proposed here, there will be substantial improvement of performance in such radio propagation prediction models, making possible the analysis of more complex wireless environments.

ACKNOWLEDGMENT

This work was supported by the National Council for Scientific and Technological Development (CNPq).

REFERENCES

[1] Richter, J., Al-Nuaimi, M.O.; Ivrissimtzis, L.P., "Optimization of radio coverage in urban microcells using a UTD based ray-tracing model," *2004 Antennas and Propagation, IEE Proceedings*, vol. 151, Issue: 3, 21 June 2004, pp. 187 - 192.

[2] Z. Chen, A. Delis, H.L. Bertoni, "Building footprint simplification techniques and their effects on radio propagation predictions," *Comput. J.*, 47 (1) (January 2004) 103-133.

[3] Kipp, R.A.; Miller, M.C., "Shooting-and-bouncing ray method for 3D indoor wireless propagation in WLAN applications," *Antennas and Propagation Society Symposium, 2004*, vol. 2, 20-25 June 2004, pp 1639 - 1642.

[4] Aryanfar, F.; Srabandi, K., "3D wave propagation simulation in complex indoor structures," *Antennas and Propagation Society Symposium, 2004*, vol. 2, 20-25 June 2004, pp 1635 - 1639.

[5] Z. Chen, A. Delis, H.L. Bertoni, "Radio-wave propagation predictions using ray-tracing techniques on a network of workstations (NOW)," *J. Parallel Distrib. Comput.*, 64 (2004) 1127-1156.

[6] T. Kurner, A. Méier, "Prediction of outdoor and outdoor-to-indoor coverage in urban areas at 1.8 GHz," *IEEE J. Selected Areas Comm.*, 20 (3) (April 2002) 496-506.

[7] Z. Chen, H.L. Bertoni, A.Delis,"Progressive and aproximate techniques in ray-tracing based radio wave propagation prediction models," *IEEE Trans. Antennas Propagation*, 52 (1) (2004) 240-251.

[8] Cavalcante, A. M., Costa, J. C. W. A., Francês, C. R. L., Souza, L. V., Sousa, M. J. and Cavalcante, G. P. S.,"Computational parallelization strategy applied in 3D ray-tracing to modeling of radio mobile channel,"(in portuguese) *Momag (XI Brazilians Simposium of Microwave and Optoelectronic -SBMO and VI Barzilians Congress of Electromagnetism -CBMag) 2004*, São-Paulo - SP, Brazil.

[9] Degli-Esposti, V,; Guiducci, D.; de'Marsi, A.; Azzi, P.; Fuschini, F.;"An advanced field prediction model including diffuse scattering," *IEEE Transactions on Antennas and Propagation*, vol. 52, Issue: 7, July 2004, pp. 1717 - 1728.

[10] Bertoni, H. L., *Radio propagation for modern wireless systems*, Prentice-Hall - Wireless Communications Series, 2000.

[11] Y. Foster, K. Kennedy, J. Dongarra, G. Fox, *Sourcebook of Parallel Computing*, Morgan Kaufmman Pub, 2002.

[12] M. F. Cátedra and J. P. Arriaga, *Cell Planning for Wireless Communications*, Artech House - Mobile Communications Series, 1999.

[13] W. Gropp, E. Lusk, A. Skjellum, R. Thakur, *Using MPI : Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation)*, 2nd Edition, MIT Press, 1999.

[14] Tanenbaum, A.S., *Computer Networks*, 4th Ed.,Prentice Hall, 2002.