Vascular Network Modeling - Improved Parallel Implementation on Computing Cluster

Krzysztof Jurczuk^{1,2,3}, Marek Krętowski¹, and Johanne Bézy-Wendling^{2,3}

 Faculty of Computer Science, Białystok Technical University Wiejska 45a, 15-351 Białystok, Poland
² INSERM, U642, Rennes, F-35000, France
³ University of Rennes 1, LTSI, Rennes, F-35000, France {k.jurczuk,m.kretowski}@pb.edu.pl

Abstract. In this paper, an improved parallel algorithm of vascular network modeling is presented. The new solution is based on a more decentralized approach. Moreover, in order to accelerate the simulation of vascular growth process both the dynamic load balancing and periodic rebuildings of vascular trees were introduced. The presented method was implemented on a computing cluster with the use of the MPI standard. The experimental results show that the improved algorithm results in better speedup thus making it possible to introduce more physiological details and also perform simulations with a greater number of vessels and cells. Furthermore, the presented approach can bring the model closer to reality where the analogous vascular processes can be also parallel.

1 Introduction

The vascular system modeling plays an important role in the field of computational biology and medicine [2]. It can help to understand complex vascular processes (e.g. blood transport, angiogenesis) and also their impact on the other living elements (e.g. cells), tissues and whole organs. In medicine, computer models of blood vessel trees may be considered as working tools that can help to understand pathological processes. Moreover, virtual organs with vascular systems can be used to perform dynamic medical imaging simulations, with no need for patients to participate and hence useful to test several acquisition protocols.

In order to build a representative model which reflects reality, we ought to take into account the most essential properties of the system and disregard all those elements whose role is insignificant. However, this selection process constitutes one of the main difficulties in model designing, particularly in modeling of living organisms and consequently vascular systems. A high quality vascular model, apart from representing an appropriate number of physiological and anatomical details, should be also effective in practical cases, i.e. the computational simulations should be performed in a reasonable time. Moreover, a lot of internal organic processes are parallel or distributed (e.g. new blood vessel growing from the pre-existing ones). Therefore, the creation of distributed vascular models seems to be useful. Firstly, they can decrease the time of simulation and

R. Wyrzykowski et al. (Eds.): PPAM 2009, Part I, LNCS 6067, pp. 289–298, 2010.

simultaneously allow us to introduce more sophisticated details. Secondly, they make it possible to reflect reality more accurately.

In our previous research [1] and [6], we used the modeling of vascular systems to search for pathology markers and improve the interpretation of dynamic medical images. We focus on vessel networks because many diseases are directly related to changes in vessel structures. Moreover, when a contrast product is administrated, these anatomical or functional vessel modifications can appear in medical images. Initially, we made use of a sequential algorithm to generate a virtual liver [7], CT simulator and MRI virtual scanner. Subsequently, we introduced a parallel solution of vascular network growth [3], which significantly reduced the computation time. In this paper, we wish to present an improved parallel algorithm of physiological vascular modeling whose aim is to further accelerate the simulation process. Moreover, the new solution is based on a more decentralized approach that can reflect the real process of vessel growth more accurately. Both parallel algorithms were implemented on a computing cluster with the use of the MPI standard [10].

Many other vascular models have been proposed, e.g. Constrained Constructive Optimization method for an arterial tree generation inside a 3D volume [5] or a fractal model [12]. However, as far as we know, all the previous solutions, published by other authors, have been using sequential algorithms to develop vascular systems.

The remaining part of the paper is organized as follows. In the next section the vascular model is briefly described and both the sequential and preliminary parallel algorithms of vascular development are recalled. In Sect. 3 the improved parallel solution of the vascular growth process is explained. An experimental validation of the proposed algorithm is performed in Sect. 4. The last section contains the conclusion and some plans for future research.

2 Vascular Modeling for Image Understanding

The discussed model consists of two main components: the tissue and the vascular network [7]. Although most of its features are not linked with any particular organ, the model expresses the specificity of the liver which is one of the most important organs. The liver plays a major role in the metabolism and has a number of functions in the body, including hormone production, decomposition of red blood cells, detoxification, protein synthesis, etc. [11]. Moreover, it stands out from other vital organs by its unique organization of the vascular system with three types of trees: hepatic artery, portal vein and hepatic vein.

The model is designated to simulate the growth of organs and certain pathological processes. However, it should be emphasized that it is oriented towards an image generation. Therefore, it concentrates on elements that are directly visible in images or have a significant influence on image formation. In our case, we focus on vessels which play a key role in the propagation of contrast material and are very visible in dynamic CT and MRI images. In the model, vessels form the vascular system and supply the tissue.

2.1 Tissue Modeling

The tissue is represented by a set of Macroscopic Functional Units (MFU) which are located inside the specified shape. An MFU is a small, fixed size part of tissue and has been assigned a class which determines most of its structural/functional properties (e.g. probability of mitosis and necrosis, size, density) and also physiological features (e.g. blood flow rate, blood pressure). Several classes of MFUs can be defined, which makes it possible to represent pathological regions of the tissue. Moreover, in order to introduce more natural variability, certain parameters (like blood flow rate) are described by defined distributions.

2.2 Vessel Network Modeling

The vascular network of the liver consists of three vessel trees. Hepatic arteries and portal veins deliver blood to individual MFUs. Whereas, the hepatic venous tree is responsible for blood transport back to the heart. In the model, the vascular trees are composed of vessels that can divide creating bifurcations. Each vessel is represented by an ideal, rigid tube with a fixed radius, wall thickness, length and spatial position. The model distinguishes vessels larger than capillaries, while the capillaries themselves are hidden in the MFUs. As a result, anastomoses (e.g. mutual vessel intersections), which occur particularly among vessels with very small radii or in pathological situations, are not taken into account. Hence, each vascular tree assumes the form of a binary tree.

In order to design realistic vascular structures, the vessel parameters (i.e. radius, blood flow, etc.) are calculated according to two basic physical laws. At each bifurcation the law of matter conservation must be observed:

$$Q = Q_r + Q_l \quad , \tag{1}$$

where Q is the blood flow in a parent vessel, Q_r and Q_l are the blood flows in the right and left daughter branches respectively. This relation ensures that the quantity of blood entering and leaving a bifurcation is equal. Moreover, the morphological law describing the dependency of the mother vessel radius (r) and the radii of its two daughters (right r_r and left r_l) is used:

$$r^{\gamma} = r_r^{\gamma} + r_l^{\gamma} \quad , \tag{2}$$

where the γ factor varies between 2 and 3 [4]. Furthermore, one more important assumption is also taken into consideration. In the model, the blood is treated as a Newtonian fluid, with constant viscosity (μ). Therefore, its flow can be modeled as a non-turbulent streamline flow in parallel layers (laminar flow) induced by the pressure difference (ΔP) between two extremities and consequently the Poiseuille's law can be used to calculate the remaining properties of vessels:

$$\Delta P = Q \frac{8\mu l}{\pi r^4} \quad , \tag{3}$$

where l is the length, r is the radius and Q is the blood flow of the vessel.

2.3 Sequential Vascular System Growth Algorithm

The growth process starts with an organ whose size is a fraction of the adult one. In discrete time moments (called cycles) the organ enlarges its size (growth phase) until it reaches its full, mature form. Additionally, each cycle consists of subcycles during which each MFU can divide and give birth to a new MFU of the same class (mitosis process) or die (necrosis process). During the growth phase the proliferation process is dominant, due to the mitosis supremacy over the necrosis (analogy to immature and not fully developed organs) [8]. Therefore, the increasing needs of the growing tissue induce the development of a vascular network which is responsible for the blood delivery. In each subcycle, the processes of mitosis and necrosis are repeated, whereas new cycle starts when the specified organ shape is filled by MFUs.

New MFUs, which appear during the mitosis process, are initially ischemic (no connection with blood vessels) as they are not perfused by the existing vascular system. As a result, they produce angiogenic factors to stimulate the closest vessels to sprout new branches that will be able to provide blood supply. In the model, for each new MFU a fixed number of the nearest vessels (candidate vessels) is found. If more than one tree is taken into account, the algorithm creates all possible combinations of candidate vessels (a single combination consists of one vessel from each tree). Subsequently, temporary bifurcations are created, which means that all combinations of candidate vessels sprout new branches that temporarily perfuse the assigned MFU. The spatial position of the bifurcation point is controlled by the Downhill Simplex procedure (minimization of additional blood volume to perfuse and supply new MFU) [7].

The above process can be regarded as a kind of competition since only one vessel in each tree can perfuse the new MFU (exactly one combination of candidate vessels) and the remaining ones retract and disappear. Moreover, in the presented approach only non-crossing configurations are taken into account, therefore the algorithm detects intersections between perfusing vessels both from the same and different trees and rejects related configurations. Finally, from among the remaining combinations, the configuration with the lowest sum of volumes is chosen to permanently perfuse the MFU.

After the mitosis and perfusion (i.e. reproduction process), comes a degeneration phase. During this part of the algorithm, few MFUs can die (necrosis process) and then all the vessels supplying these MFUs retract and disappear (retraction process). Next, the algorithm goes back to the reproduction phase again.

2.4 Previous Parallel Growth Algorithm

In the sequential algorithm of organ growth, all MFUs are perfused by the vascular system one by one. For each new MFU a lot of temporary bifurcations have to be created and tested, which requires a great number of calculations to face the imposed constraints connected with realistic vascular designing. Therefore, in order to accelerate this most time consuming part of vascular development, an algorithm to spread the computations of the perfusion process over processors was introduced (see Fig. 1) [3]. Moreover, our intention was to bring the solution closer to reality in which analogous perfusion phenomena can also occur in a parallel way.

The mitosis process is performed at the managing/principal node that, however, makes no attempt to find candidate vessels. Instead, it sends new MFUs to computational nodes. When a computational node receives the message, it then tries to find the closest vessels and an optimal point to connect the received tissue element. If the search ends with a success, it does not perfuse a new MFU permanently but sends the parameters of the bifurcation to the managing node.

Next, when the managing node receives a message with an optimal configuration, it takes a decision about a permanent perfusion. If, however, at least one of candidate vessels sent by the computational node was used earlier to perfuse another tissue element, then the MFU is rejected. It means that such a vessel no longer exists at the managing node, which is related with the tree nonuniformity between the nodes. But in other cases, the new MFU is permanently connected to the vascular system. Subsequently, if there are more MFUs to be checked, the managing node sends both all the new changes in its vascular trees as well as the next MFU to the sender of the latest considered configuration.

The remaining processes (i.e. necrosis, retraction and shape growth) are performed only at the managing node that possesses the most current vascular system and tissue. Thus, at the beginning of each subcycle, the principal node broadcasts the latest vascular trees and MFUs. In order to minimize the message size, we choose only the parameters of vessels that cannot be reconstructed by computational nodes. Furthermore, a lot of vascular structure characteristics are read from input files, which enables us to send quite a small package in comparison to the size of all modeled organ elements.



Fig. 1. The outline of the preliminary parallel algorithm of vascular growth

3 Improved Algorithm of Parallel Vascular Development

The previously presented parallel algorithm of vascular network growth can be treated as the first step of bringing the discussed model closer to reality [3]. Moreover, in spite of the fact that in this solution the efficiency of computational nodes was strongly dependent on the operation of the managing node that splitted the tasks step by step, it allowed us to gain a significant speedup (about 8 for 16 processors). This strong dependency ensured high uniformity of the vascular trees between the nodes. Furthermore, the computational nodes

were evenly loaded during the perfusion process. But, on the other hand, such an approach could result in delays connected with the additional time needed to wait for the following candidate MFUs. Therefore, a decision was made to build a more decentralized algorithm that would provide a better speedup and, at the same time, reflect reality more accurately.

Basing on the profiling results (execution times of specific methods), it was shown that the time needed to send a message with vascular structures and tissue within the framework of the message passing interface is insignificant but the tree reconstruction process at the computational nodes is a time consuming operation, especially for large configurations with thousands and more vessels. Therefore, we decided to broadcast the whole organ only once at the beginning of the simulation to assure that all the nodes possess identical vascular trees and MFUs.

Now, each new subcycle starts with the mitosis during which a list of new MFUs is created (see Fig. 2). Next, the perfusion process is performed. As it is the most time consuming operation of the vascular network growth, we also decided to improve it. After the reproduction process, the degeneration phase follows. Due to giving up broadcasting the whole organ in each of the subcycles, all the computational nodes must be informed about possible necrosis changes. Therefore, the managing node transmits essential information to all other nodes about the MFUs that have to be removed. The entire algorithm of retraction must be performed at each node and fortunately it is not so time consuming in comparison to the perfusion process. There is a possibility to model the process of MFU selection in a parallel environment but the performance analysis showed that the time needed for that part of the algorithm can be neglected.

After the degeneration process, if the organ reaches its adult size, the algorithm ends but when the another new subcycle is needed the process starts again with the mitosis. In the other case the organ enlarges its size. At each node the same operations are performed in a parallel way and only after they are completed, the solution returns to the mitosis process.



Fig. 2. The general diagram of the improved parallel vascular growth algorithm

3.1 Modified Parallel Perfusion Algorithm

After the mitosis process, when the list of new macroscopic functional units is created, the managing node spreads all candidate MFUs over computational nodes (see Fig. 3). Each computational node, after receiving the message with the group of different MFUs, attempts to find the closest vessels and optimal points to perfuse new tissue elements. Each time, when the search ends with a success, the parameters of the optimal bifurcation are sent to the managing node. Next, the computational node checks if there are any messages with vascular tree changes to be introduced in its vascular system. If there are, the changes are applied and the node continues to work with the next assigned candidate MFU.

Special attention was paid to assure the optimal load among computational nodes. Due to spreading all candidate MFUs, immediately after the mitosis process, some of processors can be idle after performing all assigned tasks while others have jobs queuing for execution. Therefore, the opportunity to leave some candidate MFUs, which are sent on demand, was introduced (dynamic load balancing). However, before the work distribution in each following subcycle, it is impossible to approximate the time needed to perform each task. Therefore, the presented solution makes it possible to leave a fixed part of new MFUs. When the computational node finishes all assigned tasks it sends a message to the managing node with the demand to get more job. Such an algorithm allows us to detect and handle load imbalances dynamically. Moreover, we introduced the possibility to spread some MFUs before the end of the mitosis process.

The managing node is responsible for gathering messages coming from the computational nodes. It can also perform calculations connected with finding optimal perfusion parameters. Owing to non-blocking communication between the managing and computational nodes, for each permanent perfusion in the vascular network a unique number is attributed. The managing node also collects tree updates in special queues for each individual node as well as related permanent perfusion numbers. Using this information, it estimates the tree uniformity and makes a decision about the permanent perfusion. The principal node has a possibility to send all collected vascular tree updates only to the sender of the



Fig. 3. The detailed diagram of the improved parallel perfusion process

last package or broadcast the permanent changes among all the computational nodes immediately as these changes appear.

4 Experimental Results

In this section the improved solution is experimentally verified. The default settings for the sequential version (about 12000 MFUs) were used and we also checked the behavior of the algorithm for large configurations with about 48000 MFUs and consequently about 300000 vessel segments. Figure 4 presents a visualization of one of the obtained vascular networks.

In the experiments a cluster of sixteen SMP servers running Linux 2.6 and connected by an Infiniband network was used. Each server was equipped with two 64-bit Xeon 3.2GHz CPUs with 2MB L2 cache, 2GB of RAM and an Infiniband 10Gb/s HCA connected to a PCI-Express port. We used the MVAPICH version 0.9.5 [9] as the MPI standard implementation [10]. Moreover, randomly selected experiments were carried out on a similar cluster of sixteen SMP servers except that each server was equipped with eight CPUs and the MVAPICH version 1.0.1. Furthermore, in order to execute the performance analysis we made use of the Multi-Processing Environment (MPE) library and the graphical visualization tool Jumpshot-4 [10].

Figure 5a presents the obtained mean speedup. It is clearly visible that the improved algorithm is about 25% faster than the previous one. Currently, the time needed to obtain an organ with about 300000 vessels equals approximately 2 hours (with 16 processors), instead of 24 hours in a single processor machine (3.2GHz CPU and 2GB of RAM). Moreover, it is worth noting that in the presented solution the acceleration is not sensitive to the increasing size of vascular structures. Furthermore, in order to assure that the vascular trees are represented in the operating memory by continuous memory segments, periodical reallocations were introduced. Namely, at the beginning of each subcycle computational nodes rebuild their vascular structures. Obviously this operation takes some time



Fig. 4. Visualization of adult liver (about 48000 MFUs and 300000 vessels): a) hepatic veins with a tumor shape, b) main hepatic arteries, portal veins and hepatic veins with liver and tumor shapes



Fig. 5. Efficiency of the parallel implementation: a) mean speedup for several configurations of MFUs, b) influence of dynamic load balancing

but the profiling results showed that exactly at the same time the mitosis process is being performed at the managing node. Owing to this vascular tree rebuilding, the time needed to find optimal bifurcations, connected with frequently executed tree search algorithms, was decreased. Prior to the introduction of this mechanism, the obtained speedup used to be even worse than in the previous solution, especially for huge vascular networks.

The detailed results concerning the optimal load balancing among computational nodes are presented in Fig. 5b. It shows how the number of left (send on demand) MFUs influences on the simulation time. We can see that it is impossible to choose one common value that gives the best gain in time. But on the other hand, it is visible that each value below 90% makes it possible to achieve a better speedup. Finally, we suggest that any value within the range from 30% to 60% is acceptable.

The introduced possibility of spreading new MFUs before the end of mitosis has a positive influence on the simulation time when the number of vascular structures is small enough (i.e. thousands of vessels maximally). Moreover, when the managing node also performs calculations connected with finding optimal bifurcations, the speedup can be decreased because of delays in tree change broadcasting (especially in experiments with more than 4 processors).

5 Conclusion

In this paper, an improved solution of vascular network modeling in a parallel environment is investigated. It was shown that the new algorithm is about 25% faster than the one used previously. As a result, it is not only possible to increase the number of MFUs and vessels but also introduce more details into the model.

Moreover, the presented algorithm, based on a more decentralized approach, can bring the discussed model closer to reality in which analogous phenomena of vascular development can occur both in a parallel and distributed way.

The presented model is still in its development stage. Continuing our research, we intend to introduce an algorithm aimed at restoring the tree uniformity among nodes during the perfusion process. On the other hand, we plan to implement the vascular growth process in the framework of multi-platform shared-memory parallel programming (OpenMP), which could eliminate the necessity of maintaining tree uniformity between nodes.

Acknowledgments. This work was supported by the grant W/WI/5/08 from Białystok Technical University.

References

- 1. Bézy-Wendling, J., Krętowski, M.: Physiological modeling of tumor-affected renal circulation. Computer Methods and Programs in Biomedicine 91(1), 1–12 (2008)
- 2. Hoppensteadt, F.C., Peskin, C.S.: Modeling and Simulation in Medicine and the Life Sciences. Springer, Heidelberg (2004)
- Jurczuk, K., Krętowski, M.: Parallel implementation of vascular network growth. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 679–688. Springer, Heidelberg (2008)
- Kamiya, A., Togawa, T.: Optimal branching structure of the vascular trees. Bulletin of Mathematical Biophysics 34, 431–438 (1972)
- 5. Karch, R., Neumann, F., Neumann, M., Schreiner, W.: Staged growth of optimized arterial model trees. Annals of Biomedical Engineering 28, 495–511 (2000)
- Krętowski, M., Bézy-Wendling, J., Coupe, P.: Simulation of biphasic CT findings in hepatic cellular carcinoma by a two-level physiological model. IEEE Trans. on Biomedical Engineering 54(3), 538–542 (2007)
- Krętowski, M., Rolland, Y., Bézy-Wendling, J., Coatrieux, J.-L.: Physiologically based modeling for medical image analysis: application to 3D vascular networks and CT scan angiography. IEEE Trans. on Medical Imaging 22(2), 248–257 (2003)
- Morgan, D.O.: Cell Cycle: Principles of Control. Oxford University Press, Oxford (2006)
- MVAPICH: MPI over InfiniBand and iWARP, http://mvapich.cse.ohio-state.edu/
- 10. Pacheco, P.: Parallel Programming with MPI. Morgan Kaufmann Publishers, San Francisco (1997)
- 11. Sherlock, S., Dooley, J.: Diseases of the liver and biliary system. Blackwell Science, Malden (2002)
- Zamir, M.: Arterial branching within the confines of fractal L-system formalism. Journal of General Physiology 118, 267–275 (2001)