XL CILAMCE
IBERO-LATIN AMERICAN
CONGRESS ON
COMPUTATIONAL
METHODS IN
ENGINEERING
NOVEMBER
11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

A parallel implementation of a thermoregulation model using the finite element method

Henrique C. C. Andrade
Ana Beatriz C. G. Silva
henriqueconde@coc.ufrj.br
anabeatrizgonzaga@coc.ufrj.br
Fernando Luiz Bastos Ribeiro
fernando@coc.ufrj.br
Federal University of Rio de Janeiro
Technological Center, Ilha do Fundão, CEP 21945-970, Rio de Janeiro, Brazil
Luiz Carlos Wrobel
luiz.wrobel@brunel.ac.uk
Brunel University
Uxbridge UB8 3PH, United Kingdom

Abstract. This paper presents a parallel implementation of a thermoregulation model using the finite element method to perform numerical analyses of brain cooling procedures in neonates and adults. The parallel method was designed for hybrid parallel machines, which is the case of PC clusters with multicore processors connected via a local network. Compressed data structures are used to store the coefficient matrices and obtain iterative solutions in a subdomain-by-subdomain approach. The MPI standard is used for distributed memory interprocess communication and for shared-memory, with intranode communication being performed simulating a virtual distributed-memory system in a multicore hardware. The method is used in a FEM package applied to the solution of the continuum bioheat Pennes equation with a blood pool approach for arterial temperature changes. The efficiency of the parallel implementation is tested in two different platforms using a complex three-dimensional geometry obtained from computed tomography medical images.

Keywords: finite element method, parallel computing, thermoregulation model, hypothermia treatment

1 Introduction

In recent decades, several models were developed to reproduce the human thermoregulatory behavior, from two node models of core and skin heat balances to more complex multi-segment models of the human body and its thermoregulatory responses [1]. These models incorporate concepts of physiological regulation to predict human thermal responses and body heat loss at various activity levels and thermal environments [2]. Practical examples can be found in various application fields such as the automotive industry, environmental comfort and biomedical engineering [1–3].

The work of [2] uses an arterial system model to calculate blood flow in the core tissue and a bioheat model to determine skin temperature for nude and clothed human body in transient non-uniform environments. The model presented in [4] incorporates the body heat losses considering a non-uniform temperature distribution in the skin, regulatory responses, properties of clothing used and various environmental conditions such as extreme temperatures, wind speed and solar radiation. [5] uses a compartimental model of 12 body segments and a blood compartment to simulate whole body hyperthermia treatments for tumours. In [6], a heat transfer model using the commercial software *FLUENT* was implemented to simulate hypothermia treatment in neonates using a three-dimensional geometry obtained from magnetic resonance imaging (MRI) scans. In [7], a numerical model for whole body intravascular cooling was developed and applied to a human body consisting of a cylinder of one material and a combination of components representing torso, head and limbs.

The consideration of realistic geometries for representing the human body makes the finite element problems computationally expensive, involving the solution of large systems, that must be solved thousands of times. The evolution of computer technology in recent years have increased the capacity of scientists, engineers and mathematicians to solve these complex problems. To take advantage of this technological improvement, the simulation method must be optimized to obtain a solution in a reasonable time.

One of the major issues concerning optimization techniques is the choice of efficient data structures used to store the matrix coefficients. In addition, there are several levels of parallelization that can be used to speed up the code, such as on-chip optimizations, distributed, shared and GPU parallelization.

In this paper we present a parallel implementation of a thermoregulation model [8] using the finite element method to perform numerical analyses of brain cooling procedures in adults. The parallel method was designed for hybrid parallel machines, which is the case of PC clusters with multicore processors connected via a local network. Compressed data structures are used to store the coefficient matrices and obtain iterative solutions in a subdomain-by-subdomain approach. The MPI standard is used for distributed memory interprocess communication and for shared-memory, with intranode communication being performed simulating a virtual distributed-memory system in a multicore hardware. The method is used in a FEM package applied to the solution of the continuum bioheat Pennes equation with a blood pool approach for arterial temperature changes. The efficiency of the parallel implementation is tested in two different platforms using a complex three-dimensional geometry obtained from computed tomography medical images.

2 Methodology

2.1 Mathematical Model

In this work, the calculation of a whole body thermal analysis was based on a continuum macroscale bioheat model based on blood perfusion developed by Pennes [9]. This bioheat model considers blood and tissue as a continuous homogeneous medium. The Pennes' equation is given by

$$\rho_t c_t \frac{\partial T_t}{\partial t} = \nabla \cdot (k_t \nabla T_t) + \rho_b c_b \omega_b (T_a - T_t) + \dot{q}_m \tag{1}$$

and represents the bioheat flux in a domain Ω . In the above equation, the left-hand side accounts for the unsteady term, the first term on the right represents the conductive heat transfer through the tissue, the second entry on the right-hand side is the blood perfusion heat convection term and the last term is the energy generation associated with metabolism. The symbol T is the temperature and the subscripts t, b, a and m represent tissue, blood, arterial blood and metabolism, respectively. The material properties defined in the equation are: k (thermal conductivity), c (specific heat), ρ (density) and ω (blood perfusion rate). The metabolic heat generation rate is represented by $\dot{q}_m$. The perfusion term and the metabolic heat generation rate are considered as isotropic heat sources. The arterial temperature T_a is obtained considering the heat exchanges during blood circulation in the body, according to [4]:

$$T_a = \frac{\dot{m}_b\, c_b T_p + h_x T_v}{\dot{m}_b\, c_b + h_x} \tag{2}$$

The symbol T_p stands for the blood pool temperature, T_a and T_v are the arterial and venous temperature and h_x is the counter current heat exchange coefficient, considered as zero in the core and with defined values for the extremities of the body [1].

The bioheat equation is defined in all parts of the body considering different properties for each tissue. The boundary conditions are prescribed temperatures $\overline{T}(\Gamma_t.t)$ in the boundary Γ_t and heat fluxes $\overline{q}(\Gamma_q, t)$ in the boundary Γ_q, $\Gamma = \Gamma_t \cup \Gamma_q$. The initial condition is,

$$T(x, t_0) = T_0, \tag{3}$$

where T_0 is the initial temperature in the tissue.

The metabolic heat generation rate in a specific tissue can be considered as a composition of the basal rate $\dot{q}_{m,0}$, representing a thermal neutrality condition, and an additional rate $\Delta\dot{q}_m$ generated by a local thermoregulation activity [4]:

$$\dot{q}_m = \dot{q}_{m,0} + \Delta\dot{q}_m \tag{4}$$

The blood perfusion rate $\omega_{b,t}$ in a specific tissue can be divided into two components:

$$\omega_{b,t} = \omega_{b,0,t} + \Delta\omega_{b,t} \tag{5}$$

where $\omega_{b,0,t}$ is the local basal blood perfusion rate and $\Delta\omega_{b,t}$ is a local variation depending on the tissue temperature, calculated as:

$$\Delta\omega_{b,t} = \omega_{b,0,t}\Big[Q_{10}^{\frac{T_t - T_0}{10}} - 1\Big] \tag{6}$$

The reference temperature T_0 is the equilibrium temperature of the body and the Q_{10} coefficient is usually considered as equal to 2.

The heat exchange between the body and the surrounding environment can be divided into three main mechanisms: convection, radiation and evaporation. The heat exchange rate varies along the body surface, and the heat flux is the sum of the contributions of convective, radiative and evaporative fluxes:

$$q_{skin} = q_{conv} + q_{rad} + +q_{evap} \tag{7}$$

The convective flux q_{conv} between the skin surface and the external environment can be calculated using the Newton cooling law, defined as

$$q_{conv} = h_{conv} \cdot (T_{ext} - T_{skin}) \tag{8}$$

The symbols T_{ext} and h_{conv} indicate the external temperature and the heat transfer coefficient, respectively.

The radiative flux between the skin and the surrounding environment can be obtained by the Stefan-Boltzmann law:

$$q_{rad} = h_{rad}(T_{skin}^4 - T_{sr,mean}^4) \tag{9}$$

where T_{skin} is the temperature at the skin surface, $T_{sr,mean}$ is the mean temperature of the surrounding radiating surfaces and

$$h_{rad} = \sigma \, \varepsilon \tag{10}$$

in which σ refers to the Stefan-Boltzmann constant and ε is the average emissivity of all radiating surfaces. The emissivity may vary depending on the surface material (skin, clothes, hair).

2.2 Numerical Model

In the numerical model, the finite element method is used to obtain an approximate solution to the Pennes equation and the circulatory model described in the previous section. To solve the transient problem, a time-marching scheme based on a semi-discrete form of the FEM is used, where the spatial discretization is performed by finite elements and the time derivative is approximated by finite difference operators using the two-point closed Newton-Cotes formula, also called Trapezoidal rule. Considering the Pennes Eq. (1) in a spatial domain Ω and a temporal interval $(0, \Pi)$, the domain Ω is discretized into elements and at each time step $t = t_{n+1}$ we adopt the approximation:

$$T(x, t_{n+1}) \cong \tilde{T}(x, t_{n+1}) = \sum_{j=1}^{k} N_j(x) \, \tilde{T}_{j,n+1} \tag{11}$$

where $N_j(x)$ refers to the spatial interpolation functions and $\tilde{T}_{j,n+1}$ are the nodal values of the approximate temperature function $\tilde{T}$ at time step t_{n+1}. The time derivative of T at t_{n+1} is approximated by

$$\frac{\partial T}{\partial t}\Big|_{t=t_{n+1}} \cong \dot{\tilde{T}}(x) = \sum_{j=1}^{k} N_j(x) \, \dot{\tilde{T}}_{j,n+1} \tag{12}$$

Introducing the approximations presented above, the following system of algebraic equations is obtained at time $t = t_{n+1}$:

$$M \, \dot{T}_{n+1} + K \, T_{n+1} = F_{n+1} \tag{13}$$

where M is the mass matrix, $\dot{T}_{n+1}$ are the nodal values of the time derivative of temperature, K is the stiffness matrix, T_{n+1} are the nodal temperatures at time step t_{n+1} and F_{n+1} is the vector of independent terms. The coefficients of these matrices are calculated as follows:

$$m_{ij} = \int_{\Omega} c_t \rho_t N_i N_j d\Omega \tag{14}$$

$$k_{ij} = k \int_{\Omega} \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right) d\Omega + \int_{\Omega} c_b \rho_b \omega_b N_i N_j d\Omega \tag{15}$$

$$f_i = \int_{\Omega} \dot{q}_m N_i d\Omega - \int_{\Gamma} \overline{q} N_i d\Gamma + \int_{\Omega} c_b \rho_b \omega_b T_a N_i d\Omega \tag{16}$$

.

2.3 Parallel Implementation

The parallel Implementation uses the *MPI* standard in conjunction with a subdomain-by-subdomain strategy (SBS) and the compressed data structure *CSRC*[10], which is a variation of structure *CSR*[11] for sparse topological matrices. The main advantage of the SBS strategy is that the sequential code is practically the same as the parallel code, and the differences are only punctual where calls to the MPI functions are required. Another important advantage is that the convergence of iterative processes do not change, except for small fluctuations caused by finite-precision arithmetic. The impact of these fluctuations on the final solution is practically negligible.

The parallelization proposed in [10] used two techniques of domain division. The main differences between the two techniques are the coefficients and array dimensions of each partition. In the non-overlaping technique the partitions do not share elements with neighboring partitions, so the coefficients on the internal interfaces of the partitions are partial and the square matrix is stored using the *CSRC*. In the overlaping technique, which will be used in this work, the partitions share elements with the neighboring partitions. Thus we only have global coefficients, which results in a rectangular matrix and the storage in *CSRC* is used in the square part of the matrix. The other coefficients are stored in the standard *CSR* format. Full details of this implementation can be found in [10] and [12]. Although initially developed only for distributed memory architectures, parallel implementation demonstrated good results in [13] for hybrid architectures, or even in purely shared memory architectures [14].

The parallel algorithm used enables the use of iterative solvers such as *CG* [15], *GMRES* [16], *BICGSTAB*[16] and *SQMR*[17] without any modification of the parallel implementation. In the case of the present work, the system of equations was solved using the Jacobi preconditioned conjugate gradient method.

2.4 Parallel platform

The cases were performed on two Clusters of *PCs* connected via *LAN*. Figure 1 represents the simplified generic configuration of the two parallel platforms used in this work. Each node is a *ccNUMA* (Non-Uniform Memory Access Cache) machine with two multicore *XEONs* processors. Each processor has 4 physical cores, which leads to a total of 8 physical cores per node, and the (*sockets*) are connected through the technology *QPI* (QuickPath Interconnect). The processors have a local memory bank, which means that access to memory bank 0 (local to *socket* 0) by *socket* 1 is more costly than access from memory bank 1 (local to *socket* 1). Nodes are connected via the network.

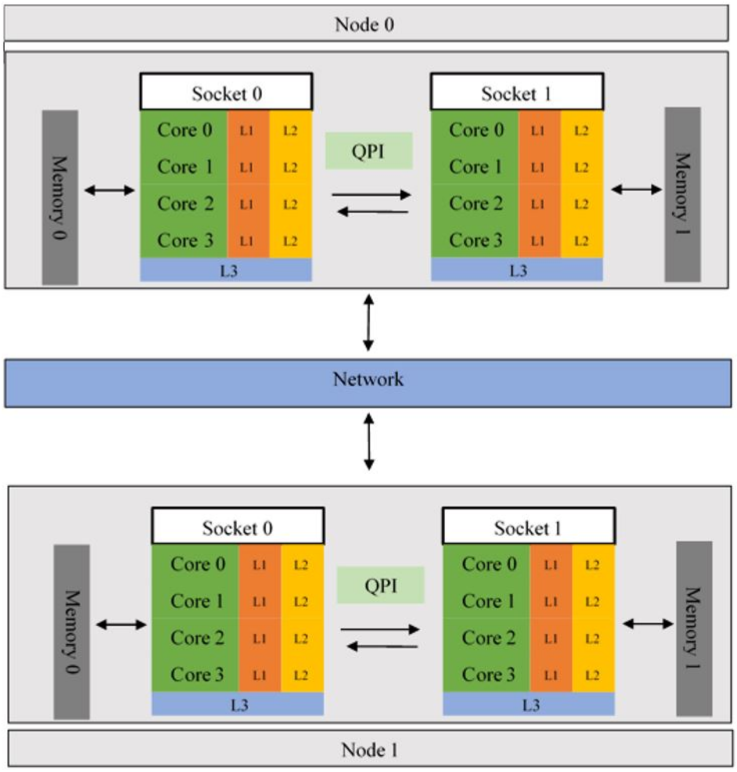


Figure 1. Simplified schematic of the platform.

The main specifications of the clusters are summarized in table 1. Platform 1 has 8 nodes with 8 cores and 16 gigabytes of RAM per machine, totaling 64 physical cores and 128 gigabytes of *RAM*. An *infiniband* 20 Gbit/s standard network is used. Platform 2 has 4 nodes with 10 cores and 256 gigabytes of *RAM* per machine, totaling 80 cores and 1,024 gigabytes of *RAM*. The network is composed by two 10 Gbit Gigabit cards running together through an operating system aggregation method called *Bonding*.

Table 1. Plataforms

	Platform 1	Platform 2
processor	E5620	E5-2630 v4
condename	Nehalem EP/Beckton	Broadwell-EP
number of nodes	8	4
Socket per node	2	2
cores/threads	4/8	10/20
cache size (L3)	12 MB	25 MB
Clock	2.4 Ghz	2.2 Ghz
QPI	5.86 GT/s	8 GT/s
Memory per node	16 GRAM (DDR3-1.067 Mhz)	256 GRAM (DDR4-2.134 Mhz)
Motherboard	S5500BC	S2600CWR
LAN	InfiniBand-Qlogic (20Gbit/s)	Gigabit (2x10 Gbit)
OS	CentOS 7	CentOS 7

Both parallel platforms are hybrid, which means there are distributed and shared memory features and the cores of each node can directly access local memory but the acess between the nodes is not possible. Considering this characteristic for the parallelization of the code we used the *MPI* standard that is implemented in this type of architecture in a simple way through the messaging approach via network or internally between the cores. The *API* is the same and the available implementations of *MPI* automatically choose the best way to perform the communication. There are numerous implementations using the libraries of the *MPI* standard. In this work the *Intel MPI* version 18.0.3 was used. The software was also generated using the *ifort* compiler of *Intel*. The only optimization option used in the build procedure was *-O3*.

Due to the hybrid nature of the platforms, the communication between the processes will occur in two ways: between processes on different nodes (*INTER-NODE*); between processes on the same node (*INTRA-NODE*). The INTRA-NODE communication can be performed between processes in the same socket (*INTRA-SOCKET*) or between different sockets (*INTER-SOCKET*). To evaluate the communication there are several benchmarks available. Here the *IMB-MPI1*, available in the *Intel* package itself, was used. In this test a point-to-point message is passed from process 0 to process 1 and returned from process 1 to 0. Figure 2 presents the results in *Gbytes* per second for messages between 2 - 1,073,741,824 bytes. It is noted that results for both the *INTRA-SOCKET* and the *INTER-SOCKET* are practically the same on both platforms, with the platform *P2* having a better performance than the *P1*. The *INTRA-NODE* communication is faster than *INTER-NODE* on both platforms. The *infiniband* (*P1*) was shown to be faster than Gigabit (*P2*) in messages between 2 - 4,194,304 bytes, while *Gigabit* was superior only for messages between 8,388,608 - 1,073,741,824 bytes.

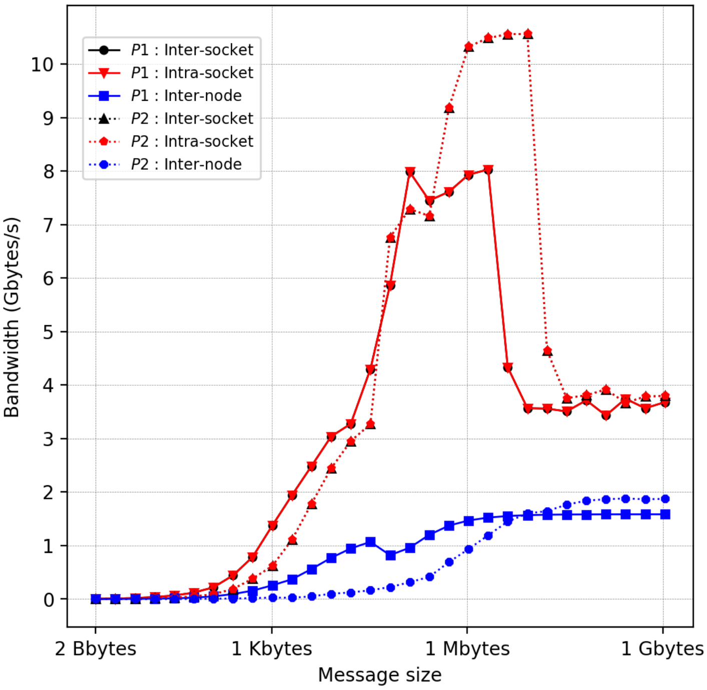


Figure 2. Benchmark IMB-MPI.

3 Application

The geometrical model used for simulations of adults was obtained by segmentation of 3D medical images (CT scans) of the Visible Human Data Set (VHD) provided by the National Library of Medicine, US Department of Health and Human Services. The medical images were used to generate the geometry using the software *MIMICS* and adapted using the packages *ANSYS Workbench* and *Trelis* to generate a 13 million mesh of a male adult with a body weight of $100kg$, body surface area of $2.27m^2$, $1.88m$ height, a cardiac output of $6l/min$ and 28% body fat content. The total basal whole body metabolism is $106W$ and basal evaporation rate from the skin of $18W$ (taken from [18]).

The geometry of the body is composed by eight different materials: skin+fat, muscle, bone, brain, viscera, lungs, eyes and cerebrospinal fluid. For the calculation of the arterial temperature, the body is divided in six sectors: trunk + abdomen, head, arm, hand, leg, foot. The simulations where performed in a geometry of half of the body, due to symmetry. The materials and division into sectors are depicted in Figures 3 to 6.

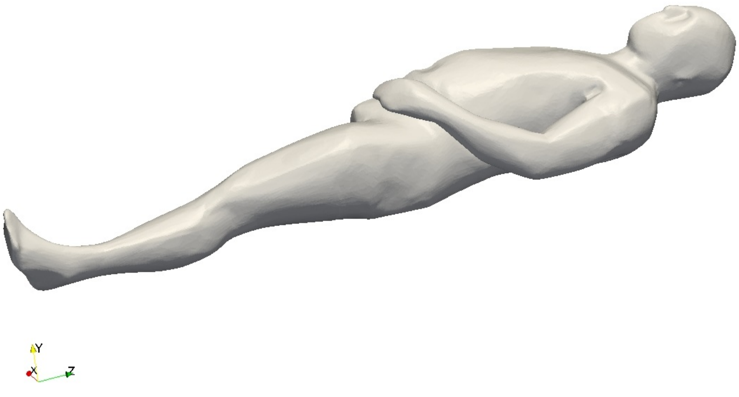


Figure 3. Geometry of the male adult.

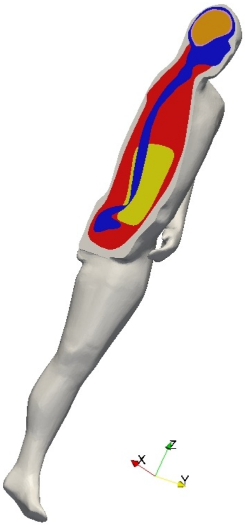


Figure 4. Geometry of the male adult - internal organs

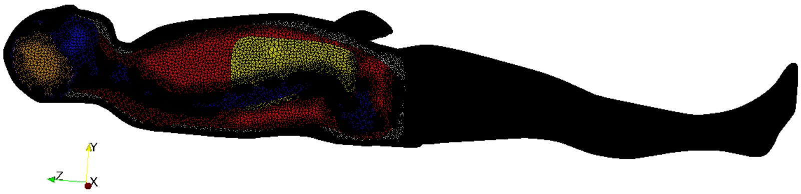


Figure 5. Mesh of 13 million elements - internal organs

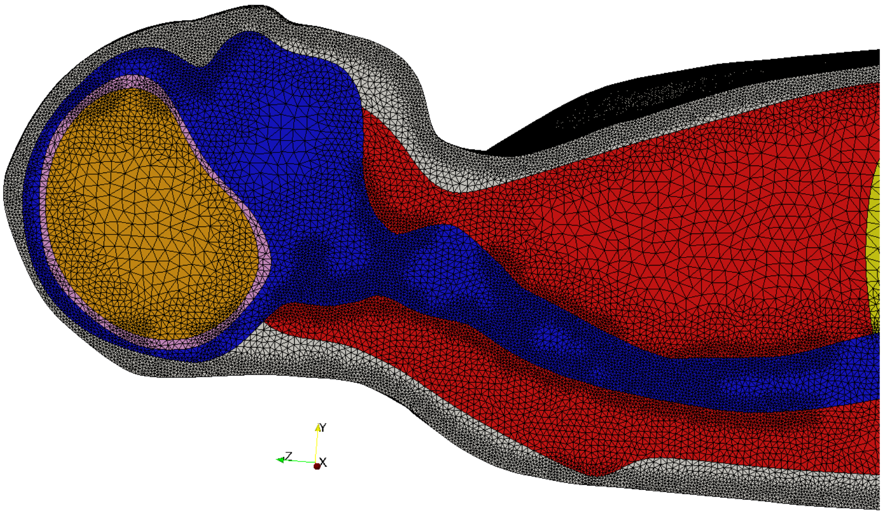


Figure 6. Zoom -Mesh of 13 million elements

The thermophysiological properties of the human tissues were taken from the literature, adapted from [4, 19, 20]. Table 2 shows the tissue thermophysiological properties used in this simulation. The counter current heat exchange coefficients were taken from [4] and are presented in Table 3.

Table 2. Thermophysiological properties of the different tissues

Material properties					
Tissue	Thermal Conductivity $(W/m.^oC)$	Density (kg/m^3)	Specific Heat $(J/kg.^oC)$	Metabolic Heat Generation Rate (W/m^3)	Blood Perfusion Rate $(1/s)$
Blood	0.5	1050	3800	-	-
Eye	0.43	1076	4200	0	0
Lungs	0.39	394	3886	1835	0.0008677
Skin + Fat	0.2	877	2727	170	0.0003146
Cerebrospinal fluid	0.57	1007	380	0	0
Bones	1.16	1300	1590	0	0
Muscle	0.5	1050	3770	528	0.0005355
Viscera	0.55	1100	3350	3160	0.004532
Brain	0.53	1360	2450	12954	0.013124

Table 3. Counter current heat exchange coefficient of the seven sectors of the body

Countercurrent heat exchange coefficient - h_{xc} $(W/^oC)$						
Sector	Head	Trunk+Abdomen	Arm	Hand	Leg	Foot
	0.000	0.000	4.13	0.57	6.2	1.45

The case chosen to measure the performance of the parallel platforms was the case performed in [21]. Only the first 10 steps of time were considered to take the time measures used in the performance analysis. The mesh characteristics and the equation systems derived from these mesh are summarized in Table 4. The total number of non-null terms are all the coefficients of the matrix, that is, the lower, upper, and main diagonal.

Table 4. Mesh data

Malha	Elements	Nodes	Equations	non-zeros
	12,387,727	2,217,562	2,217,562	32,023,448

For parallel execution, the mesh must be subdivided into partitions. Good partitioning needs to meet two key criteria: balancing work between processes, which can be obtained using similar sizes for each partition and minimization of communications between partitions, that can be obtained by reducing internal boundaries. The partitioning of *mesh* using 64 partitions is shown in Figure 7 . The partitioning library*METIS*[met] was used.

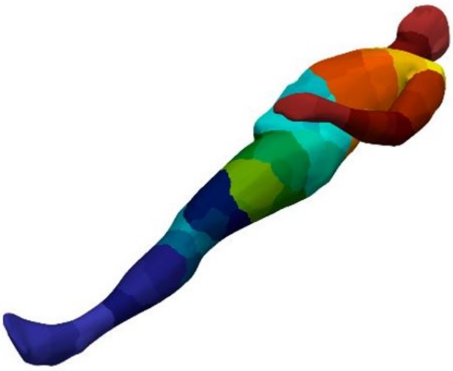


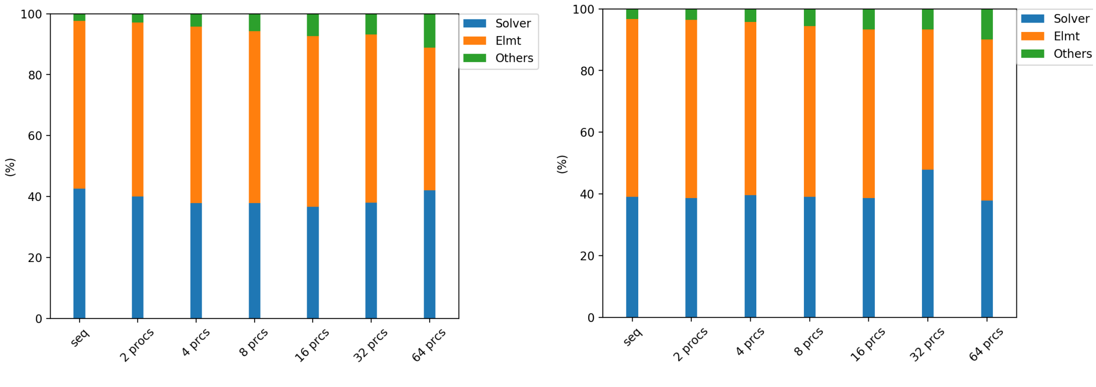
Figure 7. *Mesh* divided into 64 partitions. Each color corresponds to a partition.

The complete analysis of both cases requires 720 time steps. The total time in seconds using 64 processes is summarized in the Table 5. The advantage of the parallelized code is evident when analyzing these results. A conservative estimate of the sequential code time shows that it would take two days to complete the simulation.

Table 5. Total time (seconds)

Malha	Total Time in P1	Total Time in P2
	2151.08	1678.13

The graphs in Figure 8 show the percentage of time spent in each part of the finite element code. The graphs have 3 divisions, the calculation and assembly of the coefficient matrix (*Elmt*), solution of the system of equations (*Solver*) and remainder of the program (*Others*). It is observed that most amount of time is spent in the assembly and resolution stage of the system of equations.



(a) :*P1*

(b) :*P2*

Figure 8. Percentage of time spent in different steps code.

The graphs in Figure 9 show the speedups and the efficiency of the assembly stage of the equation systems. Both platforms demonstrated excellent results and efficiency superior to 1.0, a phenomenon known as super-speedup. Good performance in this step is expected because there is no communication, the only overheads are the overlapping elements of the overlapping technique. Platform *P1* presented the best speedups.

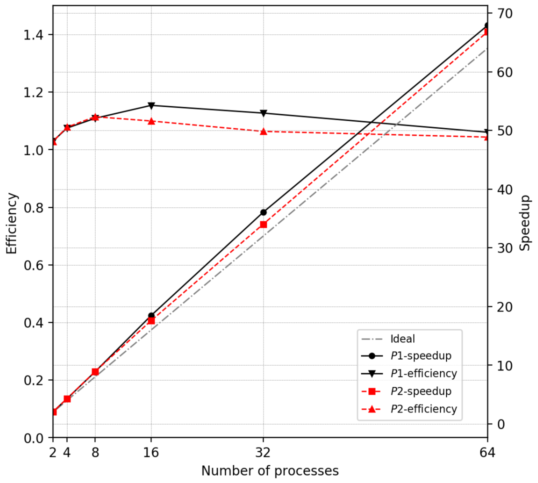


Figure 9. Speedups curves of the assembly stage of the global matrix. Black curves are from the platform *P1*. Red curves are from the platform *P2*.

The graphs in Figure 10 show the speedups and the efficiency of the system resolution step. Both platforms presented super-speedup. The platform *P2* only showed efficiency over 1.0 in 2, 4, 8 and 16 processes, the speedups at 32 were acceptable and in 64 process they were larger than at *P1* reaching almost ideal speedup.

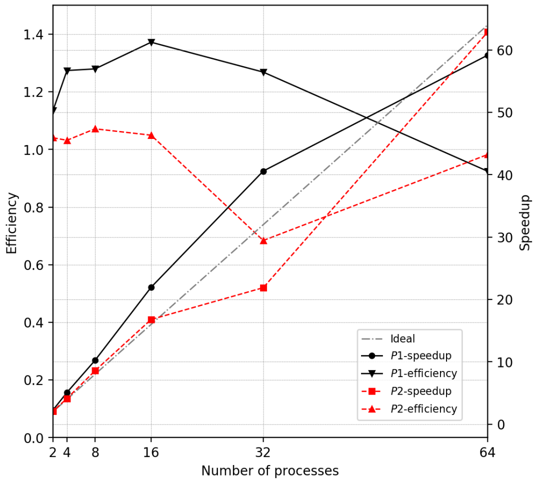


Figure 10. Speedup curves of the resolution step of the system of equations by the method *PCG*. Black curves are from the platform *P1*. Red curves are from the platform *P2*.

The graphs in Figure 11 show the communication time on the *sendrecvs* operation of the slower process, and show that the platform *P1* is superior to the platform *P2* in data transfer rate.

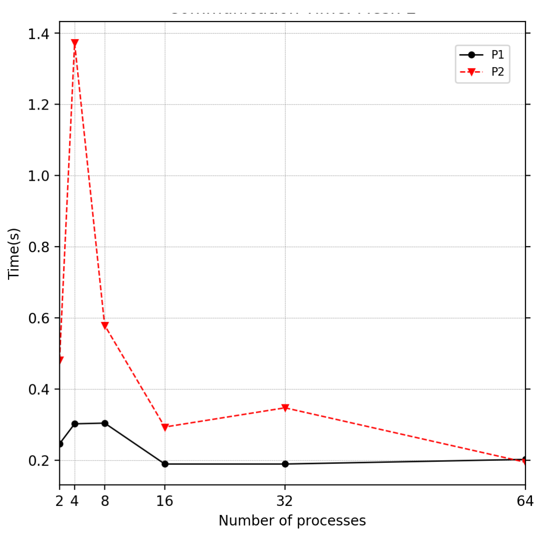


Figure 11. Curves of communication time of the slower process. Black curves are from the platform *P1*. Red curves are from the platform *P2*.

The graphs in Figure 12 show the largest message size that was reported between two partitions via *MPI*. The transmitted byte size range is exactly the region where *infiniBand* is higher, as can be seen from Figure 2, which explains the superior performance of the platform *P1* in the *solver* as seen in Figure 10.

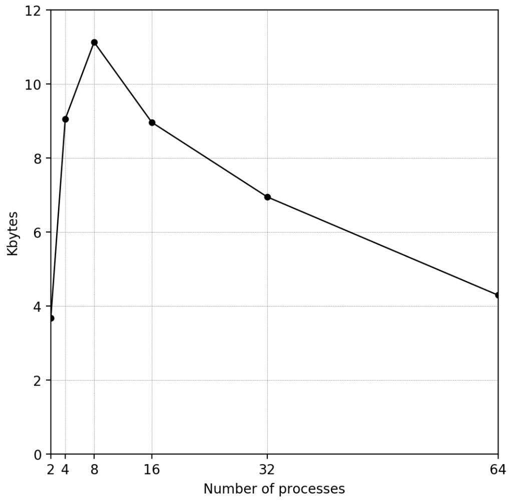


Figure 12. Larger message buffer size between two partitions for each partitioning.

The graphs in Figure 13 show the ratio *P1/P2* of the total computing time of the platforms. The *P1* was always inferior to the *P2* except in 32 processes. This result demonstrates the efficiency of the parallel implementation, and shows that the determining factor of the final performance was not the data transfer rate but the capacity (calculation time) of the local nodes.

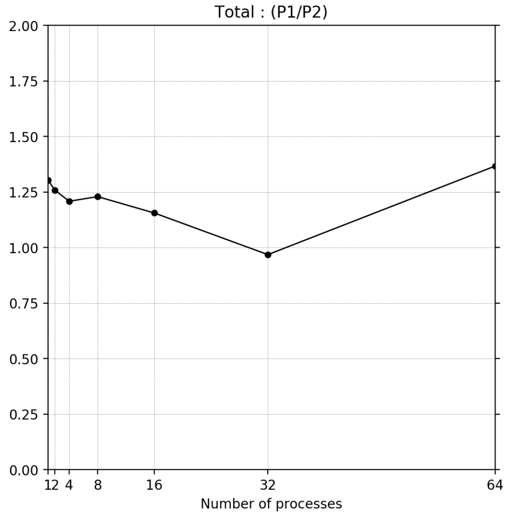


Figure 13. *P1/P2* ratio between the total time.

4 Conclusion

In this paper we presented a parallel implementation of the finite element method designed for hybrid parallel machines. The speedup results of both the assembly stage and the resolution of the system of equations were excellent. The assembly stage of elements always presented super-speedups,and during the solution stage there were no signs of speeddown.

Despite the processor frequency and better network transfer rate of the platform *P1*, the best performance was from the platform *P2*. This may be explained comparing the specifications of both platform. The cache size (*L3*), the frequency of the *RAM* (*DDR4*) and the *QPI* (Quickpath interconnect) are better in *P2*, which demonstrates the importance of these parameters in the implementation performance

Considering the results shown in this work, the implementation using only the *MPI* library were very good in a distributed/shared memory hybrid architecture. Comparing both platforms efficiency the platform *P2*, which has the latter hardware, had better results. This means that this type of implementation can be successfully used in distributed/shared memory platforms.

Acknowledgements

We thank the Coordination for the Improvement of Higher Education Personeel (CAPES) for the financial support.

References

[1] Fiala, D., Lomas, K. J., & Stohrer, M., 1999. A computer model of human thermoregulation for a wide range of environmental conditions: the passive system. *Journal of Applied Physiology*, vol. 87, n. 5, pp. 1957–1972.

[2] Al-Othmani, M., Ghaddar, N., & Ghali, K., 2008. A multi-segmented human bioheat model for transient and asymmetric radiative environments. *Journal of Heat and Mass Transfer*, vol. 51, pp. 5522–5533.

[3] Kingma, B. R., Vosselman, M. J., Frijns, A. J., Steenhoven, A. A. V., & Lichtenbelt, W. D. V. M., 2014. Incorporationg neurophysiological concepts in mathematical thermoregulation models. *International Journal of Biometeorology*, vol. 58, n. 1, pp. 87–99.

[4] Fiala, D., 1998. *Dynamic simulation of human heat transfer and thermal comfort*. Phd thesis, De Montfort University, Leicester, UK.

[5] Xiang, S. H. & Liu, J., 2008. Comprehensive evaluation on the heating capacities of four typical whole body hyperthermia strategies via compartmental model. *International Journal of Heat and Mass Transfer*, vol. 51, pp. 5486–5496.

[6] Laszczyk, J. E. & Nowak, A. J., 2015. Computational modelling of neonates brain cooling. *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 26, n. 2, pp. 571–590.

[7] Zhu, L., Schappeler, T., ordero Tumangday, C., & Rosengart, A. J., 2009. Thermal interactions between blood and tissue. *Advances in Numerical Heat Transfer*, vol. 3, pp. 191–219.

[8] Silva, A. B. C. G., 2012. *Numerical analyses of the temperature distribution in the human brain using the finite element method*. Msc dissertation, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil.

[9] Bhowmik, A., Singh, R., Repaka, R., & Mishra, S., 2013. Conventional and newly developed bioheat transport models in vascularized tissues: A review. *Journal of Thermal Biology*, vol. 38, n. 3, pp. 107–125.

[10] Ribeiro, F. L. B. & Ferreira, I. A., 2007. Parallel implementation of the finite element method using compressed data structures. *Computational Mechanics*, vol. 41, pp. 31–48.

[11] Ribeiro, F. L. B. & Coutinho, A. L. G. A., 2005. Comparison between element, edge and compressed storage schemes for iterative solutions in finite element analyses. *Numerical Methods on Engineering*, vol. 63, pp. 569–588.

[12] Ainsworth Jr, G. O., Ribeiro, F. L. B., & Magluta, C., 2011. A parallel subdomain by subdomain implementation of the implicitly restarted arnoldi/lanczos method. *International Journal of Numerical Methods in Engineering*, vol. 48, pp. 563–577.

[13] Andrade, H. C. C., Silva, A. B. C. G., Ribeiro, F. L. B., & Maghous, S., 2017. A parallel poromechanics fem model. Conference Proceedings in: XXXVIII Iberian Latin American Congress on Computational Methods in Engineering.

[14] Andrade, H. C. C., Silva, A. B. C. G., Batista, V. H. F., & Ribeiro, F., 2013. Parallelization of the finite element method for distributed/shared memory architectures. Conference Proceedings in: XXXIV - Ibero-Latin American Congress on Computational Methods in Engineering.

[15] Saad, Y., 2000. *Iterative Methods for sparse Linear Systems*.

[16] der Vorst H, V., 2003. *Iterative Krylov Methods for Large Linear Systems*.

[17] Freund, R. W. & Nachtigal, N. M., 1994. A new krylov-subspace method for symmetric indefinite linear systems.

[18] Fiala, D., Havenith, G., Bröde, P., & B. Kampmann, G. J., 2012. Utci-fiala multi-node model of human heat transfer and temperature regulation. *International Journal of Biometeorology*, vol. 56, n. 3, pp. 429–441.

[19] Vallez, L. J., Plourde, B. D., & Abraham, J. P., 2016. A new computational thermal model of the whole human body: Applications to patient warming blankets. *Numerical Heat Transfer - Part A*, vol. 69, n. 3, pp. 227–241.

[20] Hasgall, P. A., Gennaro, F. D., Baumgartner, C., Neufeld, E., Gosselin, M. C., Payne, D., Klingenbock, A., & Kuster, N., 2015. It'is database for thermal and electromagnetic parameters of biological tissues. Virtual Population Group.

[21] Silva, A. B. C. G., Wrobel, L. C., Ribeiro, F. L. B., & Nowak, A., 2018. A thermoregulation model for whole body cooling hypothermia. *Journal of Thermal Biology*, vol. 78, pp. 122–130.

[met] Metis - serial graph partitioning and fill-reducing matrix ordering. http://glaros.dtc.umn.edu/gkhome/metis/metis/overview. Accessed: 2019-01-30.