

Efficient Application of Parallel Processing with Standard Tools for Electromagnetic Transients Simulation

H. M. Barros, A. Castro, R. N. Fontoura Filho, M. Groetaers dos Santos, C. F. Teodósio Soares

Abstract-- This paper presents the improvement in overall computational efficiency for an Electromagnetic Transients (EMT) simulation assisted by user's defined network partitioning.

For this purpose, a standard EMT simulation program takes the advantage of special designed routines for providing a user defined library of time-delayed components assisted by the MS-Windows operating system resources for controlling of data exchange among different computational processes.

Instead of using a generic co-simulation interface, e. g. same way as MATLAB/SIMULINK engine cooperates with external tools, the goal here is to adopt an efficient Inter-Processes Communication mechanism (IPC) for supporting the routines that controls data transfer between clients and server processes connected by conduits or pipelines. This approach allows the simultaneous execution of each individual subsystem derived from the entire network, without changing the original EMT program algorithm and also minimizes the communication burden.

This development, called "assisted parallel simulation" together with a set of well-designed frequency dependent multi-port equivalents, has allowed the Electric System National Operator - ONS to execute off-line electromagnetic transient studies for a large power system that includes controller's modeling of two HVDC links and one Back-to-Back connection in a very efficient way.

Keywords: Electromagnetic transient program, Inter-process communication mechanism, parallel simulation, cooperative simulation, user's defined external model, interfacing for off-line EMT simulators.

I. INTRODUCTION

Transient simulation of large electrical Power Systems demands massive computational resources. Synthetic equivalents that reproduce the frequency response for parts of the system are normally adopted to reduce the computational burden. However, the support of parallel processing is

mandatory to speed up the simulation time when power electronics and controls representation are included [1], [2] and also in some electric power applications that demands special modeling [3].

The Electromagnetic Transients (EMT) simulation standards adopted by the industry: ATP, EMTP-RV, PSCAD/EMTDC and MATLAB/Simulink; do not take full advantage of the current inexpensive multi-core computer platforms. However, these off-line simulation tools can explore the decomposition of a very large or complex network in small subsystems which can be solved simultaneously, almost in the same way as for the real-time simulation. The main idea of subdividing the entire network in smaller subsystems connected by time-delayed elements (transmission lines, stub lines or modified models that include a time delay) remains valid. The basic requirement for exploring the decomposition of a complex network into a set of small systems corresponds to the availability of "hooks" in the computational algorithm that allow the inclusion of user-defined mathematical models. Fortunately, this requirement is met by all the EMT simulators in reference.

As the electrical network to be studied increases in dimension and complexity, the application of some type of parallelization for numerical solution of the EMT problem becomes desirable. Currently a number of factors greatly favors the holding of several approaches for improving computing performance of the EMT programs, among which are: availability of low-cost computing architectures with multiple processors; availability of communication networks with high data throughput and low latency; support for inter-process communication mechanisms included in the Operating Systems (OS). Unfortunately, full utilization of these facilities does not reach the most of specialists in EMT studies. In fact, a very high programming effort and a deep knowledge of the OS internals is required in order to deal with a concurrent processing not characterized as "embarrassing parallelism" quite common in optimization and/or statistical studies. Depending upon the programming requirements in order to access the user's modeling capability of a particular simulation tool, it can be also necessary to request developer's support for seeing undocumented instructions.. Even if the simulation tool is well documented about the external model integration, such a facility can require programming knowledge that goes beyond the expertise of an average user.

In order to speed-up the corresponding simulation time of

H. M. Barros (henildo@ons.org.br), A. Castro (acastro@ons.org.br), R. N. Fontoura Filho (rfontoura@ons.org.br), M. Groetaers dos Santos (marcels@ons.org.br) are with Electric System National Operator - ONS, Rio de Janeiro, Brazil.

C. F. Teodósio Soares is with Federal University of Rio de Janeiro (carlosfts@gmail.com).

Paper submitted to the International Conference on Power Systems Transients (IPST2013) in Vancouver, Canada July 18-20, 2013.

the EMT program, the developers can incorporate some modifications to the solution algorithms that can be classified in two groups: The first corresponds to replace the sequential matrix and vector operations by its parallel realization. The second corresponds to incorporate a partitioning scheme so that to decouple, either in definitive way (by time-delayed elements) or in temporary way (diakoptics techniques [4]), the entire set of network equations in small tasks. Of course, issues such as the synchronizing of calculation and the load balance it need to be treated by the solution algorithm without the help of the user. Consequently, these types of improvements require deep modifications to be implemented inside the core of the EMT simulators used for a long time in industry. This difficulty may explain, at least in part, reason why the so well established tools like PSCAD/EMTDC, ATP and the EMTP-RV do not take full advantage of the current hardware architectures and OS capabilities.

Fortunately, from the point of view of users, there is still the possibility to explore resource provided by the operating system and that is associated with the control of the information exchange between different computational processes that run on the same computer or under computers in the same network. This strategy corresponds to the use of inter-process communication mechanism that can control in a very efficient the data exchanging between processes from different EMT simulators or even between related processes from a unique EMT Simulator.

II. EMT PROGRAM CHARACTERISTICS AND ALGORITHM ESSENTIALS

As explained before, the EMT programs for analysis in the time domain as the PSCAD/EMTDC, ATP and the EMTP-RV, the latter more modern design, have not been updated by its developers for a more efficient use of the modern hardware platforms and OS resources. In general the programming efforts were focused to decrease the computer burden associated to the conductance matrix updating and also to mitigate or avoid the appearance of numerical oscillations. Regarding the ATP and PSCAD/EMTDC, whose origin of source codes dating back to the early 70's, some performance improvement is related to the reordering of the solution matrix for repositioning of the nodes where switches and variable conductance branches are connected. Such action aims to accelerate the nodal conductance matrix refactoring for topological changes that can happen over the simulation time.

Additionally, as the interpolation feature was included in the PSCAD/EMTDC's programming, as way of correcting zero-crossing instant during the switch opening time, there was some concern to limit the conductance matrix refactoring to small subsystems previously identified by the algorithm. Alternatively, the EMTP-RV mitigates the numerical oscillations that result from switching operations by changing the method of numerical integration without modifying the conductance matrix.

Anyway, strictly speaking about a parallel-type solution,

none of the EMT "standards" embeds in its computation algorithm or even in its library of electrical components the essential elements for exploring the most widespread alternative: topological subdivision through transmission lines or component representation by TLM (Transmission Line Modeling) or "stub-line". However, the EMT tools provide their users some way to embed external models for simulating of electrical components that can provide such improvement.

III. COOPERATIVE AND PARALLEL SIMULATION WITH EMT PROGRAM

Independent of the EMT "standard" adopted, it will be up to the user to provide specific routines for communication between complementary computational processes associated with either distinct simulators or parallel solution in a single simulator. Obviously, this task goes beyond the knowledge of an expert in electromagnetic transient studies that does not have a deep knowledge of OS resources and high level programming expertise. This difficulty represents a first obstacle to be overcome, aiming at the parallel processing with a single tool or the cooperative simulation with different tools.

In spite of this difficulty all the EMT "standards" meet a basic requirement for implementing of parallelism: possibility of interaction with other tools at run time; which is supported by their specific interfaces for adding user-defined external models. As a consequence, this feature can be applied with higher or lower effort of programming in order to enable a hybrid type simulation for what can be desirable, e.g.: access to a model available in the library of another EMT tool, execution of a tailor made routine for controlling of data communication with another process, realization of the both tasks listed before.

Philosophically there are two alternatives used by the different EMT simulators for including of user-defined external models to its solution algorithm. These alternatives will define how a specialized routine for data communication between computational processes will exchange the data information.

The solution algorithm of ATP program adopts the compensation method for including of mathematical models of equipment developed by the user. In this method, two network solutions for the nodal matrix equation system are performed in each integration time step, the last one including the external model effect. In between them one combined solution of the reduced equivalent network together with the external model by direct or iterative calculation is realized. Despite of the greater amount of calculations inherent in this method, it supports the partitioning of the network with or without using of "time-delayed" components for effect of parallel solution.

The solution algorithm of the program PSCAD/EMTDC allow including external models of electrical components directly into the nodal conductance matrix if they can be reduced to discrete time equivalent representation of type Norton or Thevenin. If a non-linear relationship between the voltage and current at the terminals of the external component

exists, it may be necessary to take the advantage of a predictor-corrector algorithm to provide the instantaneous equivalent. In this scheme, the partitioning of the network must be accomplished through "time-delayed" components for effect of parallel solution.

On the other hand, the EMTP-RV computational engine adopts a sophisticated way for dealing with the user defined models. In order to couple any external model directly and smoothly with its core solver, this simulation tool adopts an interface based upon Dynamic Link Library (DLL) function. This approach is not so simple to describe as the previous ones but can be roughly visualized as a set of external devices (DLL models) that interchange specific information with the host through a numerical bus. Despite to be very powerful from the point of view of model portability, the DLL interface utilization demands much more programming effort and expertise.

Table I tries to summarize the interfaces described in the previous paragraphs. Proper use of them together with "time-delayed" components can enable cooperative execution between different EMT simulators and partitioned (parallel) execution of different processes created from a single EMT simulator, without changing its original algorithm. However, it is up to the user to setup the number of inter-dependent subsystems suitable for parallel processing and also the size and complexity of such subsystems for purpose of tuning the computational load balancing.

IV. IPC MECHANISMS INCLUDED IN THE MS-WINDOWS OS

The MS-Windows operating system provides a number of IPC mechanisms for supporting data sharing among different processes [5], each one with its own advantages and disadvantages. However, not all the mechanisms described in the reference before are suitable for improving the computational performance when a partitioned or cooperative simulation approach is been used. Disregard the peculiarities of each mechanism IPC can lead the developer of computer models to conclude that the overhead of time introduced by communication between different simulators would invalidate subdividing the solution between two specialized tools cooperatively [6].

For example, according to the documentation of the MATLAB/Simulink program, its built-in MATLAB Engine library performs the communication – in the MS-Windows environment - through an IPC mechanism known as the Component Object Model (COM). The main advantage of COM mechanism is the ability to perform the exchange of data with support for a wide range of applications, due to its versatility [5]. Take into account this characteristic, the COM mechanism is quite suitable for the MATLAB Engine library, since its goal is exactly to provide, to the programmer, a communication interface of easy access by any application under developing. On the other hand, for the provision of such generality, the COM mechanism should include additional data to the messages exchanged between the processes and should

also performs data manipulation operations that overload the simulation run-time. Hence, this mechanism is not tailored for integration of applications that require great computing performance.

TABLE I

EMT SIMULATION TOOLS INTERFACE WITH EXTERNAL COMPONENTS

EMT tool	User defined component access	Solution type	User defined component codification
ATP	Type-94 element	Indirect coupling by compensation method (separated algorithm section)	FORTRAN 77 and ANSIC plus a complement in ATP proprietary language (MODELS)
PSCAD/EMTDC	Norton or Thevenin type discrete element	Direct coupling in nodal conductance matrix	FORTRAN-like language wrapped by a special template
EMTP-RV	DLL programming interface	Direct coupling of model equations and sharing of solver internals	FORTRAN 95 and C++ languages wrapped by a special template

Among the IPC mechanisms available in MS-Windows: *Clipboard*, *COM (Component Object Model)*, *Data Copy*, *DDE (Dynamic Data Exchange)*, *File Mapping*, *Mailslot*, *Pipes*, *RPC (Remote Procedure Call)* and *Windows Sockets*. Taking into account the evaluation of throughput comparison and additional criteria presented in technical literature [7], [8] as: limitation of affinity with other processes, read and write capability, limitation in the number of processes allowed to communicate, built-in data synchronization capability; the following ones present the best requirements for partitioned or cooperative numerical solution of EMT problem.

File Mapping (Named Shared Memory): The programmer must provide synchronization between the processes. It can be used only between processes on a local computer.

Named pipes (FIFOs - first-in, first-out): Named pipes provide a simple programming interface for transferring data between two processes. The pipe automatically synchronizes the data exchange between two processes. Data can be transmitted through a named pipe as either a stream of bytes or as a stream of messages.

Windows Sockets: Has the advantage of simplicity and portability for heterogeneous OS environments.

V. NAMED PIPES (FIFOs) GENERAL CHARACTERIZATION AND PERFORMANCE

Among the other available mechanisms the Named pipes was considered the best alternative for the feasibility of parallel or multi-tool simulation, in order to be simple and flexible enough so that the task of communication among the processes do not represent too much burden to computing performance [9]. A pipe is a kind of communication pipeline between two processes, through which both can transfer data among themselves. There are two types of pipes available:

Anonymous Pipes and Named Pipes. The Anonymous Pipes are employed exclusively for establishing communication between two processes related to each other, for example, a communication between a parent process and the child process, or else between child processes of the same father. The Named Pipes can be used in the communication between two processes that have no relation to each other. In this case, the pipe must receive an identifying name, so that both the processes involved can access it.

The pipes mechanism can carry out the transfer of data between processes in two different formats: message or a stream of bytes. Communication through messages has the drawback of additional information that must be encapsulated in the message along with the data. Otherwise, as a stream of bytes the data is transmitted through the pipe in its raw form (bytes), without any additional information. In this last type of data transfer, both processes should know how the data is arranged in the stream of bytes and how many bytes are to be read or written. Additionally, as this type of communication is quite simple, it makes it faster than others [9].

To establish the communication between two processes via pipes, we should have a process performing the role of the server and the other the role of client. This distinction is made because the process responsible for creating the pipe is necessarily the server. From the client point of view, it will be enough to connect itself to an existing pipe. As for the direction of the data flow, we can have three different types of pipe: one where the server only sends data to the client, one where the server only reads the data sent by the client and the one that is allowed to two-way communication, where both-server and client-can write and read data.

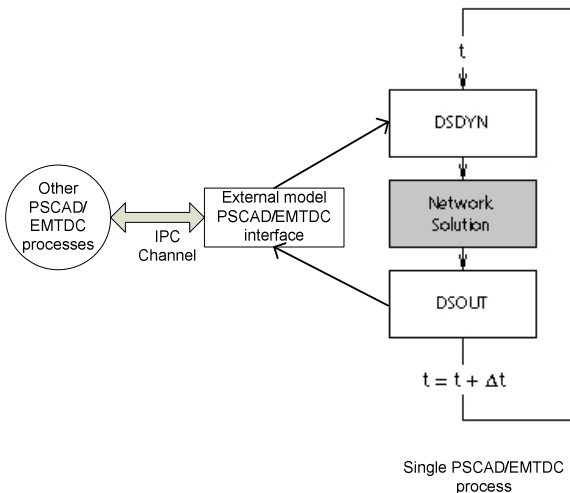


Fig. 1. Proposed approach applied to PSCAD/EMTDC program main structure.

VI. CASE STUDY

The following examples illustrate how much improvement in overall computational efficiency can be obtained when it makes use of the “assisted parallel simulation” for solving very large and complex systems. Both simulations were performed

in PSCAD/EMTDC program with assistance of a specialized library of time-delayed components including data exchanging control by named pipe IPC mechanism.

Their associated diagrams show how these networks were manually partitioned for comparison of the results between the conventional computation with only one PSCAD/EMTP process and the assisted parallel simulation with four and five PSCAD/EMTP processes for systems A and B, respectively. All computations were performed by using a PC based on INTEL i5-2400 – 3,10GHz quad-core CPU with 8GB RAM.

A. A double 500kV - 230kV back-to-back connection

The first benchmark network includes 30 busbar, 41 series elements, 03 synchronous generators, 02 complete model of SVC and 02 back-to-back connections with the corresponding controllers. This configuration corresponds to the preliminary connection of the Rio Madeira HVDC transmission system with the north part of the interconnected AC transmission system in Brazil.

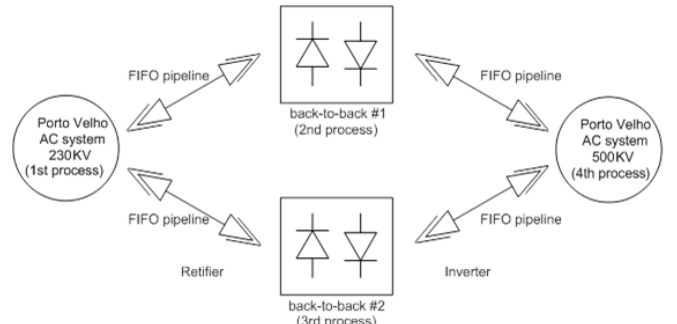


Fig. 2. Schematic of the network for the example A.

Following figures show some waveforms from the PSCAD/EMTDC simulation of a three-phase to ground fault applied to the 230kV AC system. Results for the conventional and parallel simulation are in red lines and blue lines, respectively. As it can be verified there is almost no difference caused by insertion of the “stub-lines” necessary for separation of the two back-to-back connections from the 500kV and 230kV AC systems.

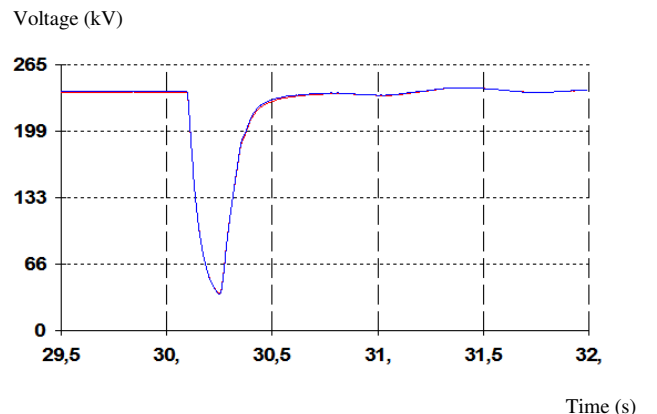


Fig. 3. RMS AC voltage (kV) at the coletora Porto Velho substation during a three-phase to ground fault.

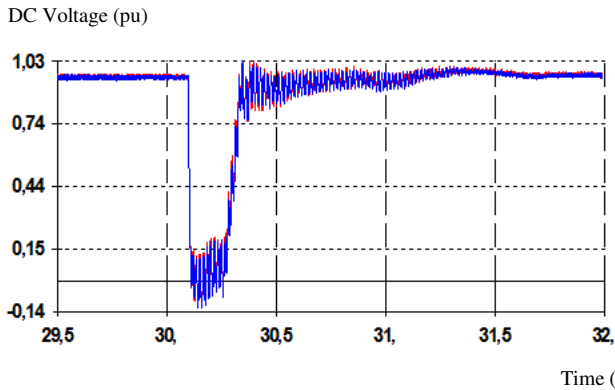


Fig. 4. DC voltage (pu) at the coletora Porto Velho substation during a three-phase to ground fault – inverter side.

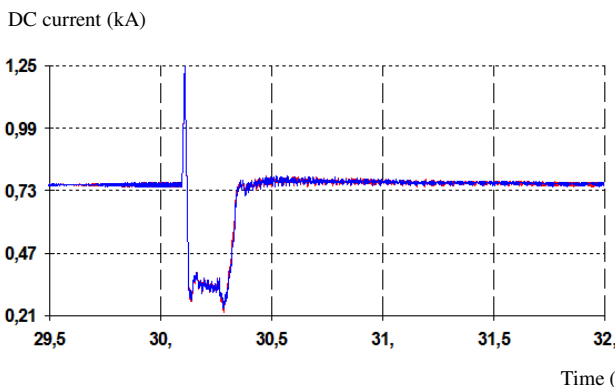


Fig. 5. DC current (kA) at the coletora Porto Velho substation during a three-phase to ground fault – inverter side.

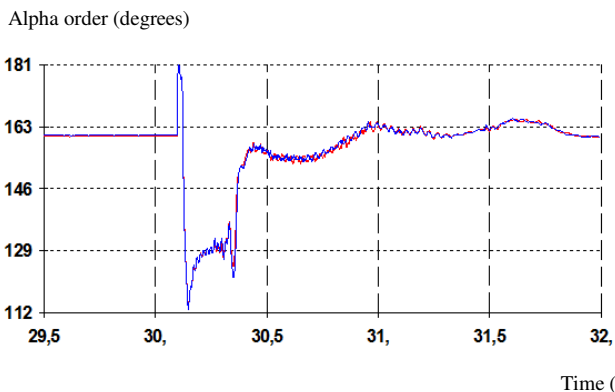


Fig. 6. Alpha order (degrees) during a three-phase to ground fault – inverter side.

Figures 7 and 8 show two screen shots of the MS-Windows task manager showing how the four processes were distributed during the course of simulation. Processes one to four were forced of execution in just one CPU. As it can be verified, the 1st process of this set is the largest consumer of its own CPU resources when comparing with the others three, what indicates that a better partitioning scheme may reduce the total time of simulation.

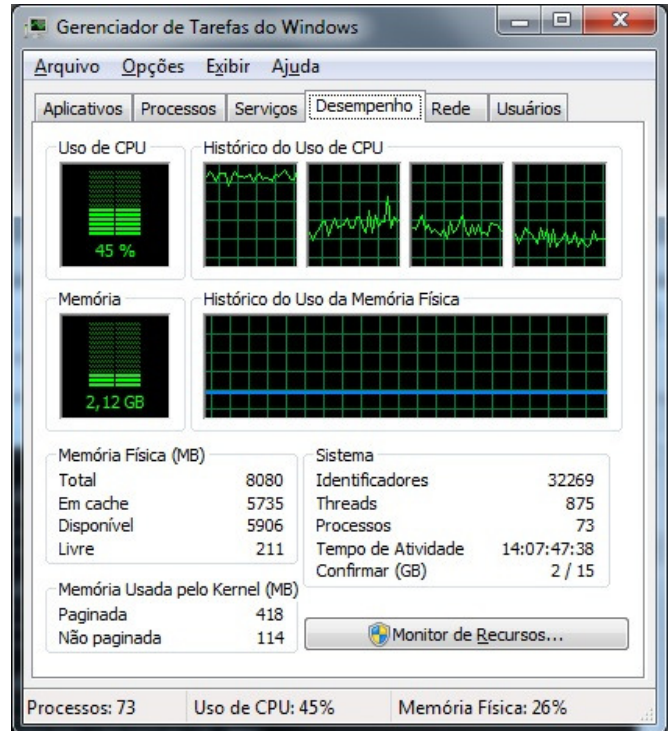


Fig. 7. Screenshot of the MS-Windows task manager for computation during the time interval of plots – individual CPU performance.

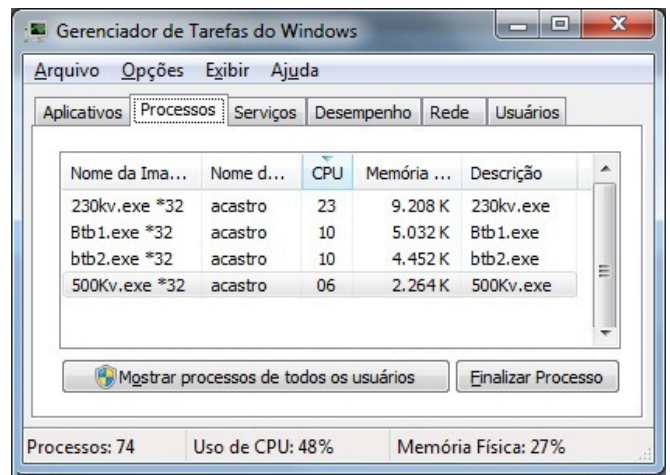


Fig. 8. Screenshot of the MS-Windows task manager for computation during the time interval of plots – processes statistics.

Table II shows the simulation time statistics presented by the PSCAD/EMTDC program where can be verified the improvement proportioned by adopting the user's "assisted parallel solution". The simulation of this real network by using a quad-core machine it was accelerated by a factor of approximately 4 times without optimization of the load balance.

TABLE II
PSCAD/EMTDC SIMULATION TIME STATISTICS

Conventional simulation (single process)	Parallel simulation (four processes)
Time Summary	Time Summary
Start Date: 02-21-2013	Start Date: 02-20-2013
Start Time: 08:45:07	Start Time: 16:36:38
Stop Time: 10:12:10	Stop Time: 16:58:24
Total CPU Time: 5223.55 sec.	Total CPU Time: 1306.17 sec.

B. A double HVDC link plus a 500kV - 230kV back-to-back connection

The second benchmark network has approximately 630 three-phase nodes of size, as the PSCAD/EMTDC program output report informs. In addition to the previous network equipment it also includes additional generation substations and 02 HVDC links for transmission of electric power from the north part of Brazil to the consumer centers in southeast part. It will be the final configuration of the Rio Madeira HVDC transmission system to be interconnected to the current AC transmission system in Brazil.

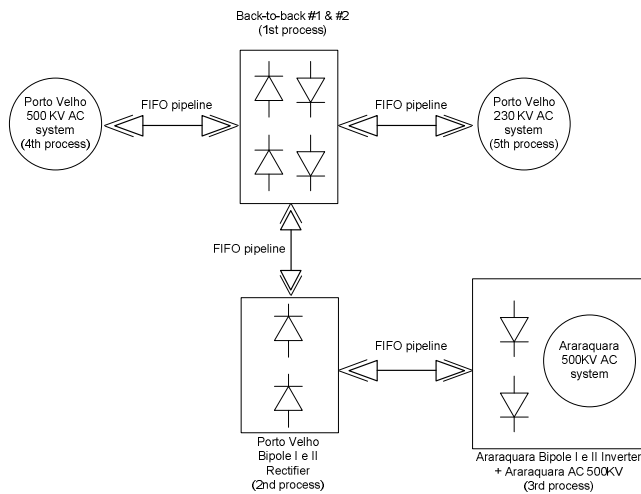


Fig. 9. Schematic of the network for the example B.

Following figures show some waveforms from the PSCAD/EMTDC simulation of a three-phase to ground fault applied to the 500kV AC system connected to inverter substation. Results for the conventional and parallel simulation are in red lines and blue lines, respectively. As it can be verified there is almost no difference caused by insertion of the “stub-lines” necessary for separation 500kV side of the two back-to-back connections (rectifier) from the 500kV side (rectifier) of the two HVDC links.

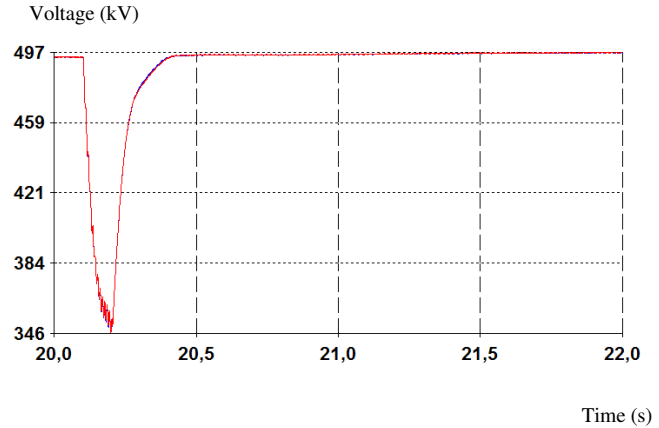


Fig. 10. RMS AC voltage (kV) at the Araraquara substation (inverter) during a three-phase to ground fault.

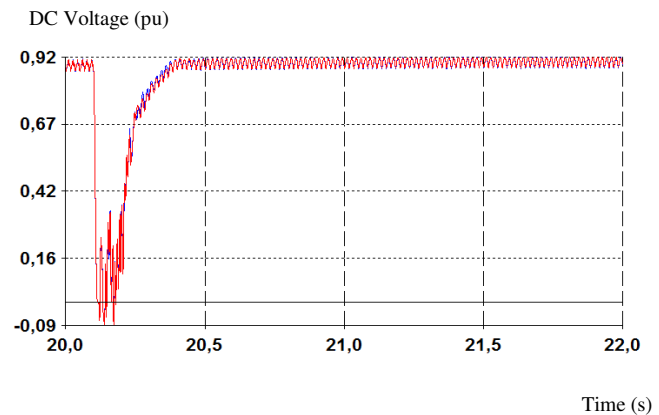


Fig.11. DC voltage (pu) at the Araraquara substation (inverter) during a three-phase to ground fault.

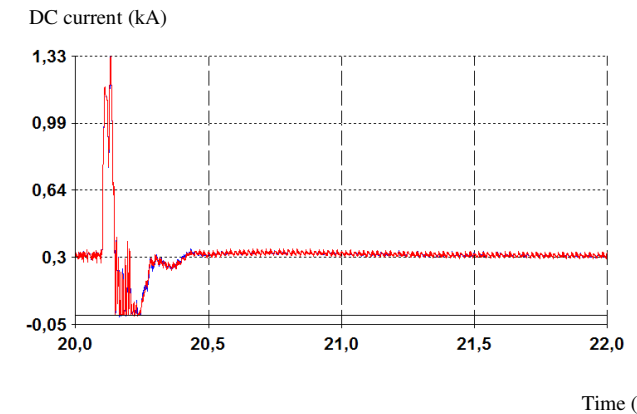


Fig.12. DC current (kA) at the Araraquara substation (inverter) during a three-phase to ground fault.

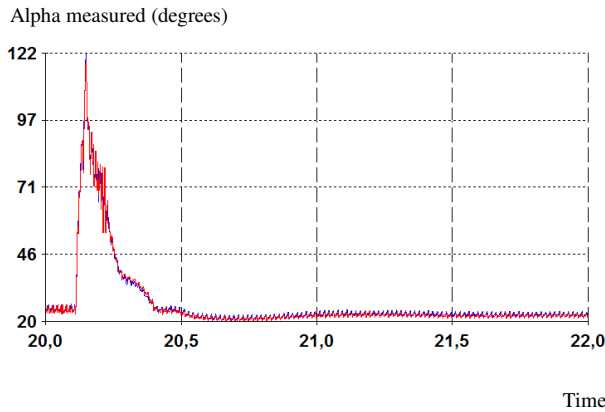


Fig. 13. Alpha measured (degrees) at the Porto Velho substation (rectifier) during a three-phase to ground fault.

Figures 14 and 15 show two screen shots of the MS-Windows task manager showing how the five processes were distributed during the course of simulation. Processes one to four were forced of execution in just one CPU and process five it was free for the OS task scheduler takes care of it. As it can be verified, the 1st process of this set is again the largest consumer of its own CPU resources when comparing with the others four, what indicates that a better partitioning scheme may reduce the total time of simulation.

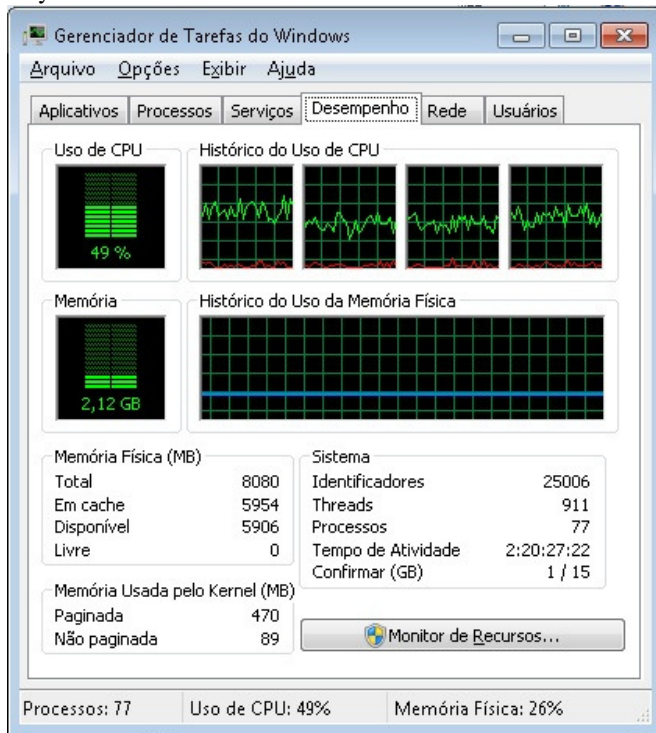


Fig. 14. Screenshot of the MS-Windows task manager for computation during the time interval of plots – individual CPU performance.

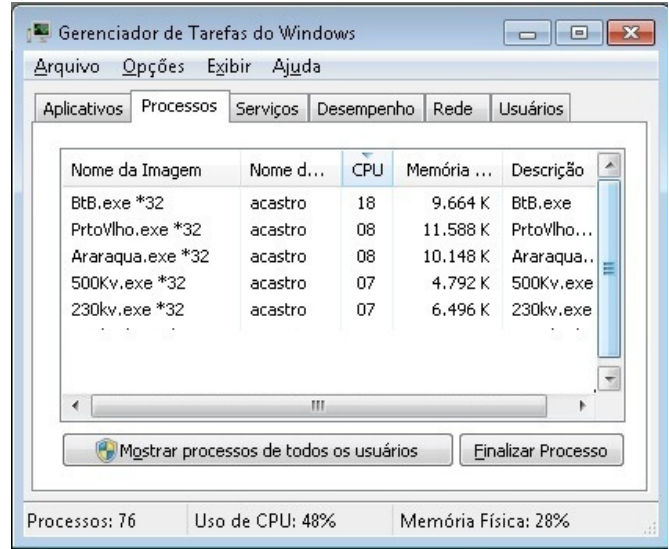


Fig. 15. Screenshot of the MS-Windows task manager for computation during the time interval of plots – processes statistics.

The table III shows the simulation time statistics presented by the PSCAD/EMTDC program where can be verified the improvement proportioned by adopting the user's "assisted parallel solution". The simulation of this real network by using a quad-core machine it was accelerated by a factor of approximately 20 times, what is an excellent result.

TABLE III
PSCAD/EMTDC SIMULATION TIME STATISTICS

Conventional simulation (single process)	Parallel simulation (five processes)
Time Summary	Time Summary
Start Time: 14:10:43	Start Time: 11:23:29
Stop Time: 19:57:15	Stop Time: 11:40:46
Total CPU Time: 20792 sec.	Total CPU Time: 1037 sec.

Of course, additional verifications need to be done in order improve some points but, for effect of a practical utilization during the off-line studies, we evaluate EMT assisted parallel simulation as valuable alternative to conventional utilization of PSCAD/EMTDC.

VII. CONCLUSIONS

The lack of native support for parallel solution of large systems simulated on standards of type "EMT" (PSCAD/EMTDC, ATP, EMTP-RV and MATLAB/Simulink) brings as a consequence considerable under-utilization of the processing power provided by the personal computing platforms currently available.

However, once such programs have specialized interfaces for adding external user-defined models, it was possible to develop a library of elements that led to the solution of the original problem into smaller parts or decoupled subsystems, corresponding to a set of concurrent processes interconnected by data conduits. For the implementation of this solution, models of elements whose mathematical description allows

topological subdivision of original electrical network were employed (constant parameter transmission line, Transmission Line Modeling - TLM - component and stub-line component) in association with routine for data exchange control based on the named pipes (FIFOs) IPC mechanism.

The improvement in overall computational efficiency for both presented examples, show the effectiveness of the so called "assisted parallel simulation".

The results obtained show that parallelizing EMT simulations is not limited only to developers of these specialized tools. From the efficient use of the IPC mechanisms included into OS resources and by the proper utilization of the external model interfaces any advanced user of EMT tool has the potential to enjoy the benefits that came from speeding-up of simulation time.

VIII. FUTURE INVESTIGATIONS

As the results presented in this work are restricted to EMT processes that are being executed on a single multi-core machine, the next step of this investigation intends to evaluate the improvements in simulation time that can be archived when applying the same solution for a cluster of PCs connected in LAN. For this application it can be necessary to review the IPC mechanism that best suits the computational architecture whose inter-process communication has to be done over the local network instead of PC data bus.

Another important improvement can be achieved by generalization of the mathematical model developed for the time-delayed elements in order to allow transient simulation of each subsystem with a different time step from the others. So, a subsystem including a component that demand a very small time step, as for example a Multiple Module or Multilevel Converter (MMC) would not need to impose such a requirement for the rest of the network solution.

Finally, it is important to note that this same solution is under use for supporting a hybrid simulation environment where either different time-frame processes or mixed domains processes need to exchange post processed information by using a proper transfer protocol. The preliminary results show that the benefits, related to the reduction of communication burden, are most or least significant depending on the type of interaction protocol to be used [11].

IX. REFERENCES

[1] C. Dufour, V. Jalili-Marandi, J. Bélanger, L. Snider. "Power System Simulation Algorithms for Parallel Computer Architectures". IEEE Power and Energy Society General Meeting, 22-26 July 2012.

[2] H. Pourrez, A. Gole, S. Filizadeh, P. Graham. "Parallelizing Power Systems Simulation for Multi-core Clusters: Design for an SME". Proceedings of the 23rd international conference on High Performance Computing Systems and Applications (HPCS'09), pp. 165-179, 2010.

[3] F. M. Uriarte. "Multicore Simulation of an Ungrounded Power System". IET Electrical Systems in Transportation, Vol. 1, Issue 1, pp. 31-40, 2011.

[4] F. A. Moreira, L. C. Zanetta. "Solution Method Based on the Theory of Diakoptics for an Efficient Time-Domain Transients Simulation of

HVDC Converters". IEEE/PES Transmission & Distribution Conference & Exposition: Latin America, 2004.

[5] Microsoft online documentation available at: <http://msdn2.microsoft.com/en-us/library/aa365574.aspx>.

[6] J. Mahseredjian, G. Benmouyal, X. Lombard, M. Zouiti, B. Bressac, L. Gérin-Lajoie. "A link between EMTP and MATLAB for user-defined modeling". IEEE Transactions on Power Delivery, pp. 667-674, vol. 13, No. 2, April 1998.

[7] P. K. Immich, R. S. Bhagavatula, R. Pendse. "Performance Analysis of five Interprocess Communication Mechanisms across UNIX Operating System". The journal of Systems and Software. pp. 27-43, issue 68, 2003.

[8] K. Wright, K. Gopalan, H. Kang. "Performance Analysis of Various Mechanisms for Inter-process Communication". Operating Systems and Networks Lab, Dept. of Computer Science, Binghamton University. 2007.

[9] H. M. Barros, C. F. T. Soares, F. D. de Jesus. "ATP and Matlab/SIMULINK Cooperative Simulation Applied to the Electrical Studies of Electromagnetic Transients Due to Switching of Power Equipments" (in Portuguese). Proceedings of the XIX SNPTEE, Rio de Janeiro, October 2007.

[10] IEEE Taskforce on Interfacing Techniques for Simulation Tools. "Techniques for Interfacing Electromagnetic Transient Simulation Programs with General Mathematical Tools". IEEE Transactions on Power Delivery, pp. 2610-2622, Vol. 23, No. 4, October 2008.

[11] IEEE Taskforce on Interfacing Techniques for Simulation Tools. "Interfacing Techniques for Transient Stability and Electromagnetic Transient Programs". IEEE Transactions on Power Delivery, pp. 2385-2395, Vol. 24, No. 4, October 2009.