# Massively Parallel Real-Time Simulation of Very-Large-Scale Power Systems

P. Le-Huy, M. Woodacre, S. Guérette, É. Lemieux

*Abstract*— **System reductions or network equivalents are frequently used to reduce the scale of simulated power systems in electromagnetic transient simulation tools, and even more so in real-time software. However, complex hardware-in-the-loop studies involving multiple devices under test or focusing on wide-area phenomena such as geomagnetically-induced currents or wide-spread ferroresonance require full modeling of large portions of the power system. The main technical challenges in very-large-scale real-time electromagnetic transient simulation reside in both hardware and software as both have to be able to support a very high degree of parallelism.**

**This paper presents how the Hypersim real-time simulator was adapted to enable massively parallel electromagnetic transient simulations. The enabling features of the hardware platform, an SGI UV300 supercomputer, are reviewed in light of Hypersim's needs. The required software modifications to achieve high levels of parallelism are then presented and discussed.**

**A very-large-scale power system counting more than 16,000 electrical nodes, using as basic construction block a representation of Hydro-Québec's power system, is used to illustrate the massively-parallel approach. Results, from both data acquisition services and IOs, are provided for a 200-core 30-μs real-time simulation. Various mapping of this very-large-scale power system on the UV300 are done to analyze performances in order to provide insights about the cost of massive parallelization.**

*Keywords*: **Electromagnetic transient, simulation, real-time, very-large-scale AC system, parallel processing, massively parallel, simulation software, high-performance computing.**

## I. Introduction

WIDELY accepted electromagnetic transient (EMT) hardware-in-the-loop (HIL) studies are traditionally limited in scope due to computational burden and the availability of real-time (RT) simulation hardware. However, there is a growing need for more complex HIL studies that take into account:

- Wider areas to assess the impact of certain phenomena (e.g. geomagnetically-induced currents, wide-area control schemes, wide-spread ferroresonance);
- More equipment that could trigger control system or power electronic harmonic interaction issues with the device under test.

Based solely on the scope of these studies, a transient stability (TS) software package would have been the intuitive solution but this is not a valid option in this case as the investigated phenomena are not properly represented in TS simulations or non-trivial to analyze due to the lower frequency bandwidth of the TS software. Furthermore, most HIL studies require EMT modeling. From this, it can be observed that the application domain of both EMT and TS simulations is not as clearly defined as it once was.

Recent advances in computer hardware and EMT simulation software facilitate large-scope EMT studies with detailed and complex modeling in RT. The current paper presents a massively parallel (MP) RT simulation hardware platform for the EMT simulation of very-large-scale (VLS) power systems. Based on traditional multi-core CPUs, in contrast to specialized hardware such as GPUs [1][2] and FPGAs [3][4], this platform is capable of RT operation with several hundred cores with all the required services for industrial grade studies such as signal acquisition, automatic testing, data logging and report generation [5]. This environment is also very flexible and allows for quick reconfiguration and customization, which is not a strong suit of application-specific solutions based on the previously mentioned specialized hardware. The first part of the paper presents the simulation hardware platform, with emphasis on the processor topology, communication engine and memory-access co-processor. The second part explains how this platform's power was harnessed for EMT simulation with Hydro-Québec's Hypersim RT Simulator [6][7]. Using several hundred computation cores in a single simulation is not a trivial task: required modifications to Hypersim software are presented and discussed. The final section of the paper illustrates the power of the MP RT simulator with a test case of a VLS power transmission network. The network composition is detailed for the readers to properly evaluate the computational burden. MP RT performances are then analyzed in terms of parallelization overhead and speedup. Finally, concluding remarks are given and future work is discussed.

## II. SGI UV300: MP Simulation Hardware Platform

For an RT EMT simulator, the most critical factor to consider in a hardware platform is not its processing power but rather its communication engine/architecture: the processing power of each processing unit will limit the size of the most complex task it will be able to solve but its communication engine performance will limit the number of computing units working together. The effect of the

Philippe Le-Huy, Sylvain Guérette and Éric Lemieux are Research Engineers with IREQ, Hydro-Québec's research institute (Power Systems and Mathematics unit), 1800 Lionel-Boulet, Varennes, Québec, Canada, J3X 1S1 (e-mail of corresponding author: le-huy.philippe@ireq.ca).
Mr. Woodacre is Chief Engineer with SGI, 900 North McCarthy Blvd., Milpitas, CA 95635 USA (e-mail: woodacre@sgi.com).

Paper submitted to the International Conference on Power Systems Transients (IPST2017) in Seoul, Republic of Korea June 26-29, 2017

communication overhead can be clearly seen from the extension [8] of Amdahl's law [9], which takes into account communication cost. It defines the speedup $\psi$ boundary of parallel processing as

$$\psi(n, p) \leq \frac{\sigma(n)+\varphi(n)}{\sigma(n)+\frac{\varphi(n)}{p}+\kappa(n, p)} \tag{1}$$

where $\sigma$ and $\varphi$ are respectively the serial and parallel part of the workload, which are a function of the problem size $n$, $p$ is the number of processing units and $\kappa(n,p)$ the parallelization cost (communication, memory and IOs access time, waiting time, etc.), a function of the problem size and of the number of processing units used. Another useful term is the total execution time, defined as

$$T(n, p) = p(\sigma(n)+\kappa(n, p))+\varphi(n). \tag{2}$$

In Hypersim's case, the sequential part of the RT execution is null. In such cases, the main limiting factor to the speedup is the parallelization overhead, which is normally a monotonically increasing function in regard to $p$. Equation (1) clearly illustrates that reducing the communication cost is essential for massive parallelization of any algorithm, even more so in a real-time execution context.

This section presents how SGI's UV300 supercomputer helps in reaching new levels of performance for RT EMT simulation by providing a very-high performance communication fabric, named NUMAlink 7 (NL7).

### A. *Topology*

The UV300 supports up to eight nodes connected with NL7 cables in an all-to-all topology, as seen in Fig. 1. This topology allows memory access anywhere in the system with very low latency as all memory is accessible through a single NL7 hop. NL7 interconnects have a total data rate capability of 56 Gb/s per direction over four lanes and packet encoding is optimized for small payloads in order to achieve latency on the scale of nanoseconds. The NL7 fabric also implements adaptive routing to ensure reliable data delivery in case of heavily- or over-utilized and/or defective links. The communication protocol is impervious to the actual path taken to reach the destination.

### B. *Node Architecture*

As shown in Fig. 2, each UV300 node consists of four Intel® E7 processor sockets, and related hardware, connected to a pair of NL7 ASICs ("HARP"), which provide connectivity to the NL7 fabric. The sockets are connected in a ring topology with Intel® Quick Path Interconnect (QPI). Each socket is connected to one of the HARP ASICs through its remaining QPI link.

### C. *Implicit Communication Engine*

The UV300 runs a single system image on all interconnected nodes, where cache coherence is assured by SGI proprietary hardware. This cache-coherent non-uniform memory access (ccNUMA) platform was first implemented by SGI in the Origin® 2000 product [10] in the late 90s. The UV300 implementation was optimized to harness the full capabilities of the latest Intel® QPI.

Concretely, this implies that any processing unit used in the simulation is able to access the simulation data on any other at any time. However, the access time will be a function of the data location in relation to the accessing entity, but this effect is mitigated by the all-to-all topology of the UV300 which reduces the non-uniformity to intra-socket, intra-node and inter-node communications. All of this is transparent to the real-time executables, hence the term implicit, but nonetheless, great care is taken to group simulation tasks communicating with each other in the same socket or in adjacent sockets to minimize inter-node communications, which are the most expensive.

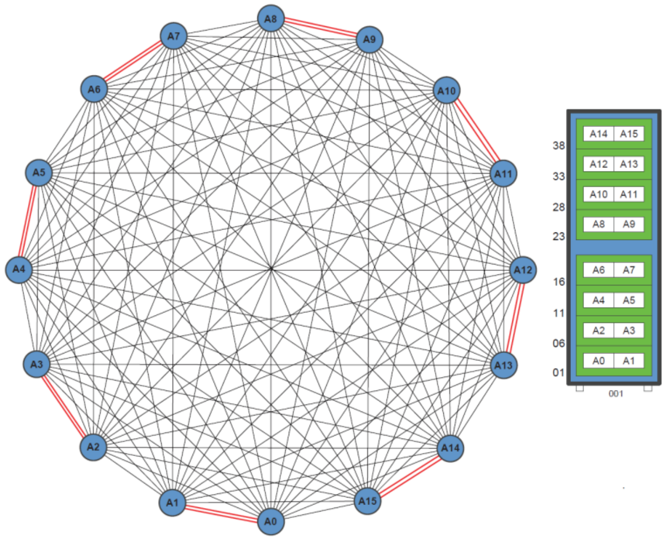

Fig. 1. SGI UV300 all-to-all topology for an eight node single system.

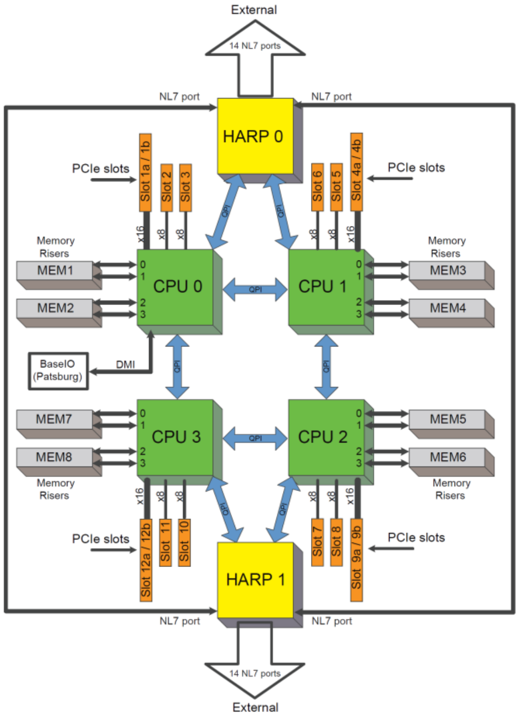

Fig. 2. UV300 node architecture.

### D. *Explicit Communication Capabilities*

In addition to providing the implicit communication engine, the NL7 fabric provides a user-accessible communication engine composed of the Global Reference Unit (GRU) and the Active Memory Unit (AMU) [11]. This explicit engine allows acceleration of data intensive operations, such as remote block copies, and atomic memory operations (AMO). Building upon the last NL generation, the NL7 explicit communication engine sports additional resources to improve remote data copy sustainable bandwidth and an improved cache for AMOs. As these communication resources are built directly into the NL7 fabric, they can operate on the complete address space. They also form the foundation of SGI's MPI and SHMEM libraries. As presented in the following section, the GRU capabilities are exploited in the synchronization barrier to great effect.

## III. Hypersim RT EMT simulator

In order to tackle VLS simulations with an MP RT hardware platform, three elements of the Hypersim software were enhanced: the automatic task mapper (ATM), the synchronization barrier and the RT software architecture.

### A. *Automatic Task Mapper*

After partitioning, a VLS power system is usually broken down into several thousand simulation tasks of various sizes. The ATM has to fairly distribute this computational burden on the available processing units in the RT hardware platform. This mapping is not trivial as communications have to be kept at a minimum and the processing core workloads have to be as balanced as possible for optimal performance. Furthermore, simulation tasks with IOs have to be placed in the processor to which the IO card is connected to in order to minimize IO communication time.

With a very high number of tasks to map, the previous ATM was found to be lacking in mapping speed and flexibility and was thus replaced by a new algorithm that was built on top of the SCOTCH library. This library was developed by the Laboratoire Bordelais de Recherche en Informatique (LaBRI) [12]. It provides a set of algorithms to partition graph structures very efficiently [13]. It runs in linear time and scales very well with very large task graphs. With the previous ATM, several minutes were required to map the application example presented in section IV while the new ATM implementation provides an optimal solution in a few seconds.

### B. *Synchronization Barrier*

As stated earlier, MP RT parallel processing puts a lot of stress on the communication engine. As can be seen in Fig. 4, this stress comes from the data exchange between the simulation threads (COMM) but most importantly from the synchronization barrier (SYNC) required to ensure RT operation. The software implementation of that barrier relies on AMOs to ensure proper operation as the barrier information is relayed to all participating cores.

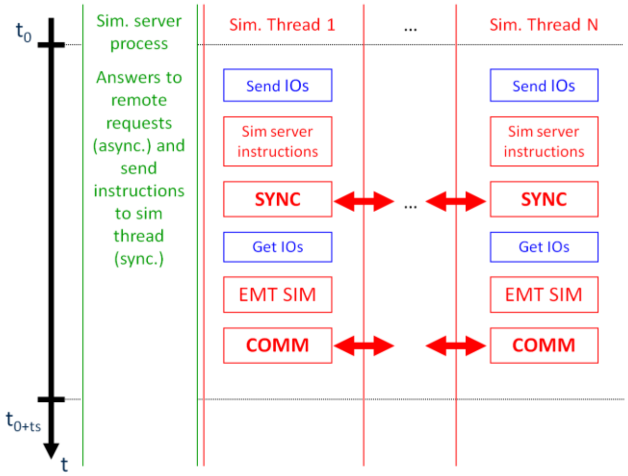

Fig. 4. Conceptual Hypersim simulation time step execution flow.

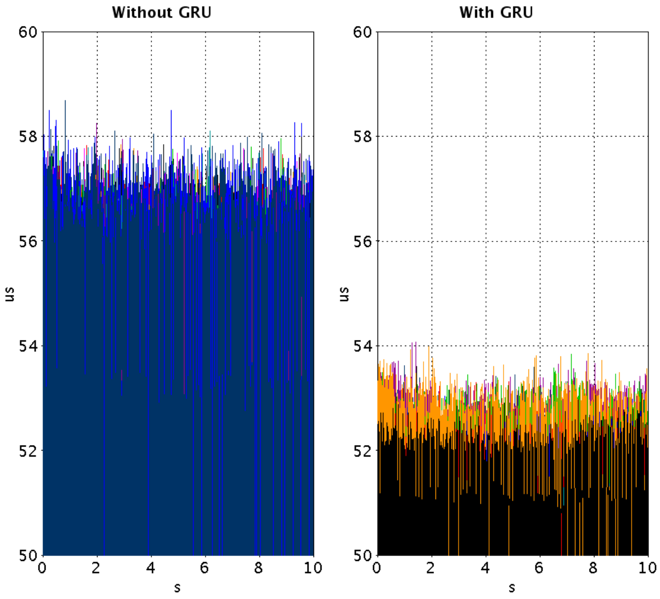

Fig. 5. Impact of GRU on synchronization barrier jitter performance (the closer to 50 µs, the better).

During recent development, it was observed that modest Intel® Xeon simulation platforms (i.e. 4-8 sockets) presented adequate synchronization barrier performance with standard AMOs. However, this was not the case for bigger systems, as shown in Fig. 5, which presents the results from a synthetic synchronization barrier test with two different types of AMO. The objective of the test is to evaluate what kind of delay, called synchronization jitter, is to be expected for all active cores to resume activities following the release of the barrier. The illustrated test results were obtained with a complete eight-node UV300 system (32 sockets, 320 cores) to compare regular AMOs to GRU-enhanced AMOs. "With GRU" and "Without GRU" results were obtained respectively with 318 and 315 active cores; the remaining cores were reserved for the operating system (OS). The timing results from all active cores are superposed.

In the case of regular AMOs, on the left side, a jitter in excess of 8 µs is observed to software synchronize more than 300 cores, distributed in 32 sockets on eight nodes, which in itself is already interesting. However, GRU-enhanced AMOs present even better performance as the jitter is reduced to slightly more than 4 µs to synchronize all 318 cores.

### C. RT Software Architecture

GRU-enhanced AMOs provide a very significant boost to performance but it was observed that the GRU is sensitive to translation lookaside buffer (TLB) shootdown resulting from address space modifications. To provide adequate isolation for the RT simulation threads, thus preventing TLB shootdowns, the RT software architecture was modified: all simulation threads are started by a separate process, as shown in Fig. 6.

Graphic user interface, acquisition and other tools from the software environment are all executed on a workstation (i.e. a regular computer (desktop or laptop)), while the rest operates on the supercomputer. The multi-user server is responsible for dispatching RT resources to the various users and to start the simulation servers. Each simulation server will then spawn the RT process, containing all the simulation threads, which are migrated to the RT cores for execution after raising the shields to restrict OS interferences. This multi-process, multi-thread architecture allows more efficient usage of GRU resources and better RT thread isolation, resulting in better RT performance.

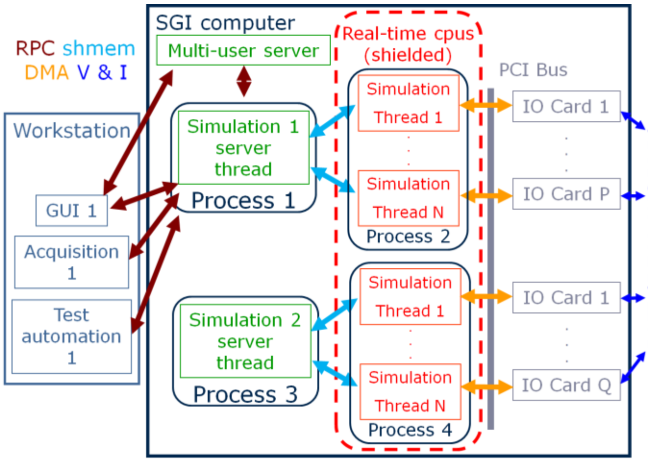

Fig. 6. Hypersim RT software architecture for maximum RT thread isolation. The communication mechanisms (RPC, shmem, DMA and real voltages and currents) are identified with different colors.

## IV. Application Example

To illustrate the power of the MP RT simulation platform, the Hydro-Québec's 735-kV power transmission system (counting a little more than 300 three-phase buses) was duplicated and assembled as illustrated in Fig. 7 to form a single VLS power system. This basic subsystem is extensively used internally for various studies and was thoroughly validated with EMTP-RV and even with a hybrid simulator [7]. The exact content of the resulting power system is listed in Table I. At over 5000 three-phase buses, this power system definitively qualifies as VLS in the current RT EMT context. No equivalent or network reduction is used and all of the 16432 nodes are simulated as well as the 656 electrical machines (with saturation) and their full control systems (exciter, governor, PSS, etc.), almost 4000 piece-wise linear saturable elements (surge-arresters and saturable inductances) and 1600 switches (breakers and thyristors; $R_{on}/R_{off}$ modeling). All voltages, currents and control signals are available for observation: the observed signal set can be changed at any time during the simulation according to the needs of the moment. Furthermore, parameter modifications are allowed as long as the topology of the power system is not changed.

The nature and the magnitude of the stress applied on the communication engine are better understood when analyzing the inter-core communications. As seen in Table I, 4561 signals are sent at each time step. All those signals are bundled into 756 data packets, all containing less than 1 kb of active payload. At each time step, which is typically in the 30-50 µs range, a burst of communication is required by the 200 active cores to transmit and receive all 756 packets. Needless to say, these intense bursts of very small packets are not trivial to handle, which is why a very-high performance communication engine, or fabric in this case, is required.

A six-cycle three-phase fault is applied in one subsystem of the VLS power system as illustrated in Figs. 8 to 10: the first two figures present the waveforms obtained through the signal acquisition services while Fig. 10 is a scope screenshot displaying the IOs related to the signals in Fig. 9.

TABLE I
Total Content of the VLS Power System

| Power system element | Number |
|---|---|
| Electrical nodes | 16432 |
| Electrical machines | 592 |
| Synchronous condensers | 64 |
| Static var compensators | 112 |
| Transmission lines | 2736 |
| Three-phase transformers | 2096 |
| RLC elements | 49988 |
| Non-linear elements | 3984 |
| Switches | 1600 |
| Inter-core communications | 4561 |

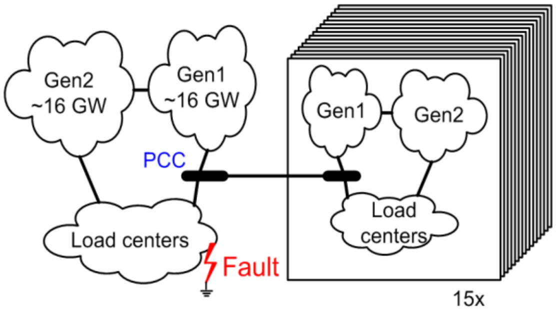

Fig. 7. Simplified representation of the very-large-scale EMT simulation.

It is important to note that no special RT-specific modeling, accuracy-relaxation or simplification is required for RT operation. In fact, as described in [14] and [15], a RT bounded-iterative engine is used to preserve a very high level of accuracy. Iterations are done according to the needs of each computational task and not globally, hence reducing the execution time cost and preserving RT operation.

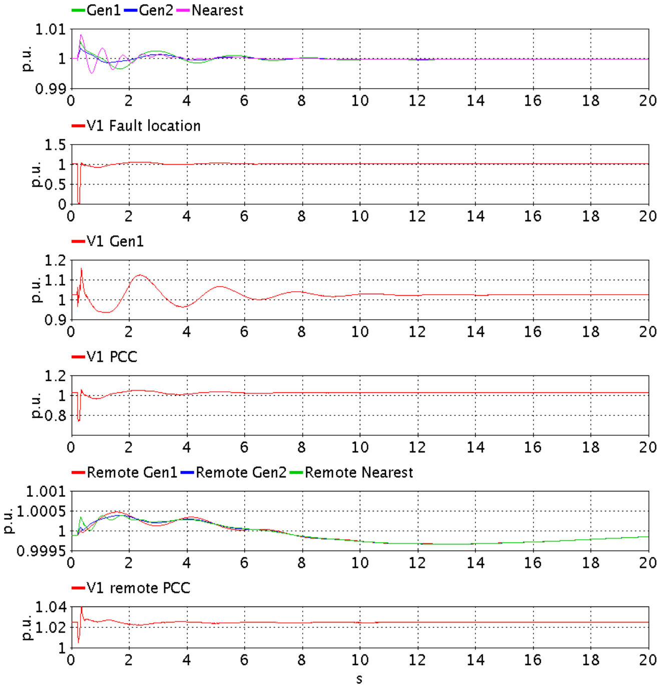

Fig. 8. System response to a six-cycle three-phase fault located in the load centers: generator speed in faulted system (main Gen1 generator, main Gen2 generator and nearest generator); pos.-seq. voltage at fault location, pos.-seq. voltage at main Gen1 generator; pos.-seq. at point of common coupling (PCC); generator speed in a remote system (same generators) and pos.-seq. voltage at PCC of the remote system (all signals in p.u.).

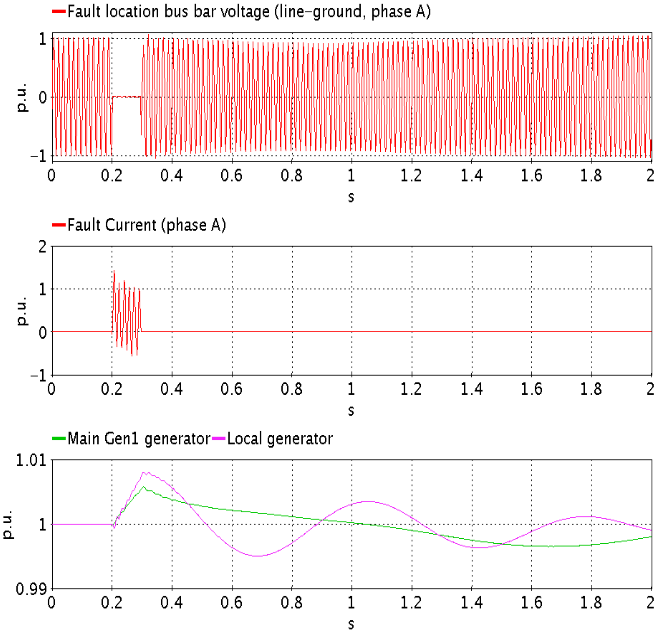

Fig. 9. System response as seen through data acquisition services to a six-cycle three-phase fault located in the load centers: fault location bus bar voltage (line-ground phase A, p.u.), fault current (phase A, p.u. selected to match scope scaling), nearest machine speed (p.u.) and main Gen1 generator speed (p.u.).

Concerning the mapping of this VLS power system, an efficient ATM is mandatory as trivially repeating the same mapping pattern for each subsystem would result in a sub-optimal mapping where resources are underutilized (CPUs and IOs). Due to IO placement, several simulation tasks representing different part of different subsystems had to be grouped together. The SCOTCH-based ATM algorithm then provided an optimal mapping minimizing the communication while maintaining uniform processing load on all cores.

Timing results for the VLS simulation are gathered in Table II. As stated earlier, the sequential part $\sigma(n)$ of the RT execution is null and the parallel part $\varphi(n)$ was empirically determined as approximately 2500 µs of processor time on Intel® Xeon E7-8891 v4 processors. Using (1) and (2), speedup $\psi(n,p)$ and parallelization overhead $\kappa(n,p)$ were determined and provided interesting results, seen in Table II. For the first two table entries, the resulting RT executable files were so large and required so much runtime memory that L3 cache memory was insufficient and RAM had to be used, drastically reducing performance. Obviously RT EMT simulation in that situation is not possible. The third entry, with 20 processing units, is interesting as cache memory is nearly sufficient as indicated by $\kappa(n,p)$ but for the problem size, raw processing power is insufficient to provide a usable time step for a RT EMT simulation.

The last three entries provide possible RT scenarios, where parallelization overhead behaves in a more traditional way. For RT operation of the presented VLS power system (more than 16 thousand electrical nodes) at 60 or 50 µs, 48 and 65 E7-8891v4 cores are required respectively. However, to reach a time step of 30 µs, 200 cores are required. This last case is interesting as parallelization overhead is actually greater than "useful" processor time for each step, which clearly illustrates the non-linear nature of the speedup factor of parallelization.

All presented simulation results were obtained at 30 µs on 200 E7-8891v4 cores.

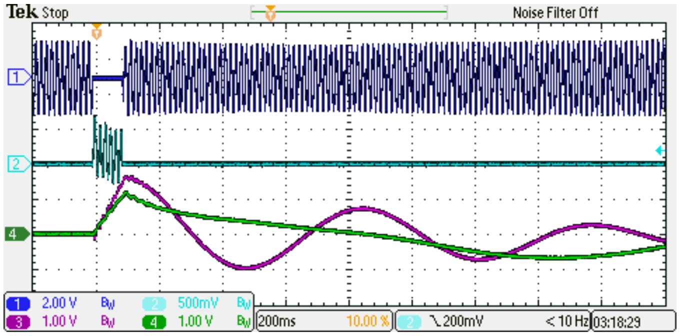

Fig. 10. System response as seen at the IOs to a six-cycle three-phase fault located in the load centers: fault location bus bar voltage (phase A), fault current (phase A), nearest machine speed deviation and main Gen1 generator speed deviation.

TABLE II
TIMING RESULTS OF THE VLS POWER SYSTEM

| Number of processing units | Min possible RT $T_s$ (µs) | Total processor time (µs) | Overhead $\kappa(n,p)$ (µs) | Speedup $\psi(n,p)$ |
|---|---|---|---|---|
| 1 | 9725 | 9725 | 7225 | 0.26 |
| 10 | 1075 | 10750 | 825 | 2.33 |
| 20 | 140 | 2800 | 15 | 17.86 |
| 48 | 60 | 2880 | 7.9 | 41.67 |
| 65 | 50 | 3250 | 11.5 | 50 |
| 200 | 30 | 6000 | 17.5 | 83.34 |

## V. Conclusions

As supercomputers continue to evolve, massive parallelization of the EMT computations of VLS power systems with standard processing units becomes a reality. This enables RT operation of EMT simulation software as presented in this paper: the MP-enabling features of the UV300 communication fabric were presented in addition to the required software modifications as VLS power system simulation is very demanding at several levels.

This approach to MP simulation presents several advantages compared to previous solution based on GPUs and FPGAs: while presenting impressive performances, they lack the flexibility and ease of use of generic CPUs. Furthermore, significant efforts have to be deployed in order to modify all the standard algorithms and models developed over the years to match the strong suit of these devices. Finally, GPU- and FPGA-based solutions have yet to demonstrate their ability to provide the services required for real-time commissioning studies as required by entities such as Hydro-Québec.

An application example was then presented: with more than 16 thousand electrical nodes and more than 650 machines, this VLS power system used for basic subsystem a representation of the Hydro-Québec network, which is, in itself, a realistic basic construction block. A brief analysis of the parallelization overhead was then presented, in which an interesting memory-related performance degradation was observed for cases with a low processing unit count: due to the sheer size of the problem, a certain partitioning of the problem is required to use only the fast L3 cache memory and not a combination of cache and RAM memory. The VLS 16 thousand-node power system was successfully simulated in RT with 200 processing units with a time step of 30 μs.

Future work will explore the possibility hinted at by the experimental parallelization overheads: the RT EMT simulation of power systems with more than 10 thousand three-phase bus bars.

## VI. References

[1] Z. Zhou, V. Dinavahi, “Parallel Massive-Thread Electromagnetic Transient Simulation on GPU,” *IEEE Trans. Power Delivery*, vol. 29, no. 3, pp. 1045-1053, 2014.

[2] J. K. Debnath, A. M. Gole, W.-K. Fung, “Graphics-Processing-Unit-Based Acceleration of Electromagnetic Transients Simulation,” *IEEE Trans. Power Delivery*, vol. 31, no. 5, pp. 2036-2044, 2016.

[3] M. Matar, R. Iravani, “The Reconfigurable-Hardware Real-Time and Faster-Than-Real_Time Simulator for the Analysis of Electromagnetic Transients in Power Systems,” *IEEE Trans. Power Delivery*, vol. 28, no. 2, pp. 619-627, 2013.

[4] Y. Chen, V. Dinavahi, “Multi-FPGA digital hardware design for detailed large-scale real-time electromagnetic transient simulation of power systems,” *IET Gener. Transm. Distrib.*, vol. 7, no. 5, pp. 451-463, 2013.

[5] Le-Huy, P., Giroux, P., Soumagne, J.-C.: “Real-Time Simulation of Large-Scale AC System with Offshore DC Grid,” IPST’13, Vancouver, Canada, July 18-20, 2013.

[6] V. Q. Do, J.-C. Soumagne, G. Sybille, G. Turmel, P. Giroux, G. Cloutier, S. Poulin. “Hypersim, an Integrated Real-Time Simulator for Power Networks and Control Systems,” ICDS’99, Vasteras, Sweden, May 25-28, 1999.

[7] D. Paré, G. Turmel, J.-C. Soumagne, V. A. Do, S. Casoria, M. Bissonnette, B. Marcoux, D. McNabb. “Validation tests of the Hypersim digital real time simulator with a large AC-DC network,” IPST’03, New Orleans, USA, Sept. 28 - Oct. 2, 2003.

[8] M. J. Quinn, “Parallel Programming in C with MPI and OpenMP,”, McGraw-Hill, 2004.

[9] G. M. Amdahl, "Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities," AFIPS’67, Atlantic City, USA, April 18-20, 1967.

[10] J. Laudon, D. Lenoski. “The SGI Origin:a ccNUMA highly scalable server,” ISCA’97, Denver, USA, June 2-4, 1997.

[11] G. Thorson, M. Woodacre. “The SGI UV2: A fused computation and data analysis,” SC12, Salt Lake City, USA, Nov. 10-16, 2012

[12] F. Pellegrini, J. Roman, “SCOTCH: A Software Package for Static Mapping by Dual Recursive Bipartitioning of Process and Architecture Graphs,” HPCN’96, Brussels, Belgium, April 15-19, 1996.

[13] F. Pellegrini, “SCOTCH and LIBSCOTCH 6.0 User's Guide”, LaBRI, University of Bordeaux, September 2014.

[14] O. Tremblay, M. Fecteau, P. Prud’Homme “Precise Algorithm for Nonlinear Elements in Large-Scale Real-Time Simulator,” in Proc. of CIGRÉ Canada Conf. on Power Systems, Montréal, Canada, Sept. 2012.

[15] O. Tremblay, R. Gagnon, M. Fecteau, “Real-time Simulation of a Fully Detailed Type-IV Wind Turbine,” IPST’13, Vancouver, Canada, July 18-20, 2013.