Review of HPC Applications for Future Power System Analysis Tools

Milana Plecas
University of Strathclyde
Glasgow, United Kingdom
milana.plecas@eee.strath.ac.uk

Ivana Kockar
University of Strathclyde
Glasgow, United Kingdom
ivana.kockar@eee.strath.ac.uk

Abstract – **One of the major challenges in the energy sector is to ensure secure and sustainable energy supply. A growth in the usage of renewable resources which are usually connected at lower distribution levels will lead towards electricity networks that are more complex. Likewise, technology that will allow active demand participation has been developed and would be introduced in networks. It will enable that the part of the power demand of active customers could be shifted away from peak periods. This means that the number of network nodes with active devices will increase rapidly affected by a number of distributed generators and, even more, by utilization of smart meters and electric vehicles that can act both as a consumer and as a source/storage device. Consequently, operation of these new Smart Grids will require a step-change in capabilities of operational tools due to a considerable rise in the number of control variables and reduced time intervals between which generation outputs and prices are calculated. Calculating frequent prices and power flow for systems with a large number of nodes will require very fast computations.**

Therefore, as the power grid networks become more granulated and more intelligent, their operation and control will become even more challenging due to the size of the underling mathematical problems that need to be solved in various power systems analysis tools.

The objective of this paper is to review ways in which High Performance Computing (HPC) can be used in power system analysis, and discuss possible further developments in this field.

Index Terms – **HPC, Smart Grid, Power System Analysis.**

I. INTRODUCTION

Secure and reliable delivery of energy is extremely important for modern society. However, due to the significant growth of a power demand as well as gradual decrease of energy resources, this task is becoming more challenging.

So far, we have mostly been dependent on traditional non-renewable fossil fuels. However, among their welfare benefits to social and economic development, they have also had a significant influence on environmental pollution and degradation. Consequently, finding sustainable alternatives has become an urgent task of modern society. One approach is the utilization of large number of smaller renewable generators, which are usually connected at lower distribution levels. Furthermore, energy demand has been experiencing a continuous substantial growth, due to industrial utilization, as well as a modern lifestyle. This trend may be expected to continue in particular through the mass introduction of electric vehicles and other ways of transport electrification, as well as electric heat pump. These new loads may have a significant effect on the load profile and network and therefore, an active demand participation as well as utilization of the smart meters is becoming essential.

However, as the power networks are becoming larger or more granulated, computation time of power system analysis is becoming longer and longer, since the models may become more complex and involve more control variables. Moreover, the integration of more renewable resources introduces newer challenges to power grids because of their distributed and volatile outputs which need to be controlled.

As a result, providing efficient and reliable power system operation of these more complex and more granulated networks requires improved computational efficiency of power system analysis tools. One of the ways to achieve this goal is to use the High Performance Computing (HPC) that has experienced a significant improvement in the last decades.

Since the last major review of the application of HPC and the power systems was published in 1997 [1], a large number of algorithms and methods for utilization of HPC in the power systems have been developed, and both fields have undergone significant changes. The aim of this paper is to present a brief overview of the HPC application in the power system analysis in the last decades and indicate some of the new developments that were recently suggested.

The remaining parts of this paper are organized as follows. Section II gives an overview of the main features of the HPC field. Its applicability in various areas of power system analysis is described in section III, while the conclusions are drawn in Section IV.

II. INTRODUCTION TO HPC

Over the last decades, the field of HPC has changed significantly. Namely, in 1965, Gordon Moore stated that the number of transistors that can be placed in an integrated circuit will double approximately every two years [2]. However, as illustrated in [3], and [4], the evolution of computer performance no longer follows Moore's Law due to physical limitations of the current chip manufactures.

Therefore, processor designers have turned to the new and exciting solutions in order to improve the computational performance. Instead of improvement in the single processor performance, emergent trends in the HPC industry include grid computing, multi-core computing, and Graphics Processing Unit (GPU) computing.

One of the suggested approaches to the HPC is the Grid computing which presents HPC technologies that changed the way of solving the complex computational problems. It refers to a centrally managed but flexible computing environment that consists of connected storage and data as well as Central Processing Units (CPU) from different systems [4]. Grid architecture is built on an open source tools which allow dynamic communication across geographically dispersed resources. Also, it can operate over the Internet or any other suitable computer networking technology. Some of the major advantages of grid computing include: parallel processing, virtual organization for collaboration, utilization of idle resources, access to shared resources, load balancing and management. However, for a wide adoption of grid computing, there are several issues that need to be overcome, such as inter-processor communication and seamless integration over heterogeneous resources. There are arguments that the grid computing could have potential benefits for future power system applications including monitoring and control, electricity market and participation, regulation, planning, reliability and security, performance improvement, scheduling, load balancing and available transfer capability as well as other applications [5-7]. However, application of the Grid Computing has to be evaluated taking into consideration the nature of some power system analysis that are used in operation, as well as considering data and communications security.

A multi-core CPU contains two or more complete functional units, and it is considered as shared memory system. All of its cores compete for the memory access which can be a problem because of the limited bandwidth for memory access. Consequently, multi-core computing is useful for an application with a regular memory access pattern, such as sparse matrix and irregular computations [3].

Currently, a graphics processing unit stands for the emerging HPC trend and, considering computation per dollar, presents probably the most powerful tools used for general purpose processing [8]. It is traditionally used to increase the performance of graphic applications, but since a Compute Unified Device Architecture (CUDA) was developed by NVIDIA, GPU has become accessible to other types of applications [9]. Thus, its utilization extends from the graphics processing unit to a general processing unit. Both multi-core CPU and GPU combine several cores into one die which share the same memory channel. However, while the multi-core CPU has stronger control capabilities and greater cache system, GPU has the massively parallel architecture, i.e. the large number of computational cores. Thread is determined as the minimum executed unit in GPU, and GPU hides the memory latency by driving many threads to concurrently access memory, so its rates of memory bandwidth increase more than memory rates of multi-core CPU.

In contrast, the GPU is controlled by the CPU, but its own specified instructions are run independently from the CPU [10]. CPU runs the main program and sends tasks (i.e. kernel functions) to GPU. Following that, these kernel functions are executed in parallel by GPU threads [11]. Consequently, GPU computation power is significantly faster than a CPU computation and GPU can deal with extremely high floating-point rates of calculation [4]. This technology is suitable for an application with simple logical controls and bulk concurrent data calculations. However, while GPUs can offer a great performance increase over CPUs for some applications, they may not necessarily always be the best choice.

The measure of the HPC's performances is floating point operations per second (FLOPS). First supercomputers, which achieved the speed of about 100 megaflops, were built in the 1970s. So far there was a significant growth of the computing performance from gigaflops in the 1980s, via teraflops in the 1990s up to petaflops in 2008. Since the growth of FLOPS depends directly on computer architecture, computer designers have turned from vector machines to massively parallel processing and parallel supercomputer. Furthermore, in order to achieve the exaflops speed, which is predicted to be realized by 2020, a new class of computers is required and a huge increase in parallelism as well as an improvement of computer reliability and power supply needs to be overcome [12-13].

III. HPC APPLICATIONS FOR POWER SYSTEM ANALYSIS

As power grids become more intelligent and larger, their operation will become more complex. This will require better and faster power system analysis tools whose computational demands will increasing significantly. As a result, complex mathematical methods, network theories, and optimization methods need to be used in order to solve these problems [3]. The aim of this section is to review HPC application for few areas of power systems analysis.

A. *Power Flow*

The power flow analysis is one of the major tools for power system operation and planning which provides the steady state characteristic of the whole power system. Mathematically, it yields the solution of a set of nonlinear algebraic power balance equations as follows:

$$P_i = V_i \sum_{j=1}^{N} V_j Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij}), i = 1, ..., N \tag{1}$$

$$Q_i = V_i \sum_{j=1}^{N} V_j Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij}), i = 1, ..., N \tag{2}$$

where P_i and Q_i are the active and reactive bus power injection; V_i and V_j are the bus voltage magnitudes at buses i and j; Y_{ij} is the element of the bus admittance matrix; δ_i and δ_j are the phase voltage angles at node i and j; θ_{ij} is the phase angle of admittance bus element Y_{ij} between buses i and j; N is the number of system buses [8, 14].

There are a wide range of direct and iterative methods for solving the resulting nonlinear system of equations. In the last century, a number of effective iterative solutions were presented, including Gauss-Seidel and Newton-Raphson method. They provide an approximate solution, and due to its good convergence and high speed efficiency, the Newton-Raphson method is the most popular technique for solving set of nonlinear equations of the power flow problem.

To improve computational time, a parallel Newton-GMRES (Generalized Minimal Residual) power flow algorithm is proposed in [15]. This algorithm is one of the Krylov subspace iterative methods. A test architecture that was used consisted of a cluster of single-processor workstations that communicate via 1.0Gbit/s Ethernet through a message passing distributed-memory. Moreover, the test power flow cases consist of 15359 and 30910 buses. The study demonstrates good performance of the new algorithm and shows that the iterative linear methods are more scalable then the direct solvers in a zero latency network. Also, it shows that the former could achieve a bigger than 30% acceleration of the flops.

The work in [11] uses the computational power of GPU in order to enhance the processing of iterative power flow solvers. It presents a parallel algorithm for power flow analysis which is based on CUDA and run on GPU. The algorithm evaluates the performance of NR and Gauss-Jacobi (GJ) iterative methods. It is computed in Matlab using the Jacket platform. The tests are performed on a few standard IEEE bus systems and one synthetic 4766-bus system. They are run on NVIDIA Tesla C1060 GPU and Intel Xeon Quad Core CPU. The paper shows good results, especially for the large-scale power systems. It demonstrates the speedup of the execution time on GPU of over 10 times for NR and GJ, and of even 140 times for the Jacobian matrix computation.

B. *Optimal Power Flow*

The Optimal Power Flow (OPF) is a tool typically used in operation and control of power system generation and network. In its most general form, it is a highly constrained and large dimensional nonlinear optimization problem used to minimize desired objective function while satisfying various equality and inequality constraints. Mathematically, the OPF problem can be represented as:

$$\min F(x) \tag{3}$$

Subject to

$$g(x) = 0 \tag{4}$$

$$h(x) \leq 0 \tag{5}$$

where x is the vector of decision variables, including the control and dependent state variables (e.g. active generation outputs, voltage magnitudes and angles, reactive power generation, transformer tap settings); $F(x)$ is the system objective function (e.g. minimum generation costs (economic dispatch), power loss or bus voltage deviation); $g(x)$ represents the equality constraints which are the nodal power flow equations; $h(x)$ includes the inequality constraints with their upper and lower limits which represent system security, physical limitations and reliability restrictions [14, 16].

Since the OPF was introduced by Carpentier in 1962 as an economic dispatch problem [17], it has been widely studied and the several conventional optimization methods, as well as heuristic optimization methods have been proposed to solve this problem.

Furthermore, a parallel Particle Swarm Optimization (PSO) method for the solution of the OPF is presented in [18]. The developed algorithm is based on a PC cluster system with 6 Intel Pentium IV 2GHz processor with 256MB RAM. The system objective function which is considered is minimization of fuel cost for all generators, and the tests have performed using the IEEE 30-bus test system. The evaluated results have demonstrated that the computing cost can be decreased by parallel processing.

A new method for the solution of the DC OPF is developed in [19]. It decomposes the large interconnected power system into independent regions around the tie-lines and not boundary nodes. The solutions of the regional OPF sub-problems are coordinated through a pricing mechanism in order to solve the original OPF problem. It is the iterative method, and tie-line information is exchanged at the end of each iteration. Also, the paper presents a completely decentralized implementation, without any central coordination. This work is extended in [20], where the DC OPF sub-problem is reduced, and algorithm is implemented on a network of workstations, using the Parallel Virtual Machine (PVM) software. Each workstation is assigned to a regional operator and a master workstation check the algorithm convergence. The both algorithms show that the computation times of both decentralized and centralized solutions are similar, but the latency is reduced in the decentralized model.

In [21], a new seamless computing method is used to solve OPF problem. This method is based on a grid platform and it divides the electric network using boundary current-ejecting method. There are considered only equality constraints and two objective functions, the power loss minimization and cost of fuel minimization. The method performances are tested on different size of "Lab-grid." Overall, it was demonstrated that the total computing speedup and efficiency were improved.

C. Contingency Analysis

Contingency analysis (CA) is one of the main functions of power system security analysis. In order to satisfy the system efficiency and reliability as well as to prevent the blackouts, CA evaluates the impact of different combination of power system component outages. It is based on estimating the critical contingencies, and can simulate both single failure events and multiple equipment failure events. CA programs involve two functions, contingency selection and contingency evaluation. Traditionally, it is limited to "$N-1$" analysis, which means that if one of the system components fails, the system has still to operate within its limits. However, some of the latest developments in power grid operation indicate that there is a need to consider "$N-2$" or even higher, "$N-x$", analysis [22-24].

Thus, the work in [8] proposes the computation of "$N-1$" contingency analysis on the GPU using the DC power flow model. A sparse-matrix Gauss-Jacobi and Gauss-Seidel algorithm are used to solve the power flow equations. In addition, in order to implement CA algorithm on the GPU, "Cg" ("C in computer graphics") programming language is used. The proposed algorithm is tested on the standard IEEE 118/300-bus systems, as well as on evolved 1000/2000-bus systems. The results show that there is a significant improvement in the calculation time when using the GPU compared to a similar implementation on the CPU, especially for large systems.

Furthermore, the potential of the HPC for a massive "$N-x$" contingency analysis is investigated in [23]. The computational performances of both static and dynamic computational load balancing are analyzed. They are implemented with both the Colony2A and MPP2 cluster machines. The former has 24 Itanium-2 Hewlett-Packard (HP) nodes which have two 1.0GHz processors, 6GB memory, and 36GB disk space, while the latter has 980 HP Longs Peak nodes with dual Intel 1.5GHz Itanium-2 processors and HP's zx1 chipset. Furthermore, the computational performances are tested on the 14000-bus Western Electricity Coordinating Council (WECC) system using Message Passing Interface (MPI). The study indicates that there is a significant improvement in the computational time, of about 500 times with the "$N-2$" analysis and slightly less with the "$N-1$" scenario.

In addition, the work in [24] demonstrates a parallel processing approach to contingency analysis. Each contingency is shared across the processor and all possible single line outages are considered. This method is implemented on a Linux Cluster, and MPI is used for sharing data between the processors. IEEE standard test systems such as 14-bus, 30-bus, 118-bus, 162-bus and 300-bus are used for an evaluation of the method performances. It is shown that the parallel processing approach does not improve the performances for small systems. However, there is a significant decrease in the execution time for large systems.

D. State Estimation

State estimation is one of the major Energy Management System (EMS) functions of the system control and stability analysis. It generates the current power grid status as well as inputs for other operational tools such as contingency analysis, optimal power flow, and economic dispatch. The purpose of state estimation is to estimate the voltage magnitude and angles at all buses of a power system, from the available measurements according to some criterion. These available real time measurements are received from a Supervisory Control And Data Acquisition (SCADA) system, and include line flows, node injections, and voltage magnitudes. Moreover, the most common used criterions are the maximum likelihood criterion, the weighted least-squares criterion, and the minimum variance criterion. Mathematically, the state estimation problem can be represented as:

$$z = h(x) + e \tag{6}$$

where z is the $(m \times 1)$ measurement vector; x is the $(n \times 1)$ true state vector; $h(x)$ is the $(m \times 1)$ state equation vector; e is the $(m \times 1)$ measurement error vector; m is the number of measurements; n is the number of state variables.

The state estimation concept was introduced into the field of power systems, by Schweppe, in the early 1970s, and a number of methods have been proposed to calculate the state vector. Traditionally, the state estimation is based on a steady state system model and the static state vector. Bus voltages and phase angles can be obtained only in an interval of a few minutes because the SCADA data is not able to capture grid dynamics. However, due to the changes in the load, the real-time power system stands for very dynamic system. Therefore, state estimation need to be performed at short intervals of time in order to estimate dynamic states, e.g. generator speed and rotor angle. Consequently, the dynamic state estimation techniques which can predict the system state one step forward have also been developed [22, 25-26].

A heuristic parallel particle swarm optimization method for solving the power system state estimation is presented in [26]. The proposed algorithm is paralleled by the PC cluster system which is consisted of 8 Intel Core 2 Duo 2.33GHz processors. The effectiveness of the proposed algorithm is verified with simulation on the IEEE 118-bus system, and the gained results show the speedup of 4.8 times for the implemented method.

The work in [27] examined the utilization of HPC to the Extended Kalman Filter (EKF) technology for a dynamic state estimation process. In order to run the algorithm on multiple processors, there is used OpenMP Application Program Interface (API). Following, the evaluated algorithm is tested on a single quad-core Xeon E5345 machine with 8 cores which run at 2.33GHz and use 16GB of shared memory. As the test cases there are used 16, 50, 100, and 200 machine models, and the results proposed that there are evident reductions in computational times, especially as the size of the system increases.

E. Transient Stability

Transient stability study has an important role in planning, operation, and control of the power systems. It describes ability of the power system to maintain synchronism when there are different transient disturbances such as a fault on transmission facilities, loss of generation, loss of a large load, line-switching operations, and sudden changes. Due to these disturbances, generator rotor angles, power flows, bus voltages, and other system variables experience a large excursion. Mathematically, transient stability study can be presented as a set of nonlinear differential algebraic equations (DAE). Large system analysis consists of thousands of these equations, so finding the solution of this equation set involves huge computationally effort.

Transient stability study became popular in the 1960s and since then several approaches have been developed to deal with these great computational demands in the real time [28].

The work in [10] uses GPU computational advantages to deal with transient stability analysis for large-scale systems. In order to make a hybrid GPU-CPU simulator, there are used NVIDIA GeForce GTX 280 and a quad-core AMD Phenom CPU. Simulator accuracy and efficiency are validated. The former is verified by comparing obtained results with results from the PSS/E software. As regard the latter, there are used the IEEE New England system with different number of buses. It is shown that the algorithm accuracy is very good and there is achieved the speedup about 340 times.

Furthermore, a spatial parallel algorithm for transient stability simulation is developed in [29]. The aim of this algorithm is to calculate transient stability of whole power system which was previously divided into subsystems and a harmony system. The algorithm is validated on the large power system of the north China using an IBM Cluster 1350. Since the high improvement in calculation time is obtained, it is shown that this algorithm is suitable for real-time transient stability simulation for large power systems.

IV. Conclusions

Recently, HPC has been recognized as the most efficient and effective means to improve performance of power system analysis tools. Keeping in mind its importance for both current and future work, this paper investigates the potential of an utilization of the HPC applications for improvement of the power systems. Although this work presents a short and not exhaustive literature review, it is shown that suitable HPC platforms can significantly improve power system operation, analysis and control. They contribute to much faster solving of the grid operation functions as well as allow the integration of the dynamic analysis into the real-time grid operation, especially for large-scale systems. Also, it is shown that there is a great potential for further improvement and it is evident that HPC holds a promise of expansion of the computational resource.

Acknowledgments

Authors would also like to thank the EPSRC for their financial support of the Novel Asynchronous Algorithms and Software for Large Sparse Systems project, under grant number EP/I006729/1.

References

[1] D. M. Falcao, "High performance computing in power system applications," in *VECPAR '96: Selected papers from the Second International Conference on Vector and Parallel Processing*, pp. 1–23, London, UK: Springer-Verlag, 1997.

[2] E. Moore, "Cramming more components onto integrated circuits," *Gordon Electron.*, vol. 38, no. 8, pp. 114–117, Apr. 19, 1965.

[3] Z. Huang, and J. Nieplocha, "Transforming Power Grid Operations via High Performance Computing," in *IEEE Power and Energy Society General Meeting 2008*, Pittsburgh, PA, USA, July 20-24, 2008.

[4] G. Vacek, D. Mullally, and K. Christensen, "Trends in High Performance Computing Requirements for Computer-Aided Drug Design," *Current Computer - Aided Drug Design*, Vol. 4, No. 1., pp. 2-12, March 2008.

[5] M. Ali, Z. Y. Dong, and X. Li, "Applications of grid computing in power systems," in *Proceedings of the Australasian Universities Power Engineering Conference*, Hobart, Tasmania, Australia, September 2005.

[6] J. Dongarra, "Trends in High Performance Computing," *IEEE Circuits and Devices Magazine*, January/February 2006.

[7] M. Irving, G. Taylor, and P. Hobson, "Plug in to grid computing," *IEEE Power and Energy Magazine,* vol. 2, no. 2, pp. 40–44, March 2004.

[8] A. Gopal, "DC power flow based contingency analysis using graphics processing units," Master's thesis, Drexel University, Philadelphia, PA, March 2010.

[9] NVIDIA, http://www.nvidia.com/object/cuda_opencl_new.html

[10] V. J. Marandi, and V. Dinavahi, "Large-scale transient stability simulation on graphics processing units," pp. 1–6, July 2009.

[11] J. Singh, and I. Aruni, "Accelerating Power Flow studies on Graphics Processing Unit," in *Proceedings of the Annual IEEE India Conference 2010 (INDICON)*, pp. 1-5, December 2010.

[12] TOP500, www.top500.org

[13] DOE (2011) "Leap to the extreme scale could break science boundaries", Available at http://ascr-discovery.science.doe.gov/feature/exa_ov1.shtml

[14] J. Z. Zhu, *Optimization of Power System Operation*, Wiley, 2009.

[15] F. Tu, and A. Flueck, "A Message-Passing Distributed-Memory Newton-GMRES Parallel Power Flow Algorithm," in *IEEE Power Engineering Society Summer Meeting 2002*, vol. 3, pp. 1477–1482, July 2002.

[16] T. Orfanogianni, and R. Bacher, "Increased OPF code development efficiency by integration of general purpose optimization and derivative computation tools," *Power Systems, IEEE Transactions on*, pp. 987-993, vol. 15, issue 3, August 2000.

[17] J. Carpentier, "Contribution to the Economic Dispatch problem," *Bull. Soc. France Elect.* vol. 8, pp. 431-437, August 1962.

[18] J. Y. Kim, H. M. Jeong, H. S. Lee, and J. H. Park, "PC cluster based parallel PSO algorithm for optimal power flow," in *Intell System Application to Power System, International Conference on*, vol. 5(8), pp. 1-6, 2007.

[19] A. G. Bakirtzis, and P. N. Biskas, "A decentralized solution to the DC-OPF of interconnected power systems," *IEEE Trans on Power System*, vol. 16, pp. 1007-1013, August 2003.

[20] P. N. Biskas, A. G. Bakirtzis, N. I Macheras, and N. K Pasialis, "A decentralized implementation of DC optimal power flow on a network of computers," *IEEE Trans on Power System*, vol. 20, pp.25-313, February 2005.

[21] F. Hualiang, M. Chengxiong, Z. Buhan, and L. Jiming, "The grid computing model for OPF," *University power engineering conference 2006*, Newcastle U.K., Sep. 6-8, 2006.

[22] A. J. Wood, and B. F. Wollenberg, *Power Generation Operation and Control*, 2nd edition, John Wiley and Sons, Inc., 1996.

[23] Z. Huang, Y. Chen, and J. Nieplocha, "Massive Contingency Analysis with High Performance Computing," in *IEEE Power Engineering Society General Meeting 2009*, Calgary, Canada, July 26-30, 2009.

[24] G. A. Ezhilarasi, K. S. Swarup, "Parallel contingency analysis in a high performance computing environment," *Power Systems, 2009. ICPS '09. International Conference on*, pp.1-6, 27-29 Dec. 2009.

[25] N. R. Shivakumar, and A. Jain, "A Review of Power System Dynamic State Estimation Techniques," *Power System Technology and IEEE Power India Conference, 2008. POWERCON 2008. Joint International Conference on*, pp. 1-6, October 2008.

[26] H.M. Jeong, H.S. Lee, and J.H. Park, "Application of Parallel Particle Swarm Optimization on Power System State Estimation," *Proc. of IEEE Transm. \& Distrib. Asia Conference*, 2009.

[27] K. Schneider, Z. Huang, B. Yang, M. Hauer, and Y. Nieplocha, "Dynamic state estimation utilizing high performance computing methods," *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pp. 1–6, March 2009.

[28] P. Kundur, *Power system stability and control*, McGraw Hill Inc., New York, 1994.

[29] J. Ye, Z. Liu, and L. Zhu, "The Implementation of a Spatial Parallel Algorithm for Transient Stability Simulation on PC Cluster," *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pp 1489-1492, May 2007.