EPSRC
Engineering and Physical Sciences
Research Council

NSFC
National Natural Science
Foundation of China

Developing Scalable Smart Grid Infrastructure to Enable Secure Transmission System Control

EP/K006487/1

UK PI: Prof Gareth Taylor (BU)

China PI: Prof Yong-Hua Song (THU)

Consortium UK Members: Brunel University (BU), Alstom Grid Ltd (AGL), Intel Corporation (IC), National Grid (NG)

Consortium China Members: Tsinghua University (THU), Sichuan University (SU), CEPRI

Document Title	Comprehensive Report and Survey of Current High Performance Computing and Related Power System Analysis Tools
Document Identifier	
Version	Version 1.0
Work package number	WP1
Task Number(s)	T1.1 &1.2
Distribution	Public
Reporting consortium member	BU, THU, SU and CEPRI
Internal reviewer & review date	Prof G A Taylor (date)

Document Information

EPSRC Project Number	EP/K006487/1	**China Funding Council**	NSFC
Full Title	Comprehensive Report and Survey of Current High Performance Computing and Related Power System Analysis Tools		
Project URL	http://www.brunel.ac.uk/sed/ece/research/bips/epsrc-smart-grids		
Document URL	N/A		
Report Number	T1.1&1.2	**Title**	Experimental feedback on proposed algorithms and sensors positioning (delivery: M35)
			N/A
Work Package Number	WP1	**Title**	Overall Project Familiarization and Detailed Survey

Date of Delivery	**Work Plan**	M6	**Actual**	M6
Status	Version 1.0			
Nature	Internal Report			
Dissemination Level	Consortium			
Author(s) (Partners)	G. Taylor (BU), M. Li (BU) M. Khan (BU), Z. Huang (BU)			
Lead Author	**Name**	Prof. G A Taylor	**E-mail**	Gareth.Taylor@brunel.ac.uk
	Partner	BU	**Phone**	+44 (0) 1895 266610
Abstract	Recently significant advancements have occurred in both High Performance Computing (HPC) and power systems domains. HPC has been improved in the form of cluster computing, multi-core computing, grid computing, cloud computing and graphic processing units. Power systems have also advanced and become more complex and computationally intensive due to the integration of renewable energy sources, the integration of the distributed power generation and storage, inclusion of electric vehicles, advancement in metering infrastructure, deployment of Phasor Measurement Units (PMUs) and the advent of smart grid. This report presents a survey of HPC applications in power systems. The report starts with a brief introduction to the different types of HPC platforms adequate to power system applications. Then existing HPC technologies are described. Finally, the report reviews existing and further potential areas of HPC application in power systems as reported over the last of 5 years.			
Keywords	High Performance Computing, Power Systems			

Contents

Abbreviations

API	Application Programming Interface
CUDA	Compute Unified Device Architecture
DFA	Detrended Fluctuation Analysis
DH	Data Historian
EMS	Energy Management System
GB	Great Britain
GPU	Graphics Processing Unit
HDFS	Hadoop Distributed File System
HPC	High Performance Computing
LAN	Local Area Network
MIMD	Multiple-Instruction Multiple-Data
MPLS	Multi-Protocol Label Switching
OGF	Open Grid Forum
OGSA	Open Grid Service Architecture
PDC	Phasor Data Concentrator
PMU	Phasor Measurement Unit
WAMS	Wide Area Monitoring System

1. Introduction

Recently the power systems have become more complex and computationally intensive due to the advent of the smart grid, rapid deployment of PMUs, integration of the renewable energy sources and inclusion of the electric vehicles. The power system community faces the challenge of finding suitable computation methods to solve the growing computational issues. Such methods can be found in the field of HPC technologies through parallel processing.

This report provides a summary overview of the various HPC technologies that are utilised and developed upon within the context of enhancing the power systems to become a system which is more flexible, secure and controllable. The relevant technology areas which are applicable within the scope of this research collaboration are contained within two subfields: the area of HPC technology, and the area of relevant power system modelling and analysis techniques. The aim of this document is therefore to review the most important HPC technologies that have been applied in the field of Power systems. Moreover, this document reviews the last 5 years research works that have employed HPC for the power systems modelling and analysis.

The remainder of this report is organised as follow: Section 2 provides an overview of trends in HPC platforms. The existing HPC technologies are described in Section 3. Section 4 reviews the potential areas of HPC application in power systems over the last of 5 years. Finally, Section 5 presents a summary of the report.

2. Trends in High Performance Computing

The term High Performance Computing (HPC) refers to the practice of combining computing power in a way that delivers high computing power than one could achieve from a single computer. The HPC technology is used to solve computationally intensive problems that require vast amounts of processing capacity and resources. The goal of the HPC implementation is to minimise computation time of large and complex problems. This section provides an overview of HPC systems which can be organised into the following categories.

2.1 Multi-Core Systems

A multi-core system is based on a MIMD (multiple-instruction multiple-data) architecture that integrates multiple cores into a single chip in order to address the limitations of single processor architecture and provide high computing power on a single chip [1]. The motivation behind the multi-core system isto scale up the performance of system by increasing the number of cores on a single chip (horizontal scale) instead of increasing clock frequency (vertical scale) of a single processor. The acceleration of complex computation on multi-core system is dependent on thread-level parallelization. If the computation is not appropriately parallelised, it does not matter how many core is available for the computation and there will be not enough performance gain [2].

2.2 Graphics Processing Units

Graphics Processing Units (GPUs) as a general purpose and highly parallel HPC system consist of thousands of processing cores for handling multiple tasks simultaneously. The GPU technology is rapidly gaining popularity as a powerful tool for computationally intensive applications. Over the past few years, GPU system has evolved from a fixed-function pipeline processor (graphics processor) into a massively parallel programmable many-core multiprocessors computing architecture. Today, GPU employs hundreds of parallel processor cores executing tens of thousands of parallel threads in order to solve computation problem of applications where computation requirements are high, parallelism is substantial. Currently, GPU technology is the most ubiquitous highly parallel and cost effective processing platform [3][4]. A heterogeneous approach i.e. combination of both the CPU and GPU (co-processing systems) further improves the performance of computations. This is because both processors have complementary attributes which allow computation to be performed efficiently using both types of processors. CUDA (see Section 3.4 for details) is a programming framework that provides facilities where both processors can be employed for a computation. CUDA executes the serial portions of a code on CPU while it executes the parallel portions of the code over GPU cores [5].

2.3 Cluster Computing

Cluster computing is referred to a system where multiple processing nodes and storage devices are connected through redundant inter-connections to form a single integrated high level system [6]. The goal of cluster computing technology is to develop an efficient and cost-effective computing platform that utilises a group of computer resources in order to improve performance and availability of the system over that a single computer system. The computer resources are integrated with the help of high speed network (LAN), hardware (fast switches) and software (middleware). The middleware is employed to hide the complexity of the underlying systems and present them to users and applications as a single unified system instead of cluster of the systems [7]. The efficiency of the cluster-based system is improved due to the integration of distributed load-balancing mechanism, scalability methodologies and parallel processing capabilities. The traditional cluster is transformed into on-demand cluster which provides better reliability, availability and serviceability [6]. The cluster computing has been employed for different applications ranging from commercial applications such as web-service support to scientific applications; however, it is mainly used for scientific and engineering computationally intensive applications. Cluster computing consists of several processing nodes, where one node acts as a server and the remaining nodes act as worker nodes. The server node is responsible for controlling the overall operation, synchronisation and tasks scheduling. The worker nodes perform the actual task execution. The single server architecture of cluster system can be subjected to single point of failure. Therefore, High-Availability (HA) option is included where multiple nodes can be configured as servers in active/passive form.

2.4 Grid Computing

A grid computing is one the most powerful and useful technologies in modern HPC and closely related to cluster computing as mentioned in [8]. Foster et al. [9] define the grid computing as “A system of coordinated resources sharing and problem solving in dynamic and multi-institutional virtual organizations”. A grid computing is a distributed infrastructure and collections of resources such as databases, storage servers, high-speed networks, supercomputers and computer clusters available for solving large-scale modern sciences problems [10] [11]. The objective of grid technology is to share the vast amount of resources that are owned by distinctive individuals through secure and reliable communication [6] [12]. The resources are shared based on a number of factors such as trust, policies and negotiation. There are certain bodies that define standardization for grid related technologies and applications. The most vital standard organization is the Open Grid Forum (OGF) [13]. The OGF has specified several technologies but the most important one is the Open Grid Service Architecture (OGSA) [14]. The OGSA is a distributed interaction and computing architecture that standardise all the services that a grid application may use, for examples, communication and security services and job and resource management services [15].

2.5 Cloud Computing

Cloud computing continues to be an attractive technology and has increasingly gained popularity because of its ability to provide scalable, virtual, shared and on-demand computing resources across the Internet or LAN to small, medium and large enterprises. The resources are delivered through services such as - *Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)* [16]. *IaaS* provides the cloud user with shared storage and computing resources, which include CPU, disk space and memory. *PaaS* provides the cloud user with a platform for testing any applications. While *SaaS* provides the user with software and applications without the user having to worry about licenses and installation, this usually runs on the cloud providers infrastructure and services are deployed through *Private, Public, Community or Hybrid* cloud deployments. The deployments ofcloud services have varying security consideration even for major issues like confidentiality, integrity (of data) and availability. A secure cloud infrastructure protects the application and subsequently the data that would be saved or have been saved on it. The security issues of the cloud computing can be addressed through the trusted cloud computing [17].

3. Existing High Performance Computing Technologies

This section describes the existing high performance computing technologies. The HPC technologies mentioned in this section are open-source and freely available. These technologies are described as follow.

3.1 Message Passing Interface

MPI, the Message Passing Interface, is a standardised and portable message passing system that automatically parallelised computations on both shared memory and distributed memory computing architecture. It is designed by a group of researchers for HPC and scalable parallel computing platforms. The MPI standard defines high-level Application Programming Interfaces (APIs) that not only hide the underlying complexity of code parallelisation from upper-level application programmers but also make the code portable across a wide variety platforms. The source code written for a MPI system can be run on different computers as long as the MPI library is available. For instance, it can be run on distributed memory parallel computers, shared memory parallel computers and it can be also run on a network of computers or as a group of processes running on a single computer. Moreover, it allows the transparent execution of code on heterogeneous systems and across computers with various characteristics (i.e. a machine having messages buffering functionality or no buffering at all functionality). Another feature offered by MPI system is that it allows both point-point communications (e.g. send and receive) and collective communication where multiple processes are involved in a single communication action. There are several implementation of MPI such as OpenMPI, MPICH, HP-MPI and Intel MPI. The HP-MPI and Intel MPI are commercially available whereas OpenPMI and MPICH are open-source and freely available [18]–[20].

3.2 Go Google’s Programming Language

Google’s Go programming language [21], also known as Golang, is an open-source, compiled and multi-paradigm language. The language supports both object-oriented programming and functional notions. The specific features of Go language include concurrent programming, fast compilation, code-formatting, statistically typed, automatic type inference, scalable, code analysis and package member scope visibility. A member that starts with lower case letter is considered as private member while if it starts with upper case letter is considered as public member. However, the Go language has a number of limitations. For example, the abstract class or constructor does not available in the Go language and hence applying default initialization is challenging. Furthermore, due to the lack of operator overloading functionality, multiple initialization functions cannot be defined [22].

3.3 Hadoop MapReduce

MapReduce programming model was originally proposed by Google in 2004. It has become a major computing model for parallel and distributed process of a large scale dataset in computer cluster environment. In MapReduce programming model, the computation is specified in the form of a map function and a reduce function. The map function processes a block of dataset as a (key, value) pair and produces map output in the form of a list of (key, value) pairs. The intermediate values are grouped together based on the same key and then passed to the reduce function. The reduce function takes the intermediate key along with its associated values and processes them to produce a new list of values as final output. The map function and the reduce function are executed independently on allocated resources which shows high parallelism [23].

Hadoop is an open source implementation of MapReduce programming model and has become the foremost computing platform for big data analytics. The Hadoop MapReduce framework is highly scalable and it can be scaled up from a single machine to tens of thousands machines, each of which offering local computation and data storage. The size of the Hadoop cluster can be shrunk or expanded dynamically based on workload. The Hadoop MapReduce framework is developed with fundamental hypothesis that machine failure is common in cluster computing and it should be handled automatically in software level by the framework. Therefore, fault-tolerance and automatically machine failure handling techniques are included in the framework. The Hadoop MapReduce framework mainly includes: MapReduce and Hadoop Distributed File System (HDFS). The MapReduce deals with computation i.e. it executes the map function and reduce function on worker nodes while the HDFS is a scalable file storage system which automatically distributes a large amounts of data across a cluster of nodes [24], [25].

3.4 CUDA

Compute Unified Device Architecture (CUDA) is the most commonly used programming framework available for users to parallelise computation across GPUs cores. It is a framework that automatically parallelises a code written in C, C++, FORTRAN, OpenCL, DirectCompute and any other languages. A program written in CUDA is organised into a host program that consist of one or more serial threads executing on a host CPU and one or more parallel threads running on parallel computing GPUs. The CUDA framework enables the development of massively scalable parallel programs that can be executed on hundreds of processor cores. It provides mechanism to perform computation on GPU, synchronise threads on GPU and efficiently move data between GPU and CPU [3][5].

3.5 GridPACK ™

GridPACK™ [26], [27] is an open-source software framework for designing power system simulation applications for HPC platform. It consists of modules and libraries that automatically set up and distribute power system networks, provide support for modelling the behaviour of individual buses and branches, convert the network models to the corresponding algebraic equation and apply parallel procedures for manipulating and solving large algebraic equations. Moreover, it provides supports to grid application developers for parallel code development. For examples, it provides generic functionality that automatically distributes data in parallel applications and exchange data between processors. It hides the complexity of code parallelization and application developers only focus on physics of their problems. There are also other functionalities, including libraries of parallel matrix and vector operations, linear and non-linear solvers, routines for designing network from standard format input file and functionality for profiling and error management.

4. Applications of High Performance Computing in Power Systems

This section reviews the last 5 years key papers that employed HPC technology to solve complex and computationally intensive power systems problems. Comprehensive survey papers that reviewed the applications of HPC in power systems domain are published in [12], [28].

4.1 Contingency Analysis

Contingency analysis is a well know security analysis function in Energy Management System (EMS) that provides information to operators at control centre about security violations and performs corrective action to eliminate the violations before they arise. Due to the integration of technologies such as renewable energy sources, PMUs, smart grid, AMI and emphasis on multiple contingency analyses (N-K), the computation demand of the contingency analysis has increased to ensure the power system security.

In literature, numerous research studies have been conducted to speed up the computation of contingency analysis through parallel approaches [22], [29]–[31]. For example, Khaitan et al. [22], [30] discussed parallel methods for the N-K contingency analysis. The authors parallelised the computation over 2, 4, 8 and 16 cores using X10 [30] and GO Google’s programming language [22]. The proposed methods achieve more than 14 times speedup compared with the sequential approach. Chen et al. [31] proposed a parallel approach for N-K contingency analysis over thousands of processing cores using counter-based dynamic load balancing scheme. The computation is parallelised across a cluster of nodes with different number of cores. The experimental results show that the proposed approach achieves a speedup of 7877 times using 10240 cores.

4.2 State Estimation

State estimation is an essential tool and widely used by system operators to analyse the status of the power systems in real-time. It has the ability to detect, identify and correct measurement errors. It plays an important role in monitoring and controlling of power systems for consistent operations. However, due to a very large number of nodes in distribution systems, the computation time of state estimation is extremely prolonged which makes the application of state estimation unpractical in power systems.

There are a number of research studies that have significantly speeded up the computation of state estimation algorithms through parallel computation techniques. For example, Chen et al. [32] parallelised the state estimation algorithms using 32 cores machine in order to speed up the computation of algorithms. Gain matrix factorization and QR orthogonal factorization algorithms were tested in the experiments. The experimental results show that the parallel algorithm is 5-20 times faster than the commercial EMS tool. Nusrat et al. [33] proposed a novel overlapping zone-based state estimation method for a large distribution network. The proposed approach first divides the whole network into sub-zones and then calculates state estimation at each local zone using MATLAB solver simultaneously. MPI technology is used to exchange information between the zones. The proposed method improves the computation time from 1.6 times to 3.7 times using a network of 711 nodes and splits the network into 2-4 zones.

4.3 Security and Reliability Assessments

The assessment of security and reliability in power systems is an import component in order to analyse the status of a power network in current and contingency conditions. It can be used to avoid overload, voltage instability and blackout. Real-time security and reliability assessments are computationally intensive because they require a large number of simulations and analysis of many contingency scenarios. Due to a high computation demands, the conventional computing platform is no longer sufficient for the comprehensive security analysis. Huang et al. [34] presented a computation framework that combines the use of phasor data and advance computing platform for dynamic security assessments. The framework parallelised dynamic state estimation algorithms and contingency analysis cases over a cluster of 64 cores in order to assess the power network dynamic security in a predictive manner. The reference work [35] has discussed power system probabilistic and security analysis on commodity high performance computing systems. The work considered distribution probabilistic load flow, transmission probabilistic load flow and AC contingency calculation. A high performance Monte Carlo simulation and AC contingency calculation were parallelised on commodity high computing platform for fast and comprehensive steady state security analysis. The experimental results show that one million load flows can be solved within 4 seconds, which is less than the update time interval of most SCADA systems.

4.4 Wide Area Monitoring Systems

Wide Area Monitoring System (WAMS) is a network of PMUs. The PMUs are used to collect data from various grid substations. The data collected by PMUs are delivered over fast and modern communication channels to local databases called Phasor Data Concentrator (PDC) and then transmitted to a high-capabilities master database called super-PDC. The consolidated data collected by super-PDC are fed into analytics applications such as state estimation, oscillation mode identification, stability assessments, data visualization, real-time monitoring and control [36]–[39]. We have published quite large number of papers [25], [40]–[48] in the domain of WAMS. The work [40] reviewed the developments of WAMS on the Great Britain (GB) transmission system. The work also outlined that new technologies would require to facilitate the availability of real-time data to determine the operating state of the power system at an improved accuracy and resolution compared to that of traditional state estimation. The critical evaluation of WAMS from a GB transmission operator perspective and its future requirements are investigated in [41], [42]. The application of PMUs to estimate power system inertial frequency response and transient event detection and analysis of the GB transmission system are presented in [43], [44]. In WAMS, the data are transmitted to PDC through data communication channels and for establishment of a reliable WAMS, a high performance communication infrastructure is becoming essential. Therefore, the performance of real-time WAMS communication infrastructure with Multi-Protocol Label Switching (MPLS) enabled capability has been evaluated in [46], [47] using discrete event simulation tools. Furthermore, Ashton et al. [45] proposed a method that employs Detrended Fluctuation Analysis (DFA) algorithm for detection of power system transient events captured on small PMU measurements, more specifically for the detection of instantaneous generation losses. As WAMS consists of a large number of PMUs which generates massive amounts of data leading to a major computation challenges. In order to deal with computation challenges, Khan et al. [25] proposed a novel parallel DFA (PDFA) algorithm for fast event detection on massive volumes of PMU data. The approach is implemented in Hadoop MapReduce environment which making the use of a cluster of inexpensive commodity computers. The PDFA is tested and demonstrated in two stages. The first stage provides the details of a laboratory based online setup, using PMUs installed at the domestic supply and configuration of openPDC platform with a localised Data Historian (DH) to collect and store 50 HZ resolution data. The second stage details the application to the WAMS installed on the transmission system of the GB, whereby an offline data mining approach is demonstrated. In the offline data mining, the PDFA work by detrending a dataset of PMU frequency measurements on sample-by-sample sliding window. The window is configured to be 50 samples long, this is to detect events over a 1 second period time. The PDFA first divides the massive dataset into data blocks with 2 MB size each and then a number of Map tasks and Reduce tasks process the data blocks in parallel. The number of Map tasks and Reduce tasks depend upon the number of Map slots and Reduce slots specified in the cluster configuration file. The performance of PDFA is compared with original DFA [45] in term of efficiency and accuracy, using PMU data from the GB WAMS. The performance results show that PDFA is 5.4 times faster than the DFA and it is highly scalable and resilient.

4.5 Transient and Dynamic Stability Analysis

Stability analysis can be used as a preventive approach that detects the system stability margin in normal condition and predicts the system performance if disturbance occurs. Stability analysis includes dynamic stability analysis, which deals with power system response when continuous small disturbance is occurred, and transient stability analysis which concerns with power system response in the case of large system disturbances [49]. However, the stability analysis of a large interconnected power system is limited to analyse a large number of contingencies in near real-time due to a high computation cost.

Parallel technique and HPC technology can be used to speed up the computation of stability analysis. For example, a massive parallel GPUs-based architecture for large-scale transient stability simulation has been investigated in [50]. The authors employed three levels of parallelism: algorithm-level, task-level and data-parallelism in order to speed up the computation of transient stability analysis. Smith et al. [51] utilised HPC for on-line transient stability analysis. The authors used MPI-based distributed memory parallelization approach and parallelised thousands of contingencies on 64 to 4096 cores. The experimental results show that the parallel approach provides high scalability and computation gain. Other works [52][53] presented the parallel implementation of dynamic simulation for large power system in order to improve its computation efficiency through HPC for real-time analysis. The simulation is parallelised over 64 cores and achieved computation speed suitable for real-time.

5. Conclusion

The current power systems have become more complex and computationally intensive. The traditional technologies are unable to deal with computationally intensive applications. The HPC technologies can be deployed to solve computationally intensive problems. This report has presented an overview of HPC technologies that can be suitable for complex and computationally intensive applications. Moreover, this report reviews the last five years papers that have employed HPC for power system applications.

References

[1] X.-H. Sun, Y. Chen, and S. Byna, “Scalable Computing in the Multicore Era,” in *In Proc. of International Symposium on Parallel Algorithms, Architechures and Programming*, 2008, pp. 1–18.

[2] K. Bosschere, W. Luk, X. Martorell, N. Navarro, M. O’Boyle, D. Pnevmatikatos, A. Ramirez, P. Sainrat, A. Seznec, P. Stenström, and O. Temam, “Transactions on High-Performance Embedded Architectures and Compilers I,” P. Stenström, Ed. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 5–29.

[3] S. Potluri, D. Bureddy, H. Wang, H. Subramoni, and D. K. Panda, “Extending OpenSHMEM for GPU Computing,” in *Parallel Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, 2013, pp. 1001–1012.

[4] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, “GPU Computing,” *Proc. IEEE*, vol. 96, no. 5, pp. 879–899, May 2008.

[5] J. Nickolls and W. J. Dally, “The GPU Computing Era,” *Micro, IEEE*, vol. 30, no. 2, pp. 56–69, Mar. 2010.

[6] H. Hussain, S. U. R. Malik, A. Hameed, S. U. Khan, G. Bickler, N. Min-Allah, M. B. Qureshi, L. Zhang, W. Yongji, N. Ghani, J. Kolodziej, A. Y. Zomaya, C.-Z. Xu, P. Balaji, A. Vishnu, F. Pinel, J. E. Pecero, D. Kliazovich, P. Bouvry, H. Li, L. Wang, D. Chen, and A. Rayes, “A Survey on Resource Allocation in High Performance Distributed Computing Systems,” *Parallel Comput.*, vol. 39, no. 11, pp. 709–736, Nov. 2013.

[7] R. Buyya, T. Cortes, and H. Jin, “Single System Image (SSI),” *Int. J. High Perform. Comput. Appl.*, vol. 15, no. 2, pp. 124–135, 2001.

[8] M. Irving, G. Taylor, and P. Hobson, “Plug in to grid computing,” *Power Energy Mag. IEEE*, vol. 2, no. 2, pp. 40–44, Mar. 2004.

[9] I. Foster, C. Kesselman, and S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” *Int. J. High Perform. Comput. Appl.*, vol. 15, no. 3, pp. 200–222, Aug. 2001.

[10] F. Berman, G. Fox, and A. J. G. Hey, *Grid Computing: Making the Global Infrastructure a Reality*. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[11] "Grid Computing," *wikipedia.org*. [Online]. Available: http://en.wikipedia.org/wiki/Grid_computing. [Accessed: 29-Nov-2015].

[12] R. C. Green, L. Wang, and M. Alam, "High performance computing for electric power systems: Applications and trends," in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1–8.

[13] "Open Grid Forum." [Online]. Available: https://www.ogf.org/ogf/doku.php. [Accessed: 29-Nov-2015].

[14] "Open Grid Services Architecture." [Online]. Available: http://en.wikipedia.org/wiki/Open_Grid_Services_Architecture. [Accessed: 29-Apr-2015].

[15] M. Li and M. Baker, *The grid core technologies*. John Wiley & Sons, 2005.

[16] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft) Recommendations of the National Institute of Standards and Technology," *NIST Spec. Publ.*, vol. 145, no. 6, p. 7, 2011.

[17] M.-J. Sule, M. Li, G. A. Taylor, and S. Furber, "Deploying trusted cloud computing for data intensive power system applications," in *Power Engineering Conference (UPEC), 2015 50th International Universities*, 2015, pp. 1–5.

[18] B. Barney, "Message Passing Interface (MPI)," *Lawrence Livermore National Laboratory*. [Online]. Available: https://computing.llnl.gov/tutorials/mpi/. [Accessed: 09-Dec-2015].

[19] S. Marc, O. Steve, H. Steven, W. David, and D. Jack, *MPI: The Complete Reference*. London, England: The MIT Press, 1996.

[20] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall, "Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation," in *Proceedings, 11th European PVM/MPI Users' Group Meeting*, 2004, pp. 97–104.

[21] Google, "The Go Programming Language." [Online]. Available: https://golang.org/. [Accessed: 04-Jan-2016].

[22] S. K. Khaitan and J. D. McCalley, "PARAGON: an approach for parallelization of power system contingency analysis using Go programming language," *Int. Trans. Electr. Energy Syst.*, vol. 25, no. 11, pp. 2909–2920, 2015.

[23] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[24] M. Khan, Y. Jin, M. Li, Y. Xiang, and C. Jiang, "Hadoop Performance Modeling for Job Estimation and Resource Provisioning," *Parallel Distrib. Syst. IEEE Trans.*, vol. PP, no. 99, p. 1, 2015.

[25] M. Khan, P. M. Ashton, M. Li, G. A. Taylor, I. Pisica, and J. Liu, "Parallel Detrended Fluctuation Analysis for Fast Event Detection on Massive PMU Data," *Smart Grid, IEEE Trans.*, vol. 6, no. 1, pp. 360–368, Jan. 2015.

[26] "GridPACK." [Online]. Available: https://www.gridpack.org. [Accessed: 01-Dec-2015].

[27] B. Palmer, W. Perkins, Y. Chen, S. Jin, D. Callahan, K. Glass, R. Diao, M. Rice, S. Elbert, M. Vallem, and Z. Huang, “GridPACK: A framework for developing power grid simulations on high-performance computing platforms,” *Int. J. High Perform. Comput. Appl.*, pp. 1–18, 2015.

[28] R. C. Green, L. Wang, and M. Alam, “Applications and Trends of High Performance Computing for Electric Power Systems: Focusing on Smart Grid,” *Smart Grid, IEEE Trans.*, vol. 4, no. 2, pp. 922–931, Jun. 2013.

[29] Z. Huang, Y. Chen, and D. Chavarria-Miranda, *High-Performance Computing for Real-Time Grid Analysis and Operation*. Berlin Heidelberg: Springer Berlin Heidelberg, 2013.

[30] S. K. Khaitan and J. D. McCalley, “Achieving Load-balancing in Power System Parallel Contingency Analysis Using X10 Programming Language,” in *Proceedings of the Third ACM SIGPLAN X10 Workshop*, 2013, pp. 20–28.

[31] Y. Chen, Z. Huang, and M. Rice, “Evaluation of Counter-Based Dynamic Load Balancing Schemes for Massive Contingency Analysis on over 10,000 Cores,” in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, 2012, pp. 341–346.

[32] Y. Chen, S. Jin, M. Rice, and Z. Huang, “Parallel state estimation assessment with practical data,” in *Power and Energy Society General Meeting (PES), 2013 IEEE*, 2013, pp. 1–5.

[33] N. Nusrat, P. Lopatka, M. R. Irving, G. A. Taylor, S. Salvini, and D. C. H. Wallom, “An Overlapping Zone-Based State Estimation Method for Distribution Systems,” *Smart Grid, IEEE Trans.*, vol. 6, no. 4, pp. 2126–2133, Jul. 2015.

[34] Z. Huang, R. Diao, S. Jin, and Y. Chen, “Predictive dynamic security assessment through advanced computing,” in *PES General Meeting | Conference Exposition, 2014 IEEE*, 2014, pp. 1–5.

[35] T. Cui and F. Franchetti, “Power System Probabilistic and Security Analysis on Commodity High Performance Computing Systems,” in *Proceedings of the 3rd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*, 2013, pp. 2:1–2:10.

[36] M. Hurtgen and J.-C. Maun, “Advantages of Power System State Estimation using Phasor Measurement Units,” 2008. [Online]. Available: http://www.pscc-central.org/uploads/tx_ethpublications/pscc2008_520.pdf. [Accessed: 05-May-2015].

[37] N. Kakimoto, M. Sugumi, T. Makino, and K. Tomiyama, “Monitoring of interarea oscillation mode by synchronized phasor measurement,” *Power Syst. IEEE Trans.*, vol. 21, no. 1, pp. 260–268, Feb. 2006.

[38] M. Zhao and A. Abur, “Multi area state estimation using synchronized phasor measurements,” *Power Syst. IEEE Trans.*, vol. 20, no. 2, pp. 611–617, May 2005.

[39] B. Milosevic and M. Begovic, “Voltage-stability protection and control using a wide-area network of phasor measurements,” *Power Syst. IEEE Trans.*, vol. 18, no. 1, pp. 121–127, Feb. 2003.

[40] P. M. Ashton, G. A. Taylor, M. R. Irving, A. M. Carter, and M. E. Bradley, “Prospective Wide Area Monitoring of the Great Britain Transmission System using Phasor Measurement Units,” in *Power and Energy Society General Meeting, 2012 IEEE*, 2012, pp. 1–8.

[41] P. M. Ashton, G. A. Taylor, A. M. Carter, and I. Pisica, “Critical evaluation of wide area monitoring

systems from a GB transmission system operator perspective," in *Universities Power Engineering Conference (UPEC), 2012 47th International*, 2012, pp. 1–6.

[42] P. M. Ashton, G. A. Taylor, and A. M. Carter, "Future wide area monitoring requirements for the GB transmission system," in *AC and DC Power Transmission (ACDC 2012), 10th IET International Conference on*, 2012, pp. 1–6.

[43] P. M. Ashton, G. A. Taylor, A. M. Carter, M. E. Bradley, and W. Hung, "Application of phasor measurement units to estimate power system inertial frequency response," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, 2013, pp. 1–5.

[44] P. M. Ashton, G. A. Taylor, and A. M. Carter, "Transient event detection and analysis of the GB transmission system using synchrophasor measurements," in *Power Engineering Conference (UPEC), 2013 48th International Universities'*, 2013, pp. 1–6.

[45] P. Ashton, G. A. Taylor, M. R. Irving, I. Pisica, A. Carter, and M. E. Bradley, "Novel Application of Detrended Fluctuation Analysis for State Estimation Using Synchrophasor Measurements," *Power Syst. IEEE Trans.*, vol. 28, no. 2, pp. 1930–1938, 2013.

[46] M. Golshani, G. A. Taylor, I. Pisica, and P. M. Ashton, "Performance evaluation of MPLS-enabled communications infrastructure for wide area monitoring systems," in *AC and DC Power Transmission, 11th IET International Conference on*, 2015, pp. 1–7.

[47] M. Golshani, G. A. Taylor, I. Pisica, P. M. Ashton, C. Chen, J. Liu, and J. Lin, "Performance analysis of wide area network communications using discrete event simulation tools," in *Power System Technology (POWERCON), 2014 International Conference on*, 2014, pp. 1098–1105.

[48] M. Khan, M. Li, P. Ashton, G. Taylor, and J. Liu, "Big data analytics on PMU measurements," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2014 11th International Conference on*, 2014, pp. 715–719.

[49] J. Zhang and D. Chen, "On the application of Phasor Measurement Units to power system stability monitoring and analysis," in *Power and Energy Conference at Illinois (PECI), 2012 IEEE*, 2012, pp. 1–6.

[50] V. Jalili-Marandi, Z. Zhou, and V. Dinavahi, "Large-Scale Transient Stability Simulation of Electrical Power Systems on Parallel GPUs," *Parallel Distrib. Syst. IEEE Trans.*, vol. 23, no. 7, pp. 1255–1266, Jul. 2012.

[51] S. Smith, C. Woodward, L. Min, C. Jing, and A. Del Rosso, "On-line transient stability analysis using high performance computing," in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, 2014, pp. 1–5.

[52] Z. Huang, S. Jin, and R. Diao, "Predictive Dynamic Simulation for Large-Scale Power Systems through High-Performance Computing," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, 2012, pp. 347–354.

[53] S. Jin, Z. Huang, R. Diao, D. Wu, and Y. Chen, "Parallel implementation of power system dynamic simulation," in *Power and Energy Society General Meeting (PES), 2013 IEEE*, 2013, pp. 1–5.