Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A High Performance Computing Platform for Performing High-Volume Studies with Windows-based Power Grid Tools

Yousu Chen*, Zhenyu Huang**

** Pacific Northwest National Laboratory, Battelle Seattle Research Center, Seattle, WA, 98109 USA (Tel: 206-528-3062; e-mail: yousu.chen@pnnl.gov).*
*** Pacific Northwest National Laboratory, P.O. Box 999, Richland 99352 USA (e-mail: zhenyu.huang@pnnl.gov)}*

Abstract: Serial Windows-based programs are widely used in power utilities. For applications that require high-volume simulations, the sequential runtime can be in the order of days or weeks. The lengthy runtime, along with the availability of low-cost hardware, is leading utilities to seriously consider high performance computing (HPC) techniques. However, the majority of the HPC computers are Linux-based, and many HPC applications have been custom developed without considering existing simulation tools and ease of use. This has created a technical gap for applying HPC-based tools to today's power grid studies using Windows-based tools. To fill this gap and accelerate the acceptance and adoption of HPC for power grid applications, this paper presents a prototype of a generic HPC platform for running Windows-based power grid tools on the Linux-based HPC environment. The preliminary results show that the runtime can be significantly reduced from weeks to hours to improve work efficiency.

Keywords: High performance computing, power grid, power system operation

1. INTRODUCTION

Single-processor-based—especially Windows-based—programs are widely used in today's power utilities. For example, General Electric Energy Consulting's Positive Sequence Load Flow software (GE PSLF) [General Electric online], PowerWorld Simulator [PowerWorld online], and Dynamic Security Assessment (DSA) Tools™ [PowerTech online] are extensively used at power utilities for transmission planning and operation studies. Given the increasing complexity and size of today's power system, the processing time for certain applications can be on the order of days, or even weeks, for applications such as a full assessment of transient stability. The advances in renewable technologies and more pervasive control technologies will lead to an increasing number of scenarios with more detailed network representation. This situation will further increase the computational time of running these simulations. This lengthy runtime with single central processing units (CPU), along with the availability of low-cost hardware, is leading utilities to seriously consider high performance computing (HPC) to reduce runtime and increase work efficiency.

While HPC techniques have been identified as a key driver for enabling fast computation, the vast majority of the HPC computers are still Linux-based. From the authors' observations, power engineers are more familiar with Windows-based interface, rather than the command-line type in the Linux environment, to use a software tool. Many power system researchers have presented their work in the area of HPC for different power system applications. References [Falco 1997] and [Green 2013] provide a good general review on the HPC applications for power system analysis. A few examples are: [Luo 2004] discusses a parallel approach to compute power flow; [Huang 2009] and [Chen 2010] describe a framework for massive contingency analysis based on a counter-based dynamic load balancing scheme; [Falco 1995 and [Nieplocha 2006] present parallel state estimation results on distributed HPC machine and shared-memory based machine, respectively; [Shu 2005] presents a multilevel partition scheme and a hierarchical block bordered diagonal form algorithm, for parallel transient stability analysis; and [Jalili-Marandi 2010, 2012] summarize their work on graphics processing units.

While all the examples above showed some promising and attractive results, many of these kinds of HPC applications have been custom developed external to the core simulation engine without consideration for ease of use, in particularly for power system engineers who are the targeted end-users. Most of them are not familiar with HPC techniques and/or the Linux environment.

The situation described above has created a technical gap for applying HPC-based tools to today's power grid studies: the power of HPC has been recognized, but the barrier of using HPC is high in terms of learning a completely new tool in an unfamiliar environment. Furthermore, power engineers normally have their customized scripts/programs for different applications. If using a new HPC tool requires significant time to modify those existing scripts, additional overhead and investment would be introduced that could slow down the adoption of HPC tools.

To accelerate the acceptance and adoption of HPC for power grid applications, the authors developed the concept of a generic HPC platform which can not only improve

computational performance, but is easily used in the users' familiar environment, without modifying their current working programs/scripts. The objective of this paper is to present a generic HPC platform for facilitating Windows-based power system applications for high-volume tasks. The GE PSLF tool for dynamic simulation will be used as an example of a Windows-based function to show the usability and effectiveness for the platform.

The rest of the paper is organized as follows: Section 2 discusses the benefit of accelerating high-volume dynamic simulations for improving path rating studies. Section 3 describes the general procedure for using PSLF to run a sequential, single-core-based simulation. Section 4 introduces the components in the proposed HPC platform. Section 5 presents some preliminary simulation results for two sets of test cases using Western Electricity Coordinating Council (WECC)-approved power flow models. Section 6 discusses the advantages and future work of the HPC platform.

2. HIGH-VOLUME STUDY EXAMPLE: PATH RATING STUDY

Power system path rating studies include a large amount of dynamic simulations. Dynamic simulation solves a set of differential algebraic equations that describe the electro-mechanical interaction of generators, as well as power electronic devices and their controllers. These simulations determine the time-series dynamic trajectory of a power grid when it is subject to disturbances such as short-circuit faults, generator tripping, or line switching. It plays a critical role in power system transient stability studies for power system operation, planning, and control.

An important objective of computing a transient stability limit is to determine whether the system can withstand a set of contingencies in the dynamic mode. This is a key element in path rating studies. Determining transmission path rating is a task of great importance to the reliable and efficient operation of a bulk power system. Traditionally, path rating studies are performed off-line with assumed worst-case operating conditions for different seasons. The resulted rating could be lower than the actual rating of the path in real-time conditions, as the assumed operating conditions are intentionally selected to be conservative. One important reason for the off-line conservative rating studies is the computational time needed to determine path ratings. In today's studies, power system engineers need to study hundreds of dynamic cases for a few hundred contingency configurations. For example, transmission planning studies for Bonneville Power Administration (BPA) include 250 contingencies for 200 different base cases to study transient stability limits, which can take weeks to complete a full assessment.

The disadvantage of using a conservative rating is obvious: it can cause overly conservative operation of a transmission path and therefore the transmission assets are being under-utilized. If the path rating can be computed in hours or near real-time, then more accurate path ratings can be applied to grid operation for more efficient congestion management. It is estimated that a 1000-MW rating increase for one transmission path could generate annual revenue of millions of dollars. Path rating studies are a good application example to show the benefit of using HPC techniques.

While there are many researchers working on how to parallelize an individual dynamic simulation run, power engineers at utilities are mostly using commercial Windows-based power application tools that are designed for the sequential single-CPU environment. Most of the commercial tools have been available for more than a decade, well-tested, and optimized for what they are designed to do. Therefore, a task-level HPC program should be a good starting point for HPC adoption. Task-level means that each individual task run is still in a sequential mode, but the tasks are assigned to different CPUs. Path rating studies are a perfect candidate for task-based execution.

3. DYNAMIC SIMULATION TOOL MECHANISM IN THE GL PSLF

GE PSLF is one of the popular power simulation tools widely used in the power utilities to perform dynamic simulation. Inside GE PSLF, a toolbox called "DYTOOLS" is used. This tool has a built-in transient stability batch processor for running multiple cases and contingencies written in EPCL, the PSLF's user programming language. This batch processor has four main functions: 1) loading power flow databases; 2) loading dynamic model databases; 3) initializing the dynamic simulations; and 4) running the dynamic simulations. The batch simulation control file contains all the information needed to run transient stability analysis. The key control parameters include: powerflow database; dynamic model database; EPCL scripts for pre-run, in-run, post-run, and fault conditions; the simulation ending time; output sampling rate; and simulation time step. An example of the control file can be found in Fig. 1**Error! Reference source not found.**.

```
DATACHECK     0                    /* DATA CHECK FLAG:  (0) OFF        (1) READ ONLY */
MODSTAT       0                    /* RUN MODEL STATUS: (0) DON'T RUN  (1) RUN       */
TIMEUNIT      0                    /* TIME UNITS FLAG:  (0) SECONDS    (1) CYCLES    */
ENDTIME       3.0                  /* TOTAL SIMULATION END-TIME                      */
CLEARTIME     3.00  0.0667  0.1000 /* TRANSIT END TIME; CLEARING TIME 1; CLEARING TIME 2  */
SAVEPATH      CASES\               /* FOLDER WHERE *.SAV ARE LOCATED      */
DYDPATH       DYDS\                /* FOLDER WHERE *.DYD & EPCL *.P  ARE LOCATED */
CHFPATH       CHANS\               /* FOLDER WHERE CHANNEL *.CHF OUTPUTS ARE LOCATED*/
SORTDYD       0  1                 /* SORT FLAG FOR *.DYD FILES */
FIXDATA       1                    /* FIX BAD DATA FLAG */
NUMPLOT       2  2  2              /* PLOT NUMBER INTERVALS OF PRE-, DURING-, AND POST-FAULT */
CONVMON       0                    /* CONVERGENCE MONITOR FLAG: (0) OFF, (1) ON            */
STEPSIZE      0.0041667            /* TIME STEP SIZE (SECONDS) */

CASES
# SAVE CASE NAME   PRE-EPCL   POST-EPCL   IN-RUN EPCL   FAULT EPCL   CHANNEL OUTPUT       DYD[1]       DYD[2]
OC1000.SAV         NONE        NONE        NONE          WECC_F_1.P   OC_1000_F_1.CHF     WECC2.DYD    PMU.DYD
OC1000.SAV         NONE        NONE        NONE          WECC_F_2.P   OC_1000_F_2.CHF     WECC2.DYD    PMU.DYD
OC2000.SAV         NONE        NONE        NONE          WECC_F_3.P   OC_1000_F_3.CHF     WECC2.DYD    PMU.DYD
OC2000.SAV         NONE        NONE        NONE          WECC_F_4.P   OC_1000_F_4.CHF     WECC2.DYD    PMU.DYD
END
```

Fig. 1: An example of PSLF control file for batch transient simulation runs.

These simulations can be launched in a batch mode without launching the PSLF Graphical User Interface (GUI). To setup high-volume transient stability analysis simulations, the user only needs to specify all the control parameters in the control file in a pre-defined tabular form representation.

GE PSLF DYTOOLS reads all the control parameters at the beginning of the simulation and runs each case one by one,

sequentially. Depending on the complexity of power flow models and the dynamic models, the simulation time for an individual 30-second simulation can range from 60 seconds to a few minutes. A transient stability study normally requires running a few hundred cases for each pre-selected contingency, while the number of the pre-selected contingencies can be in the order of hundreds to thousands. Such a large number of runs can take weeks to complete.

This batch processing tool, and many others, is well-tested and trusted by power engineers. The HPC platform presented in this paper leverages these tools but changes the sequential process into a task-level parallel environment to reduce the total simulation time.

4. THE STRUCTURE OF THE HPC PLATFORM

Based on the motivation above, a prototype of HPC platform for Windows-based tools has been developed. It includes four main functions: 1) split tasks to available cores using a pre-processor; 2) launch Windows-based power grid tools in Linux environment with the help of the "wine" program [win online]; 3) submit jobs on the HPC machines using an HPC resource management tool; and 4) collect output files and perform post-processing. The general view of the platform is shown in Fig. 2.

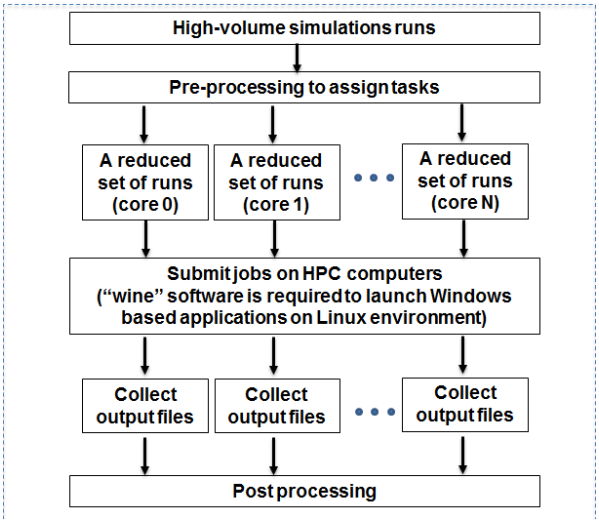

Fig. 2: The structure of the HPC-based platform for running high-volume studies with Windows-based power grid tools.

The first step to reduce simulation runtime is to optimally distribute the simulation runs to the available cores. As discussed in Section 3, GE PSLF DYTOOLS relies on the control file to setup the running environment. A pre-processing script written in Perl has been developed to create one control file per available core to split tasks and setup input/output file paths. At the current stage, the same number of cases is evenly distributed to each core for the purpose of proving the concept. A dynamic load balancing scheme [Huang 2009] based on the availability of cores should have better performance than the current implementation.

The second step is to run Windows-based programs in Linux environment. A virtual machine is one available option for this purpose. Typically, the virtual machine emulates a physical computing environment in software. To run multiple PSLF simulations on independent hardware would require multiple licenses for operating systems and analysis tools, which makes this option too expensive. Another option is to use the third-party software "wine" to install and execute Windows-based applications. "Wine" software can enable Linux, Mac, FreeBSD, and Solaris users to run Windows applications in those environments.

The next step is to submit jobs on an HPC machine using Simple Linux Utility for Resource Management (SLURM). After all simulations are completed, the output files can be collected at a specified location, where post-processing tasks run to extract and report the critical information for further analysis. All of these steps do not require a modification of existing EPCL scripts, which allows users to utilize the power of HPC computers with a minimum learning curve.

Some case studies results are reported in the following section to further prove the effectiveness of the HPC platform.

5. CASE STUDIES

The prototype of the HPC platform has been developed on the Olympus HPC computer, supported by the PNNL Institutional Computing (PIC) program [PIC online]. The Olympus machine has approximately 22,100 cores in 692 nodes. Each node is dual socket with 16 cores per socket Interlagos processors running at 2.1 GHz (32 cores/node). Each node has 64 GB of 1600 MHz memory (2GB/socket). The operating system running on Olympus is Red Hat Enterprise Linux Client release 5.7. A QDR Infiniband high speed network (40 GB/s) is used for internal communication.

Two sets of cases were studied to show the performance of the HPC platform. The detailed test case description and test results are presented in the next subsection.

- Test case 1: one dynamic model, 160 contingencies;
- Test case 2: three different dynamic models for a total of 500 dynamic contingency cases to mimic a typical transmission planning study. The size of the output files (I/O operation) is different for different dynamic models.

5.1 Test case 1: one power flow model and one dynamic model

A 2009 approved WECC operating case contains approximately 16,000 buses, 3,200 generators 14,000 lines, and 6,330 transformers. There are 160 contingencies selected: 117 three-phase to ground faults followed by line tripping, sorted by line flow; 34 load loss contingencies sorted by load size; and 9 shunt trips, sorted by shunt size. The simulation length is 30-seconds with a time step of 0.25 cycles of 60 Hz.

The control file has been pre-processed automatically for testing the code with 1, 2, 4, 8, 16, 32, 80, and 160 cores. The same number of cases was assigned to each core for each run. A part of the HPC job submission file is listed below:

```
#sbatch -t 01:30:00
#sbatch –p shared
#sbatch –n 1
wine C:/upslf181/jre/bin/java -jar C:/upslf181/jclasses/pslf.jar runEpcl_t80_80.bat
```

The first three lines are SLURM commands for setting time limits, partition running, and the number of requested cores. The last line is to launch PSLF runs. The configuration of each run is defined in the corresponding control file, invoked by *runEpcl_t80_80.bat.*

The test results for these 160 dynamic contingency cases are listed in Table 1, and the curve of the speedup vs. number of cores is shown in Fig. 3.

Table 1: Computational time and speedup vs. the number of cores for 160 dynamic contingency cases

# of cores	Time (min)	Speedup
1	535	1.00
2	280	1.91
4	150	3.57
8	93	5.75
16	44	12.16
32	28	19.11
80	15	35.67
160	12	44.58

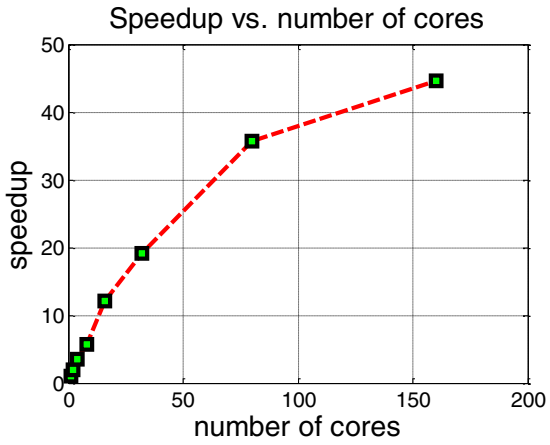

Fig. 3: Speedup vs. number of cores for test case 1.

The following observations are obtained for test case 1:

- The accuracy of the output files is the same as those in the Windows-based version.
- The computational time for running all 160 dynamic contingency simulations can be reduced from about 9 hours to 12 minutes, which makes near-real-time path rating studies possible. It also proves the usability and effectiveness of this HPC platform.
- The speedup slows down when the number of cores is greater than 16. There are two main possibilities for this unfavorable observation: 1) there is more communication overhead when more cores are requested; 2) the HPC resources, mainly memory, at each node are saturated when more cores are requested.

Therefore, to achieve better performance, hardware optimization/resource management is needed. Better speedup is expected when the number of simulation cases is large [8].

5.2 Test case 2: multiple power flow cases and dynamic models

For the second test, a 2012 approved WECC operating case with about 18,200 buses, 3,500 generators 15,200 lines, and 7,200 transformers was used. This case is similar in complexity to that in test case 1 (2009), though the size is slightly larger. It was selected for the availability of the different dynamic models that are described in the next paragraph.

Three different dynamic models were used for this test case study. The main difference is in regards to load modeling details and the size of output files. The first dynamic model contains a normal PSLF dynamic load model (denoted by "DYD"), the second one contains the composite load model (CMPLD) with full data output, and the third one contains the composite load model, with partial data output.

Two simulation time lengths were studied, 30 seconds and 75 seconds, with the same simulation time step of 0.25 cycles. Therefore, there are six individual cases as shown in Table 2. The computational time and the size of the output file for each case can be found in Table 2.

Table 2: The execution time and the size of the output file for the six individual cases.

Simulation length (sec)	Dynamic model	Execution time (sec)	Output size (MB)
30	Normal DYD	330	174
30	CMPLD with full outputs	950	215
30	CMPLD with partial outputs	938	124
75	Normal DYD	732	419
75	CMPLD with full outputs	3747	833
75	CMPLD with partial outputs	3707	480

From Table 2, we can first see that:

- The composite load model requires more simulation time than the normal DYD model, especially for the 75-second simulation. The reason is that for the 75-second timeframe, some slow dynamic actions, such as those of load tap changers, start to take effect after approximate

40 seconds and can induce a noticeable slowdown in the simulation time.

- There is no significant difference in the computational time between CMPLD with full outputs and CMPLD with partial outputs. Therefore, the time associated with output operations is not as significant as the pure simulation time.

Based on these six individual cases, contingency studies were performed to evaluate the impact of different dynamic models, different data output sizes, and different simulation lengths on the platform. Two sets of 500 contingency cases were simulated with lengths of 30 seconds and 75 seconds, respectively. Each set included a mixture of the three dynamic models. Different numbers of cores were used for the simulation. Completing all 500 simulations with a small number of cores requires an exorbitant amount of time. Of most interest to power utilities is small HPC computers with 64 cores or less, given the availability of commodity HPC computers. Therefore, these 500 cases are tested with 20, 32, and 64 cores on the PIC machine.

One of the observations from test case 1 is that the hardware resource management can be a key factor impacting speedup. The resource on each HPC node is not adequate enough to utilize the full power of all 32 cores in one node. Thus, in performing the 500-case simulation, to make more resources available to a single core, only eight, instead of 32, cores per node were requested to better optimize hardware resources. This "hardware optimization" mainly depends on the memory requirements of the simulation cases. More systematic hardware optimization requires further studies and would be important guidance for hardware procurement, which will be reported in a future companion paper.

The performance with 30-second simulation cases and 75-second simulations is shown in Table 3 and Table 4, respectively. The execution time with one core in these two tables is estimated based on the individual runtime shown in Table 2.

Table 3: The computational performance for 30-second simulation cases.

# of cores	Time (hrs)	Speedup
1	105 (estimated)	1.0
20	9	11.7
32	4.3	24.4

Table 4: The computational performance for 75-second simulation cases.

# of cores	Time (hrs)	Speedup
1	385 (estimated)	1.0
64	7	55

The speedup achieved in test case 2 meets our expectation. For 30-second cases, the execution time for 500 cases can be reduced from about 4.5 days to 4 hours; while for 500 75-second cases, the simulation time can be reduced from 16 days to 7 hours. Comparing against the numbers from test case 1 (Table 1), a better speedup was obtained because of more hardware resources requested in test case 2 and more runs that were performed.

All these studies only require a minor change on existing EPCL scripts in specifying the name of the corresponding file, which can be automatically done by a pre-processor. Such a pre-processor has been developed with the function of creating individual control files based on the number of cases and available computer cores. Therefore, use of this HPC platform is very straightforward in today's simulation environment at power companies, even if users are not familiar with HPC and/or the Linux environment. The easy-to-use feature and the speedup gained from this platform can significantly reduce the time required to compute large-scale studies. Examples of analysis that would benefit from this implementation include: online security assessment to enhance situational awareness; near-real-time path rating studies for more efficient congestion management; and transmission planning studies with much less turnaround time.

While the dynamic simulation tool in GE PSLF is used as an example for this platform, other PSLF-based or non-PSLF-based simulations can be run on this platform as well. The platform is designed to be generic in nature and can be extended to any other Windows-based power grid tools, as long as they can be launched in a batch mode by "wine."

6. CONCLUSION

This paper presents a generic HPC platform for running a large amount of simulations with Windows-based power grid tools on a Linux-based HPC computer. Test results using WECC approved power flow cases have shown excellent speedup performance, which is able to reduce the simulation time from hours to minutes and from weeks to hours. More computer cores will further reduce the simulation time. In the meantime, the platform allows the users to continue use their existing scripts/programs, minimizing the impact on the engineer's current workflow process. The runtime reductions enable engineers to focus on the analysis of results rather than the setup and processing of results.

Future work for enhancing this platform includes a consistent workflow for faster runtimes, better job management, easier use with GUI, more effective post-processing data analysis, and more efficient hardware optimization. The platform is generic in nature and can be applied to running most Windows-based power grid tools on HPC hardware. This can result in highly improved work efficiency to aid grid operation and planning.

Power utilities have recognized the potential of HPC techniques, but most HPC computers use the Linux operating system. Windows-based applications are dominant for power grid studies, which slows the adoption of HPC techniques for

utilities. The development of this HPC platform allows utilities to use HPC techniques with minimal efforts. The benefits utilities will gain from this platform can facilitate the acceptance and adoption of HPC for power grid applications.

ACKNOWLEDGEMENT

This work is supported by the U.S. Department of Energy through its Advanced Grid Modelling Program. The Pacific Northwest National Laboratory (PNNL) is operated by Battelle for the U.S. Department of Energy under Contract DE-AC06-76RL01830.

REFERENCES

Chen, Y., Huang, Z. and Chavarria-Miranda D. (2010) *Performance evaluation of counter-based dynamic load balancing schemes for massive contingency analysis with different computing environments*. in Proc. IEEE Power Energy Society General Meeting, 2010.

Dynamic Security Assessment (DSA) Tools. [online]. Available: http://www.dsatools.com/.

Falcao, D.M, Wu, F.F.and Murphy, L. (1995). *Parallel and distributed state estimation*. IEEE Trans. Power Syst, vol.10, no.2, pp.724-730.

Falcao, D.M. (1997). *High performance computing in power system applications*. in Selected Papers from the 2nd International Conference on Vector and Parallel Processing. London, U.K. Springer-Verlag, pp. 1–23.

General Electric Energy Consulting Positive Sequence Load Flow (PSLF). [online]. Available: http://geenergyconsulting.com/practice-area/software-products/pslf.

Green, R, Wang, L. and Alam, M. (2013). *Applications and Trends of High Performance Computing for Electric Power Systems: Focusing on Smart Grid*. IEEE Transactions on Smart Grid, vol. 4, No. 2, June.

Huang, Z., Chen, Y., and Nieplocha, J. (2009). *Massive contingency analysis with high performance computing*. in Proc. IEEE Power Energy Society General Meeting.

Jalili-Marandi, V. and Dinavahi, V. (2010). *SIMD-based large-scale transient stability simulation on the graphics processing unit*. IEEE Transactions on Power Systems, vol. 25, no. 3, pp. 1589–1599.

Jalili-Marandi, V., Zhou, Z. and Dinavahi, V. (2012). *Large-scale transient stability simulation of electrical power systems on parallel GPUs*. IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 7, pp. 1255–1266.

Luo, P., Yang, F., and Rao, M. (2004). *A parallel approach to computing load flow equations*. Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms, vol. 11, no. 1–2, pp. 229–238.

Nieplocha, J., Marquez, A., Tipparaju, V., Chavarria-Miranda, D., Guttromson, R. and Huang, Z. (2006). *Towards efficient power system state estimators on shared memory computers*. In Proc. IEEE Power and Energy Society General Meeting, Montreal, Quebec, Canada

PNNL Institutional Computing (PIC) Program. [Online]. Available: http://pic.pnnl.gov/.

PowerWorld Simulator. [online]. Available: http://www.powerworld.com/.

Shu, J., Xue, W., and Zheng, W. (2005). *A Parallel Transient Stability Simulation for Power Systems*. IEEE Transactions on Power Systems, vol. 20, no. 4, 2005

Wine project. [online]. Available: http://wiki.winehq.org/