Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 55–60

Contingency Analysis Post-Processing With Advanced Computing and Visualization

Yousu Chen*, Kurt Glaesemann, Erin Fitzhenry*****

* 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: kurt.glaesemann@pnnl.gov)
*** Pacific Northwest National Laboratory, Battelle Seattle Research Center, Seattle, WA, 98109 USA (e-mail: erin.fitzhenry@pnnl.gov)}

Abstract: Contingency analysis is a critical function widely used in energy management systems to assess the impact of power system component failures. Its outputs are important for power system operation for improved situational awareness, power system planning studies, and power market operations. With the increased complexity of power system modeling and simulation introduced by increased energy production and demand, the penetration of renewable energy and fast deployment of smart grid devices, and the trend of operating grids closer to their capacity for better efficiency, more and more contingencies must be executed and analyzed quickly to ensure grid reliability and accuracy for the power market. Currently, many researchers have proposed different techniques to accelerate the computational speed of contingency analysis, but not much work has been published on how to post-process the large amount of contingency outputs quickly. This paper proposes a parallel post-processing function that can analyze contingency analysis outputs faster and display them in a web-based visualization tool to help power engineers improve their efficiency by fast information digestion. Case studies using an ESCA-60 bus system and a Western Electricity Coordinating Council planning system are presented to demonstrate the functionality of the parallel post-processing technique and the web-based visualization tool.

Keywords: Contingency analysis, high performance computing, parallel posting processing, power system operation, visualization

1. INTRODUCTION

Contingency analysis (CA) is a key function in an energy management system (EMS). It is a “what-if” analysis that is used to assess the capability of a power grid to sustain various combinations of power grid component failures (contingencies). The outputs of contingency analysis are used to provide the basis for operators to take actions to maintain system reliability. They are also extensively used in power market operation to perform feasibility tests of locational marginal price solutions.

In today’s practice, most utilities maintain a list of selected contingencies for their operations. One main reason is the heavy computation involved in contingency analysis. For example, the Bonneville Power Administration (BPA) runs ~500 contingency cases in a time interval of 5 minutes. However, the complexity of power system modeling and simulation is increasing, due to increased energy production and demand, penetration of renewable energy and fast deployment of smart grid devices, and the trend of operating grids closer to their capacity for better efficiency. With this increased complexity, a short list of contingencies is not adequate for assessing the vulnerability of a grid at various operating conditions. Thus, more contingencies must be executed and analyzed quickly to ensure safe and reliable operation of today’s power grid.

If probabilistic look-ahead contingency analysis is considered for handling the uncertainty introduced by forecasting renewable energy and load, the number of contingencies required will increase significantly. Meanwhile, on the power market side, contingency analysis is critical because system security has to be considered in all categories in market operations. Therefore, the total number of contingency cases in a market is larger than the number used in operation. This computational intensity will further increase with the fast development of the power grid if nothing changes to expedite contingency analysis simulation and post-processing.

Researchers already recognize the need to accelerate massive contingency analysis (MCA) computation using high-performance-computing (HPC). There are many publications in the area of applying HPC techniques to MCA. For example, Santos, et al. [1999] discussed a distributed client-server scheme using a fast decoupled load flow algorithm. Balduino and A. Alves [2004] developed an implementation in a microcomputer cluster with parallel virtual machines (PVMs) and a message passing interface (MPI) mechanism. Morante et al. [2006] leveraged a pervasive grid middleware to assign contingency cases with a master-slave model.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.010

Khaitan et al. [2013] used a master-slaves scheme with proactive task scheduling and stealing. Mittal, et al. [2011] presented a scalable implementation of a probabilistic contingency analysis scheme. A speedup of 6,513 and 5,630 was achieved with 8,192 cores using the IEEE 300- and 118-bus models, respectively.

The authors and their colleagues have published a substantial body of work in the area of massive contingency analysis (MCA). For instance, a framework of "*N-x*" MCA was established with a counter-based dynamic load-balancing scheme, illustrated via case studies of a 300,000-contingency case analysis using a 2005 Heavy Summer Western Electricity Coordinating Council (WECC) power grid model containing nearly 20,000 buses with up to 512 cores [Huang et al. 2009]. Because the scalability of counter-based dynamic load-balancing schemes with large numbers of processors is likely to be limited by counter congestion, a multiple-counter scheme with task-stealing capability has been proposed in the work of Chen et al. [2010], and its performance with varying numbers of contingency cases, cores used, and computing environments has been studied. In Chen [2012], a performance evaluation of the "*N-x*" parallel MCA framework with up to 10240 cores was conducted, reporting a speedup of 7877 with 10,240 cores for 1 million *N*-2 contingency cases using the 2005 WECC model. This was the first publication to report that contingency analysis can be scaled up to more than 10,000 cores. The authors' most recent work [Chen et al. 2015], reported a better near-linear speedup of 9,877, using 10,000 cores for 1 million WECC *N*-2 contingency cases, which took only 25 seconds.

Besides the acceleration of contingency analysis computation using HPC techniques, contingency analysis visualization is also an important research area for wide-area situational awareness because it can help users to easily develop a big picture of the system status. Two-dimensional (2D) visualization with a one-line diagram or geospatial layout is generally used. Researchers have also been developing three-dimensional (3D) visualizations. For instance, Sun [2004] discussed the transition from a 2D visualization to an interactive 3D visualization, displaying power system element vulnerability levels and outage severity information separately to represent the overall state of a system. The 3D visualization techniques can be found at Sun and Overbye [2003a, 2003b] and Overbye et al. [2013]. [Wong 2014]

In general, 3D visualization techniques usually require a larger visualization footprint, especially for large systems. Disadvantages of 3D techniques include perceptual ambiguities in depth, size, and distance, and the blocking of background objects by foreground objects. With these issues in mind, the authors have chosen to develop a 2D graphical contingency analysis (GCA) tool based on a one-line diagram [Chen et al. 2009], [GCA online]. The tool visualizes a large amount of data using different color schemes to represent the system's vulnerability with various contingencies. Rice et al. presented a feature for alleviating contingency analysis violations with suggested operation actions through the GCA tool [Rice et al. 2012]. Wong et al. [2014] proposed a visual-analytics pipeline that can transform approximately 100 million contingency scenarios to a manageable size and form.

While the achievements in accelerating massive contingency analysis and contingency analysis visualization are impressive, there is a missing step: extracting useful information from a large trunk of contingency outputs created by a large number of computer cores and making it available to users through visualization. There is no such work reported in the area of how to process contingency analysis quickly and effectively.

This paper is a continuation of the authors' previous work on MCA programming and visualization. This paper presents a parallel post-processing technique used in combination with the MCA application, as well as a new web-based visualization tool for the processed contingency analysis. To the authors' best knowledge, the parallel post-processing technique presented in this paper is the first reported in the area of contingency analysis. The developed algorithm can be easily customized for other power system applications. The web-based visualization tool will also adapt, enabling users to conduct in-depth analyses on currently unforeseen contingency outputs.

The rest of the paper is organized as follows: Section 2 discusses the implementation of parallel post-processing techniques with a brief introduction to the MCA application; Section 3 presents some case study results of the ESCA 60-bus system and the WECC 2005 Heavy Summer case; Section 4 presents the web-based visualization tool with examples; and Section 5 summarizes the paper with future work.

2. PARALLEL POST-PROCESSING TECHNIQUES

With the increased number of contingency cases, contingency analysis outputs for large power systems grow large enough that it is not efficient to digest the useful information using traditional sequential methods. Extracting useful information from all contingency outputs generated by multiple cores at different locations becomes critical to perform in a timely manner. To bridge the gap between massive contingency analysis computation and contingency analysis visualization, a suite of contingency analysis parallel post-processing applications has been developed.

Currently, the applications are being used with the MCA software developed by the authors [Chen 2013] and [Chen et al. 2015]. The MCA software is executed in parallel with each and every CPU core writing to its own output file. To reduce I/O contention (in particular file-system meta-data), each CPU core writes its files to a specific subdirectory. MCA writes a binary file for post-processing and/or a human-readable text file if necessary.

The computational time for each contingency case can vary greatly based on the number of iterations required, so a dynamic load-balancing scheme is used with cores requesting work from a central counter process on node zero [Huang et al. 2009]. A file containing all binary filenames is then created and used as an input for the parallel post-processing algorithm.

The parallel post-processing code uses a message passing interface (MPI), with static load balancing to distribute input files between cores. The maximum number of usable MPI