ON-LINE TRANSIENT STABILITY ASSESSMENT USING MATLAB PARALLEL PROGRAMMING ENVIRONMENT

Gürkan Soykan

Computational Science and Engineering Program,
Istanbul Technical University, Istanbul, Turkey
soykan@itu.edu.tr

Hasan Dağ

Information Technologies Department,
Kadir Has University, Istanbul, Turkey
hasan.dag@khas.edu.tr

ABSTRACT

Transient Stability Assessment (TSA) is the major part of Dynamic Security Assessment (DSA) in an electric power system. On-line TSA has gained more importance for power system security in last two decades. The decreasing costs of hardware and advanced package programs and techniques encourage researchers to apply parallel programming to the problems that are either too much time consuming or impossible to solve using serial computers. The real-time evaluation of TSA is now almost possible thanks to such tools and techniques. We demonstrate the use of the parallel computing toolbox of Matlab for TSA. In this regard, IEEE 57 and 118 bus systems are tested with 57 and 56 contingencies respectively. Each contingency requires solution of a transient stability problem. The parallel performance of this implementation is expected to spur researchers to use parallel programming in Matlab environment. It requires minimal knowledge regarding parallel programming.

1. INTRODUCTION

Transient Stability is defined as the study of dynamic behavior of an electric power system after a severe disturbance such as transmission line fault or loss of a generation unit. After the disturbance, the system is termed stable if it is able to operate at the initial operating condition or settle down to a new operating condition. Available methods to assess transient stability of a power system are summarized in [1]. Those proposed methods can be classified in three parts: simulation techniques, heuristic techniques, and training techniques [1]. The simulation techniques can be categorized as: numerical integration methods, direct methods (extended equal area criterion and transient energy function) or Lyapunov methods, and probabilistic methods. The heuristic techniques are called expert systems, which consist of pattern recognition and neural network methods.

Any severe transient disturbance is called a contingency in an electric power system. Dynamic Stability Assessment (DSA) is defined as evaluating the voltage transients and stability of a power system for a large number of contingencies. Transient Stability Assessment (TSA) is the main part of the DSA problem. The objective of TSA is to determine the dynamic behavior of an electric power system after a large disturbance. TSA is based on studies at the planning and operation stages of a power system. Also, assessing the power system dynamic response to a contingency is an important step in power system security analysis. TSA can be thought of as an off-line study if all contingencies are analyzed. However, on-line TSA analysis become a necessity for on-line security analysis [2].

On-line transient stability assessment has become an inevitable method in last two decades in energy management centers [3–5]. Several studies have been published at this point. Fast stability analysis techniques can be used to obtain real-time transient stability assessment. Time domain numerical integration methods require large amount of computation. This is a bottleneck for on-line TSA. To surmount this bottleneck, parallel programming can be a solution method. Parallel programming approach has been used in transient stability [6–9]. A distributed computing approach was implemented on TSA problem [10]. On the other hand, direct methods [11, 12], dynamic equivalent method [13, 14] and Neural Network based methods [15–18] have been proposed to attain on-line TSA.

This paper presents a parallel implementation of TSA on a Matlab environment using the parallel computing toolbox to achieve real-time performance. The decreasing cost of hardware and increased capability of processor architecture enables us to use parallel computation with existing time-domain methods to carry out on-line TSA. Different contingencies can be run on different processors concurrently. Matlab environment is an easy way to implement parallel processing. Also multi-core technology is appropriate for parallel computation in Matlab environment.

2. TRANSIENT STABILITY ASSESSMENT USING TIME DOMAIN SIMULATION METHOD

Conventional methods are called simulation techniques and one of them, the time domain simulation, uses to assess

9781-4244-3428-2/09/$25.00 ©2009 IEEE

TSA. A differential algebraic model is used to describe the power system in this simulation method. Dynamic devices are modeled as a set of differential equations and power system network is modeled as a set of algebraic equations. The differential-algebraic equations are given below in a general form.

$$\dot{x} = f(x, V, u), \qquad x(0) = x_0 \tag{1}$$

$$I(x, V) = \mathbf{Y_N} V \tag{2}$$

where x is a vector of state variables, V is a vector of complex bus voltages, u is a vector of inputs, I is an injected current vector to the buses, $\mathbf{Y_N}$ is the $N \times N$ complex bus admittance matrix of the network, and N is the number of buses in the system.

$$\begin{aligned} F_{n+1} \quad &\triangleq x_{n+1} - x_n - \\ &\frac{h}{2}[f(x_{n+1}, V_{n+1}) + f(x_n, V_n)] = 0 \end{aligned} \tag{3}$$

$$G_{n+1} \triangleq \mathbf{Y_N} V_{n+1} - I(x_{n+1}, V_{n+1}) = 0 \tag{4}$$

Numerical integration methods can be used to solve (1). These methods are classified as explicit and implicit methods. One of the implicit methods, trapezoidal integration method, is preferred to integrate (1) in this study. After the integration of (1), the differential equations turn to algebraic equations as in (3). Here, h is an integration time step, n and $n+1$ are time steps in (3) and (4). If (3) and (4) are solved separately and iterated between each other, the solution method is called alternating solution or if equations are solved simultaneously, the solution is called simultaneous solution. The latter method is preferred in this paper. Combined algebraic equations must be solved using Newton-Raphson method.

$$\begin{bmatrix} \frac{\partial F_{n+1}}{\partial x_{n+1}} & \frac{\partial F_{n+1}}{\partial V_{n+1}} \\ \frac{\partial G_{n+1}}{\partial x_{n+1}} & \frac{\partial G_{n+1}}{\partial V_{n+1}} \end{bmatrix} \begin{bmatrix} \Delta x_{n+1} \\ \Delta V_{n+1} \end{bmatrix} = - \begin{bmatrix} F_{n+1} \\ G_{n+1} \end{bmatrix} \tag{5}$$

Using the solution of (5), the unknown variables can be updated for the next Newton iteration. In time domain solution method, differential-algebraic equations are resolved with the help of simultaneous implicit method on each time step.

Time-domain simulation method for transient stability analysis, as explained above, is used in TSA study. For large electric power systems, performing a detailed analysis for all contingencies takes a lot of time in TSA. This is an obstacle to an on-line TSA. So, the concept of contingency selection has been used to decrease the number of contingencies. Selected contingencies, which are potentially detrimental for power system, are simulated to achieve an on-line TSA. The type of contingencies considered in this study are the three-phase balanced faults created at various locations in the system at any one time.

3. DOMAIN DECOMPOSITION BASED METHOD FOR ON-LINE TSA

Different contingencies can be handled on different processors concurrently. The following algorithm is used to achieve an on-line transient stability assessment in Matlab environment. Algorithm 1 demonstrates how parallel programming in Matlab environment applies to TSA.

Algorithm 1

1: Create initial data for contingency evaluation,
2: Decide a contingency distribution depending on the number of available processors,
3: Open the single program multiple data (SPMD) block
   - Evaluate Transient Stability Analysis for contingencies
4: Close the SPMD block
5: Results

In Matlab environment, parallel computing toolbox is available to implement parallel computation of serial solution for any problem [19]. This is very important to use the computer power on nowadays. The multi-core and multiprocessors systems have more capacities in terms of computing power. To utilize this power in Matlab, popular for all engineering calculations, parallel programming is an unavoidable technique by means of parallel computing toolbox. This toolbox has different key features to develop parallel applications efficiently. In TSA, data parallel algorithm is a suitable way to acquire real-time solution. Thus, single program multiple data (spmd) construction from parallel computing toolbox is used to create a domain decomposition method. In spmd block, each processor simulates concurrently its share of contingencies, which are decided initially.

Algorithm 2

1: Matlabpool open
2: spmd
   - For each worker do statements in this spmd block
3: end
4: Matlabpool close

Algorithm 2 shows the usage of spmd block in Matlab environment. Matlabpool is used to request parallel resources. Resources are called workers in parallel computing toolbox. Cores and processors behaves as workers in Matlab. Each worker has a unique identity, called labindex, to customize the execution of parallel jobs. After getting parallel resources, parallel jobs are created in spmd block. Then, workers do the same jobs in this block. At the end of parallel jobs, Matlabpool is closed to release parallel resources. The following algorithm, Algorithm 3, is a simple example to show the usage of spmd. A matrix is multiplied by different vectors on workers at the same time.

Algorithm 3

Require: Matrix A and vector t.

Matlabpool open
spmd
$x \leftarrow labindex * t$
$b \leftarrow A * x$
end
Matlabpool close

4. TEST RESULTS

Simulations are conducted on a cluster node, which has 4 Dual Core Xeon 2.33 GHz processors and 32 GB RAM.

IEEE 57 and 118 bus test systems are used to observe the performance of domain decomposition based method for parallel implementation. The former system has 7 generators and 80 branches. The latter has 54 generators and 186 branches. All contingencies, three phase faults, are used in TSA simulation. All together 57 contingencies created for the 57 bus system and 56 contingencies for the 118 bus system are simulated. The transient disturbance started at 0.5 seconds and cleared at 0.55 seconds. The total simulation time is 2 seconds in each contingency. The parallel performance of the implementation is denoted by speedup definition. It is the ratio of the cpu-time taken for one processor to the cpu-time taken for a given number of processors on parallel algorithm. It is called the relative speedup. The results of the applied method are given in Table 1 for both systems. Fig.1 shows speedup changes with respect to number of cores on 57 and 118 bus systems. Thanks to the domain decomposition based method, the speedup curve is closer to the ideal speedup.

Table 1: Relative Speedup for parallel TSA

Test System	2 cores Speedup	4 cores Speedup	8 cores Speedup
57	1.98	3.94	7.67
118	1.91	3.75	7.10

5. CONCLUSION

The results show that domain decomposition based method for on-line TSA is quite effective and very user friendly. Because of the decreasing cost of hardware and growing computing power of processors, the parallel programming is the essential way to reveal compute power of computer systems. Parallel programming in Matlab environment is easy to use for anyone who does not have any prior knowledge of parallel computations. Multi-core technology will also encourage/require the usage of parallel programming.

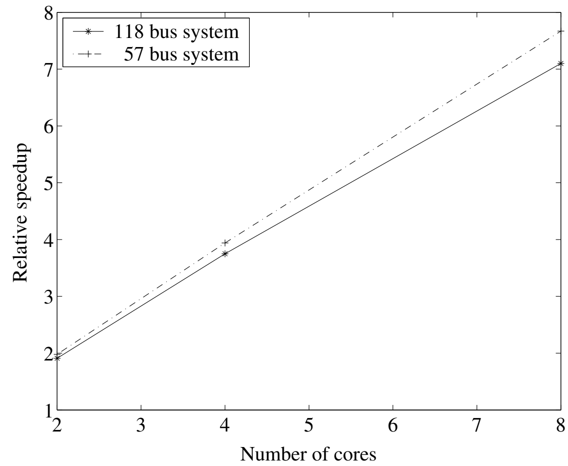

Fig. 1: Parallel Speedup In Terms of TSA

6. REFERENCES

[1] M. Moechtar, T. C. Cheng, and L. Hu, "Transient stability of power system - a survey," in *Proceedings of WESCON/95*, 7-9 November 1995, pp. 166–171.

[2] E. Vaahedi and K. W. Cheung, "Evolution and future of on-line dsa," in *IEEE Power Engineering Society Summer Meeting*, 18-22 July 1999, pp. 291–293.

[3] G. C. Ejebe, C. Jing, B. Gao, J. G. Waight, G. Pieper, F. Jamshidian, and P. Hirsch, "On-line implementation of transient stability assessment in an energy management system," in *IEEE Power Engineering Society Winter Meeting*, 23-27 January 2000, pp. 100–104.

[4] P. Kundur, G. K. Morison, and L. Wang, "Techniques for on-line transient stability assessment and control," in *IEEE Power Engineering Society Winter Meeting*, 23-27 January 2000, pp. 46–51.

[5] J. Giri and R. A. Rosalaes, "Transient stability assessment: integration with ems and control center requirements," in *IEEE Power Engineering Society Winter Meeting*, 23-27 January 2000, pp. 94–99.

[6] D. J. Tylavsky, A. Bose, F. L. Alvarado, R. Betancourt, K. Clements, G. Heydt, G. Huang, M. Ilic, M. LaScala, M. Pai, C. Pottle, S. Talukdar, J. VanNess, and F. Wu, "Parallel processing in power system computation," *IEEE Transactions on Power Systems*, vol. 7, no. 2, pp. 629–637, May 1992.

[7] J. Wu, A. Bose, J. Huang, A. Valette, and F. Lafrance, “Parallel implementation of power system transient stability analysis,” *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1226–1233, August 1995.

[8] M. Scala, G. Sblendorio, A. Bose, and J. Wu, “Comparison of algorithms for transient stability simulations on shared and distributed memory multiprocessors,” *IEEE Transactions on Power Systems*, vol. 11, no. 4, pp. 2045–2050, November 1996.

[9] J. Shu, W. Xue, and W. Zheng, “A parallel transient stability simulation for power systems,” *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1709–1717, November 2005.

[10] G. Aloisio, M. A. Bochicchio, M. L. Scala, and R. Sbrizzai, “A distributed computing approach for real-time transient stability analysis,” *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 981–987, 1997.

[11] Y. Xue, T. V. Cutsem, and R. M. Pavella, “A simple direct method for fast transient stability assessment of large power systems,” *IEEE Transactions on Power Systems*, vol. 3, no. 2, pp. 400–412, 1988.

[12] A. A. Fouad, V. Vittal, and T. Kyoo, “Critical energy for direct transient stability assessment of a multi-machine power system,” *IEEE Transactions on Power Apparatus and Systems*, vol. 103, no. 8, pp. 2199–2206, 1984.

[13] A. M. Miah, “Simple dynamic equivalent for fast online transient stability assessment,” *IEE Proceedings Generation Transmission and Distribution*, vol. 145, no. 1, pp. 1267–1274, 1998.

[14] R. A. Schlueter and P. A. Rusche, “Dynamic equivalents in rapid analysis of transient stability methods,” in *IEEE Power Engineering Society Winter Meeting*, 1987, pp. 30–36.

[15] L. Chunyan, T. Biqiang, and C. Xiangyi, “On-line transient stability assessment using hybrid artificial neural network,” in *Industrial Electronics and Applications 2007*, 23-25 May 2007, pp. 342–346.

[16] N. I. A. Wahab, A. Mohamed, and A. Hussain, “An improved method in transient stability assessment of a power system using probabilistic neural network,” *Journal of Applied Sciences Research*, vol. 11, no. 3, pp. 1267–1274, 2007.

[17] H. Sawhney and B. Jeyasurya, “On-line transient stability assessment using artificial neural network,” in *LESCOPE-04*, 28-30 July 2004, pp. 76–80.

[18] S. Krishna and K. R. Padiyar, “Transient stability assessment using artificial neural networks,” in *IEEE International Conference on Industrial Technology*, 19-22 January 2000, pp. 627–632.

[19] “Parallel computing toolbox,” World Wide Web electronic publication. [Online]. Available: http://www.mathworks.com/products/parallel-computing/