An adaptive scheme for distributed dynamic security assessment of large scale power systems

XING-ZHI WANG, ZHENG YAN and LI LI
Dept. of Electrical Engineering
Shanghai Jiao Tong University
Shanghai, 200240
CHINA
wangxingzhi2002@yahoo.com, xzwang.sjtu@gmail.com, and lili5396088@yahoo.com.cn

Abstract: - The requirements for significant computational resources imposed by dynamic security assessment applications have led to an increasing interest in the use of parallel and distributed computing technologies. This paper presents an adaptive scheme that involves user-friendly flat application program interfaces for scripting and an object-oriented programming environment for distributed dynamic security assessment implementation. Functional parallelism and data parallelism are supported by each of the message passing interface model and TCP/IP model. Adaptive stochastic-based objectives and conservative parameter prediction techniques are used to produce more efficient data parallelism. Tests for a 39-bus network and a 3872-bus network are reported, and the results of these experiments demonstrate that the proposed scheme is able to execute the distributed simulations on either stand-alone personal computers, cluster systems, or a computational grid infrastructure and provide efficient parallelism for the given environment.

Key-Words: - Adaptive systems, graph partitioning, parallel processing, parameter estimation, power system security.

1 Introduction

Graph partitioning plays a fundamental role in parallel computing by identifying the concurrency in a given problem when the computation can be modeled by a graph. A partition of the graph into sub-graphs leads to a decomposition of the data and/or tasks associated with a computational problem and the sub-graphs can then be mapped to the processors of a multiprocessor.

Graph partitioning also has an important role to play in the design of many serial algorithms by means of the divide and conquer paradigm. Two important examples of this algorithmic paradigm are in the solution of partial differential equations by domain decomposition and in the computation of nested dissection orderings for solving sparse linear systems of equations. Other applications include circuit partitioning and layout, VLSI design, and Computer-Aided-Design [1-4].

Two objectives are usually stated in the partitioning problem: partition a given graph into a specified number of sub-graphs such that the sub-graphs have roughly equal numbers of vertices, and few edges join different sub-graphs to each other. More general objective functions may need to be considered for many problems [5-7]. The work associated with a sub-graph may be modeled more accurately by attaching a weight to each vertex, and then equi-partitioning the weights. The communication costs in the algorithm might be modeled more accurately by the number of sub-graphs a given sub-graph is connected to, or the number of boundary vertices, or similar variants.

The simulation of a complex power system for stability analysis is difficult due to the large number of components that need to be considered such as governors, exciters, loads, electronic converters, etc., thus increasing the number of the state variables and consequently the complexity of the dynamic model [8-10]. In order to develop a practicable approach utilizing limited computational and technical resources, it is often necessary to use the parallel and distributed computing technologies [11-17].

During the last decades, a number of power system simulators implementing dynamic security assessment with parallel computing have been developed. One of the most common schemes is the data parallelism [18-20]. While these low-level infrastructures are extremely powerful, they are not compatible with each other, nor are they readily accessible to an average computational electrical engineer. On the other hand, higher-level parallelization systems with a Web-based user interface may help computer neophytes, but these systems lack programming flexibility to implement a user's analysis algorithm for various research purposes [21-23].

In this paper, an adaptive architecture was developed to produce higher-level application program interfaces (APIs) to provide users with a scripting environment and to distribute dynamic security assessment on stand-alone personal computer (PC), cluster and grid environments. Functional parallelism and data parallelism are supported by each of the message passing interface (MPI) model and TCP/IP model.

The standard approach for data parallelism has been to divide the vertices of a power gird into approximately equal-weight partitions and minimize the number of cut edges between partitions. It's proved that minimizing edge-cut may not reduce the parallel execution time of a dynamic security assessment application [24]. This paper presents a weighted form of stochastic-based objective to directly optimize various weighted power grid cuts. This makes the proposed algorithm scalable to irregular tasks than standard power grid partitioning methods.

During the dynamic security assessment, there exist both very fast and very slow dynamic devices. An important concern arises here is how to consider the contribution of the slow-varying subsystem when solving for the fast one throughout a complete interval. The initial attempts to exploit latency were made in very large-scale integration circuit simulation in connection with the waveform relaxation method. In the early 1990s, the waveform relaxation method was used for the time domain simulation of power systems in transient stability studies. In the latency technique proposed in this paper, boundary subsystem uses the self and neighboring historical iterate parameters as guesses for the current state of subsystems.

The remainder of this paper is organized as follows. Section 2 presents the adaptive architecture for distributed dynamic security assessment. Section 3 briefly introduces the functional parallelism and data parallelism algorithms. In Section 4, the adaptive stochastic-based clustering algorithm and conservative parameter prediction techniques are described. In Section 5, two power system networks are used to test the effectiveness of the proposed approach. Finally, conclusions are drawn in Section 6.

2 Adaptive Architecture

The architecture for distributed dynamic security assessment is composed of three layers: 1) a class library for dynamic security assessment and its C++ APIs, 2) a Python language wrapper of core simulation engine and lower-level middleware and 3) python front-end utilities (Fig.1).

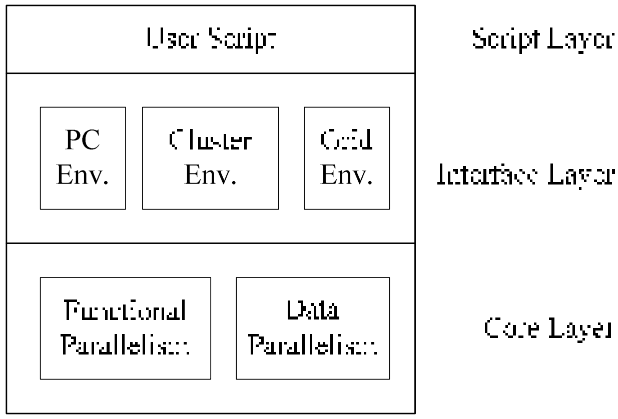

Fig. 1. Adaptive architecture for distributed dynamic security assessment.

A distributed dynamic security assessment application consists of a number of related threads or jobs. As illustrated in Figure 2, the proposed architecture uses various components to hide and management heterogeneity.

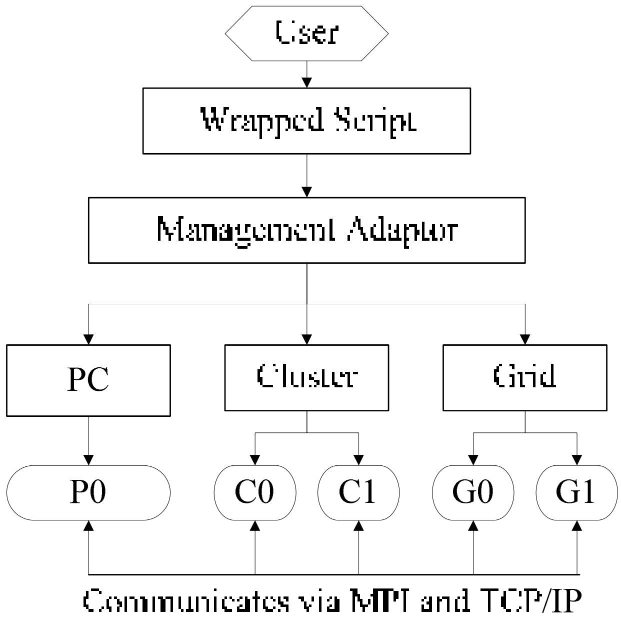

Fig. 2. Schematic of the start-up.

When an application is submitted to the simulation system, the individual steps are as follows:

a) The simulation system creates a wrapper script and the status of the thread or job is ready.

b) Management adaptor conceals the difference of distributed environments and communicates to the computer operating system or to the low-level middleware of the computing environment.

c) Execute adaptor executes a task in stand-alone PC, or cluster, or grid environments and holds the status.

d) If the task ends without errors, the output is transferred back and the status of the task is done.

3 Distributed Dynamic Security Assessment

3.1 Functional Parallelism of Errors or Attacks

A power system can be modeled as a set of generators and a set of loads interconnected via a transmission network. The stability analysis of a power system is a simulation in the time domain, lasting several seconds or minutes. First the power system is simulated in its operating state, then large disturbances and protective actions are simulated and finally the simulation is continued for a few more seconds or minutes. Different components of the power system have their greatest influence on the stability of the system at different points in time of the response simulation. The dynamic security assessment of an electrical power system emphasizes the rapidly responding electrical components of the system, for instance voltage and currents at the loads.

Here we represent power systems as networks and large disturbances are simulated by errors or attacks [25-28]. By error or failure we mean the breakdown of the randomly selected nodes or branches. Instead, we call attack the targeted breakdown of the most important nodes or branches. Different criteria can be considered to determine the importance of a node or a branch. For example, degree, betweenness and load of the node can be adopted for node attack purpose, while betweenness, admittance and power flow of the branch can be considered as criteria for branch attack purpose.

Since there are independent tasks applying different disturbances to power grids, a functional parallelism can be used for parallel dynamic security assessment of large-scale power systems. For a given power system with a given contingency, if parameters change, one would like to know as quickly as possible how these changes will affect system stability indices. The transient stability indices at different fault clearing times can be obtained by combining the transient energy function method with the time domain simulations [30]. According to the parallel simulation results of instability and stability indices, we can show how the topology of a network may have important consequences on error and attack tolerance of the power system.

3.2 Data Parallelism of the Given Breakdown Event

The dynamical model of an interconnected power system can be completely described by a set of highly nonlinear differential equations:

$$\dot{X} = f(X,V) \tag{1}$$

subject to the initial conditions:

$$X(0) = X_0$$

and a set of nonlinear algebraic equations

$$0 = g(X,V) \tag{2}$$

where X are the state variables of the differential equations, and $\mathbf{V}$ are the network variables.

Eq.(1) can be discretized using the implicit trapezoidal rule:

$$X_t = X_{t-1} + \frac{h}{2}\left(f(X_t,V_t) + f(X_{t-1},V_{t-1})\right) \tag{3}$$

where $t = 1,2,\cdots,T$ denotes time steps, while h represents the step length. Eq.(3) can be rewritten in residual form:

$$R_G = X_t - X_{t-1} - \frac{h}{2}\left(f(X_t,V_t) + f(X_{t-1},V_{t-1})\right) \tag{4}$$

where R_G is the residual with respect to the generator variables. Similarly, Eq.(2) can be written as follows:

$$R_N = g(X_t,V_t) \tag{5}$$

where R_N denotes the residual with respect to the network variables. Eq.(4) and Eq.(5) are linearized using Newton's method and then combined to form the linear system:

$$\begin{bmatrix} R_G \\ R_N \end{bmatrix} = -\begin{bmatrix} A_G & B \\ C & Y + Y_{ld} \end{bmatrix}\begin{bmatrix} \Delta X \\ \Delta V \end{bmatrix} = -J\begin{bmatrix} \Delta X \\ \Delta V \end{bmatrix} \tag{6}$$

where

$A_G = \frac{\delta R_G}{\delta X}$, $B = \frac{\delta R_G}{\delta V}$, $C = \frac{\delta R_N}{\delta X}$, and $J_N = \frac{\delta R_N}{\delta V}$.

Note that Y is the network admittance matrix and Y_{ld} is a diagonal matrix obtained from the derivation of nonlinear load currents with respect to the voltages at the nodes. In addition, the structure of the time-variant Jacobian J can be written in the following form:

$$J = \begin{bmatrix} A_{G1} & 0 & \cdots & 0 & B_1 \\ 0 & A_{G2} & 0\cdots & 0 & B_2 \\ 0\cdots 0 & A_{Gi} & & 0 & B_i \\ 0 & \cdots & 0 & A_{Gm} & B_m \\ C_1 & \cdots & C_i & C_m & J_N \end{bmatrix} \tag{7}$$

After taking the Schur complement of the linear system in Eq.(6), we obtain the set of equations:

$$\hat{R}_N = R_N - CA_G^{-1}R_G \tag{8}$$

$$\hat{J}_N = J_N - CA_G^{-1}B \tag{9}$$

$$-\hat{J}_N \Delta V = R_N \tag{10}$$

where Eq.(10) can be solved for ΔV . Finally, we perform backward block substitution to find ΔX ,

$$\Delta X = A_G^{-1}\left(R_G - B\Delta V\right) \tag{11}$$

Handling the solutions of distributed linear algebraic equations is one of the most difficult tasks in transient stability problem. Many parallel algorithms tried to solve linear equations, such as those incorporate the multiple factoring methods, the **W** matrix method, parallel factorization and substation method, conjugate gradient method, and bordered blocked diagonal form (BBDF) class methods. Among these methods, **BBDF** class is preferred for its light communication [18-19]. **BBDF** linear equation can be solved by parallel **LU** factorization and parallel forward-backward substitution.

Consider a power system network with n buses, which is torn apart into s regions. Each region is referred as non-overlapping subsystem of size $n_l\left(l = 1,2,\cdots,s\right)$ which is connected through m interconnected tie lines. An unknown current exchange $\hat{I}_{ex}$ will be appropriately injected back to the subsystems for modeling the other neighboring subsystems at any boundary buses, i.e.

$$\mathbf{V}_l^k = \mathbf{Y}_l^{-1}\mathbf{I}_l^k + \mathbf{C}_l\hat{\mathbf{I}}_{lex}^k,\ \left(l = 1,\cdots,s\right) \tag{12}$$

where l is the subsystem identifier, k is iteration number and $C_l\left(n_l \times m\right)$ is the incidence matrix of the l-th subsystem. If bus i is a boundary bus and currents outgoing, then c_{ij} is 1. While, if bus i is a boundary bus and currents incoming, then c_{ij} is -1. Moreover, if bus i is not a boundary bus, then c_{ij} is 0. Since there are independent tasks applying the same operation to different elements of the power grid data set, a data parallelism can be used for distributed dynamic security assessment.

4 Adaptive Stochastic-based Objectives and Conservative Parameter Predictions

4.1 Adaptive Stochastic-based Objectives

Considering the computational and communication cost of the subsystem, the cost of l can be described as follows

$$x_l = \left(N_{vl} + 2\cdot N_{el} + N_{dl}\right) + \left(N_{cvl}^2 + 2\cdot N_{cel} + N_{cdl}\right) \tag{13}$$

where N_{vl} is the number of internal nodes, N_{el} is the number of internal weighted branches, N_{dl} is the equivalent weight of internal dynamic units, N_{cvl} is the number of boundary nodes, N_{cel} is the number of boundary weighted branches, and N_{cdl} is the equivalent weight of boundary dynamic units.

Performance models are often parameterized by values that represent power system simulation and load characteristics. In dedicated settings it is often sufficient to represent these characteristics by a single value, or point value. However, point values are often inaccurate or insufficient representations for characteristics that change over time [15]. One way to represent this variable behavior is to use a stochastic value, or distribution. To allow for the use of stochastic information, we focus on the f value (minimum load to compute one subsystem data on the given breakdown event) in the load-balancing equations. In the non-stochastic settings, a single value (maximum load of subsystems) is minimized for load-balancing purpose. In the stochastic setting, we still use a single value for f but choose that value from the range given by the stochastic prediction of performance. This allows us the flexibility to choose larger or smaller values of f in the range. One approach to determine which value to choose for f is to consider the mean and standard deviation information of the computational and communication cost, that is

$$\min \quad f = TF_m \cdot m_x + TF_{sd} \cdot sd_x \tag{14}$$

where

m_x: The mean of the predicted computational and communication cost x for subsystems.

sd_x: The standard deviation of the predicted computational and communication cost x for subsystems.

TF_m: The Tuning Factor of mean, used to determine the number of means to add to (or subtract from) the standard deviation value.

TF_{sd}: The Tuning Factor of standard deviation, used to determine the number of standard deviations to add to (or subtract from) the mean value.

Given a set of vectors $a_1, a_2, \cdots, a_n$, Eq.(14) may be rewritten as

$$D\left(\{\pi_c\}_{c=1}^{k}\right) = TF_m \cdot m_c + TF_{sd} \cdot \sqrt{\frac{1}{(k-1)} \sum_{c=1}^{k} \sum_{a_i \in \pi_c} \omega_i \left\| \phi(a_i) - m_c \right\|^2} \tag{15}$$

where $m_c = \dfrac{\sum_{a_i \in \pi_c} \omega_i \phi(a_i)}{\sum_{a_i \in \pi_c} \omega_i}$.

Let s_c be the sum of the weights in cluster c, that is, $s_c = \sum_{a_i \in \pi_c} \omega_i$. Define the $n \times k$ matrix $\mathbf{Z}$

$$Z_{ic} = \begin{cases} \dfrac{1}{s_c^{1/2}} & if \quad a_i \in \pi_c \\ 0 & otherwise \end{cases} \tag{16}$$

Suppose $\mathbf{\Phi}$ is the matrix of all the $\phi(a_i)$ vectors, $i = 1, \cdots, n$, and $\mathbf{W}$ is the diagonal matrix of weights. The matrix $\mathbf{\Phi W Z Z}^T$ has column i equal to the mean vector of the cluster that contains a_i. Thus, Eq.(15) may be written as

$$D\left(\{\pi_c\}_{c=1}^{k}\right) = TF_m \cdot m_c + TF_{sd} \cdot \sqrt{\frac{1}{(k-1)} \sum_{i=1}^{n} \omega_i \left\| \mathbf{\Phi}_{\cdot i} - \left(\mathbf{\Phi W Z Z}^{\mathrm{T}}\right)_{\cdot i} \right\|^2} \tag{17}$$

where $\mathbf{\Phi}_{.i}$ denotes the i-th column of the matrix $\mathbf{\Phi}$.

Let $\widetilde{\mathbf{Y}} = \mathbf{W}^{1/2}\mathbf{Z}$, observe that $\widetilde{\mathbf{Y}}$ is an orthonormal matrix. Thus,

$$D\left(\{\pi_c\}_{c=1}^{k}\right) = TF_m \cdot m_c + TF_{sd} \cdot \sqrt{\frac{1}{(k-1)} \left\| \mathbf{\Phi W}^{1/2} - \mathbf{\Phi W}^{1/2} \widetilde{\mathbf{Y}} \widetilde{\mathbf{Y}}^{\mathrm{T}} \right\|_F^2} \tag{18}$$

Since $trace(\mathbf{A}+\mathbf{B}) = trace(\mathbf{A}) + trace(\mathbf{B})$, $trace(\mathbf{AB}) = trace(\mathbf{BA})$ and $trace\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right) = \|\mathbf{A}\|_F^2$, we can rewrite $\left\| \mathbf{\Phi W}^{1/2} - \mathbf{\Phi W}^{1/2} \widetilde{\mathbf{Y}} \widetilde{\mathbf{Y}}^{\mathrm{T}} \right\|_F^2$ as $trace\left(\mathbf{W}^{1/2}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi W}^{1/2}\right) - trace\left(\widetilde{\mathbf{Y}}^{\mathrm{T}}\mathbf{W}^{1/2}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi W}^{1/2}\widetilde{\mathbf{Y}}\right)$. If TF_m is set to 0, the minimization of $\dfrac{(k-1)}{(TF_{sd})^2}\left[D\left(\{\pi_c\}_{c=1}^{k}\right)\right]^2$ is equivalent to

$$\max_{\widetilde{\mathbf{Y}}} trace\left(\widetilde{\mathbf{Y}}^{\mathrm{T}}\mathbf{W}^{1/2}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi W}^{1/2}\widetilde{\mathbf{Y}}\right) \tag{19}$$

Setting $\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$ to $\mathbf{W}^{-1}\mathbf{A}\mathbf{W}^{-1}$, the trace maximization for $trace\left(\widetilde{\mathbf{Y}}^{\mathrm{T}}\mathbf{W}^{1/2}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi W}^{1/2}\widetilde{\mathbf{Y}}\right)$ is seen to equal $trace\left(\widetilde{\mathbf{Y}}^{\mathrm{T}}\mathbf{W}^{-1/2}\mathbf{A}\mathbf{W}^{-1/2}\widetilde{\mathbf{Y}}\right)$, which is exactly the trace maximization for the following weight graph association [29]

$$WAssoc(G) = \max_{v_1, \cdots, v_k} \sum_{c=1}^{k} \frac{links(\upsilon_c, \upsilon_c)}{\omega(\upsilon_c)} \tag{20}$$

The weight graph cut problem can be expressed as the association problem by noting that

$$trace\left(\mathbf{Y}^{\mathrm{T}}\mathbf{W}^{-1/2}(\mathbf{W}-\mathbf{L})\mathbf{W}^{-1/2}\mathbf{Y}\right) = k - trace\left(\mathbf{Y}^{\mathrm{T}}\mathbf{W}^{-1/2}\mathbf{L}\mathbf{W}^{-1/2}\mathbf{Y}\right) \tag{21}$$

Hence, optimizing $WAssoc$ on the matrix $\mathbf{W}-\mathbf{L}$ is equivalent to optimizing $WCut$ on $\mathbf{A}$.

Thus, a weighted form of our objective is seen to be mathematically equivalent to a general weighted graph clustering objective. Furthermore, our multilevel algorithm removes the restriction of equal cluster size and uses the weighted stochastic-based algorithm during the refinement phase to directly optimize various weighted power grid cuts. This makes the proposed algorithm scalable to irregular tasks than standard graph partitioning methods.

4.2 Conservative Parameter Predictions

In Section 3, Eq. (1) describes the machine dynamics and Eq. (2) represents the network static behavior. The alternating scheme can be used to solve the equations. An important concern arises here is how to consider the contribution of the slow-varying subsystem when solving for the fast one throughout a complete interval. To achieve such an outcome two prediction methods are presented.

The first prediction method updates the fast subsystem contribution in injection current by using self historical boundary voltages. For simple prediction, the boundary voltage can be calculated from self one-step historical value:

$$\hat{V}_{i+1} = V_i \tag{22}$$

For first order Adams prediction, the boundary voltage can be estimated from self two-step historical values:

$$\hat{V}_{i+1} = V_i + \left(V_i - V_{i-1}\right) \tag{23}$$

For modified first order Adams prediction, the boundary voltage can be updated from self modified two-step historical values:

$$\begin{cases} \hat{V}_{i+1} = V_i + \left(V_i - V_{i-1}\right) \\ V_{i+1} - \hat{V}_{i+1} = \dfrac{1}{2}\left(V_i - \hat{V}_i\right) \end{cases} \tag{24}$$

In the first prediction method which already described, the fast acting subsystem use the self historical boundary values for the current state of the other subsystems. This may require more iteration to converge and therefore degrade the efficiency. It can be enhanced by using the historical boundary values of neighboring subsystems. The idea here is to estimate contribution of other subsystems in injection currents from its previous quantity.

Based on Diakoptics technique, Eq.(12) is preserved for every cut branch between two adjacent subsystems i and j:

$$\left(1-F\right)\cdot y_{ij}\cdot\frac{\left[\left(\hat{V}_i^k - \hat{V}_j^k\right)+\left(V_i^{k-1}-V_j^{k-1}\right)\right]}{2}+ F\cdot y_{ij}\cdot\left(\hat{V}_i^k - \hat{V}_j^k\right) - \hat{I}_{ijex}^k = 0 \tag{25}$$

where F is a scalar that represents the operating state of the cut branch. If F is 1, then the cut branch is closed. While, if F is 0, then cut branch is open and either side of the boundary bus has different quantities. Eq.(25) can be written in matrix form as follows,

$$\sum_{i=1}^{s}\left[\frac{\left(1+F\right)\cdot D_l\cdot\hat{V}_l^k}{2}+\frac{\left(1-F\right)\cdot D_l\cdot V_l^{k-1}}{2}-\hat{I}_{lex}^k\right]=0 \tag{26}$$

where $D_l\left(n_l \times m\right)$ is the matrix with element $d_{ij} = c_{ij}\cdot y_{ij}$. Combining Eq.(12) and Eq.(26) into a common frame work, the general expression for the piecewise linear equation solution will be obtained.

5 Numerical Experiments

5.1 Experimental Environment

The architecture presented in Section 2 was coded using Python for front-end scripting, MPICH and Globus middleware, and C++ programming language for core simulation engine. The MPI and TCP/IP versions of functional parallelism were executed using three PCs. And the grid-enabled data parallelism was tested on an IBM p690. Two networks of 39-bus and 3872-bus were choose and used to validate the proposed algorithms.

A few statistical properties of the two power grids are listed in Table 1. The quantities measured are: number of vertices *N*v, number of edges *N*e, average degree <*k*>, clustering coefficient <*C*>, and average path length <*L*>.

Table 1. Statistical properties of the power system networks.

Network	N_v	N_e	<k>	<C>	<L>
NetworkA	39	46	2.359	0.039	4.749
NetworkB	3872	4788	2.473	0.037	21.604

In networks, the nodes are the buses and the links represent the physical connections among them. All these graphs are sparse with a small average degree and high clustering coefficient. They all show the small-world property, having a characteristic average path length very small as compared to their sizes.

5.2 The Functional Parallelism Case

The functional parallelism implementation has been tested on NetworkB, and random errors are considered. All the errors involve a three-phase-to-ground fault on a line near a busbar and the fault is cleared by opening the faulted line. For a communication environment comparison, we now compare the simulation time using MPI to that of TCP/IP. Table 2 and Table 3 show the comparison of the simulation time using MPI and TCP/IP with multi-computers. It verifies that the use of the functional parallelism is able to reduce the solution time, even though the total communication time is increased under different communication environments.

Table 2. The time consumed in dynamic security assessment using MPI (s).

Number of erros	1PC	2PCs	3PCs
5	89.3	58.2	37.3
10	185.2	95.3	80.6
20	352.0	187.5	144.3
30	520.3	272.3	188.2

Table 3. The time consumed in dynamic security assessment using TCP/IP (s).

Number of erros	1PC	2PCs	3PCs
5	92.1	60.5	39.1
10	187.7	98.5	82.3
20	374.1	189.1	146.1
30	523.1	274.3	191.8

5.3 The Data Parallelism Case

The speedup and efficiency measure for the grid-enabled data parallelism was firstly measured by using NetworkA. The speedup and efficiency of computations is presented in Fig.3 and Fig.4. This represents the practical situation where inter-processor communication has a great influence on the overall performance in a distributed dynamic security assessment. Although the use of data parallelism did reduce the calculation time, part of the gain in time was nullified by the additional communication step and programming overheads. The measured speedup for using 3 processors is 1.17 and is generally comparable with that obtained by other parallel algorithm for a similar size network [6]. For the large branch-to-bus ratio system of NetworkB, the speedup will be higher, as there is a greater degree of parallelism available and the ratio of computation to communication is relatively higher.

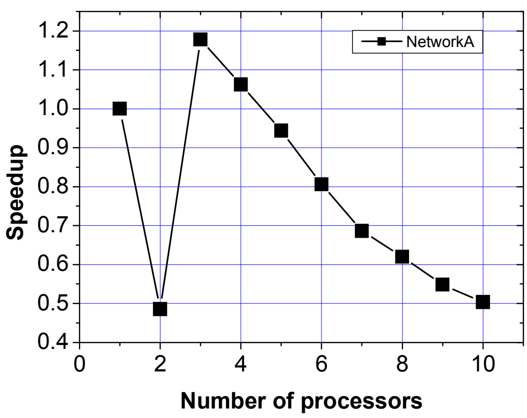

Fig. 3. Speedup of NetworkA.

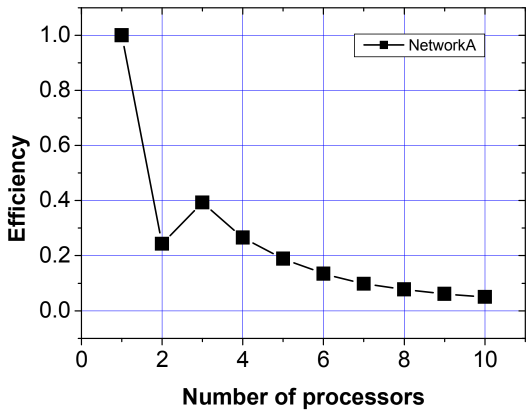

Fig. 4. Efficiency of NetworkA.

The proposed multilevel algorithm uses the adaptive stochastic-based algorithm during the refinement phase to optimize weighted NetworkB cuts. The Metis is considered as the most accurate graph clustering toolkit. Therefore the proposed method is compared against Metis. If TF_m and TF_{sd} are set to 0 and 1 respectively, the clustering quality results can be shown in Fig.5 and Fig.6. Fig.5 shows that the proposed method produces a little higher ratio association values than Metis. From Fig.6, we see that the proposed method gives a lower normalized cut values than Metis. Note that all our experiments show that the proposed method consistently produces better partitions than Metis, which is not surprising since Metis minimizes the edge cut value instead of the normalized cut or ratio association. This indicates that it is effective to employ the proposed stochastic-based method to solve the large-scale power grid partition problem.

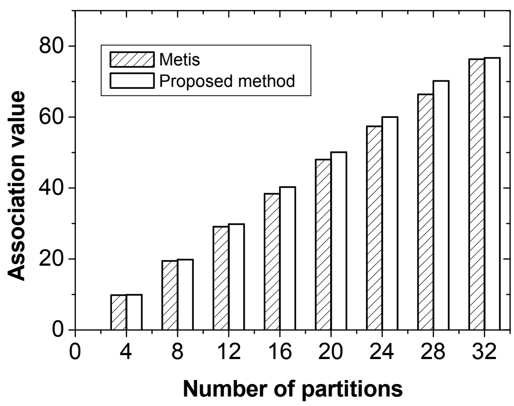

Fig. 5. Quality comparison between the Metis and the proposed method in terms of ratio association values for NetworkB.

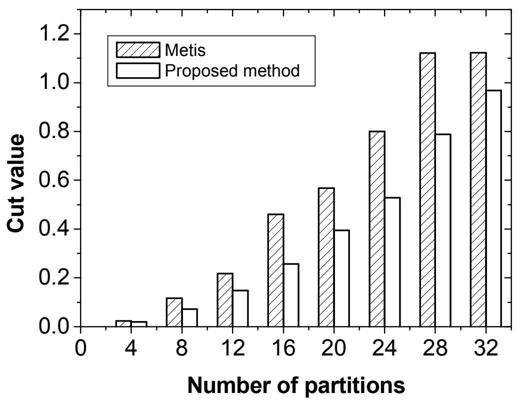

Fig. 6. Quality comparison between the Metis and the proposed method in terms of normalized cut values for NetworkB.

In Fig.7 we show simulation times of different estimation schemes: traditional estimation (black square), first order Adams estimation (red circle), and modified first order Adams estimation (green upper triangle). It is clear that the simulation times of modified first order Adams estimation are smaller than that of others. An application to NetworkB shows that by the modified first order Adams prediction method the simulation time is reduced by up to 17.89% compared to the traditional approach. This picture can serve as validations of the efficiency of our conservative parameter prediction algorithm.

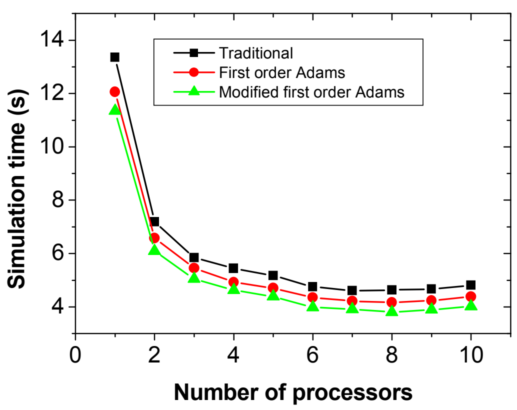

Fig. 7. Simulation times of different self historical parameter estimation schemes for NetworkB.

In addition to the self historical parameter estimation, the neighboring historical parameter estimation also contributes to the increase of efficiency. Fig.8 and Fig.9 show the simulation times (parallel simulation time does not include the data I/O time) for NetworkB with various numbers of processors being used. As illustrated in Fig.8 and Fig.9, the use of the neighboring historical parameter estimation is able to reduce the solution time. Besides that, with the increasing of the processor number, the parallel simulation time and total simulation time have some minimum values. It can be explained that there are not enough computations to parallel and too many processors may contribute to the loss of efficiency by some additional communications.

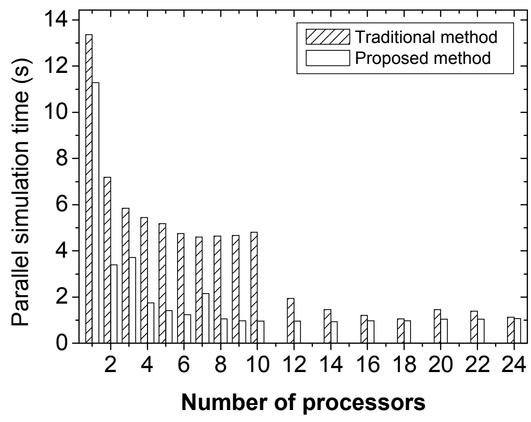

Fig. 8. Parallel simulation times of different neighboring historical parameter estimation schemes for NetworkB.

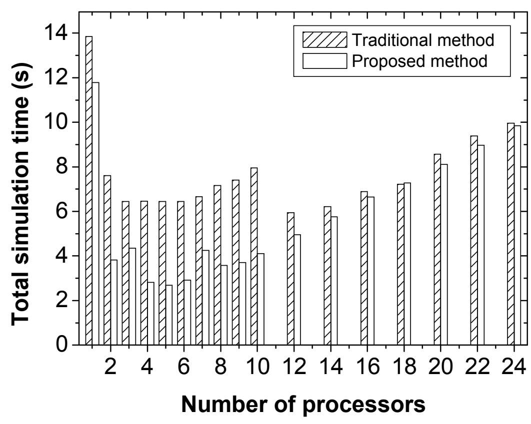

Fig. 9. Total simulation times of different neighboring historical parameter estimation schemes for NetworkB.

4 Conclusions

This paper presents an adaptive scheme that involves user-friendly flat APIs for scripting and an object-oriented programming environment for distributed dynamic security assessment implementation. The experimental results show that the proposed architecture can support functional parallelism and data parallelism in each of the MPI model and TCP/IP model. In addition to the mathematical equivalent to a general weighted graph clustering objective, our weighted stochastic-based algorithm is comparable to Metis in terms of speed and produces better weighted power grid cuts. Moreover, a conservative parameter prediction technique has been proposed to achieve efficient execution of distributed dynamic security assessment. The results reported here are promising in that they demonstrate that the adaptive scheme can be used to successfully to promote adaptation and improve application performance in complex and challenging environments.

References:

[1] T.Lengauer. *Combinatorial algorithms for integrated circuit layout*. John Wiley & Sons, Chichester, 1990.

[2] J.Garbers, H.J.Promel, and A.Steger. Finding clusters in VLSI circuits, *Proceedings of the IEEE International Conference on Computer-Aided Design*, 1990, pp.520-523.

[3] L.Hagen and A.B.Kahng. A new approach to effective circuit clustering. *Proc.IEEE Intl.Conf.on Computer-Aided Design*, Santa Clara, Nov.1992.

[4] J.Cong and M.L.Smith. A parallel bottom-up clustering algorithm with applications to circuit partitioning in VLSI design. *Proceedings of the 30th Conference on Design Automation*, 1993, pp.755-760.

[5] G.Karypis and V.Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs, *Technical Report 95-035*, Depart.of Computer Science, Univ.of Minnesota, Minneapolis,1995.

[6] B.Hendrickson and R.Leland. The Chaco user's guide, version 2.0, *Technical Report SAND95-2344*,Sandia National Laboratory, Albuquerque, NM,1995.

[7] C.Walshaw and M.Cross. Mesh partitioning: a multilevel balancing and refinement algorithm, *SIAM J.Scientific Computing*, Vol.22, 2000, pp.63-80.

[8] Jan Machowski, Janusz W.Bialek, James RIBumby. *Power system dynamics and stability.* John Wiley & Sons Ltd, 1997.

[9] N.J.Balu and et.al. On-line power system security analysis. *Proceedings of the IEEE*, Vol.80,No.2,1992,pp.262-280.

[10] Y.Sekine, K.Takahashi, and T.Sakaguchi. Real-time simulation of power system dynamics. *Int.J.of Electrical Power and Energy Systems*, Vol.16, No.3, 1994, pp.145-156.

[11] J.Q.Wu and A.Bose. Parallel solution of large sparse matrix equations and parallel power flow. *IEEE Transactions on Power Systems*, Vol.10, No.3, 1995, pp.1343-1349.

[12] J.Q.Wu and A.Bose. A new successive relaxation scheme for the W-matrix solution method on a shared-memory parallel computer. *IEEE Transactions on Power Systems*, Vol.11, No.1, 1996, pp.233-238.

[13] G.T.Vuong,R.Chahine,G.P.Granelli, and R.Montagna. Dependency-based algorithm for vector processing of sparse matrix forward/backward substitutions. *IEEE Transactions on Power Systems*, Vol.11, No.1, 1996, pp.198-205.

[14] D.M.Falco. High performance computing in power system applications. *Proceedings of the 2nd international meeting on vector and parallel proceeding*, September 1996.

[15] J.Schopf and F.Berman. Stochastic scheduling. *Proc. Supercomputing*, 1999.

[16] F.Berman, R.Wolski, H.Casanova, W.Cirne, H.Dail, M.Faerman, S.Figueira, J.Hayes, G.Obertelli, J.Schopf, G.Shao, S.Smallen, N.Spring, A.Su, and D.Zagorodnov. Adaptive computing on the grid using AppLeS. *IEEE Transactions on Parallel and Distributed Systems,* Vol.14, 2003, pp. 369-382.

[17] Liang-The Lee, Der-Fu Tao, Chieh Tsao. An adaptive scheme for predicting the usage of grid resources. *Computers and Electrical Engineering*, Vol.33, 2007, pp.1-11.

[18] K.W.Chan, R.W.Dunn, and A.R.Daniels. Efficient heuristic partitioning algorithm for parallel processing of large power systems network equations. *IEE Proc.-Gener.Transm.Distrib.* Vol.142, 1995, pp.625-630.

[19] W.Xue, J.W.Shu, J.F.Yan, X.F.Wang, and Y.D.Han. Cluster-based parallel simulation for power system transient stability analysis. *Proceedings of the CSEE*, Vol.23, 2003, pp.38-43.

[20] Jian Wu, Noel N.Schulz, Wenzhong Gao. Distributed simulation for power system

analysis including shipboard systems. *Electric Power Systems Research*, Vol.77, 2007, pp. 1124-1131.

[21] Michele Di Santo, Alfredo Vaccaro, Domenico Villacci, and Eugenio Zimeo. A distributed architecture for online power system security analysis. *IEEE Transactions on Industrial Electronics*, Vol.51, 2004, pp.1238-1248.

[22] M.Ali, Z.Y.Dong, X.Li, and P.Zhang. RSA-Grid: a grid computing based framework for power system reliability and security analysis. *Proceedings of the 2006 IEEE power engineering society general meeting*, June 2006.

[23] L.B.Shi, N.C.Chang.Z.Lan,D.P.Zhao,H.F.Zhou, Peter T.C.Tam, Y.X.Ni, Felex F.Wu. Implementation of a power system dynamic security assessment and early warning system. *Proceedings of the 2007 IEEE power engineering society general meeting*, June 2007.

[24] X.Z.Wang, Z.Yan and W.Xue. An adaptive clustering algorithm with high performance computing application to power system transient stability simulation. *Proceedings of the 3rd International Conference on Electric Deregulation and Restructuring and Power Technologies*, 2008, pp.1137-1140.

[25] Watts D.J. Strogatz S.H. Collective dynamics of small-world network. *Nature*, Vol.393, 1998, pp.440-2.

[26] R.Albert and A.L.Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, Vol.74, 2002; pp.47-97.

[27] Dorogovtsev S.N. Mendes JFF. *Evolution of networks:from biological nets to the internet and WWW*. Oxford University Press, 2003.

[28] P.Crucitti, V.Latora, M.Marchiori, and A.Rapisarda. Error and attack tolerance of complex networks. *Physica A*, Vol.340, 2004, pp.388-394.

[29] I.Dhillon, Y.Guan, and B.Kulis. Kernel *k*-means, spectral clustering and normalized cuts. *Proceedings of the 10th ACM Knowledge Discovery and Data Mining Conference*, 2004.

[30] Xingzhi Wang, Zheng Yan, Li Li, and Dong Xie. A new kernel-based clustering algorithm for the multi-machine equivalent of large power systems. *Proceedings of the 11th International Conference on Electrical Machines and Systems*, 2008.

Xing-Zhi Wang was born in Jiangsu in 1979. He received Bachelor degree of Electrical Engineering from Southwest Jiaotong University of China in 2002 and Master degree in 2005. Now he is a Ph.D student of Electrical Engineering of Shanghai Jiao Tong University. His current research interests include power system analysis and the real time simulation of power system.

Zheng Yan was born in Zhejiang in 1964. He received his Ph.D in Electrical Engineering from Tsinghua University. In 2004 professor Yan joined the faculty of Department of Electrical Engineering of Shanghai Jiao Tong University. His research interest includes power system stability analysis, optimization algorithm design and high performance computing.

Li Li was born in Jilin in 1985. She received her Bachelor degree of Electrical Engineering from Tianjin University. She is currently pursuing her Master's degree of Electrical Engineering at Shanghai Jiao Tong University. Her research interest includes power system stability analysis and emergency control.