A Distributed Computing Environment for Probabilistic Transient Stability Analysis.

K.Jayashree and K.S.Swarup, Member IEEE,
Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai

***Abstract* — This paper proposes a new method for risk assessment of power system due to transient instability using distributed computing approach. As the size of the electric power system continues to grow, the risk assessment has to be more computationally efficient and robust. Transient stability analysis plays an important role in planning and secure operation of power system whose parameters are stochastic in nature. Probabilistic transient stability analysis will be more comprehensive that requires lot of computations for accurate results. This paper presents a distributed computing approach for transient stability analysis in terms of measuring critical clearing time and the overall risk index for various uncertainties. A prototype system is developed in our lab to deploy this method. The entire problem is divided into jobs and distributed among the processors. Parallelism is achieved by sharing each task across processors. The effectiveness of distributed parallel computing is demonstrated by performing Transient stability analysis on standard IEEE test systems. The performance of the algorithm is analyzed in terms of computation speed and efficiency in comparison with the sequential approach.**

Keywords — Transient Stability Analysis, Distributed Computing, Parallel processing.

I. INTRODUCTION

The ever increasing demand for electrical energy requires larger interconnected power systems with greater generation output. This means that the security of these power systems when subjected to large disturbances has to be examined. Therefore, transient stability becomes increasingly important in system planning and operation. Power system engineers in many countries are facing increasing pressure on increasing the computational capability to handle large data of power systems with constant expansion. The need of power system assessment can be for planning purpose or it can be for consistent and reliable energy supply especially after deregulation. With constant expansion of power system, more uncertainties are evolving and require sufficient analysis such as probabilistic analysis. Probabilistic approach is considered as more comprehensive and rational for realistic power system problems especially for today's power industry. However the probabilistic analysis requires high performance computation environment which reduces computation time particularly for analysis of large scale complex power systems.

Traditionally, power systems have been planned and operated using deterministic transient stability criteria. The utilities will need to know the level of risk associated with their observed criteria so that they adjust their service quality based on consumer expectation i.e. the acceptable level of risk and corresponding price. The computer aided transient stability analysis using deterministic approach [1] gives a clear formulation of transient stability studies of multi machine system. Since the real time power system is stochastic in nature, it is necessary to do probabilistic transient stability analysis by modeling the various constraints in probabilistic manner [2].

For large scale system, probabilistic approach is done by using the historic real time data of load models and fault models [4]. Historical statistics on the probabilistic states of load level factor, fault type, fault location, fault clearing, and automatic reclosing can be used in a Monte Carlo formulation to generate sample states for the studies. Instead of graph theory, a conditional probabilistic approach can also be used as [2]. Hence it becomes necessary to compute the overall risk index of the power system whose parameters are probabilistically varying .It is necessary to adopt some efficient method, so that the computation time can be reduced. [5] & [6]. To achieve this, distributed parallel processing algorithm has been introduced [7] .

Conventional transient stability analysis with energy function based approach which evaluates transient energy function provides only approximate results. Time domain simulation is one of the most reliable methods for power system transient stability analysis. For large scale power systems the time domain simulation is still a challenging issue due to huge computational burden[10].

Even though the time domain simulation provides accurate results, it needs large number of simulations and also any change in system needs the simulation to be carried out from the beginning. To achieve better computational efficiency, a Distributed Computing approach can be used. So that all the idle systems can be utilized in an efficient way to reduce the processing time by distributing the job/large operation among the PC's in the cluster. Computation of Overall risk index of the power system due to transient instability with probabilistic modeling of loads and faults has been done in this work using distributed computing algorithm.

The application of this methodology reduces the complexity of probabilistic transient stability analysis problem. On the other hand, considerable research efforts have also been done to develop dedicated computer architectures based on supercomputers or network of workstations for the fast solution of the machine swing equations.

II. Probabilistic Transient Stability Analysis

Transient stability refers to ability of the power system to remain in synchronism when subjected to large disturbances. A major disturbance upsets the balance between the mechanical input and electrical output of the generator, with the result that some generators may accelerate while others may decelerate. The rotor angles will undergo wide variations and in this process the synchronism of the system gets affected. If the individual machines in a multi machine system are operating in steady state equilibrium and a large disturbance of any kind is imposed on the system, then the system is transiently stable if relative rotor angles reach steady state values. Any machine can be chosen as reference, generally the machine with the largest inertia, namely the machine connected to swing bus, is taken as reference for computing relative rotor angles.

In probabilistic transient stability analysis, the risk evaluation of the overall system due to transient instability has to be found. A prefault system state is defined by a network topology, a generation pattern, and a system load level. These system factors can be deterministically specified or randomly selected depending on the study purpose. A load curve model has to be considered and essentially a discrete probability distribution of the system load, with each load level having a probability value is used in this work.Also the probability of fault locations such as sending end bus, near end of the line, middle line, far end of the line and receiving end bus have been considered. The entire simulation is done for three phase to ground fault.

In this work, demand modeling is done by means of normal distribution. The simulations are done for three-phase to ground fault for different fault locations and load variations and the overall risk index of the system is evaluated. The Probability distribution of the system load levels and various fault locations used for this work are as given below.

TABLE-I : PROBABILISTIC MODELING.

System Model	Description
System Load levels	125% 100% 95% 90% 80% 70%
Fault Locations	Sending end bus (0%) Close end (first 20% of the line) Middle end (60% of the line) Far end(80% of the line) Receiving end bus (100%)

The flow diagram which depicts the entire procedure for Probabilistic transient stability analysis has been shown here. The various steps are depicted clearly and the transient stability simulation is done using Modified Euler's method. The fault clearing time is taken as 0.25 seconds and the critical clearing time is computed for all the cases to obtain the overall risk index of the system.

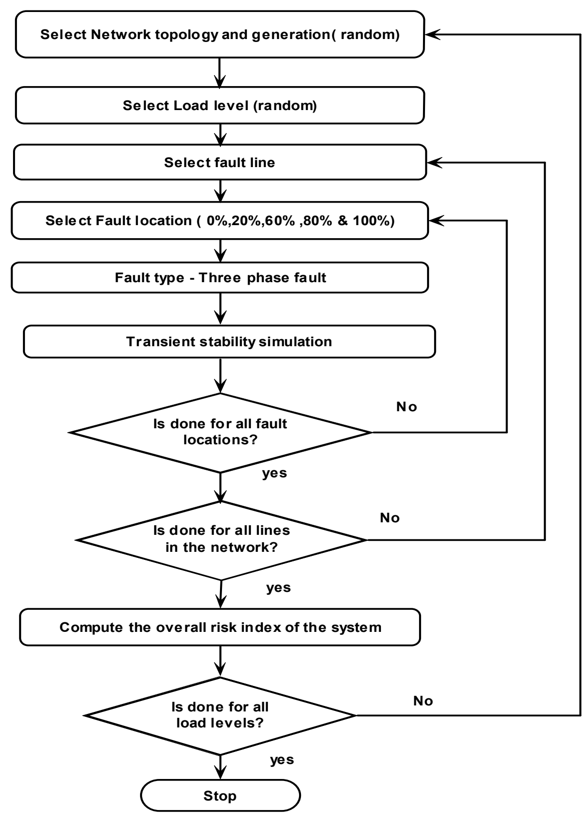

FIG: 1 FLOW DIAGRAM OF PROBABILISTIC TSA

If the critical clearing time is greater than fault clearing time, then the stability index is set as 1. Otherwise, the system will be unstable and in this case the stability index is set as 0. Using this information for all the cases the probability of stability and instability are found and the overall risk index of the system is evaluated. The overall risk index of the system is given by the following expression as,

$$\text{Overall risk index} = \frac{N_{us}}{N_s + N_{us}} * 100 \quad (1)$$

Where, 'N_s' is the number of stable events and' N_{us}' is number of unstable events.

III. Distributed Computing and Parallel Processing

The power system has a complex infrastructure. It is composed of highly interactive, non linear, dynamic entities that spread across vast areas. In any situation, including disturbances caused by natural disasters, high demands, centralized control require multiple, high-speed, two-way communication links, powerful central computing facilities and an elaborate operations control centers to give rapid responses. We can use Networked computers which are not well utilized in the industries/institutions. These unutilized computers can be used well and optimally using this Distributed computing. Using this technology we don't need super computers. In large organizations super computers are used. These super computers are more costly comparing with Personnel computers. But these Networked computers can be well utilized and can be perform tasks that are commonly given to the super computers. A primary focus of this Distributed environment harness idle CPU cycles and storage space of tens, hundreds, or thousands of networked systems to work together on a particularly processing-intensive problem. The growth of such processing models has been limited, however, due to a lack of compelling applications and by bandwidth bottlenecks, combined with significant security, management, and standardization challenges. In this work MATLAB distributed computing environment has been used both for transient stability simulation and distributed parallel processing.

MATLAB Distributed Computing Environment enables us to co-ordinate and to execute independent operations simultaneously on a cluster of computers, speeding up execution of large jobs. A job is some large operation that you need to perform in your MATLAB session. A job is broken down into segments called tasks. You decide how best to divide your job into tasks. You could divide your job into identical tasks, but tasks do not have to be identical. The MATLAB session in which the job and its tasks are defined is called the client session. This is on the machine where we program MATLAB. The MATLAB Distributed Computing Server performs the execution of job by evaluating each of its tasks and returning the result to the client session. The job manager is the part of the server software that coordinates the execution of jobs and the evaluation of their tasks. The job manager distributes the tasks for evaluation to the server individual MAT LAB sessions called workers.

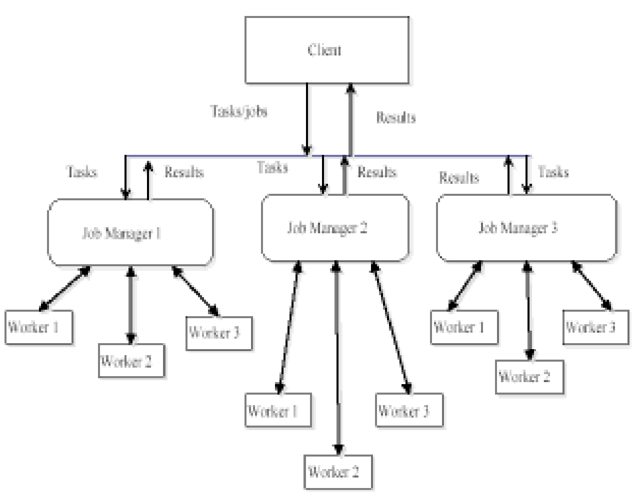

FIGURE: 2 MATLAB DISTRIBUTED COMPUTING ENVIRONMENT

The configuration of a distributed parallel job is as follows.

Identification of scheduler
Myscheduler =findResource('scheduler', 'Configuration', Jobmanager')

Creating a job
Job1 = createJob(Myscheduler)

Creating Tasks within the job

Submitting the job to the job managers
submit(Job1)

Wait for the jobs to get completed
waitForState(Job1)

Retrieve the results
results = getAllOutputArguments(Job1)

Destroying the job from scheduler's data location
Destroy(Job1)

Before configuring the job the configurations have to be validated and the proper network requirements are to be checked out. The test connectivity between the client to nodes, nodes to nodes and nodes to clients are being done by admin centre of MATLAB Distributed computing server.

IV DISTRIBUTED PROBABILISTIC TSA

The Probabilistic Transient stability assessment using serial computation is shown in the figure 3. There are six load models and four fault models. For each load level, the transient stability program has to run for all the four fault models. Suppose, a system has n-number of buses, m-transmission lines and L-load levels, each fault model needs (n+3m) number of simulations for one particular load level. For 6 load levels, we need 6(n+3m) number of simulations.

In Distributed processing, the number of jobs is same as the number of load levels and these distributed jobs are submitted to each job manager who has 4 workers each. For a particular load level, with four workers, simulation is done with 4 parallel computations. Hence the numbers of simulations are reduced to (n+3m).Even these tasks can be fine grained depending on the size of the system and number of processors available. The prototype setup for distributed computing environment consists of 6 job managers (PC's) each having 4 workers. Totally 24 workers are simultaneously executing the job once the corresponding job managers submit the tasks to them. Here the jobs are coarse grained and are independent on each other. Once the computation is over, the workers pass the output arguments to their job managers.

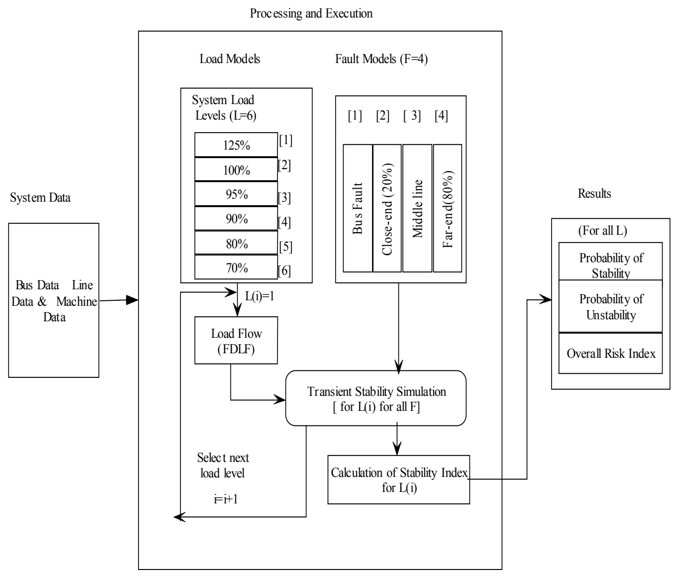

FIG: 3 SEQUENTIAL PROBABILISTIC TSA

After retrieving all the results, the job managers submit the results to the client machine. It is observed that as the number of processors gets increased, the efficiency of the distributed processing also improves. The speed up of the processing depends on the size of the system too. Hence the effectiveness of the distributed parallel processing depends on job scheduling, granularity, number of processors, communication between the processors, and size of the system or data. The Probabilistic Transient Stability Analysis using MATLAB Distributed Computing is given in figure (4).

V. Performance of Parallel Processing

A. Granularity

In parallel computing granularity is a qualitative measure of computation to communication. It can be categorized into two categorized based on granularity as coarse grained parallelism and fine grained parallelism. In coarse grained parallelism relatively large amount of computation work is done between communication events, whereas in fine grained parallelism small amounts of computational work is done between communication events. In this work coarse grained parallelism is adopted.

B. Scalability

Scalability refers to a parallel systems ability to demonstrate a proportionate increase in parallel speed with the addition of more processes. The factors that contribute to the scalability include hardware, application algorithm and the parallel overhead

C. Speedup

The speedup is the ratio of the runtime of a serial solution time to a problem to the parallel execution time [6].

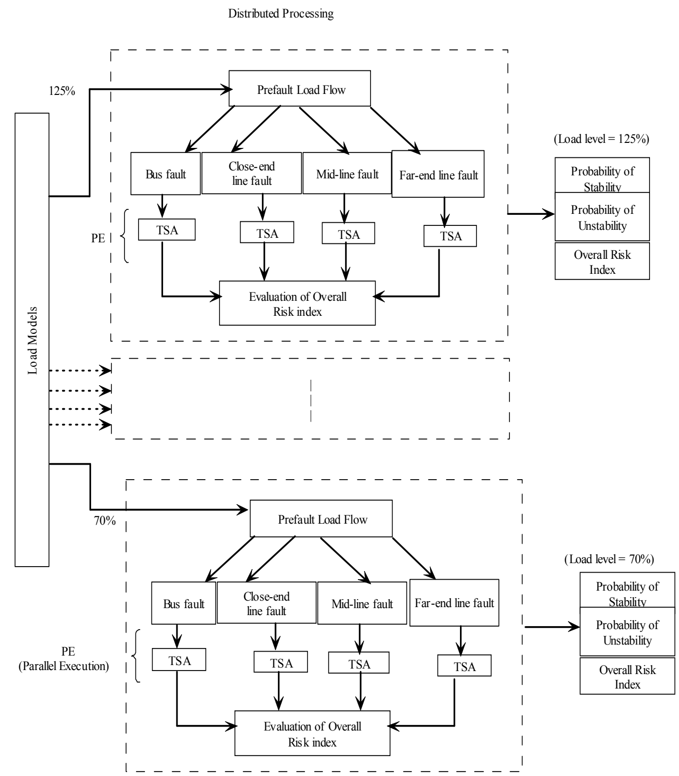

FIG: 4 DISTRIBUTED PROBABILISTIC TSA

.

It can also be defined as the ratio of the fastest known serial program on one processor to that of the parallel program running on 'p' processors of the parallel system. Speedup is given by

$$S(n,p) = \frac{T(n)}{T(n,p)} \qquad (2)$$

where,

p *is the number of processes*

$T(n)$ *is the serial execution time*

$T(n,p)$ *is the parallel execution time*

For a fixed value of p, $0 < S(n,p) \le p$

If $S(n,p) = p$, a program is said to have a linear speedup. This is a very rare condition since most parallel solutions add some communication overhead among other processors. A common occurrence is Slowdown, in which case the parallel program running on more than one processor is actually slower than the serial program. The run time of a serial program is the actual time a program takes to solve a problem from the beginning to the completion of execution including the I/O operations.

$$T(n) = T_{calc} + T_{I/O} \qquad (3)$$

The run time of a parallel program is the time that has elapsed from the moment the first process actually begins execution to the moment the last process completes execution of its last statement. In order to make good estimate of the performance of the parallel program the cost of communications must be included along with the I/O operations.

$$T(n) = T_{calc} + T_{I/O} + T_{comm} \quad (4)$$

D. Efficiency

Efficiency is the measure of the processor utilization in the parallel program, relative to the serial program. It is defined as

$$E(n,p) = \frac{S(n,p)}{p} \quad (5)$$

As $0 < S(n,p) \le p, \quad 0 < E(n,p) \le 1$

If $E(n,p) = 1$, then the program has a linear speedup.

If $E(n,p) < 1/p$, then the program has slowdown.

VI. Results and Discussion

The distributed parallel processing approach for probabilistic transient stability analysis to compute overall risk index of the system has been implemented with a prototype cluster having 6 nodes with each node having 4 workers. The data communication is done by means of LAN among the PC's in the cluster. This facilitates to run a large number of processes simultaneously, thereby reducing the computation time.

TABLE: 2 OVERALL RISK INDEX

Sl.No	Load levels	Overall Risk index		
		IEEE 14 bus system	IEEE 118 bus system	IEEE 300 bus system
1.	125%	38.67%	53.6%	69.6%
2.	100%	33.689%	43.76%	58.76%
3.	95%	32.00%	48.68%	53.68%
4.	90%	32.00%	49.53%	51.53%
5.	80%	30.23%	34.32%	37.32%
6.	70%	16.12%	18.2%	20.32%

The effectiveness of the parallel processing approach is justified by the speedup and efficiency in comparison with the sequential algorithm. The performance is tested on IEEE Standard test cases such as 14 Bus, 118 Bus, and 300 Bus Systems. The simulation results are given in Tables 2 & 3.

TABLE: 3. EXECUTION TIME, SPEEDUP AND EFFICIENCY OF TEST

Test Case	Number of Procs.	Serial Time (sec)	Parallel Time (sec)	Speedup	Efficiency (%)
14 Bus 20 Lines	24	1333.01	144.52	9.18	32.47%
118 Bus 179 Lines	24	49408.52	5319.2	9.789	40.72%
300 Bus 411 Lines	24	99109.79	11825.16	15.088	62.87%

As the size of the system increases, the execution time for the complete contingency analysis increases. The distributed parallel processing approach shows a considerable reduction in the total execution time for large systems. However there is no favorable improvement for small systems like IEEE 14 Bus Fig. 5 shows the comparison of execution time of serial and parallel processing. Scalability is good, as the size of the system increases the speedup also increases as shown in Fig 6.

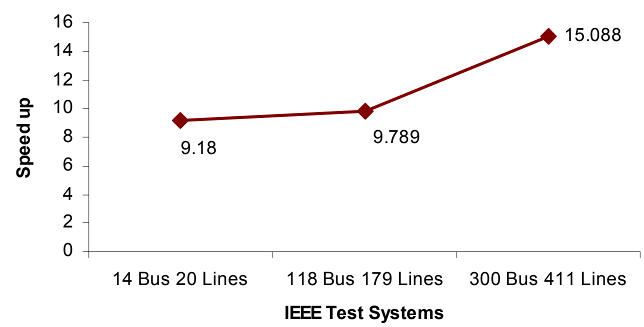

FIG:5 PLOT OF SPEED UP FOR VARIOUS TEST CASES

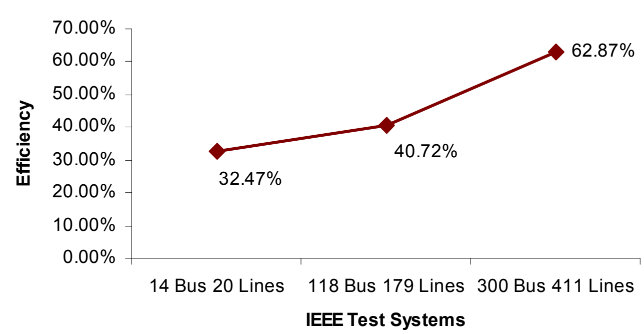

FIG. 6 PLOT OF EFFICIENCY FOR VARIOUS TEST CASES

The scalability of the algorithm is verified from the case studies. It is noted that as the size of the system increases the number of iteration involved in load flow process increases, but still a good parallel speedup is achieved using the proposed method. As defined in the earlier section, Speedup as well as Slowdown in exhibited in the test cases [6]. Lower order system suffer slowdown due to the inter process data communication. Since the data communication is only a small percentage of the computations involved in the large scale systems, a good speedup is achieved and efficiency is also high. Hence Slowdown is observed in small test cases, whereas speedup is achieved for large scale systems.

VII. CONCLUSION

The Transient Stability study with probabilistic modeling of loads and fault locations are carried out using MATLAB Distributed Computing. The proposed scheme reduces the computation time depending on the number of workers (or) no of nodes connected to the client. The computational efficiency increases when the system size increases.

The Risk index of the system is computed using probabilistic modeling of load and fault locations. The models of fault types and successful automatic reclosure of circuit breakers can also be considered in the future work.

This will enhance the real time simulation and make it simple, fast and accurate. Tests have been performed on IEEE standard test systems like 14 Bus, 118 Bus and 162 Bus and 300 Bus. It shows appreciable performance in terms of speedup and efficiency for large systems. However for very small systems like 14 Bus System it reports a slowdown in performance. This can be further implemented for real time systems and will be a suitable tool for smart grid applications and planning.

REFERENCES

[1] H. Yuan-Yih and C. Chung-Liang, “Probabilistic transient stability studies using the conditionalprobability approach,” IEEE Transactions on Power Systems, vol. 3, no. 4, pp. 1565–1572, 1988.

[2] P. Anderson and A. Bose, “A probabilistic approach to power system stability analysis,” IEEE Transactions on Power Apparatus and Systems, pp. 2430–2439, 1983.

[3] E. Vaahedi, W. Li, T. Chia, H. Dommel, B. Hydro, and B. Burnaby, “Large scale probabilistic transient stability assessment using BCHydro’s on-line tool,” IEEE Transactions on Power Systems, vol. 15, no. 2, pp. 661–667, 2000.

[4] N. Al-Rawi, A. Anwar, and A. Abdul-Majeed, “Computer Aided Transient Stability Analysis,” Journal of Computer Science, vol. 3, no. 3, pp. 149–153, 2007.

[5] W. Li, Risk assessment of power systems: models, methods, and applications. Wiley-IEEE Press, 2005.

[6] M. Shahidehpour and Y. Wang, Communication and control in electric power systems: applications of parallel and distributed processing. Wiley-IEEE Press, 2003.

[7] M. Ten Bruggencate and S. Chalasani, “Parallel implementations of the power system transient stability problem on clusters of workstations,” in Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM). ACM, 1995, p. 34.

[8] G. Zi-jian, S. Li-bao, and L. Zhi-bo, “Design of a Distributed Parallel Architecture for Total Transfer Capability With Transient Stability Constraints.”

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

[10] M. Pai, Computer techniques in power system analysis. Tata McGraw-Hill, 2005.

[11] P. Anderson and A. Fouad, Power system control and stability. Wiley-India, 2008.