Distributed Security Constrained Optimal Power Flow Integrated to a DSM based Energy Management System for Real Time Power Systems Security Control

Juliana M. T. Alves[1*], Carmen L. T. Borges[2], Ayru L. Oliveira Filho[1]

[1]CEPEL – Centro de Pesquisas de Energia Elétrica, Caixa Postal 68007, CEP 21944-970 – Rio de Janeiro RJ – Tel: +55-21-25986419
juliana.timbo@cepel.br, ayru@cepel.br
[2]COPPE/UFRJ – Programa de Engenharia Elétrica, Caixa Postal 68504, CEP 21941-972 – Rio de Janeiro RJ – Tel: +55-21-25628027 – Fax: +55-21-25628080
carmen@dee.ufrj.br

Abstract. This paper presents the development of the distributed processing based Security Constrained Optimal Power Flow (SCOPF) and its integration to a Distributed Shared Memory Energy Management System (EMS) in order to enable real time power systems security control. The optimization problem is solved by the Interior Points Method and the security constraints are considered by the use of Benders Decomposition techniques. The SCOPF is initially parallelized using MPI and then integrated to the actual DSM based SCADA/EMS system SAGE, thoroughly used in the Brazilian power system including the National System Operation Center (CNOS). Results obtained on both the MPI and DSM platforms are presented for actual large size Brazilian power systems analyzed over a list of about a thousand contingencies. The results obtained demonstrate the high efficiency and applicability of the developed tool at Control Centers for real time security control.

Topics of Interest. Cluster Computing, Large Scale Simulations in Engineering, Parallel and Distributed Computing.

1. Introduction

Modern Control Centers of electrical power systems are equipped with computational tools to help the operators to provide high quality service with a minimum number of supply interruptions and at a minimum cost. The operation is done in a way to maintain the system in a secure mode, i.e., ensuring that the system will be operating continually even when components of the network fail, what are called contingencies [1]. The electric system is monitored by the Supervisory Control and Data Acquisition (SCADA) System, which periodically acquires analog measurements and status of switching devices from the network. The monitoring system also allows the operator to act in the system through remote controls, changing switches status and position of

[*] **Candidate to the Best Student Paper Award**.

transformers tap, etc. The inherent complexity of the electric system operation makes it necessary to have sophisticated functions of diagnosis, analysis and advising available at the Energy Management System (EMS), such as Network Topology Configurator, State Estimator, Contingency Analysis, Emergency Control, etc.

This paper deals with the integration of the Static Security Control to the functions of an EMS, through the solution of the Security Constrained Optimal Power Flow (SCOPF) problem. This function will give as result a set of control actions that should be taken by the operator to maintain the system in a secure mode even if any contingency of a predefined list occurs. However, one of the problems of SCOPF is that for large systems, the processing time is elevated. In that sense, this paper proposes the application of distributed processing in order to make feasible the use of SCOPF in a real time EMS environment. The SCOPF is initially parallelized using MPI – Message Passing Interface [2] and then integrated to an actual SCADA/EMS system, SAGE [3], thoroughly used in the Brazilian power system including the National System Operation Center (CNOS). The basic program used for development of the distributed tool is the software FLUPOT [4], whose solution of SCOPF is based on the Non-Linear Interior Points Method and in the Benders Decomposition technique for the security constraints consideration.

Recently, some papers have been published reporting implementations of OPF for real time application. In [5], the authors use the Unlimited Point Algorithm for the solution of the OPF. The parallelization is made at the matrix solution level. They use MPI for the distributed implementation. In [6], the class of genetic algorithms is used for the solution of the OPF, also using MPI for the parallel implementation. In [7] and [8], the concept of decentralized solution of the OPF problem is used, where the electric network is divided into areas and each area is optimized in a separate computer. The OPF is solved by the Non Linear Interior Points Method and PVM – Parallel Virtual Machine is used for the distributed implementation. In [9], the concept of decentralized solution is also used. In this case, however, the solution of the OPF is given by linear programming techniques. The distributed implementation is made using PVM. An implementation of SCOPF using distributed processing is found in [10], using linear programming techniques for the solution of SCOPF and considering only active generation rescheduling to obtain a secure solution. However, none of these papers deals with real time application of SCOPF with the same complexity as the present paper nor even considers the integration to an actual, commercial and operational SCADA/EMS system. The results obtained and reported here for an actual power system demonstrate the applicability of the developed distributed tool at Control Centers for real time Security Control.

2. Security Constrained Optimal Power Flow

The Security Constrained Optimal Power Flow has the objective to determine a feasible point of operation that minimizes an objective function, guaranteeing that even if any of the contingencies obtained from a list occurs, the post-contingency state will also be feasible, i.e., without limits violations [11]. From a given list of N possible contingencies, the SCOPF problem can be represented mathematically as:

$$\min f(z_o) \quad s.t. \quad a_0(z_0) \leq b_0 \quad a_i(z_i) \leq b_i \quad for \quad i = 1,2,\ldots,N \tag{1}$$

Where:

$f(.)$ is the objective function;

$a(.)$ represents the non-linear balance equations of the electric network together with the operative constraints;

z represents the variables that will be optimized in the solution of the problem (state and control variables).

Each set of constraints $a_i(z_i) \leq b_i$, for $i = 1, 2, \ldots, N$, is related with the configuration of the network under contingency and must respect the operations constrains in this condition.

The objective function to be minimized in the problem depends on the purpose of the utilization of the tool. For the use in control centers as a security control tool, the common objective functions are minimum loss, minimum deviation of the programmed operation point and minimum deviation of the scheduled area interchange. Other objective functions are also used in SCOPF problems, such as minimum reactive allocation, minimum load shed, minimum generation cost, etc. The state variables are, usually, the busbars voltages and angles. The control variables, which are modified in order to obtain the optimal operation point, are the generators active injection, terminal voltages and reactive injection, transformers tap position, areas interchange, etc.

The SCOPF can be interpreted as a two-stage decision process [12]:

- In the first stage, find an operation point z_o for the base-case problem, $a_o(z_o) \leq b_o$;
- In the second stage, given the operating point z_o, find new operating points z_i that meet the constraints $a_i(z_i) \leq b_i$, for each contingency configuration.

The solution method used in this work is based on Benders Decomposition, which allows handling separately the base-case problem and each of the N contingency sub-problems. To represent the possible unfeasibility of each contingency sub-problem, penalty variables are added to the problem in order to represent the amount of violation associated with the contingency operation point. Therefore, the minimization of the constraints violations can be defined as a new objective function and the contingency sub-problem can be formulated as:

$$w(z_o) = \min d^r.r \quad s.t. \quad a(z_0) \leq b \tag{2}$$

Where $r \geq 0$ is the vector of penalty variables for the group of operative constraints and d^r is the cost vector. From this formulation, it can be seen that if $w(z_0) = 0$, the sub-problem is feasible and if $w(z_0) > 0$, the sub-problem is unfeasible.

The SCOPF can, then, be re-written in terms of z_o as follows, where the scalar functions $w_i(z_o)$ are the solutions of the contingency sub-problems (2) for the given operation point z_o.

$$
\begin{aligned}
&\min\ f(z_o) \\
&s.t. \\
&\qquad a_0(z_0) \le b_0 \\
&\qquad w_i(z_0) \le 0 \\
&\qquad\qquad for \quad i = 1,2,\ldots,N
\end{aligned}
\tag{3}
$$

The Benders Decomposition method consists in obtaining an approximation of $w_i(z_o)$ based on an iterative solution of the base-case and the N contingencies sub-problems. The Lagrange multipliers associated with the solution of each contingency sub-problem are used to form a linear constraint, known as Benders Cut, which are added to the base-case problem solution. Figure 1 shows the flowchart of the SCOPF solution algorithm based on Benders Decomposition.

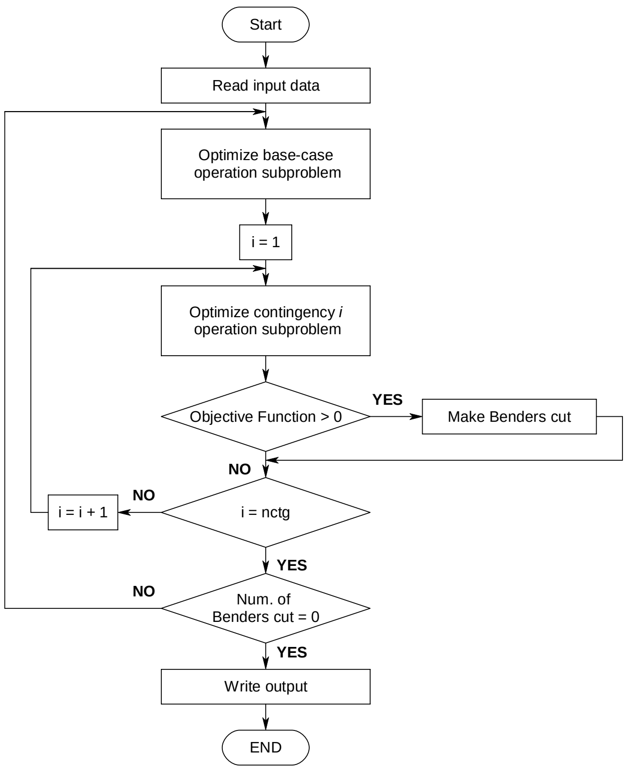

Fig. 1. SCOPF Solution Flowchart

The SCOPF solution algorithm consists, then, in solving the base-case optimization problem and then, from the operation point obtained in the base-case solution, to solve each of the contingency sub-problems. For each unfeasible contingency, a Benders Cut is generated. At the end of all contingencies solution, the generated Benders Cuts are introduced in the new solution of the base-case. The convergence is achieved when no contingency sub-problem generates Benders Cut. The contingencies sub-problems correspond to conventional OPF problems representing the configurations of the network under contingency. The base-case sub-problem is formulated as an OPF augmented by the constraints relative to the unfeasible contingencies (Benders Cuts). Each OPF problem is solved by the Interior Points Method and the network equations are formulated by the non linear model.

3. Distributed SCOPF Based on Message Passing

It can be noticed that the N contingencies sub-problems can be solved independently, once they only depend on the incoming operation point of the base-case, z_o. In that sense, the solution of the SCOPF based on Benders Decomposition can be directly benefited from the use of distributed processing, due to the natural parallelism that exists in the problem.

The parallelization strategy developed in this work is based on the master-slaves computation topology. When the parallel processing begins, each processor receives an identification number, the master being processor number zero and the slaves, processors number 1 to (*nprocs*-1), where *nprocs* is the total number of processors available. Figure 2 shows an example of contingencies allocation for a list of 10 contingencies distributed among 3 processors (1 master and 2 slaves).

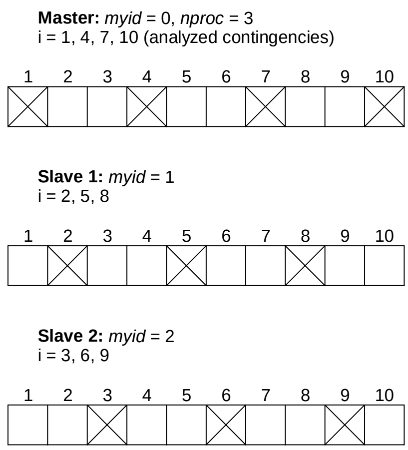

Fig. 2. Example of Contingencies Distribution among the Processors

The allocation of the contingencies sub-problems to the processors is performed by an asynchronous algorithm, based on the identification number of the processors (*myid*). In this way, the list of contingencies to be analyzed is distributed evenly among the participating processors in accordance with the value of *myid*, each one being responsible for the analysis of the contingencies of numbers (*myid* + 1 + *i*. *nprocs*), *i* = 0, 1, 2, 3,... until the end of the list. It is important to emphasize that the master processor also participates in the contingencies analysis task, guaranteeing a better efficiency for the whole process, once the processor is not idle while the slaves work.

Depending on the number of contingencies in the list and the number of processors, it can happen that some processors receive one contingency more than the others. The processing time for each contingency can also vary, since the number of iterations required for the OPF of the contingencies to converge varies from case to case. However, in the contingencies distribution strategy adopted, each processor, after finishing the analysis of a contingency, immediately begins the analysis of another without needing the intervention of a control process. In that way, the computational load of each processor is, on average, approximately the same for large contingencies lists, ensuring an almost optimal load balancing.

3.1. Distributed Algorithm based on Message Passing

The algorithm of the developed distributed application based on message passing can be summarized in the following steps:

Step 1: All processors read the input data.
Step 2: All processors optimize the base-case sub-problem.
Step 3: Parallel contingency sub-problems optimization by all processors.
Step 4: Master processor collects from all processors the partial Benders Cuts data structure (Synchronization Point).
Step 5: Master processor groups and reorganizes the complete Benders Cuts data structure.
Step 6: Master processor sends the complete Benders Cuts data structure to all processors (Synchronization Point).
Step 7: Verify if any of the N contingencies sub-problems is unfeasible. In the positive case, return to step 2.
Step 8: Master processor generates output reports.

All processors read the input data simultaneously, since the input files can be accessed by all via a shared file system, what eliminates the need to send the data read by just one processor to the others. The solution of the base-case sub-problem is also done simultaneously by all processors in order to avoid the need to send the results calculated by just one processor to the others. The reorganization of the Benders Cut data structure is a task introduced due to the distributed processing. After the analysis of their lists, each slave processor has its own partial Benders Cut data structures, which are sent to the master processor to be grouped and reorganized and later sent back again to all slave processors.

The flowchart of the developed parallel algorithm is shown in Figure 3. The two synchronization points of the algorithm, associated with the collection of the partial Benders Cuts structures and the distribution of the updated complete structure to all processors, are the only communication points of the algorithm, and for that reason, a high efficiency is expected from the distributed implementation. However, the efficiency will also depend on other factors, such as the communication technology used and the number of base-contingencies interactions (base-case plus contingencies sub-problems) necessary for convergence, since more interactions cause more communication among processors.

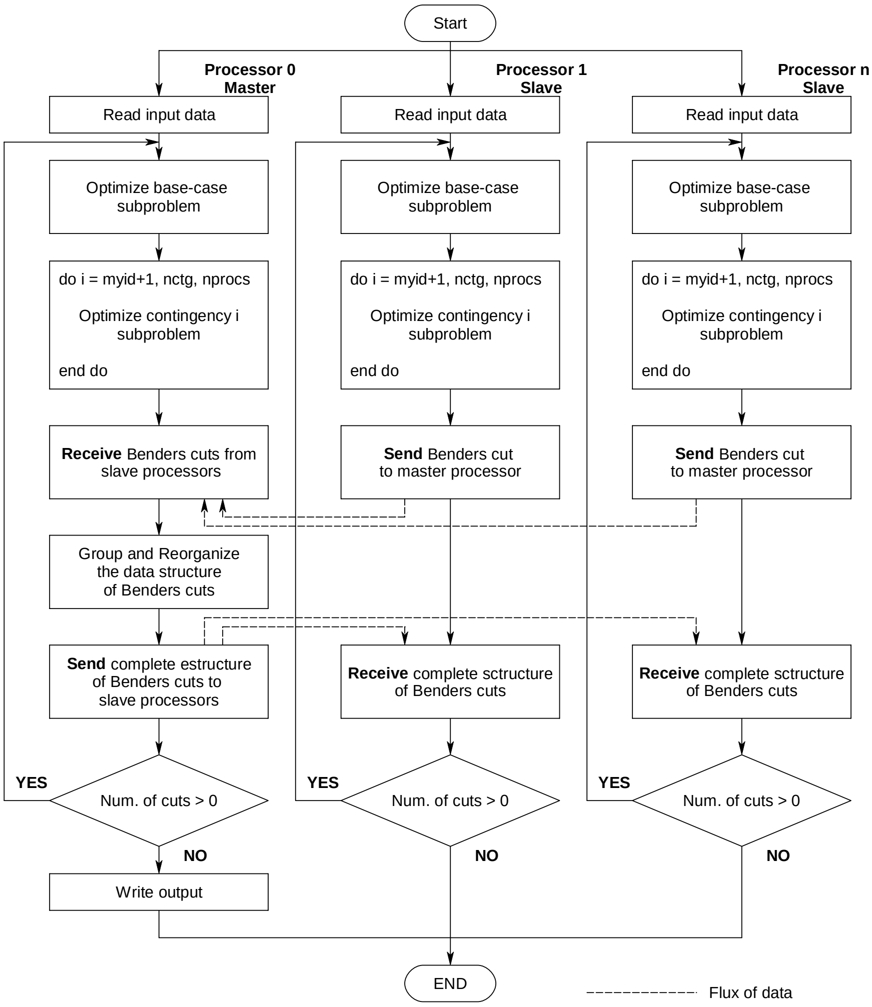

Fig. 3. Distributed SCOPF based on Message Passing Flowchart

4. Distributed SCOPF Integrated to the DSM Real Time System

SAGE [3] is a SCADA – Supervisory Control and Data Acquisition and EMS – Energy Management System, designed and developed by CEPEL, the Brazilian National Utility Energy Research Center. It includes modern energy management functions, such as State Monitoring (Network Configuration and State Estimation), Emergency Control (OPF Solution) and Security Monitoring (Contingency Analysis). The system is based on a distributed and expandable architecture. The use of redundant configurations and sophisticated control software ensures high reliability and availability for the system.

SAGE was designed to make possible the easy integration of additional modules directly to the real time database, which is build over a Distributed Shared Memory (DSM) support. The availability of common shared memory space and synchronization and control functions makes it a potential platform for distributed/parallel applications. For the implementation of applications that need access to the real time database, an API – Application Program Interface is made available. This API provides the interface routines for the communication and alarms subsystems.

4.1. Distributed Algorithm using the DSM System Resources

The integration of the distributed application to the DSM real time systems was done exploring the resources provided by the system. The information that needs to be accessed by all processors during the distributed SCOPF solution is written in DSM modules instead of being exchanged via MPI, as before. After each processor has solved its contingencies list, each one has its own Benders Cut data structure locally stored. The master processor reads the data structures generated by each slave processor, after each one has copied them to the DSM by the master processor request. After reading these Benders Cut structures, the master processor reorganizes them into a single complete structure. Soon afterwards, the master writes the new complete Benders Cut structure in the DSM, so that all processors can access it and continue with the solution process, in a parallelization strategy similar to the previously described for the MPI implementation.

The algorithm of the developed distributed application using the DSM real time system resources can be summarized in the following steps:

Step 1: All processors read the input data directly from the Real Time Database.
Step 2: All processors optimize the base-case sub-problem.
Step 3: Parallel contingency sub-problems optimization by all processors.
Step 4: Master processor asks each Slave processor to write its partial Benders Cuts data structure on the DSM.
Step 5: Each Slave processor writes its partial data structure on the DSM.
Step 6: Master processor reads the DSM and reorganizes the complete Benders Cuts data structure.
Step 7: Master processor writes the complete data structure on the DSM.
Step 8: Slave processors read the complete data structure from the DSM .

Step 9: Verify if any of the N contingencies sub-problems is unfeasible. In the positive case, return to step 2.
Step 10: Master processor generates output reports.

From steps 4 to 8 there is a synchronization process for the information exchange among the processors, via the access to the DSM, in order for the master to read all the partial Benders Cuts data structure generated by the slaves and to write the complete reorganized Benders Cuts data structure on the DSM.

Figure 4 shows the EMS functions organization at the real time environment of SAGE, already including the developed distributed tool for the Security Control module.

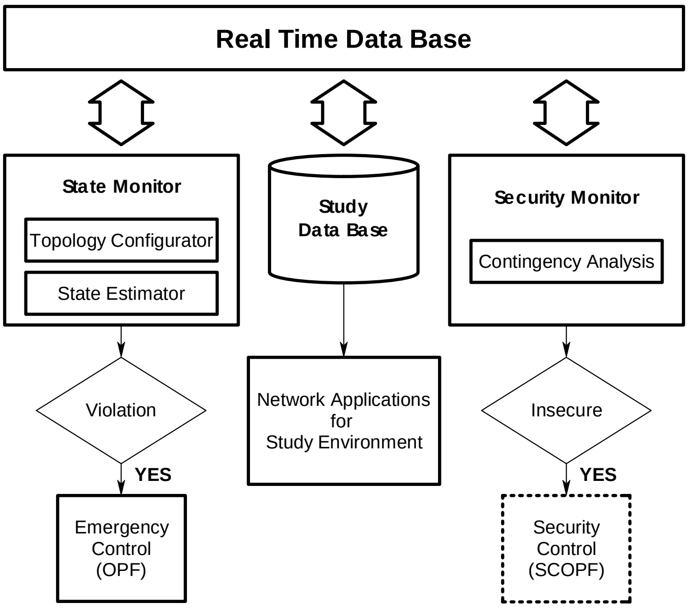

Fig. 4. SAGE EMS Functions Organization

The integration of the distributed solution of the SCOPF problem into SAGE makes it possible to use all available features of this SCADA/EMS: process control, high availability, cluster management, graphic interface, access to real time data, event triggering, alarms and logs. With the distributed tool integrated to the database of the EMS system, the SCOPF activation can be made by a request from the operator using the graphical interface, periodically or triggered by an event, such as the result of the Security Monitoring function. In the case it is detected that the system is insecure, the Security Control function (distributed SCOPF) is executed automatically afterwards. The use of SCOPF functionality in an EMS can potentially improve the operator online decision making process. The tool can advise the operator on which controls he should actuate to maintain the electrical system in a secure state.

5. Message Passing Implementation Results

5.1. Computational Platform and Test System

The computational platform used for the validation tests of the distributed implementation based on MPI was the Cluster Infoserver-Itautec [13], composed of 16 dual-processed 1.0GHz Intel Pentium III with 512MB of RAM and 256 KB of cache per node, Linux RedHat 7.3 operating system and dedicated Fast Ethernet network. The test system used is an equivalent of the actual Brazilian Interconnected System for the December 2003 heavy load configuration. The studied system is composed of 3073 busbars, 4547 branches, 314 generators and 595 shunt reactors or capacitors. The total load of the system is 57,947 MW and 16,007 Mvar, the total generation is 60,698 MW and 19,112 Mvar and the total losses are 2,751 MW and 3,105 Mvar.

For the security control, the objective function used was minimum losses together with minimum number of modified controls. The controls that could be modified for optimization of the base-case were: generated reactivate power, generator terminal voltage and transformer tap. The number of available controls is 963 controls, where 515 are taps, 224 are reactivate power generation and 224 are terminal voltages. The list of analyzed contingencies is formed by 700 lines or transformers disconnection. The contingency list was formulated in a way to obtain a good condition for tests, that is, the size of the list can be considered between medium and large, and some contingencies generate Benders Cuts during the optimization process.

5.2. Results Analysis

The SCOPF solution process converged in 3 base-contingencies iterations. In the first base-contingencies iteration, 12 contingencies generated Benders Cuts, in the second iteration, only 1 contingency generated Benders Cuts and, finally, in the third iteration, no contingency generated cut. The total losses after the optimization were 2,706 MW and 2,935 Mvar, what represents a reduction of about 2% in the active losses of the system.

The number of controls modified to lead the system to an optimal secure operation point was small, only 7 modifications of generator terminal voltages. That is due to the use of the objective function of minimum number of modified controls together with the minimum losses. This is an important issue for the use of SCOPF in the real time system operation. If the list of control actions is very long, it becomes unfeasible for the operator to perform them in time to turn the system secure.

The performance of the distributed implementation has been evaluated using from 1 to 12 nodes of the Cluster, obtaining exactly the same results as the sequential program. Table 1 shows the execution time, while Table 2 shows the Speedup and the Efficiency, for different numbers of processors.

It can be observed that the distributed implementation presents an excellent efficiency, superior to 92% using 12 processors of the distributed platform. The processing time is significantly reduced, changing from 8 minutes 25 seconds of the sequential processing to 45.43 seconds in parallel using 12 processors.

Table 1. Execution Time

No. Processors	Time
1	8 min 25 s
2	4 min 16 s
4	2 min 10 s
6	1 min 27 s
8	1 min 7 s
10	54.26 s
12	45.43 s

Table 2. Speedup and Efficiency

No. Processors	Speedup	Efficiency (%)
1	1	-
2	1.97	98.56
4	3.88	97.10
6	5.79	96.49
8	7.50	93.70
10	9.31	93.06
12	11.11	92.62

Figures 5 and 6 show the Speedup evolution and processing time reduction with the number of nodes used for the distributed solution, respectively.

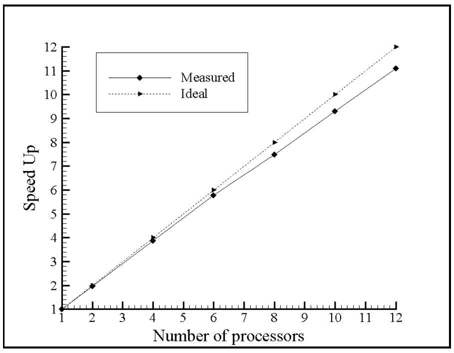

Fig. 5. Speedup Curve (MPI)

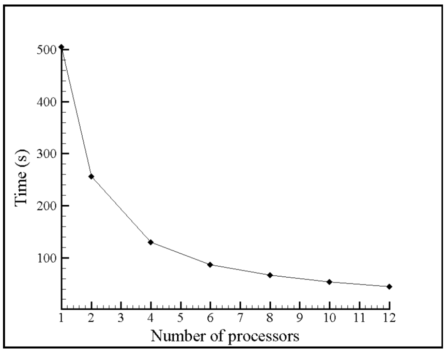

Fig. 6. Processing Time (MPI)

It can be also observed that the Speedup curve is almost ideal (linear). The Efficiency is only slightly reduced as the number of processors increases, what indicates that the parallel algorithm is scalable. From the good performance obtained, it can be expected that, if it is necessary to obtain a smaller response time for the real time application, this objective can be reached using a larger number of processors.

6. DSM Real Time System Implementation Results

6.1 Computational Platform and Test System

The computational platform used for the validation tests of the DSM integrated implementation was a Fast Ethernet network of computers at the Supervision and Control Laboratory at CEPEL, composed of eight 3.0GHz Intel Pentium IV microcomputers with 1GB RAM each and Enterprise Linux operational system. The test system used was generated based on real time operation data for the CNOS managed network of January 2006 on medium load level. This equivalent system is composed of 1419 busbars, 2094 branches, 388 generators and 92 shunts reactors and capacitors. The total load of the system is 43,654 MW and 14,582 Mvar, the total losses are 1,901 MW and 2,293 Mvar. The tests of the DSM platform were not based on the same system used in the MPI implementation in order to explore the real time database available on SAGE SCADA/EMS system.

For the security control, the objective function used was minimum losses together with minimum number of modified controls. The controls that could be modified for optimization of the base-case were: generated active and reactivate power, generator terminal voltage and transformer tap. All the variable tap transformers and generators were considered as control equipments in the optimization process. The list of analyzed contingencies is formed by 1012 simple contingencies involving lines, transformers, reactors, capacitors, load, generators and compensators disconnection.

6.2. Results Analysis

The SCOPF solution process converged in 3 base-contingencies iterations. In the first base-contingencies iteration, 16 contingencies generated Benders Cuts, in the second iteration, 11 contingencies generated Benders Cuts and in the third iteration, no contingency generated cut. The total losses after the optimization were 1,704 MW and 1,603 Mvar, what represents a reduction of about 8,5% in the active losses of the system.

A group of 122 control variables (about 10% of the total available) were modified to lead the system to an optimal secure operation point, 28 being active generations, 88 generators/synchronous terminal voltages and 6 transformers taps modifications.

Table 3 shows the execution time while Table 4 shows the Speedup and the Efficiency, for different numbers of processors, obtaining exactly the same results in parallel as in the sequential program.

Table 3. Execution Time

No. Processors	Time
1	11 min 48s
2	5 min 58s
4	3 min 2 s
6	2 min 2 s
8	1 min 34s

Table 4. Speedup and Efficiency

No. Processors	Speedup	Efficiency (%)
1	1.00	-
2	1.98	98.89
4	3.89	97.25
6	5.80	96.72
8	7.53	94.14

The execution time reduces from 11 minutes 48 seconds of the sequential simulation to about 1.5 minute on 8 processors of the distributed platform. The parallel implementation presents an efficiency superior to 94% using 8 processors, what can be considered a very good performance.

Figures 7 and 8 show the Speed up evolution and the processing time reduction with the number of processors used for the distributed solution, respectively.

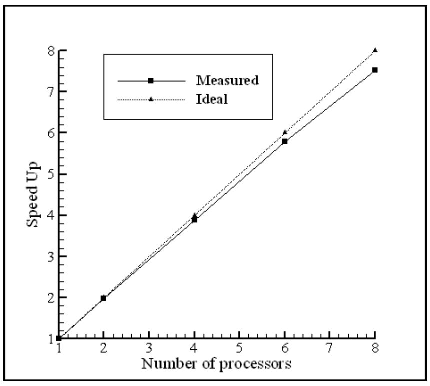

Fig. 7. Speedup Curve (DSM)

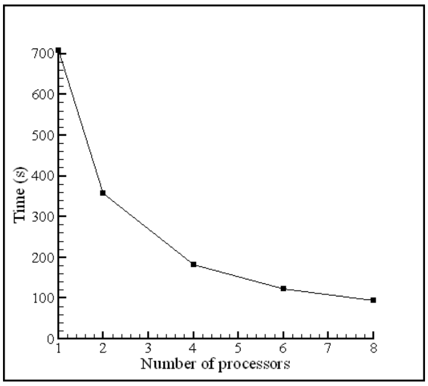

Fig. 8. Processing Time (DSM)

It can be observed that the Speedup curve is almost ideal. The Efficiency is only slightly reduced as the number of processors increases, what indicates that the parallel algorithm is scalable and a good performance can be expected when using a larger number of processors in the DSM based distributed platform.

Although the test systems are not the same in the MPI and DSM implementations, the results obtained for the two environments show similar performance. This can be verified comparing the characteristic of the speedup curves and the efficiency obtained for 8 processors, which are about 94% on both platforms.

7. Conclusions

This paper presented a proposal for enabling the system static Security Control in real time operation, based on the solution of SCOPF and in the use of distributed processing techniques. The use of this type of tool in the real time operation increases the security level of the system, since the operator, based on a list of control actions supplied by the tool, can preventively act on the system, avoiding that it evolves to a severe operative condition in the case some contingency happens.

The distributed application developed based on the MPI system is an autonomous tool, ready to be used in control centers if it is supplied, periodically, with the real time data. The integration of the distributed SCOPF directly to an DSM based EMS system added to the developed tool all the computational support offered by this type of environment, besides allowing the use of the computers already available at the control center for the SCADA/EMS functions.

In that sense, the Security Control based on SCOPF is made possible in real time application, using low cost and easily scalable platforms as a cluster of PCs or workstations. Although there is no consensus about which execution time is acceptable for the use of this type of tool in the real time operation, certainly the execution time obtained in the two test cases of the Brazilian System are acceptable. The smallest time obtained was about 45 seconds with 12 processors on the MPI environment and 1.5 minute with 8 processors on the DSM environment, which is perfectly compatible with the real time requirements of the Brazilian CNOS nowadays, where a time slice of two minutes is available for the security control task.

However, if a more time constrained response time is required, this objective than the reached not only upgrading the cluster platform but also de network technology, increasing networking speed and the overall system performance. Grid technology may also be considered for use between the several control centers available in the power system, in order to share the computational load with other clusters installed in other controls centers and also to low the overall cost in upgrading the computing platform.

References

1. Wood, A. J., Wollenberg, B.F., Power Generation, Operation, and Control. 2 ed. New York, John Wiley & Sons, 1996.
2. Gropp, W., Lusk, E. e Skjellum, A. (1996). Using MPI – Portable Parallel Programming with the Message Passing Interface, The MIT Press, Cambridge, UK.
3. http://www.sage.cepel.br/ – SAGE – Sistema Aberto de Gerenciamento de Energia, CEPEL (In Portuguese).
4. CEPEL, Optimal Power Flow Program – FLUPOT: User Manual (In Portuguese), Rio de Janeiro, RJ, Brazil, 2000.
5. Huang, Y., Kashiwagi, T., Morozumi, S., "A Parallel OPF Approach for Large Scale Power Systems". Fifth International Conference on Power System Management and Control, pp. 161-166, April 2002.
6. Lo, C.H.; Chung, C.Y.; Nguyen, D.H.M.; Wong, K.P., "A Parallel Evolutionary Programming Based Optimal Power Flow Algorithm and its Implementation", In: Proceedings of 2004 International Conference on Machine Learning and Cybernetics, Vol. 4, pp. 26-29, August 2004.
7. Baldick, R., Kim, B.H., Chase, C., Luo, Y., "A Fast Distributed Implementation of Optimal Power Flow", IEEE Transactions on Power Systems, Vol. 14, pp. 858-864 , August 1999.
8. Hur, D. Park, J., Balho Kim, H., "On the Convergence Rate Improvement of Mathematical Decomposition Technique on Distributed Optimal Power Flow", Electric Power and Energy Systems, No 25, pp. 31-39, 2003.
9. Biskas, P. N., Bakirtzis, A. G., Macheras, N. I., Pasialis, N. K., "A Decentralized Implementation of DC Optimal Power Flow on a Network of Computers", IEEE Transactions on Power Systems, Vol. 20, pp. 25-33, February 2005.
10. Rodrigues, M., Saavedra, O. R., Monticelli, A. "Asynchronous Programming Model for the Concurrent Solution of the Security Constrained Optimal Power Flow", IEEE Transactions on Power System, Vol. 9, No 4, pp. 2021-2027, November 1994.
11. Monticelli, A., Pereira, M.V. F., Granville, S. "Security-Constrained Optimal Power Flow With Post-Contingency Corrective Rescheduling", IEEE Transactions on Power System, Vol. PWRS-2, No 1, pp. 175-182, February 1987.
12. Granville, S., Lima, M.C. A., "Application of Decomposition Techniques to VAr Planning: Methodological & Computational Aspects", IEEE Transactions on Power System, Vol. 9, No 4, pp. 1780-1787, November 1994.
13. http://www.nacad.ufrj.br/ – NACAD – Núcleo de Atendimento de Computação de Alto Desempenho, COPPE/UFRJ, (In Portuguese).