North American Power Symposium (NAPS), San Luis Obispo, California, October 1999.

Web-based Computing for Power System Applications

Hong Chen Claudio A. Cañizares Ajit Singh
University of Waterloo
Department of Electrical & Computer Engineering
Waterloo, ON, Canada N2L 3G1
c.canizares@ece.uwaterloo.ca

***Abstract*—This paper discusses the possible applications of web-based computing, which is based on Internet protocol, distributed processing and Java programming, for the analysis of large scale interconnected power systems. The discussion is carried out in view of the new open energy market demands and the low cost, high availability and expandability, interoperability, and computational speed-up that can be achieved by using web-based computing, as this technology permits sharing of data and computing resources across the whole system, as well as allows for the possibility of running applications in parallel. A web-based distributed application for short-term load forecasting is proposed, implemented and tested as an example to demonstrate the feasibility and advantages of using web-based computing in power system applications.**

Keywords: Web-computing, distributed processing, Java, SCADA/EMS, short-term load forecasting.

I. Introduction

Large power systems usually expand and cover large geographical regions comprising multiple areas, which are referred to as districts in this paper. Each district has its own power system analysis packages and Supervisory Control and Data Acquisition (SCADA) and Energy Management System (EMS) in operation. These computer systems, provided by different vendors, run on different hardware and software platforms, and usually have different configurations and applications. With the new open power market polices being implemented throughout the world, and particularly in North America, higher computational and communication demands are being put onto existing SCADA/EMS systems to allow for feasible and profitable energy transactions among the different market players [1]l; this is due to the need for speedy data sharing and increased demand for real-time control applications for the "optimal" operation of the grid. These computational problems must be resolved at the minimum possible costs, as utilities would like to protect large investments in existing legacy hardware and software.

Web-based computing, i.e., Internet protocols, distributed processing and Java programming, is at a stage that it now allows to provide an adequate solution to the concerns expressed above, as it makes the following ideas possible:

1. Build a widely distributed open system made up of a variety of computational systems operating on dissimilar platforms, so that more extensive data and applications can be shared easily and flexibly.

2. Build a distributed computing environment so that operation and computing can be performed in parallel by using geographically distributed resources.

Distributed processing has been studied and used in power system applications for quite some time now (examples of some recent publications in the area are [2, 3, 4]) . The use of web-based computing in power systems, on the other hand, is a more recent trend that is slowly gaining popularity. For example, in [5], Internet-based client/server concepts are used to monitor transmission substations. In [6], the authors introduce an Internet-based energy trading system which allows buyers and sellers of energy to freely engage in trading activities over a wide range of energy sources, products and geographical regions. The authors in [7] apply Java for the interactive learning of power system stability on the web. In [8], a remote power flow is introduced as a simulation environment for power system analysis and for educational purposes via the Internet. The authors in [5, 6] mainly use web technologies for information access and exchange, whereas in [7, 8], Java applets are used to run client applications via the Internet.

The current paper first introduces some basic concepts of web-based computing and then discusses the possible applications of this technology for power system analysis and operation. The implementation of a short-term load forecasting program is discussed and an example is run to demonstrate the feasibility and advantages of web-based computing in power system applications.

The paper is structured as follows: Section II briefly introduces basic concepts of web-based computing and Java. Potential web-based power system applications are discussed in Section III. Section IV describes the implementation and use of a web-based, distributed short-term load forecasting application as an example of web-based computing in power systems. Finally, Section V draws conclusions and discusses future research directions.

II. Web Technologies

A. *Web-based Computing*

Web technologies enable communication between dissimilar computers over a large geographical region via Intranet or Internet. These provide a general distributed computing environment, so that distributed applications can be implemented on it to exploit cheap but powerful parallel virtual machines [9]. Thus, web-based computing permits data sharing and computing over a large system range on heterogeneous hardware and software platforms, permitting the execution of a number of operations simultaneously. This technology is basically based on client/server paradigms and concurrent programming.

A simple mode of web-based computing is the case where a client machine invokes a program installed on a server through the Internet; in this case, the data has to be passed from the client to the server while the program still runs on the server. Another simple mode is the case where a client machine requests the server to deliver a program to it via the Internet; in this mode, the program runs on a client and uses the client's local data. A more attractive and important mode suitable for computational intensive

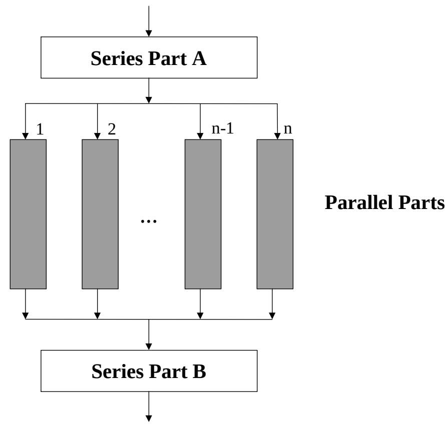

Fig. 1. Parallel web-based computing mode.

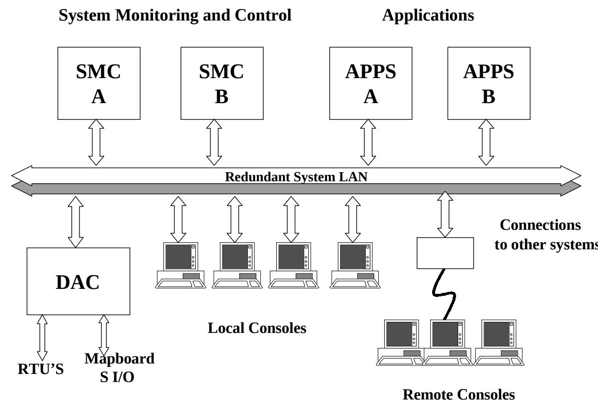

Fig. 2. Open architecture of SCADA/EMS systems [12].

applications is shown in Fig. 1. The application is divided into several subtasks; some are executed in parallel and others in series. The whole application is implemented on the web, with one machine acting as the master and the others as slaves. Parallel parts are executed simultaneously by different slaves while series parts run on the master; the slaves are coordinated by the master.

Compared to traditional parallel/distributed computing, web-based computing can be performed at much lower costs and at comparable computational speed-up, is easily available and highly expandable, and is relatively easy to maintain and update.

B. *Java*

Java is a general-purpose, high-level programming language and a powerful software platform [10]. The Java platform provides a fundamentally new model of computing that is very appropriate for large, geographically distributed, dissimilar, loosely interconnected computational systems, as is the case with power networks.

It is an object-oriented portable language that can run on heterogeneous operating systems, computing and hardware platforms. Java Remote Method Invocation (RMI) allows an object running in a Java Virtual Machine (JVM) to invoke methods on an object running in a different JVM [10]. These features make web-based applications possible and easy to implement, allowing sharing of computing resources via Intranet or Internet. Furthermore, Java allows building applets that can be used as network clients that communicate with the host where they come from, as well as built-in concurrency support for parallel operating and/or computing. Thus, the Java programming language together with its run time environment is the suitable computational environment for implementing web-based distributed applications.

III. Web-based Power System Applications

SCADA/EMS systems were introduced to help with the operation and control of power grids over half a century ago [11]. The architecture of these systems has evolved from centralized systems, to functionally distributed systems, to the current open systems [12, 13]. The traditional SCADA/EMS open architecture is based on a LAN [11], as depicted in Fig. 2.

Large power systems usually cover large geographical regions composed of several districts. Typically, each district has its own SCADA/EMS system in operation. The SCADA/EMS for the whole interconnected system is located on a centralized system control center, whereas for a district, the SCADA/EMS is located on a district control center. Due to the growth characteristics of power systems, the various SCADA/EMS systems that control the overall network and its districts are usually built during different years by different vendors; hence, these run on dissimilar hardware and software platforms and have different system configurations. The various SCADA/EMS systems used to control the grid are typically independent systems that run separate applications based on their own data resources, with only limited data communication between them.

With the deregulation/privatization of the power system industry, system data is becoming increasingly distributed, with more stringent and complex operational and control requirements. Thus, better and faster network applications are required that allow for wide data sharing and computing on heterogeneous platforms over a large geographic region. To be able to meet these increased computational demands, similar SCADA/EMS systems have been traditionally required; however, with the break up of integrated utilities and the need for fair participation of all market agents, this is just not feasible and/or cost effective any more. In this situation, feasible solutions can be readily provided through web-based computing based on well-developed Internet protocols, distributed processing and Java programming.

A possible interconnected computational network for power system operation and control is shown in Fig. 3. The network is divided into different districts and market participants; the LANs that belong to the different districts, system control center and other market participants are all connected to an Intranet or the Internet. Using Java applets, a machine in one LAN can access programs from the server in another LAN to execute them locally on its own data base. This simple mode for sharing computing

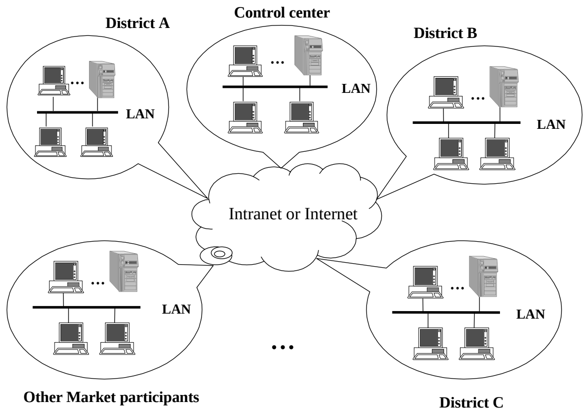

Fig. 3. Web-based distributed SCADA/EMS system.

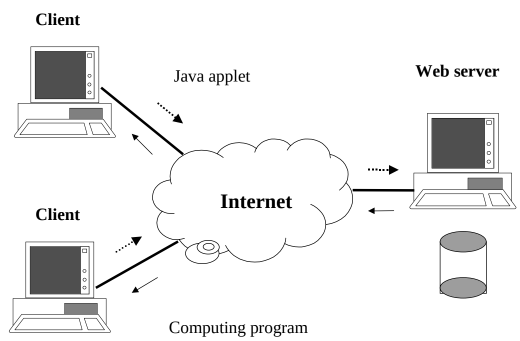

Fig. 4. Web-base share computing mode.

resources is illustrated in Fig. 4. All front ends are implemented as Java applets that connect to servers in the back end [15]. Hence, application programs do not need to be installed on each district, making it easier to maintain and update, and all system computer resources are available for applications to run, leading to low computing costs and high reliability. Computationally intensive power system analysis applications, such as optimal load flows [2], state estimation, security analysis [3], stability analysis [4], load forecasting, etc., can be decomposed into several subtasks according to districts. These subtasks are solved in parallel on each district's local data by machines on the corresponding district, and then coordinated on the system control center. This is feasible since only small amounts of data need to be exchanged among the different districts, resulting in speed-up to meet higher computational requirements.

Load forecasting is one of the EMS functions of particular importance in electric spot markets. Hence, the next section describes the implementation of a web-base distributed short-term load forecasting application to show the feasibility and advantages of web-based computing for power system applications.

IV. Web-based Distributed Short-term Load Forecasting

A. *Short Term Load Forecasting*

Power system short-term load forecasting is an important EMS function aimed at predicting system load with a lead time ranging from 15 minutes to one or seven days [14]. Since this is the basic resource for predicting the system's future load data, it is needed for scheduling of power systems, which is particularly important in electric spot markets to schedule day-ahead and/or hour-ahead power exchanges. Accurate load forecasting does not only provide timely information to operate power systems economically and reliably, but is certainly crucial to determine electricity prices in energy brokerage schemes.

At present, load forecasting is mainly performed at the main system control center, and complete and accurate system information is difficult to obtain. To speed up the process and generate more accurate load forecasts, especially in large systems, parallelism can be readily introduced by using the natural system division by districts to carry out the necessary computations. Load forecasting can be performed in this case using distributed processing, generating the total system load forecast based on the load forecast generated by each district first.

The weighted multi-model method is exploited here to forecast the district load; this method is based on using weighting factors corresponding to the forecasting time to predict the load [14]. In practice, any forecasting method such as time series, linear regression, expert systems, artificial neural networks, can be used. Since the relationship between the load and its factors is complex and non-linear, the accuracy of the different methods is basically system dependent. Notice that it is certainly feasible to have a different load forecasting method for each district; this way, districts can use their own existing load forecasting programs with only small changes. Here we use the same simple and efficient model to forecast the load for each district.

B. *Implementation*

1) Architecture The web-based distributed load forecasting is configured as a Master/Slave concurrent program, using the familiar client/server paradigms. A block diagram of the chosen architecture is depicted in Fig. 5. Master and slaves reside in different virtual machines, and these machines are connected through an Intranet or the Internet. The load forecasting is decomposed based on districts. The master is located at the system control center and the slaves are located at the district centers. The forecast entity for each district is encapsulated in a slave process, and each slave works on its own local data obtained from district SCADA and other historical data resources.

Java is chosen as the implementation platform. A Java applet is used as the graphic user interface to enable real-time interaction with the system and to display the forecasting result.

2) Concurrency The master invokes slaves to do district-wise load forecasting concurrently. Multi-threadding is used to allow concurrency in execution; the parent thread

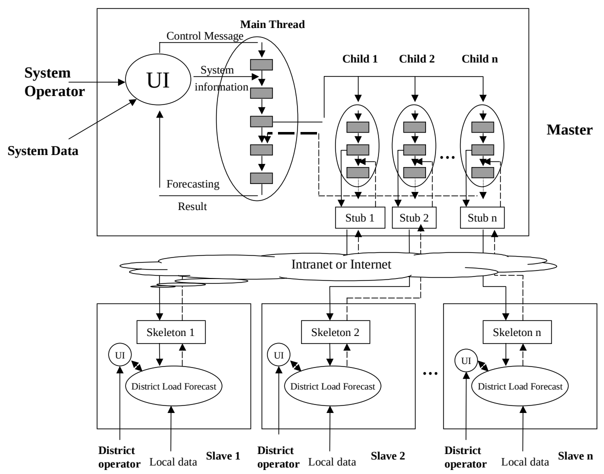

Fig. 5. Web-based distributed short-term load forecasting architecture.

Remote Interface

```
Public interface Forecast extends Remote{
        ForeDay executeForecast (Control c)
                throws RemoteException;
}

public interface Control extends Serializable {
        //interface of control function
}
```

Remote Object

```
    Public class DistrictForecast extends UnicastRemoteObject
    implements Forecast{
        …
        public ForeDay executeForecast(Control c){
                //district load forecasting
        }
        public static void main(String[] args){
                …
                //bind a name
        }
}
```

Fig. 6. Remote interface and remote object.

on the master generates the child threads, and each child thread invokes the district load forecasting on a slave. These child threads run asynchronously since there is no communication among them.

3) Synchronization In order to keep the system consistency, the slaves need to synchronize with the master. Control messages are sent from the master to the slaves; the slaves take action depending on the type of message received. The master waits until all slaves finish their computations and return results, and then generates the whole system forecasting results.

4) Message Passing Message passing is used for the master and slaves to share data and also to maintain synchronization between them. The master sends control messages and system-wide information to the slaves and invokes the forecasting program on each slave; after the salves are done processing, they return the district forecasting results to the master. Since master and slaves reside on different virtual machines, remote interfaces and remote objects are created on each slave, as shown in Fig. 6. The master gets remote references to remote objects in the slaves and then invokes methods that are responsible for district load forecasting. Java RMI provides the mechanism by which master and slaves can communicate and pass information back and forth.

C. Performance and Result

The web-based distributed short term load forecasting proposed here has been tested on a LAN network of up to 6 SUN 170-UltraSparcs workstations, one working as a master and the others as slaves; performance tests were done by varying the number of slaves. To simplify the implementation, the district systems represented at each slave were all assumed to be the same. The data for all these systems correspond to the distribution system data of the Jiangsu Province in China.

The resulting execution times are listed in Table I. From the results, we can see that the use of increasing number of slaves does not lead to a significant increase in the execution time, as expected. The processing burden on each machine is nearly even and just a small amount of Inter-process Communication (IPC) time is required. This is a typical coarse-grained parallel process, since communication cost is much less than the computation cost, and hence very suitable for distributed implementations.

TABLE I
Execution Time

Number of slaves	Execution time (ms)
2	335
3	339
4	353
5	359

Given the fact that the system used for these tests is concocted from repeating the same distribution system data several times, it is not possible to test the accuracy of the proposed load forecasting technique. However, compared to traditional centralized load forecasting, the web-based distributed load forecasting is expected to achieve higher forecasting accuracy, as district historical data used is typically more complete and up-to-date than centralized data.

V. Conclusions

This paper discusses the possible use of web-based computing, which is based on well-developed Internet protocols, distributed processing and Java programming language, for power system applications. The advantages of using web-based computing are highlighted in view of the distributed nature of large power grids. A new web-based distributed load forecasting application is implemented and tested, demonstrating its advantages over centralized forecasting, especially in large systems and new open market environments, where market participants need load forecasting with higher accuracy and speed.

To use web-based computing in real power systems, we need fast, reliable and secure networks and computing platforms. Current programs written in Java are not fast enough to satisfy the real-time requirements of some power system applications. Furthermore, there is a need for developing more efficient and suitable algorithms for web-based distributed implementation of power system applications.

References

[1] W. H. Dunn Jr., M. A. Rossi, and B. Avramovic, "Impact of Market Restructuring on Power Systems Operation," *IEEE Computer Applications in Power*, Vol. 8, No. 1, Jan. 1995, pp. 42–47.

[2] B. H. Kim and R. Baldick, "Coarse-Grained Distributed Optimal Power Flow," *IEEE Transactions on Power Systems*, Vol. 12, No. 2, May 1997, pp. 932–939.

[3] A. B. Alves and A. Monticelli, "Static security analysis using pipeline decomposition,"*IEE Proceedings Generation, Transmission and Distribution*, Vol. 145, No. 2, Mar. 1998, pp.105–110.

[4] G. Aloisio, M. A. Bochicchio, M. La Scala, and R. Sbrizzai, "A Distributed Computing Approach for Real-time Transient Stability Analysis," *IEEE Transactions on Power Systems*, Vol. 12, No. 2, May 1997, pp.981–987.

[5] W. L. Chan, A. T. P. So, and L. L. Lai, "Internet Based Transmission Substation Monitoring," *IEEE Transactions on Power Systems*, Vol. 14, No. 1, Feb. 1999, pp. 293–298.

[6] I. Slutsker, K. Nodehi, S. Mokhtari, K. Burns, D. Szymanski, and P. Clapp, "Market Participants gain Energy Trading Tools," *IEEE. Computer Applications in Power*, Vol. 11, No. 2, Apr. 1998, pp.47–52.

[7] F. Shahzad and A. Z. Khan, "An Application of Java for Power System Stability," *IEEE Power Engineering Review*, Vol. 9, No. 2, Apr. 1998, pp. 45–46.

[8] "Remote Power Flow Analysis," *http://power.elec.kitami-it.ac.jp/ ueda/java/pf/*

[9] F. M. T. Branzier and D. Johansen, "Distributed Open System," *IEEE Computer Society Press*, Los Alamitos, California, 1994.

[10] "The Java Tutorial," *http://www.javasoft.com/docs/books/tutorial*

[11] Y. Kokai and F. Masuda, S. Horrike, Y. Sekine, "Recent Development in Open Systems for EMS/SCADA," *Electrical Power & Energy Systems*, Vol. 20, No. 2, Jan. 1997, pp. 111–123.

[12] G. Ockwell and R. Kreger, "The Impact of Hardware on Open Architecture Design," *IEEE Trans. Power Systems*, Vol. 9, No. 1, Feb. 1994, pp.1–5.

[13] L. Murphy, F. F. Wu, "An Open Design Approach for Distributed Energy Management Systems," *IEEE Transactions on Power Systems*, Vol. 8, No. 3, Aug. 1993, pp. 1172-1179.

[14] C. Hong and L. Jianwei, "A Weighted multi-model Short-term Load Forecasting System," *Proc. IEEE International Conference on Power System Technology*, New York, NY, Vol. 1, 1998, pp.557–561.

[15] E. Ly, "Distributed Java Applets for Project Management on the Web," *IEEE Internet Computing*, Vol. 1, No. 3, May-June 1997, pp.21–26.

Hong Chen received her Bachelor (1992) and Master(1995) degree in Electrical Engineering from Southeast University in China. She worked in NARI (Nanjing Automation Research Institute, P.R.China) from 1995 to 1998, and was engaged in developing EMS power application software. Now she is a Ph.D student in the Department of Electrical & Computer Engineering at the University of Waterloo.

Claudio A. Cañizares received in April 1984 the Electrical Engineer diploma from the Escuela Politécnica Nacional (EPN), Quito-Ecuador, where he held different teaching and administrative positions from 1983 to 1993. His MS (1988) and PhD (1991) degrees in Electrical Engineering are from the University of Wisconsin–Madison. Dr. Cañizares is currently an Associate Professor at the University of Waterloo, E&CE Department, and his research activities are mostly concentrated in studying stability, modeling and computational issues in ac/dc/FACTS systems.

Ajit Singh completed his B.Sc. degree in Electronics and Communication Engineering (1979) at BIT, India, and M.Sc. (1986) and Ph.D. (1991) degrees in Computing Science at University of Alberta, Canada. From 1980 to 1983, he worked at the R & D department of Operations Research Group, the representative company for Sperry Univac Computers in India. From 1990 to 1992, he was involved with the design of telecommunication systems at Bell-Northern Research, Ottawa. He is currently an Associate Professor at Department of Electrical and Computer Engineering, University of Waterloo. Dr. Singh has published several research papers in the areas of software engineering, network computing, database systems, and artificial intelligence.