

e - DIAGNOSTIC FRAMEWORK FOR ELECTRICAL DRIVE SYSTEM IN DISTRIBUTED ENVIRONMENT

R.Subramaniyan

Research Scholar
DEEE, CEG
Anna University
Chennai, India

Dr.R.Ramesh

Associate Professor
DEEE, CEG
Anna University
Chennai, India

Abstract— The main objective of this paper is to propose and implement effective architectural models for monitoring and making remote diagnostics of electrical drive system in a completely secured, distributed, platform-independent and language-independent environment. Although significant progress has been made in this research area, still efficient distributed environment is not yet been attained. The Web-enabled architectural models are emerging as basic technology to support the integration of different applications using open Internet standard. An attempt is being made to convert existing architectural models into Web based model for monitoring and making remote diagnostics of electrical drive system in distributed environment. An embedded controller is used to acquire the parameter of the system.

Keywords—*e-diagnostic, electrical drive, Web and data acquisition*

I. INTRODUCTION

As a result of the recent advent of distributed computing and Internet technologies has given way to monitor the system in distributed environment. Condition monitoring of induction motors is commonly used in industrial applications to maintain safe and reliable operation of plants. The current practice in condition monitoring primarily involves using various forms of mobile devices usually with a single sensor input to perform tests at regular intervals. However, such devices and monitoring services are expensive and require an experienced operator for reliable decisions. This paper investigates an alternative low cost solution for continuous condition monitoring of induction machines using multiple sensors, which can be located next to a machine under test and can provide condition information using indicator lights for quick diagnosis (Bakhri et al.)

Bayindir and vadi proposed a novel remote monitoring and control system for induction motors using GPRS/GSM communication. Elavenil and Kalaiyani presented the integration of embedded system with a ZigBee wireless sensor networks technology for the speed and torque monitoring of three phase induction motor. Meliones et al. developed a flexible modular network-based application which supports all key services for automatic remote control and management of photovoltaic parks.

Leite et al. has shown a laboratory set up for real-time analogue monitoring instrument developed for an integrated test bench for teaching as well as for R&D activities, in power electronics and variable speed drives of

induction machines. The instrument is based on electronic modules and has been developed for monitoring the stator, rotor and magnetizing fluxes space phasors, electromagnetic torque as well as active and reactive powers.

II. ARCHITECTURE

A. Conventional Model

First, One-tier architecture has a huge advantage of its simplicity, since it doesn't need to handle any network protocols, so their code is simpler. Such code also benefits from being part of an independent operation. It doesn't need to guarantee synchronization with faraway data, nor does it need exception-handling routines to deal with network failure, bogus data from a server, or a server running different versions of a protocol or program. Figure 1 shows the conventional model for monitoring electrical drive system.

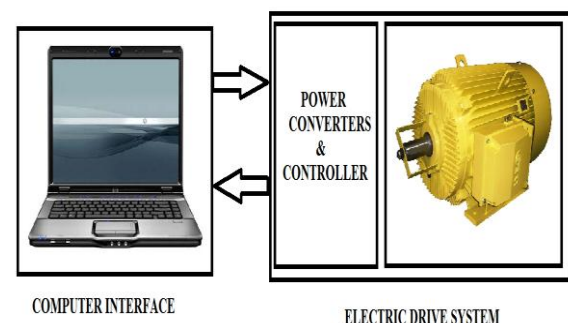


Fig. 1 . Conventional Model

B. Two Tier/Three Tire: Client – Server Model

A 2-tier architecture is Client server application and 3-tier architecture is Web based application. The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server. Because of tight coupling a 2 tiered application will run faster. Here the direct communication between client and server, there is no intermediate between client and server.

The evolution of distributed architecture resulted in distributed computing; the client-server paradigm refers to a model of network applications. Even though the client-server

model provides a more cost-effective solution of computing, developing the client-server programs is more expensive than traditional approaches with respect to engineering applications. Distributed computing overcomes the drawbacks with the monolithic and client-server applications and performs computations by a more flexible and cost effective approach. Real-time monitoring of electrical drive system can be carried out in distributed environment using remote procedure calls. Each electrical machine present in the drive system is considered as a client and hence multiple clients / single server architecture can be implemented, as shown in Fig. 2 to carry out the real-time monitoring of electrical drive system in distributed environment.

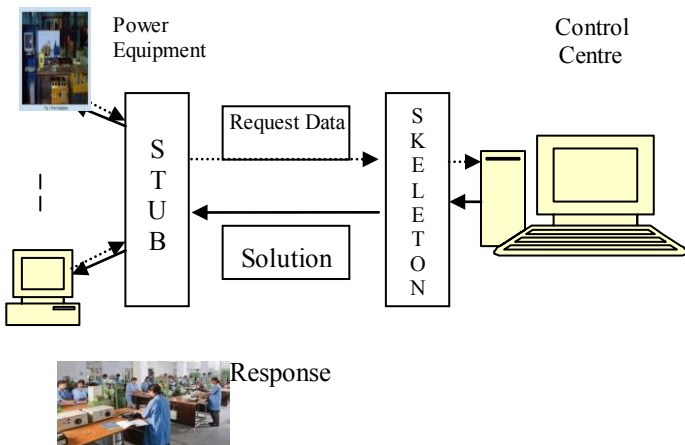


Fig. 2. Single-server / multiple-client architecture for real-time monitoring electrical drive system in distributed environment

A control centre basically monitors the various electrical machines through an applet for every specific period of time and frequently exchanges data with the client computer which is interfaced with machine. The server presents in the control centre does the required analysis and then distributes the results. Chronologically, the server process should be started first so that it can take the initiative to set up a connection link. It then starts waiting till it receives a connection request from the client. A client can register itself with the remote object just by invoking the registration procedure on the server object, when it needs a service from it. The remote object obtains the necessary information from the registered clients (electrical machine) and responds them correspondingly with the results at specific intervals of time. The total processes can be automated by making the server to get the measured values of various electrical parameters from each registered client at every specific period of time.

C. Web-Based model for monitoring electrical drive system

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 in it to exploit cheap but powerful parallel virtual machines. Web-based computing permits data sharing and computing over a large system range on heterogeneous

hardware and software platforms, permitting the execution of number of operations simultaneously. The concept of Web sharing mode is shown in Figure 3. This technology is basically based on client-server paradigms and concurrent programming. The features of proposed architectural models are detailed as follows:

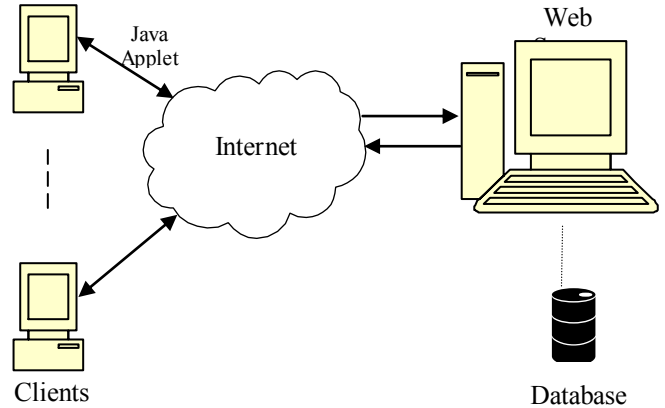


Fig. 3. Web-based share computing model

- In RMI mechanism, the client can communicate with server only if the stub is present and the availability of stub should be taken care by the developer. But in the case of Web service, automatically it takes care the availability of stub.
- Java RMI use optimized connection-oriented communications protocols that are either language specific, or have detailed rules defining how data-structure and interfaces should be realized. In contrast, Web services are based on the ubiquitous technologies that have grown up to support WWW-services (human-via-browser-to-application).
- There is no concept of an object reference; instead a service is defined by an end-point that supports various operations.
- Web services provide interoperability between various software applications running on disparate platforms
- Web services use open standards and protocols. Protocols and data formats are text-based where possible, making it easy for developers to comprehend.
- By utilizing HTTP, Web services can work through many common firewall security measures without requiring changes to the firewall filtering rules. Other forms of RPC may more often be blocked
- Web services allow software and services from different companies and locations to be combined easily to provide an integrated service
- Web services allow the reuse of services and components within an infrastructure.
- Web services are loosely coupled thereby facilitating a distributed approach to application integration

The proposed models use the built-in security mechanism and hence the distributed on-line monitoring of electrical drive

system through an applet definitely secures the safety of the server as well as the electrical drive system.

III. PERFORMANCE COMPARISONS AND DISCUSSIONS

The performance analyses are carried out in distributed environment. The distributed architectural models for real-time monitoring electrical drive system has been constructed using different technologies like RMI, RMI-IIOP, CORBA, XML-RPC and Web service. The consolidated views of all these approaches and their performance analysis have been presented. The major factor that influences the performance of the proposed models is the Round Trip Time (RTT) that includes the convergence time. The RTT measures the time needed from the point when the client sends the request and receives the response. The RTT is measured for the different system networks that invoke the services.

Java socket is an endpoint for communication between two processes i.e., server process and client process running in the same network. RMI provides sophisticated distributed and platform-independent environment to solve the engineering problems. The location transparency is the key frame of CORBA/RMI-IIOP. Location transparency of the proposed model is the ability to access and invoke operations on the CORBA server object without needing to know where the object resides. The Web service provides platform independence, language independence and location independence (Gray 2004). RTT has been measured for each request and response for various client applications by using above mentioned distributed technologies. The primitive data types like int, float, char, double, long, Boolean, byte and string have been considered for analyzing the proposed models.

To measure the performance of different models, different technologies have been used for implementation and a simple test client was prepared to obtain the desired performance results. The test has been carried out in the following different cases.

A. Server and Client on the Same Machine (Local)

The performance analysis was carried where all the services and the clients are running in the same machine. RTT has been measured. The graph is plotted between the round trip time calculated and for the different systems, as shown in Fig. 4.

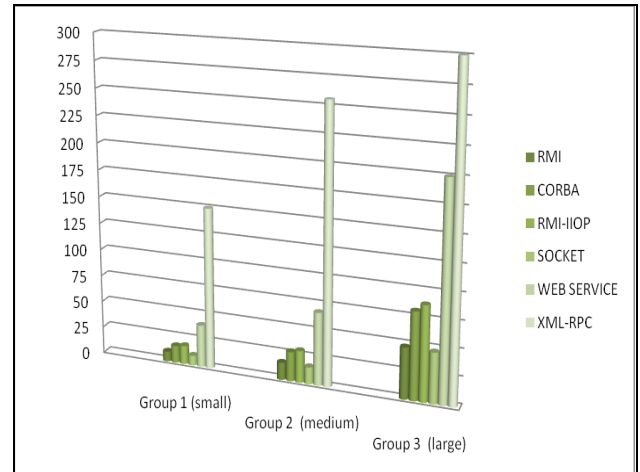


Fig.4. Performance result – Conventional method (Bar chart) Server and Client on Different Machines (Remote through Proxy Server Invocation)

The server and the client are running in different machines, where the remote machine is configured with firewalls. RTT is measured for different cases and the graph is plotted and shown in Fig.5.

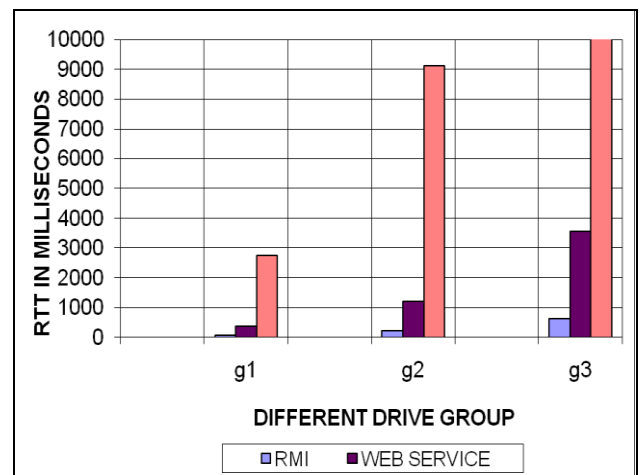


Fig. 5. Performance results - on remote invocation through proxy server

B. Different Web Services Invocation - Performance

The performance analysis was carried using the Web service, where the server and the clients are running in the same machine. RTT has been measured and the graph is plotted between the round trip time calculated and for the different system, as shown in Figure 6

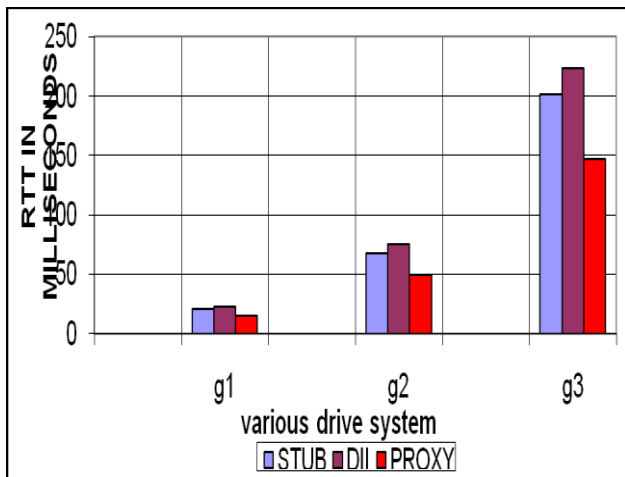


Fig .5. Performance results – Web Services invocation

Similarly, the Web service model on remote invocation and proxy server invocation has been developed to carry out the real time monitoring of drive system. The performance analysis was carried out and RTT has been measured. It is found that the time taken to invoke the service and to run the results increase as number of client increases.

IV. DISCUSSION

Since both SOAP and WSDL are XML-based, XML messages have to be parsed on both the sides. The client side proxies have to be generated on the client side before any communication can take place. XML parsing at runtime requires additional processing time which may result in longer response time of the server in case of a Web service server. The overhead of the Web service stems mainly from the usage of XML producing human readable text and is employed when interoperability with other Web services and applications is essential (Caincarini et al 1999). Even though Web services suffer from poor performance compared to other distributed computing approaches such as RMI and CORBA, the Web enabled model provides a cross-platform, cross-language data model that facilitates developing heterogeneous distributed applications. XML-RPC is found a useful way to tie together the systems written in different languages on different operating systems and enabling them to co-operate. The real advantage is that the structure of XML-RPC is flexible enough to be put to different engineering applications.

A. Pervasive Computing for monitoring electrical drive system

Pervasive computing can be defined as “The overall infrastructural support needed to provide proactively a rich set of computing capabilities and services to a user every time everywhere in a transparent, integrated and convenient way.” Pervasive computing also called ubiquitous computing, the growing idea towards embedding microprocessors in everyday objects so they imbedded with chips to connect the device to an infinite network of other devices. The idea that technology is moving beyond the

personal computer to everyday devices with embedded technology and connectivity as computing devices become progressively smaller and more powerful. The words pervasive and ubiquitous mean "existing everywhere."

Pervasive computing is about four things; users, applications, middlewares and networks. In between Personal and pervasive computing there are three more evolutionary steps namely distributed, web and mobile computing. For successful implementation of pervasive computing, protocol layering, packet switching, information caching, distributed file and database systems and encryption systems are also needed in pervasive computing. Pervasive computing defines a major paradigm shift from “anytime anywhere” computing to “all-time everywhere” computing. The support for pervasiveness will come from interoperability, scalability, smartness and invisibility on the top of mobility.

The technological advances that are needed to build a pervasive computing environment can be framed into four broad areas: devices, networking, middleware and applications.

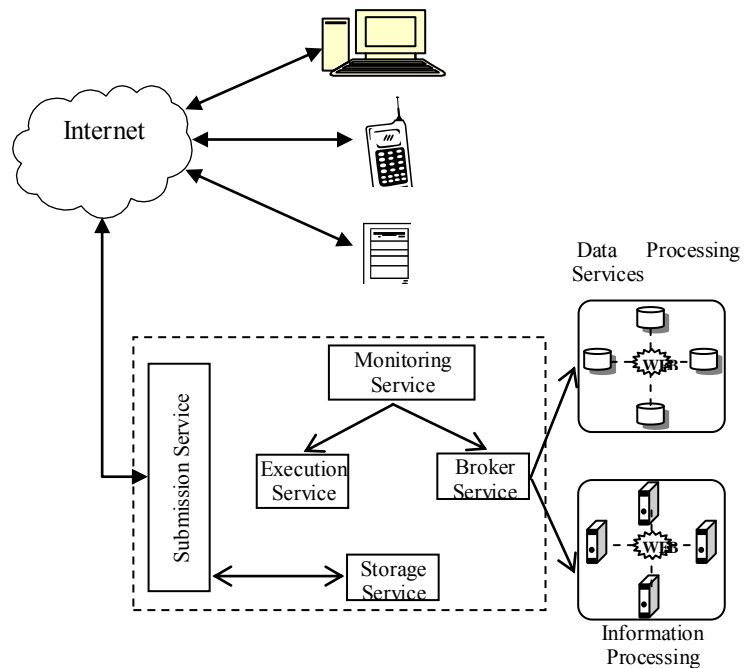


Fig. 7. Pervasive computing for monitoring electrical drive system

New intelligent devices called smart devices are embedded with microprocessors that allow users to plug into intelligent networks and gain direct, simple and secure access to both relevant information and services. In case of mobiComp, distributed Computing, a shell of middleware is essential to interface between them and pervasive network and end-user applications running on pervasive devices. Some of the functionalities of pervasive middleware are smartness, context-awareness, proactivity, transparency, mobility management, invisible interface and adaptability.

V. CONCLUSION AND FUTURE WORK

This study has a remarkable and significant importance for the reason that it can be used in monitoring, controlling and determination of system cases where the places are distant from the residential areas and have no access to the Internet. Moreover, it can be used as a training tool in professional and technical education. Future extensions of the presented work will study more novel approaches for monitoring the photovoltaic modules based on Wireless Sensor Networks (WSN) technology.

References

- [1] Meliones, A., Apostolacos, S., & Nouvaki, A. (2014). A web-based three-tier control and monitoring application for integrated facility management of photovoltaic systems. *Applied Computing and Informatics*, 10(1), 14-37.
- [2] G. Bayındır, R., & Vadi, S. (2013). Real-Time Monitoring and Control of the Parameters of an Induction Motor. *Elektronika ir Elektrotechnika*, 19(10), 145-150
- [3] Elavenil, P. E., & Kalaiivani, R. (2015). Overload and Speed Monitoring of Induction Motor Using ZigBee Wireless Sensor Networks and GSM Technology. In *Power Electronics and Renewable Energy Systems* (pp. 1007-1016), Springer India
- [4] Leite, V., Casais, F., Teixeira, H., Araújo, R., & Freitas, D. (2003). A real-time monitoring instrument of induction machines for an integrated teaching test bench.
- [5] Bakhri, S., Ertugrul, N., Soong, W. L., & Al-Sarawi, S. (2007, December). Investigation and development of a real-time on-site condition monitoring system for induction motors. In *Power Engineering Conference, 2007. AUPEC 2007. Australasian Universities* (pp. 1-6). IEEE.
- [6] Qiu, L., Padmanabhan, V. N., & Voelker, G. M. (2001). On the placement of web server replicas. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* (Vol. 3, pp. 1587-1596). IEEE.
- [7] Gray, N. A. (2004, April). Comparison of Web Services, Java-RMI, and CORBA service implementations. In *Proceedings of the 5th Australasian Workshop on Software and System Architectures at ASWEC* (pp. 52-63).
- [8] Ciancarini, P., Tolksdorf, R., & Zambonelli, F. (2002). A survey of coordination middleware for XML-centric applications. *The knowledge engineering review*, 17(04), 389-405.

Authors' information

¹Research Scholar, subbutvm@yahoo.com

²Associate Professor, ramesh@annauniv.edu



Subramaniyan R pursued his B.E. Degree in Electronics and Communication Engineering at University of Madras, Chennai, and completed his M.E Degree in Electrical Drives and Embedded Control at College of Engineering, Guindy, Anna University, Chennai, India. He is currently a PhD research scholar at CEG, Anna university, Chennai, India. His areas of interest includes Signal Processing and Embedded Control. Mr. Subramaniyan is a member of IEEE & IETE professional chapters.



Dr Ramesh R pursued his B.E. Degree in Electrical and Electronics Engineering at University of Madras, Chennai, and completed his M.E degree in power systems Engineering at Annamalai University Chidambaram. He received Ph.D degree at Anna University Chennai. Dr. Ramesh is currently working as an Associate Professor in the Department of Electrical and Electronics Engineering, College of Engineering Guindy, Anna University, Chennai, India. His areas of interest are Real-Time Distributed Embedded Control, On-line Power System Analysis and solar power system. He has published more than 20 research papers in reputed international journals.