UDC 621.313.12:519.68
Original scientific paper
Received: 08.09.2005.

A distributed model for capacitance requirements for self excited induction generators

Kannan Nithiyananthan(1) and **Velimuthu Ramachandran**(2)
(1)Birla Institute of Technology and Science - Pilani, Dubai Campus, Knowledge Village, Dubai, P.O. 500022, UNITED ARAB EMIRITATES
e-mail: nithi@bitsdubai.com
(2)Department of Computer Science and Engineering, College of Engineering, Guindy, Anna University, Chennai - 600 025, INDIA
e-mail: rama@annauniv.edu

SUMMARY

The main objective of this paper is to construct a distributed environment through which the capacitance requirements of self excited induction generators can be monitored and controlled. A single-server/multi-client architecture has been proposed which enables the self excited induction generators to access the remote server at any time, being able with their respective data to get the minimum capacitance requirements. An RMI (Remote Method Invocation) based distributed model has been developed in such a way that for every specific period of time, the remote server obtains the system data simultaneously from the neighbouring self excited induction generators with which the clients are registered and the server sends back the capacitance requirements as a response to the respective clients. The server creates a new thread of control for every client request and hence a complete distributed environment has been exploited.

***Key words**: distributed processing, self excited induction generator, RMI, client-server model.*

1. INTRODUCTION

The key aspect of the new development in on-line control of the self excited induction generator is the enhancement of the security maintaining a high reliability of electric power supply. The capacitance required to maintain excitation at all times irrespective of the value of speed of the machine as well as the load and power factor variations is known as the Cmin (minimum capacitance) [1, 2]. A minimum capacitance requirement of the self excited induction generator requires the proper integration of both automatic and manual control functions. Self excited induction generators are not connected to the grid. The excitation required is provided by an external capacitor. In the self excited machine the frequency of the output varies with the speed of the prime mover. So if it is connected to wind mill as the speed of the wind varies the frequency of the generated power varies too. As a result of this variation the capacitance required for exciting the machine may become insufficient which will result in loss of excitation. In order to avoid such a fault the capacitance connected to machine must be always monitored in order to vary it if necessary. Both the steady state [3, 4] and dynamic state conditions of a self excited induction generator have to be characterized by keeping the system operating optimally based on reactive power. As the number of self excited induction generators increases, minimum capacitance requirements by conventional client-server architecture is complicated, memory management is difficult, source code is bulk and exception-handling mechanism is not so easy.

This paper outlines a new approach to develop a solution for capacitance requirement based on on-line steady state analysis by distributed processing. RMI based client-server architecture overcomes the difficulties associated with sequential computation and it is easy to implement.

Since RMI uses built in Java security mechanism, the distributed capacitance requirements analysis through RMI is safe and secure to the server as well as to the data transfer from the induction generator.

2. RMI BASED ARCHITECTURE FOR AUTOMATED Cmin REQUIREMENTS

In this present work, a distributed environment has been set up using RMI to estimate and to monitor capacitance requirements evaluation for different induction generators of an integrated group. Each self excited induction generator has been considered as a client so that multi - induction generator clients - single Cmin server model is implemented. The self excited induction generator clients are interconnected with the Cmin server as shown in Figure 1.

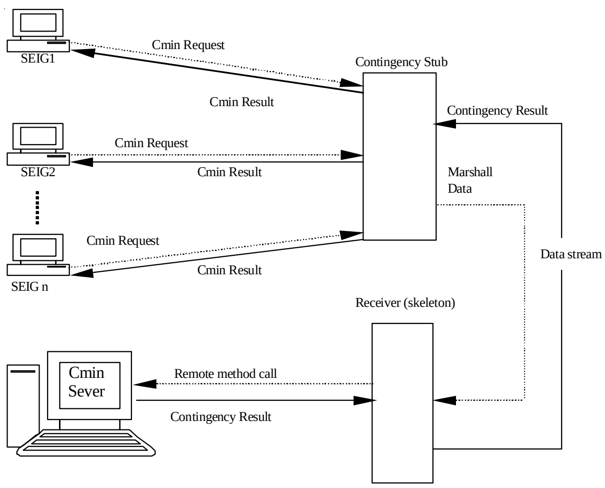

Fig. 1 RMI based client-server architecture for Cmin calculation

The induction generator client basically does the capacitance requirement analysis for every specific period of time by exchanging frequently the machine data with the server. The server does the Cmin calculation and distributes the results. Chronologically the server process should be started first, so that it can take initiative to setup a connection link. It then starts waiting and receiving a connection request from the client. A client can register itself with the remote object (server object), just by invoking the registration procedure on the server object, when it needs a service from it. The remote object obtains the necessary data from the registered client objects and responds back to them respectively with the results. This total process can be automated through the server to get the input data for every specific period of time. Transaction of data among clients and server takes place several times and so the possibilities of errors may be high and hence it must be handled properly.

2.1 RMI Data flow Model

In the proposed model, each neighboring self excited induction generator is considered as a remote client object. The induction generator client calls a method on an object that represents the remote object, which is called a stub. The stub contains a method for each of the methods in the remote object. The stub always resides on the client side and it packages the induction generator data into a block of bytes that can be communicated through the network. This process of marshalling [5] makes the entire machine data into a suitable format for transporting one virtual machine to another. The Cmin stub on the client side builds the information block that consists of an identifier of the remote object to be used, a description of the method to be called and then marshalled the system data. When the Cmin stub sends the information to the Cmin server, a receiver object (skeleton) on the server side receives and unmarshals the parameters. This receiver locates the object to be called and then calls the desired method with those parameters. When the method returns, the receiver object captures the return value and marshals the capacitance requirement result as packets on to a marshal stream and thus sends the evaluation results to the stub. The stub unmarshals the stream and returns it to the original caller. The associated RMI data flow model is shown in Figure 2.

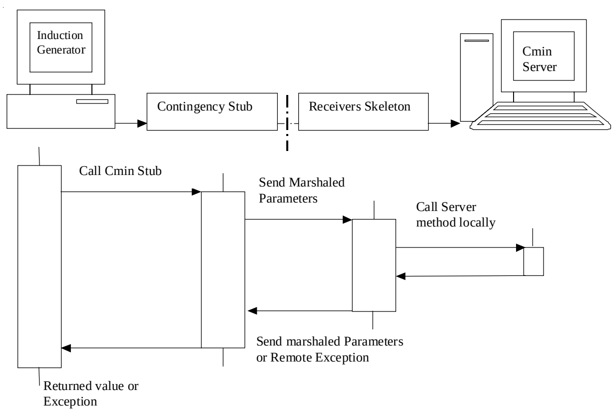

Fig. 2 Invoking a Cmin method on a remote object

2.2 Cmin server's self registry service and dynamic class loading [6]

RMI provides bootstrap registry service to locate remote server objects. Server program registers remote objects with the bootstrap registry service and the induction generators as a client's retrieve stubs to those objects. In this proposed method, the Cmin server creates its own registry and it maintains the stubs for the remote objects by itself, therefore the server no longer needs to depend on the bootstrap registry service provided by RMI protocol.

In RMI client-server architecture, clients can communicate with the remote object only when the Cmin server side stub is available with the client. The stub can be loaded on the client side dynamically by an external web server as shown in Figure 3.

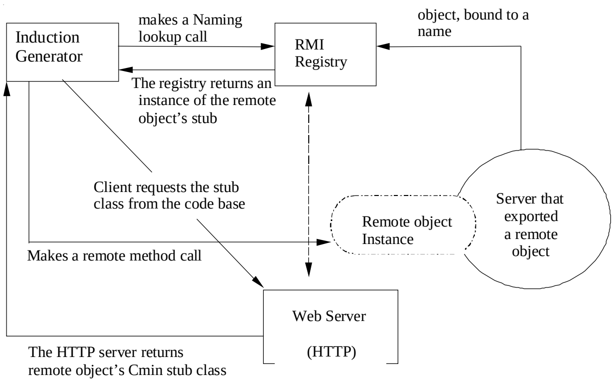

Fig. 3 Dynamic class loading

The steps involved in dynamic downloading RMI stubs are as follows:

i. The remote object's codebase is specified by the remote object's server by setting the java.rmi.server.codebase property. The RMI Cmin server registers a remote induction generator client object, bound to a name, with the RMI registry.
ii. The induction generator client requests a reference to a named remote object. The reference to the remote object's stub instance is what the client will use to make remote method calls to the induction generator server object.
iii. The RMI registry returns the stub instance reference to the requested class.
iv. The codebase which the induction generator client uses, is the URL that is annotated to the stub instance when the stub class was loaded by the registry.
v. The class definition for the stub is downloaded to the client dynamically.
vi. Now the induction generator client has all the information that it needs to invoke remote method on the Cmin server object. The stub instance acts as a proxy to the remote object that exists on the server.

Any changes in the implementation of the server side results in the modification of the stub and it will be made available for all clients by dynamic class loading through an external server.

3. RMI BASED AUTOMATIC Cmin ALGORITHM

When a remote client object registers with remote Cmin server object, the server uses the remote client reference to invoke its method to obtain the system data from that client and then provides the service through its methods. Both client and server objects are considered as remote objects and hence inter-remote object communication is achieved. The server object uses a single thread of control to distribute the minimum capacitance requirements simultaneously to the clients registered with it. The proposed model is dynamic which allows a new induction generator client that can register with the Cmin server object at run-time and gets serviced. Cmin server and its clients have to store in them the necessary object codes required for the capacitance requirement calculation. Stubs for both client and server must be kept at a common location like web server for distribution. Subsequently, the following steps are to be carried out:

i. Start Cmin server,
ii. The server should invoke its own registry service,
iii. Register an induction generator client and load a server's stub dynamically from the common location,
iv. Cmin server uses the client's reference to receive the induction generator data from the client periodically,
v. The server computes the capacitance required based on the eigen value method and returns it to the client,
vi. The client obtains the result through an Induction generator stub,
vii. For every specific period of time, the server automatically receives system data from the client, thereby providing automatic capacitance requirement evaluation.

4. RESULTS

The above distributed algorithm has been implemented in Windows NT based HP workstations connected in an Ethernet LAN. Figure 4 shows the minimum capacitance requirements for a specific induction generator client at various load impedances.

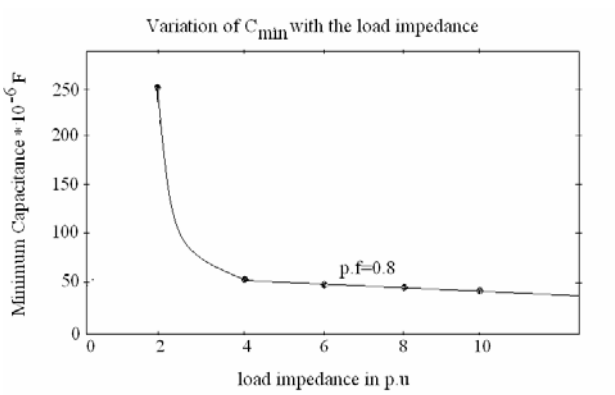

Fig. 4 Variation of Cmin with the load impedence

When each induction generator client is loaded, it registers with the Cmin server, the server stub will be downloaded dynamically through which the client sends the request and receives the output. Using this approach, different induction generator clients can monitor continuous updated capacitance requirements at regular time intervals.

5. CONCLUSION

An effective RMI based distributed model has been developed to do the on-line steady state analysis of self excited induction generators. It has been tried out to overcome the overheads associated with sequential evaluation through this model. Although on-line control architecture for capacitance requirements ranking is well established, this paper emphasizes a unique methodology based on remote method invocation to serve a large number of clients in a distributed induction generators environment, across various platforms based on communication between virtual machines. A practical implementation of this approach suggested in this paper was assessed based on sample systems. Accordingly the proposed model can be implemented for large induction generators network spread over geographically.

6. REFERENCES

[1] N.H. Malik and A.H. Al-Bahrani, Influence of the terminal capacitor on the performance characteristics of a self excited induction generator, *Generation, Transmission and Distribution, IEE Proceedings C*, Vol. 137, No. 2, pp. 168-173, 1990.

[2] L. Shirdhar, B. Singh, C.S. Jha, B.P. Singh and S.S. Muruthy, Selection of capacitors for the self regulated short shunt self excited induction generator, *IEEE Transactions on Energy Conversion*, Vol. 10, No. 1, pp. 10-17, 1995.

[3] T.F. Chan, Capacitance requirements of self–excited induction generator, *IEEE Transactions on Energy Conversion*, Vol. 8, No. 2, pp. 304-311, 1993.

[4] T.F. Chan, Steady state analysis of self–excited induction generators, *IEEE Transactions on Energy Conversion*, Vol. 9, No. 2, pp. 288-296, 1994.

[5] C.S. Horstmann and G. Cornell, *Core Java, Volume II: Advanced Features*, The Sun Microsystem Press Java Series, Prentice Hall, 1999.

[6] Dynamic code downloading using Remote Method Invocation, <http://java.sun.com/j2se/1.3/docs/gulde/rmi/codebase.html>

RASPODIJELJENI MODEL ZA ZAHTJEVE KAPACITETA NA ASINKRONIM SAMOUZBUDNIM GENERATORIMA

SAŽETAK

Cilj ovog rada je stvaranje raspodijeljene okoline kroz koju se mogu promatrati i kontrolirati zahtjevi kapaciteta asinkronih samouzbudnih generatora. Predlaže se arhitektura jednog servera u odnosu na više klijenata koja omogućava da asinkroni samouzbudni generatori pristupe udaljenom serveru u bilo koje vrijeme te da zadovolje minimalne zahtjeve kapaciteta pomoću svojih odgovarajućih podataka. Raspodijeljeni model zasnovan na RMI (Remote Method Invocation) metodi razvio se na takav način da za svaki specifični vremenski period udaljeni server simultano dobiva podatke sistema od susjednih asinkronih samouzbudnih generatora s kojima se klijenti registriraju, a server vraća zahtjeve kapaciteta kao reakciju određenim klijentima. Server stvara novi tok kontrole za svaki zahtjev klijenta te se tako iskorištava cijela raspodijeljena okolina.

Ključne riječi: *raspodijeljeno procesiranje, asinkroni samouzbudni generator, RMI, korisnik-server model.*