Journal of Computer Science 3 (5): 323-328, 2007
ISSN 1549-3636
© 2007 Science Publications

Performance Monitoring of Energy Flow in the Power Transmission and Distribution System Using Grid Computing

[1]C.Senthamarai, [2]A.Krishnan and [3]R. Hemalatha
[1]Department of MCA, K.S.Rangasamy College Of Tech.,
Tiruchengode-637 209, India.
[2]K.S.Rangasamy College of Engineering, Tiruchengode, India
[3]Key Skill/Key Soft Solutions, Bangalore, India

Abstract: Grid Computing is a promising infrastructure and technology that involves the integrated and collaborative use of Computers, networks, databases and scientific instruments owned and managed by multiple organizations. In this context our research primarily aims at developing an application framework for monitoring the Performance of Power Transmission and Distribution System using Grid Computing. Power Transmission and Distribution System of Electrical Network supplies power to millions of consumers. The demand for electrical energy is ever increasing. Today over 21% of the total electrical energy generated in India is lost in transmission and distribution. It is possible to bring down the distribution losses to the minimum level in India with the help of newer technological options in information technology called Grid Computing in the electrical power distribution sector which will enable better monitoring and control. By monitoring and controlling the parameters such as voltage, load, frequency and power factor in electrical power transmission and distribution systems the line loss will be reduced. When the Power factor is monitored using this Grid computing method, the Substations can maintain their norms by giving proper instructions to the operators dynamically to add the capacitor banks to maintain the power factor so that the line loss will be reduced and the revenue will automatically be increased.

Key Words: Grid computing, Transmission and Distribution system, monitoring and control, Line loss

INTRODUCTION

The Growth of the Internet, along with the availability of powerful computers and high-speed Networks as low-cost commodity components, is changing the way scientists and engineers do computing, and are also changing how society in general manages information. Grid Computing has been identified as an important new technology by a remarkable breadth of scientific and engineering fields as well as many commercial and industrial enterprises. Grid Computing is a form of distributed computing that involves coordination and sharing of computing application, data storage or network resources across dynamic and geographically dispersed organizations. In short, it involves virtualizing computing resources. [2]
The Grid is defined as

1. The flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions and resources [4,5,6]
2. A type of Parallel and distributed system that enables the sharing, selection and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost and users quality of service requirements[1]

While the notion of grid computing is simple enough, the practical realization of grids poses a number of challenges. The key issues that need to be dealt with are security, heterogeneity, reliability, application composition, scheduling, and resource management [1]. The Microsoft .NET Framework provides a powerful toolset for all of these, in particular support of remote execution, multithreading, security, asynchronous programming, managed execution and cross-language development, making it an ideal platform for grid computing middleware. Alchemi [3] is a windows-based desktop grid computing framework implemented on Microsoft .NET platform and developed at the University of Melbourne. Alchemi is implemented on

Corresponding Author: C.Senthamarai, Assistant Professor/Department of MCA, K.S.Rangasamy College of technology, Tiruchengode-637 209, Namakkal Dt. TamilNadu India

top of the Microsoft .NET framework and provides the runtime machinery of constructing and managing desktop grids. It also provides and object oriented programming model along with web service interfaces that enable its services to be accessed from any programming environment that supports SOAP-XML abstraction.

While transmitting and distributing power to meet the consumers demand, Power losses occur in the EHV, HV and LV lines and also in the transformers. This technical loss in the power system is an inherent characteristic and it cannot be totally eliminated but can be reduced to an optimum level which will increase the revenue when monitoring and controlling of lines are done as and then by using the emerging technology Grid Computing.

Our work focuses on developing an application for monitoring the Power Transmission and Distribution system using the Alchemi Framework.

Architecture of Alchemi Framework: Alchemi follows the master-worker parallel programming paradigm[3] in which a central component dispatches independent units of parallel execution to workers and manages them. This smallest unit of parallel execution is a grid thread, which is conceptually and programmatically similar to a thread object (in the object-oriented sense) that wraps a "normal" multitasking operating system thread. A grid application is defined simply as an application that is to be executed on a grid and that consists of a number of grid threads. Grid applications and grid threads are exposed to the grid application developer via the objectoriented Alchemi .NET API .

Alchemi offers four distributed components as in Fig.1 designed to operate under three usage patterns.

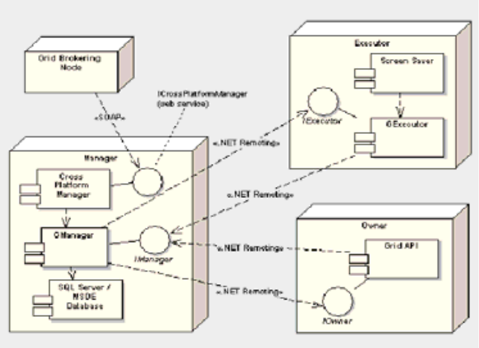

Fig.1: Alchemi Architecture

Manager: The Manager manages the execution of grid applications and provides services associated with managing thread execution. The Executors register themselves with the Manager which in turn keeps track of their availability. Threads received from the Owner are placed in a pool and scheduled to be executed on the various available Executors. A priority for each thread can be explicitly specified when it is created within the Owner, but is assigned the highest priority by default if none is specified. Threads are scheduled on a Priority and First Come First Served (FCFS) basis, in that order. The Executors return completed threads to the Manager which are subsequently passed on or collected by the respective Owner.

Executor: The Executor accepts threads from the Manager and executes them. An Executor can be configured to be dedicated, meaning the resource is centrally managed by the Manager, or non-dedicated, meaning that the resource is managed on a volunteer basis via a screen saver or by the user. Thus, Alchemi's execution model provides the dual benefit of:

1. flexible resource management and
2. flexible deployment under network constraints

Owner: Grid applications created using the Alchemi API are executed on the Owner component. The Owner provides an interface with respect to grid applications between the application developer and the grid. Hence it "owns" the application and provides services associated with the ownership of an application and its constituent threads. The Owner submits threads to the Manager and collects completed threads on behalf of the application developer via the Alchemi API.

Cross-Platform Manager: The Cross-Platform Manager, an optional sub-component of the Manager, is a generic web services interface that exposes a portion of the functionality of the Manager in order to enable the execution of platform independent grid jobs. Jobs submitted to the Cross-Platform Manager are translated into a form that is accepted by the Manager (i.e. grid threads), which are then scheduled and executed. Thus Alchemi is used to create different grid configurations like Desktop cluster grid, multi-cluster grid, and cross-platform grid (global grid).

Cluster (Desktop Grid): The basic deployment scenario – a cluster consists of a single Manager and multiple Executors that are configured to connect to the Manager as shown in Fig. 2. One or more Owners can execute their applications on the cluster by connecting to the Manager. Such an environment is appropriate for deployment on Local Area Networks as well as the Internet.

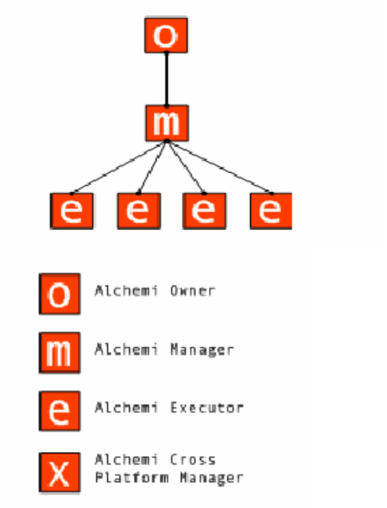

Fig. 2: Cluster(Desktop grid)

Multi-Cluster: A multi-cluster environment is created by connecting Managers in a hierarchical fashion as shown in Fig. 3. As in a single-cluster environment, any number of Executors and Owners can connect to a Manager at any level in the hierarchy. An Executor and Owner in a multi-cluster environment connect to a Manager in the same fashion as in a cluster and correspondingly their operation is no different from that in a cluster. The key to accomplishing multi-clustering in Alchemi's architecture is the fact that a Manager behaves like an Executor towards another Manager since the Manager implements the interface of the Executor. A Manager at each level except for the topmost level in the hierarchy is configured to connect to a higher-level Manager as an "intermediate" Manager and is treated by the higher level-Manager as an Executor. Such an environment is more appropriate for deployment on the Internet.

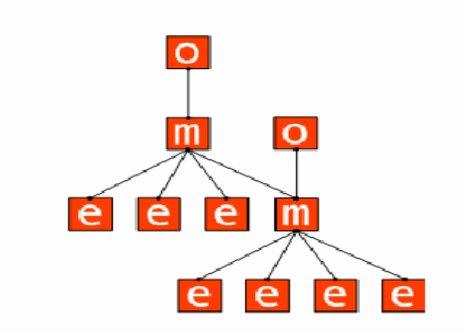

Fig. 3: Multi-Cluster grid

Cross-Platform Grid: The Cross-Platform Manager can be used to construct a grid conforming to the classical global grid model (Fig 4). A grid middleware component such as a broker can use the Cross-Platform Manager web service to execute cross-platform applications (jobs within tasks) on an Alchemi node (cluster or multi-cluster) as well as resources grid-enabled using other technologies such as Globus.

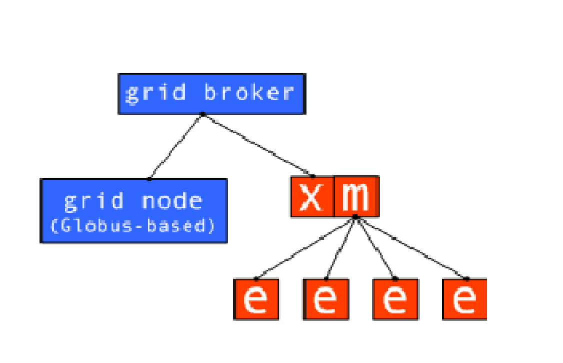

Fig. 4: Cross-Platform Grid

Grid Thread Programming Model: Alchemi simplifies the development of grid applications by providing a programming model that is object oriented. To develop and execute a grid application the custom grid class is created which is derived from the abstract GThread class. An instance of the GApplication object is created and any dependencies required by the application are added to its Dependencycollection. Instances of the Gthread derived class are then added to the GApplication's ThreadCollection.

The Lifecycle of Grid application is shown in Fig.5.

Model of Transmission and Distribution System: Electric power is normally generated at 11-25KV in a power station[7]. To transmit over long distances, it is then stepped-up to 400KV, 220KVas necessary. Power is carried through a transmission network of high voltage lines. Usually, these lines run into hundreds of kilometers and deliver the power into a common power pool called the power grid. The power grid is connected to load centers (cities) through a sub-transmission network of normally 33KV (or sometimes 66kV) lines. These lines terminate into a 33KV (or 66KV) substation, where the voltage is stepped-down to 11KV for power distribution to load points through a distribution network of lines at 11KV and lower. The power network, which generally concerns the common man, is the distribution network of 11KV lines or feeders downstream of the 33KV substation. Each 11KV feeder which emanates from the 33KV branches further into several subsidiary11KV feeders to carry power close to the load points (localities, industrial areas, villages, etc.,).

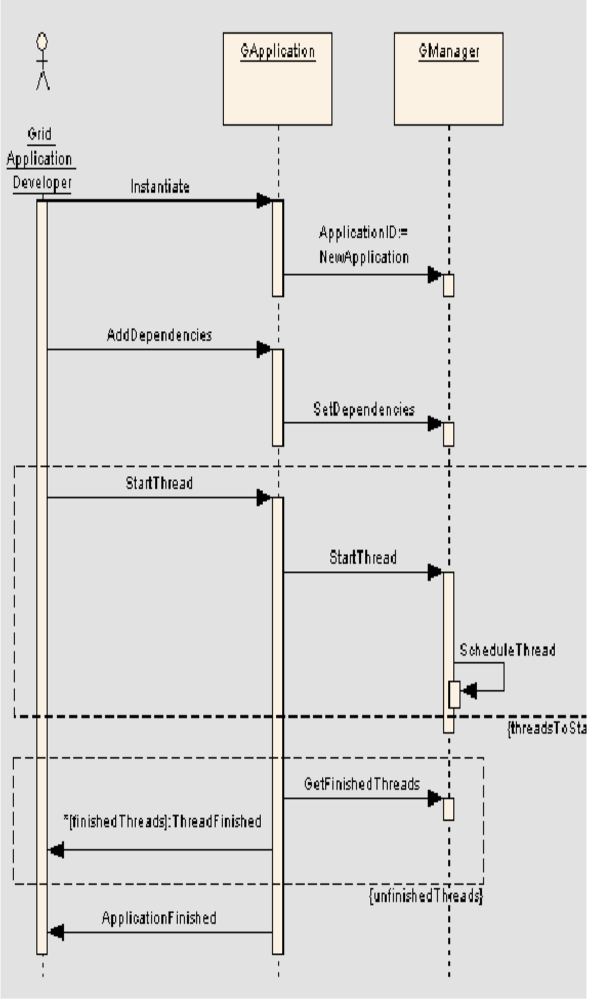

Fig. 5: Interaction between owner and Manager nodes

At these load points, a transformer further reduces the voltage from 11KV to 415V to provide the last-mile connection through 415V feeders (also called as Low Tension (LT) feeders) to individual customers, either at 240V (as single-phase supply) or at 415V (as three-phase supply).

Lack of information at the base station (33KV sub-station) on the loading and health status of the 11KV/415V transformer and associated feeders is one primary cause of inefficient power distribution. Due to absence of monitoring, overloading occurs, which results in low voltage at the customer end and increases the risk of frequent breakdowns of transformers and feeders.

The Model of Transmission and distribution System is shown in Fig. 6.

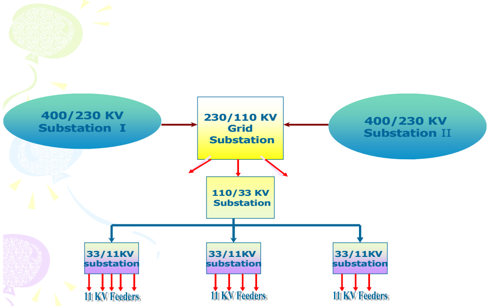

Fig. 6: Model of Transmission and Distribution System

RESULTS AND DISCUSSION

Installing our grid application framework on the substations as in Fig. 7

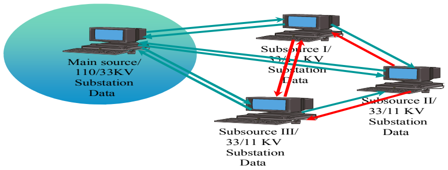
Main source/ 110/33KV Substation Data
Subsource I/ 33/11 KV Substation Data
Subsource II/ 33/11 KV Substation Data
Subsource III/ 33/11 KV Substation Data

Fig. 7: Proposed Model

The data present in any substation can be accessed using our application which will give the following benefits [8,9]

Real Time monitoring and control of substations
Load Management
Minimum interruption
Improved power quality
Load forecasting
Voltage profile improvement
Fault localization
Power factor control
Load survey data and Network Analysis of HT systems
Reduction in line loss
Revenue improvement

The method used in TNEB (Tamilnadu Electricity Board)is implemented using Alchemi framework to calculate the Line Loss

Line Loss calculation

- Connected Load in KVA = CL
- Maximum Load in the feeder in Amps = P
- Energy sent out in the feeder in units = E
- Feeder Voltage in Kilovolts = V
- Power Factor = PF
- Study Period in Hours = H
- Diversity Factor = DF
- DF = CL/1.732 * V * P
- Load Factor = LF
- LF = E/1.732 * V * P * PF *H
- Load Loss factor = LLF
- LLF = $0.8 * LF^2 + 0.2$ LF

- Name of the Feeder = x
- Cumulative capacity = KVA
- Distance in KM = L
- Resistance of conductor per Km = R
- Feeder Voltage in KV = V
- A = $(KVA/ V * DF)^2 * L * R/ 1000$
- Energy loss in a feeder during the study period Hrs = L1
- L1 = A * LLF * HRS
- Energy Loss in DTS during the study period Hrs = L2
- Total Energy loss in Kilowatt Hrs

  B = L1 + L2

Input to the Feeder in KW

C = (1.732 * V * P * PF) + A

- Energy sent out
- D = C * LF * H
- Percentage of HT Line Loss = L
- L = B * 100 **/ D**

The data were collected from the feeder X . The Sample Data and results are given for the proposed model

By varying the Load and keeping PF=0.98 and voltage=11.0KV. The % of HT Line loss is calculated and the results are given in the Table 1.

Table 1: Load Vs, HT Line Loss

Load In Amps.	HT Line Loss in %
180	15.35
200	15.42
220	15.49
240	15.55
260	15.61
280	15.67
300	15.73
320	15.78
340	15.82
360	15.89
380	15.97
400	16.00

Fig. 8 shows the % of HT Line Loss for the Table 1. It can be understood that when the Load increases the Line Loss also increases.

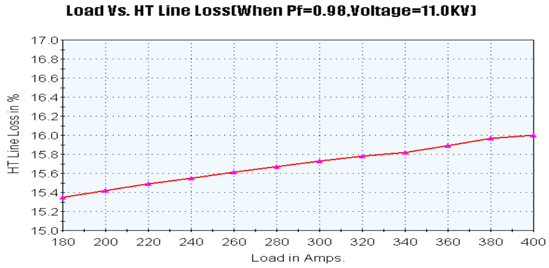

Fig. 8: Load Vs. HT Line Loss

By varying the Voltage and keeping PF=0.98 and Load=200Amps the % of HT Line loss is calculated and the results are given in the Table 2.

Table 2: Voltage Vs. HT Line Loss

Voltage in KV	HT Line Loss in %
10.0	17.29
10.2	16.87
10.4	16.48
10.6	16.11
10.8	15.76
11.0	15.44
11.2	15.10
11.4	14.8
11.6	14.50
11.8	14.23
12.0	13.80

The Fig. 9 Shows % of HT Line Loss for the above Table 2. It can be understood that when the voltage is increased the Line Loss can be reduced.

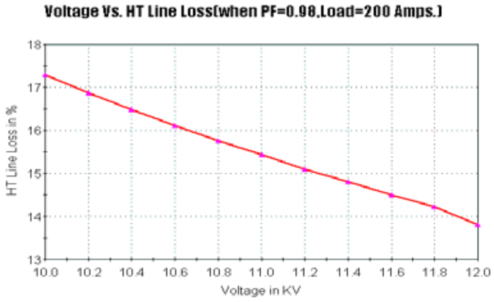

Fig. 9: Voltage Vs. HT Line Loss

iii. By varying the Power Factor and keeping the voltage=11 KV and Load = 200Amps the % of HT Line loss is calculated and the results are given in the Table 3.

Table 3: Power Factor Vs. HT Line Loss

Power Factor	HT Line Loss in %
0.70	23.57
0.72	22.70
0.74	21.89
0.76	21.14
0.78	20.44
0.80	19.78
0.82	19.17
0.84	18.60
0.86	18.06
0.88	17.55
0.90	17.07
0.92	16.62
0.94	16.20
0.96	15.80
0.98	15.42
0.99	15.24

The Fig. 10, shows the % of HT Line Loss for the Table 3. It can be understood that when the Power Factor is increased the Line Loss is decreased.

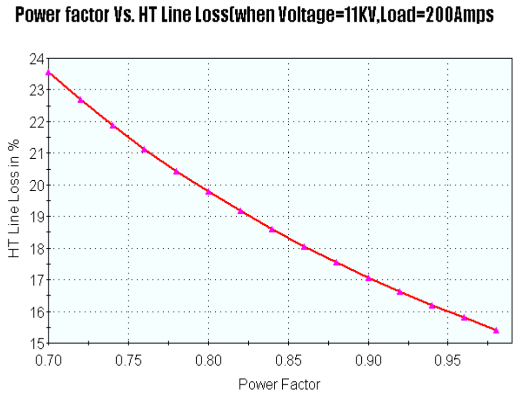

Fig. 10: Power factor Vs. HT line loss

Existing HT Line loss in TNEB is 18 %. At present HT line loss is calculated for every Quarter of the Year. By using our Grid Computing framework it is easy to monitor and control the Energy Flow and line loss in the Electrical Power grid dynamically. Also decisions can be made faster than the existing method. This will improve the Electrical System reliability, availability and Maintainability.

Suppose if the line loss is reduced to 1% from 18 % using Grid computing method revenue will be increased to an appreciable level.

CONCLUSION

By monitoring and controlling the parameters such as voltage, load, frequency, power factor, KVA, KW, KWH, KVAR in electrical power transmission and distribution systems using our grid computing application the line loss is reduced. Our experimental results prove that when the power factor and voltage are increased the Line loss can be reduced. Thus by reducing the line loss the performance of the system is improved and the revenue will automatically get increased.

REFERENCES

1. Buyya R, Abramson D, and Giddy J, 2000, Economy Driven Resource Management Architecture for Computational Power Grids, International Conference on Parallel and Distributed Processing Techniques and Applications, (PDPTA'2000), Las Vegas, USA June 26-29,2000.
2. F. Berman, G. Fox, and T. Hey, 2003, Grid Computing: Making The Global Infrastructure a Reality, John Wiley & Sons, New York, pp: 30-41
3. Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, Srikumar Venugopal, 2006, Peer-to-Peer Grid Computing and a .NET-Based Alchemi Framework Wiley series on Parallel and Distributed computing Published online pp: 403-429
4. I. Foster and C. Kesselman:2003 The GRID2: Blueprint for a new computing Infrastructure Morgan Kaufmann, 2nd Edition, pp: 38-41
5. Foster, C. Kesselman, and S. Tuecke, 2001 "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of High Performance Computing Applications, 15 (3). pp:200-222
6. K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, 2001"Grid Information Services for Distributed Resource Sharing," Proceedings of the Tenth IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), IEEE Press, pp: 181-194
7. Prem k. Kalra :Transmission and distribution Loss minimization IIT, Kanpur. http://www.electricityindia.org/dca_stanford/presentations/Prem_Kalra.pdf.
8. www.tneb.in
9. Tamil Nadu Electricity Board Statistics at a Glance 2005-2006 http://www.tn.gov.in/department/energy.htm