IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

Reducing Consumption of Power in Peer to Peer System with Load Balancing of Processes

Mr. Sumit V. Dusane[1], Mr. Pranay Chauhan[2]

Research Scholar, CSE Dept., Swami Vivekanand College of Engineering, Indore, India [1]

Assistant Professor, IT Dept., Swami Vivekanand College of Engineering, Indore, India [2]

Abstract: As we know today in the world information in which sending and receiving the information is popular and which is based on peer to peer system. In this type every computer in the network acts as the peer i.e. node and there is no centralized server in the peer to peer system. As we know the required data is distributed among the different sites and servers so many clients are trying to access that data concurrently. As simultaneous requests arises at the server its performance deviates due to heavy load. Information Systems are composed of various types of computers interconnected in network. In information system, traditional client-server model move towards P2P model. P2P systems are fully distributed i.e. absence of centralized coordinator. P2P system is another type of information system which is composed of huge number of peer computers. There is need to reduce power consumption of server. Here, it is getting more significant to discuss how to reduce power consumption of server in P2P systems. In this paper we presented how proposed algorithm reduces the power consumption of server in P2P system and how it is more efficient than Round robin algorithm and Consumption Laxity Based algorithm. Proposed Algorithm overcomes all drawbacks of Round robin and Consumption Laxity Based algorithm.

Keywords: Round Robin, Consumption Laxity Based.

I. INTRODUCTION

A. Peer to Peer System

Peer-to-peer (P2P) system is composed of peer computers which are interconnected in overlay networks. Here, each peer computer can play both roles of server and client and the P2P system is fully distributed, i.e. no centralized coordinator. In information systems, total electric power consumption has to be reduced. Various types of hardware technologies like low-power CPUs are now being developed [1].

P2P is a distributed application architecture that partitions tasks or workloads among peers. Peers are equally privileged participants in the application. Each computer in the network is referred to as a node. The owner of each computer on a P2P network would set aside a portion of its resources - such as processing power, disk storage, or network bandwidth - to be made directly available to other network participant, without the need for central coordination by servers or stable hosts. With this model, peers are both suppliers and consumers of resources, in contrast to the traditional client–server model where only the server supply (send), and clients consume (receive) [1] [2]. A server peer is a peer which can provide other peers with some service. A clientpeer issues a request to a server peer and server peer give response to the client request. Each client peer has to find a server peer which not only satisfies service requirement but also spends less amount of electric power [2]

There are two types of applications, transaction-based and transmission-based applications. In the transaction-based applications, a client peer issues a request to a server peer and the server peer mainly consumes CPU resources to process the request, e.g. encode multimedia data in Web pages.

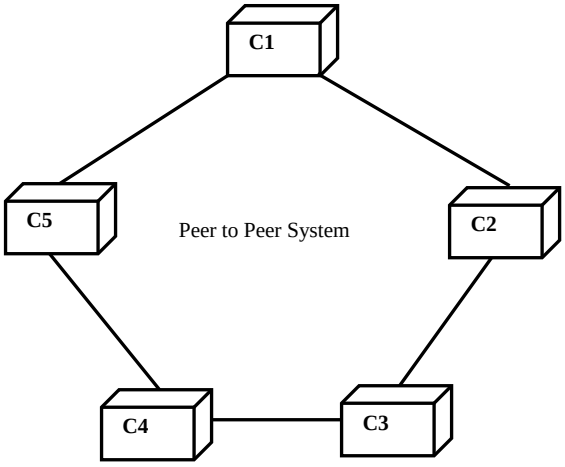

Figure 1.A Peer-to-Peer System of Nodes without Central Infrastructure

Web applications are the typical examples and in transmission based, a server peer transmits a large volume of data to a client peer like file transfer protocol (FTP) applications [3].

B. Load Balancing

In the computer system load is a measure of the amount of computational work that a computer system performs. The load average represents the average system load over a period of time. If the average load on that peer is beyond the average value then that situation creates number many problems like slow processing (Computation) of processes and it will directly affect on power consumption of that peer. So reducing the power of that peer node we have

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

distributed the load on other peer node and we can achieve this using load balancing.

"Load balancing is method to distribute workload across multiple computers or a computer cluster, network links, central processing units, disk drives, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, and avoid overload".

A client peer issues a request to a server peer to obtain the service. On receipt of a request, a server peer performs the request and sends a reply to the client. Suppose a client peer would like to obtain a service like a data from server and there are a set C of multiple server peers C1…Cn which can provide the service. A client peer issues a request to a server peer Ci in the server set. Here, the server peer spends CPU resources to perform the request. For example, the CPU resources are consumed to encode multimedia data of Web pages in Web applications. The request is performed as a process Ps in the server peer. Thus, there are a set of P application processes P1….Pn to be performed on server peers. We assume each process Ps can be performed on any server peer in the server set C[1][2].

A client peer first issues a request to perform a process to a load balancer K. In Figure 2 load balancer selects one server peer Ci in the server set C for the process and sends a request to the server peer. On receipt of the request, the process Ps is performed on the server peer and data is sent to the client peer. The load balancer is a logical process.

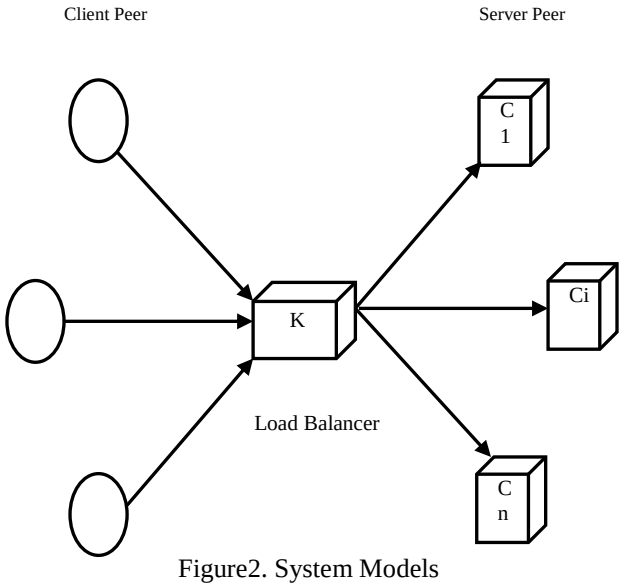

Figure2. System Models

The load balancer K having, priority queue and unsorted priority queue. The priority queue of load balancer maintains the server list or maintains the queue of server as per the machine or server capacity. On the other hand, unsorted priority queue maintains client request for the data, in queue.

In Load balancer we suggest 3 sub module for sending request to server peer.

1) Module contains information of Servers maximum no of request processing capacity and no of request currently running on Server peer. Is used to Track server capacity and No of currently running Requests. If server is working within its maximum capacity then the process immediately send to server. But if server is working with maximum capacity then all the process generated by client is transfer to module 2.
2) Module contain queue of request generated by client peer. Is used to contain all the process generated by client. This module contains the waiting request queue until server is not free.
3) Module contain queue of request processed by Server and send to client peer.

II. RELATED WORK

A. Round robin Algorithm

Round-robin (RR) is one of the simplest scheduling algorithms for processes in an operating system. The name of the algorithm comes from the round-robin principle known from other fields, where each person takes an equal share of something in turn.

In round robin algorithm server S1…..Sn are arranged in order. A client request is first issue to the server S1.If the server S1 is overloaded then request is sent to another server S2. If server S2 is also overloaded then request is sent to next server. Thus servers S1…..Sn in round robin methods are overloaded then request is issue to server Ci+1 where i<n. Normally in round robin algorithm servers are ordered as per the fix weight and having two types weighted least connection (WLC) and weighted round robin (WRR). This weight factor is depends on two factor like powerand performance of the server. When higher the performance of the server then we can assign more number of process to processor for computation and if server consumption is low then we can allocate more processes. Performance is calculated using estimated time of process execution on server and this information is stored in file for feature execution of same type of process[1][2].

A.1Weighted Round Robin (WRR) Algorithm

The weighted round-robin scheduling is designed to better handle servers with different processing capacities. Each server can be assigned a weight, an integer value that indicates the processing capacity. Servers with higher weights receive new connections first than those with less weight, and servers with higher weights get more connections than those with less weight and servers with equal weights get equal connections. In the implementation of the weighted round-robin scheduling, a scheduling sequence will be generated according to the server weights [3].

A.2 Weighted Least Connection Algorithm

The weighted least connection scheduling is a superset of the least-connection scheduling, in which you can a performance weight to each real server. The servers with a higher weight value will receive a larger percentage of connections at any one time. The Server Administrator can assign a weight to each server, and network connections are schedule to each server in which the percentage of the current number of live connections for each server is a ratio to its weight [4].

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

Conclusions of Round Robin Algorithm are as,
Suppose,

Each Server Process Capacity = 20
No of Server in network =4
Total no. of Process dump by user = 300

Then,

Load on server = 300/4 = 75 process

Means that server can process this total number of process but server can take more time to process these 75 processes and finally server consumes more power (Watts) than it ideal state, here ideal state is 20 processes.

B. Consumption Laxity based Algorithm
The computation laxity (CL) of a process Ps shows how long it takes to perform up the process Ps from time t on a server Ci. Suppose a process Ps is issued to the load balancer K at time t. The Consumption Laxity of a process on each server Ci is given as follows:

(Consumption Laxity) clis(t) =ETps (t) – t

In the CLB algorithm, a server Ci which can most early terminate a process Ps is selected for the process Ps.

Conclusion of Consumption Laxity Based Algorithm is as, Consumption laxity based algorithm calculates estimated termination time of processes. Once the estimation time is calculated for process use this information for the next same process and dump that process on same server whether that server is under load or overload. But drawback is that there is no situation when the server is under load, but server is overloading then it takes more time to process the request and consumes more power.

III. PROPOSED SYSTEM

Proposed algorithm contains three sub modules one is input for taking input data for process scheduling and Watt calculation model. Second is process scheduling using queue system model, calculating time for each process and assign queue. Finally Watt calculation module calculates total power consumption for each process as well as whole processes. Mainly this proposed algorithm schedule the load on server using queue system so it is work as load balancer.

A. Input Data
Input data for proposed system mainly divided into five sections-

I. Total number of nodes
II. Frequency of processor
III. Total Number of processes
IV. Maximum processes processing capacity of processor
V. Individual process arrival time and service time

B. Scheduling with Queue System
This is the second part of the proposed system, using that model we can schedule the server load using scheduling technique with queue system. Queue system is nothing but the array; in that we can store the processes which require the related server but at that point server is not free means server acquire its maximum capacity. So at that condition this queue system play important role.

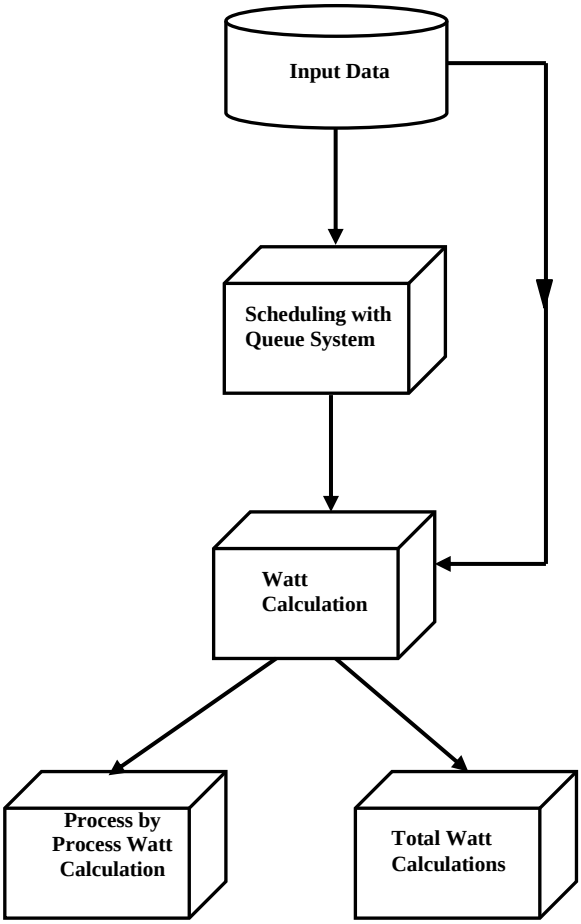

Figure3. Block Diagram for Proposed System

C. Watt Calculation:
For calculating the power consumption of server following formula can be used.
Energy Consumption = Frequency of Processor X Processing Time of Process.
Actual this processing time of process is nothing but the turnaround time for each individual process. After multiplication, we get energy consumption in joule for each individual process. So needto convert this energy consumption in joule into Watt. So,

1 Watt hour = 3600J
Joules = Watt x Seconds

Therefore,

Watts = Joules/ Second

e.g.Process X's turnaround time is 19 microsecond (Work done). Then,
Energy Consumption = 750 * 19 = 14250 J
So, Watt = 14250/3600 = 3.95 Watt.
Similarly, calculate power consumption in watt for all processes. And finally for total watt calculation do the summation of all power consumption (in watt) which are calculated for all processes.

IV. METHODOLOGY

Proposed System Algorithm with Explanation
1. Start
2. System gets Arrival time and Burst Time for coming processes.

DOI 10.17148/IJARCCE.2015.41218

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

3. System gets server capacity in MHz.
4. Check all jobs/Process is finished.
If Yes Then,
Go to End of algorithm.
If No Then,
Check timer < Max (Arrival) time
If Yes Then,
Go to step number 5
If No Then,
Then Go to step number 4
// This Step insured that whether the coming processes is already finished or new one. If the process is current working process control goes to the step no 5.
5. Check is the process coming for the first time
If No Then,
Assign Priority
Burst with different level with time quantum, and then go to step no 6.
// This step insured that the process is working process, then assign priority and execute it with different quantum time and move toward the step no 6
If Yes Then,
Check load on Server <= Max Capacity
If Yes Then, Burst with level 1 time quantum.
If No Then, Checks next server load < Max Capacity
If Yes Then, Burst with level 1 time Quantum
If No Then, Assign Waiting Queue.
Burst With different level with time quantum
Then go to step no 6.
// This step checks the server load means checking the maximum process handling capacity of the server for coming process if first server is overloaded then checks the next related server unfortunately that server is also overloaded then transfer the process to waiting queue and indirectly maintain the server load i.e. maintain the server ideal state.
6. Assign and Cumulative addition of time quantum to turn around time.
7. Calculating waiting Time.
8. Stop.

V. RESULT

We are executing three algorithms Round Robin, Consumption Laxity based algorithm and Proposed System in simulation environment. For that we consider some parameters like frequency of that node means we consider that node is treated as the sever node. Next of frequency we consider capacity of the node that capacity is nothing but shows maximum load can be handled by that node or server. After that next parameter is total number of node required for that simulation. After these parameters one important parameter is total number of processes is required for the simulation. Once this parameters is set simulator automatically generates or assign arrival time and service for these processes here arrival time is nothing but the actual time when the process will execute and service time represent tentative time required to complete the processes.
Figure 4 shows that using buttons we can create node, create processes. Using the set frequency button we can change the default frequency of the node. Simulation window also shows the arrival time and service time for the processes.

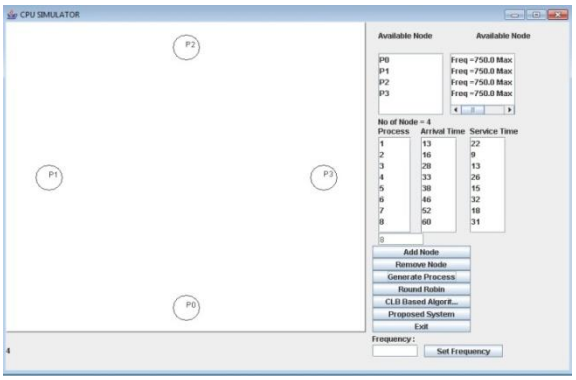

Figure4. Window of CPU Simulator

A. Round Robin Algorithm:
Figure 5 shows calculations of round robin algorithm.

Round Robin

Process	Arrival Time	Service Time	Finish Time	Turnaround Time	Tq/Ts	Watt...
Round Robin						
1	3	11	26	23.0	2.0	4.79...
3	19	11	49	30.0	2.0	6.25
4	25	5	50	25.0	5.0	5.20...
5	38	11	81	43.0	3.0	8.95...
10	67	5	98	31.0	6.0	6.45...
7	48	14	128	80.0	5.0	16.6...
2	3	29	137	134.0	4.0	27.9...
9	55	17	146	91.0	5.0	18.9...
6	45	21	147	102.0	4.0	21.25
8	49	41	168	119.0	2.0	24.7...

Figure5. Round Robin Algorithms Calculations

Turn Around time =finish time – arrival time = 26-3 = 23
Now 23 microsecond is required for process one. Then calculate the Watts required for process one.
Power consumption (Joule) = Freq of processor * Processing time of process= 750 * 23= 17250 J
Now, convert that power consumption in joule to Watts.
Power consumption in Watts = 17250 / 3600 = 4.79 Watts.
In this way we can calculate the power consumption of all other processes.

B. Consumption Laxity Based Algorithm:
Figure 6 shows the calculation for the consumption laxity based algorithm.

Computation Laxity Based (CLB) Algorithm

Process	Arrival Time	Service Time	Finish Time	Turnaround Time	Tq/Ts	Watt...
Computatio...						
1	3	11	14	11.0	1.0	2.29...
2	3	29	43	40.0	1.0	8.33...
4	25	5	48	23.0	4.0	4.79...
3	19	11	59	40.0	3.0	8.33...
5	38	11	70	32.0	2.0	6.66...
6	45	21	91	46.0	2.0	9.58...
10	67	5	96	29.0	5.0	6.04...
7	48	14	110	62.0	4.0	12.9...
9	55	17	127	72.0	4.0	15.0
8	49	41	168	119.0	2.0	24.7...

Figure6. CLB Algorithm Calculations

DOI 10.17148/IJARCCE.2015.41218

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

C. Proposed System Algorithm:
Figure 7 shows the calculation for the proposed system algorithm.

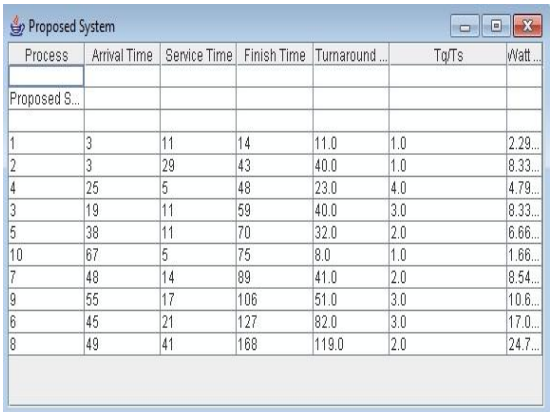

Figure7. Proposed System Algorithm Calculation

Result Analysis:
Figure 8 shows the total power consumption in Watts for three algorithms like Round Robin, Consumption Laxity Based algorithm and proposed system algorithm.

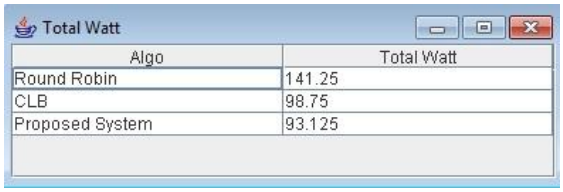

Figure8 Total Power Consumption in Watts for Three Algorithms

We can say that average power consume by each algorithm is calculate as,
Average Power Consumption = Power consume by total no. of process / Total number of processes.
Average power consumption by RR = 141.25/10 = 14.12 Watt
Average power consumption by CLB = 98.75/10 = 9.87 Watt
Average power consumption by PS = 93.12 / 10 = 9.31 Watt.
So we can say that proposed system consumes less power as compared with other algorithms. Figure 9 shows graph that shows the total power consumed by three algorithms.

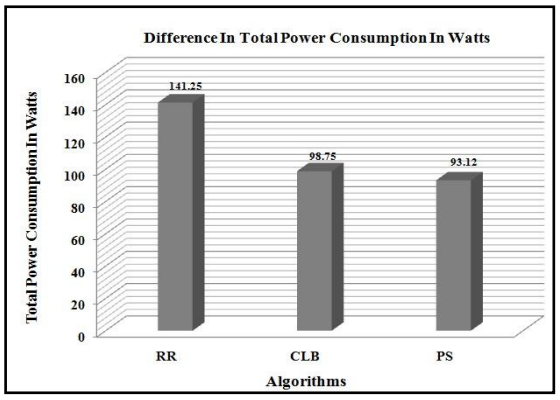

Figure 9 Total Power Consumed by Different Algorithms

S. NO	Parameters	Round Robin (RR)	Consumption Laxity Based (CLB)	Proposed System (PS)
1	Total number of nodes	4	4	4
2	Frequency in megahertz	750	750	750
3	Total number of processes	10	10	10
4	Total Power consumption in Watts	141.25	98.75	93.12
5	Difference in total power consumption in Watts with respect to proposed system	48.13	5.63	—

Table1. Comparative Study of Power Consumed By Algorithms

Table 1 contains comparison of different parameters with respect to three algorithms. Here first three parameters are constant during execution of all algorithms. But fourth parameter changes during execution of algorithms i.e. the value of parameters is decreases continuously. Also the fifth parameter showing the difference in total power consumption in Watts with respect to proposed system. So we can say that third algorithm is better than previous two algorithms.

VI. CONCLUSION AND FUTURE WORK

In this paper we discussed that by using the queue system server's load can be reduce. This proposed system balances the load of server using queue system which stores those processes when respective server is busy to process those particular processes. As soon as existing processes released by server next process from the queue system assigned to the respective server. Day to day we are observing the same in which server presents one message "Request Timed Out" or the situation in which it is busy to process that request instead of starting a new session we can store already started but not processed processes in to queue system and processes from queue system will be dumped on to the server.

If we bypass the queue system then traditional algorithms directly dump that process on respective server due to this server will be overloaded. Once this situation occurs, it consumes more power than its ideal state, so in this situation queue system is useful to reduce the power consumption of server peer.

In future we can analyze combination of different type of servers with different parameters. This type of approach will be most suitable in which network traffic load is not properly managed. If we apply queue system in such environment then traffic load can be easily managed as compare to traditional system.

REFERENCES

[1] Tomoya Enokido, Ailixier Aikebaier, and Makoto Takizawa "A Model for Reducing Power Consumption in Peer-to-Peer Systems", IEEE System Journal, Vol. 4, No. 2, pp. 221-229, June 2010.

[2] Tomoya Enokido, Ailixier Aikebaier, and Makoto Takizawa "Process Allocation Algorithms for Saving Power Consumption in Peer-to-Peer Systems", IEEE Transaction on Industrial Electronics, Vol. 58, No. 6, pp. 2097-2105, June 2011.

[3] Tomoya Enokido, Kota Suzuki, Ailixier Aikebaier "Laxity Based Algorithm for Reducing Power Consumption in Distributed Systems", International Conference on Complex, Intelligent and Software Intensive Systems, pp. 321-328, 2010.

[4] Tomoya Enokido, Kota Suzuki, Ailixier Aikebaier, "Algorithms for Reducing the Total Power Consumption in Data Communication-based Applications", IEEE International Conference on Advanced Information Networking and Applications, pp. 142-149, 2010.

[5] Andreas Merkel, Frank Bellosa, "Balancing Power Consumption in Multiprocessor Systems", EuroSys'06, April 18–21, 2006.

[6] Min Yang, and Yuanyuan Yang, "An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing", IEEE Transcation on Computers, Vol. 59, No. 9, pp. 1158-1171, September 2010.

[7] Ajit Singh, Priyanka Goyal, Sahil Batra, "An Optimized Round Robin Scheduling Algorithm for CPU Scheduling", (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 07, pp. 2383-2385, 2010.

[8] Tomoya Enokido, Ailixier Aikebaier and Makoto Takizawa, "An Algorithm for Reducing the Total Power Consumption Based on the Computation and Transmission Rates", International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 233-240, 2011.

[9] Yaashuwanth, Dr. R. Ramesh, "Design of Real Time scheduler simulator and Development of Modified Round Robin architecture" IJCSNS International Journal of Computer Science and Network Security, Vol.10 , No.3, pp.43-47, March 2010.

[10] Akshat Verma,GargiDasgupta,Tapan Kumar,NayakPradipta, De Ravi Kothari,"Server Workload Analysis for Power Minimization using Consolidation" IBM India Research Lab.

[11] Vinicius Petrucci, Enrique V. Carrera, OrlandoLoques, Julius C. B. Leite, "Optimized Management of Power and Performance for Virtualized Heterogeneous Server Clusters", 11th IEEE/ACM International Symposium on Cluster, cloud and Grid Computing, pp. 23-32, 2011.

DOI 10.17148/IJARCCE.2015.41218