Mobile Agent Based-Approach for Enhancing Energy Efficient Cloud Data Centre Network

Ogechukwu M Okonor, Mo Adda, Oliver Spear and Alex Geogov
School of Computing
University of Portsmouth
United Kingdom
ogechukwu.okonor@port.ac.uk, mo.adda@port.ac.uk, oliver.spear@myport.ac.uk and alex.geogov@port.ac.uk

Abstract: Cloud computing has become the bedrock of new, emerging smart technology. It has numerous benefits that encourage small, medium and large enterprises to use other technology through the internet. Despite the benefits of cloud technology, the challenge of the high power consumption rate it incurs as its leverages its promised attributes remains. This paper used mobile agent technology to solve the complexity problem in the data centre system, which happens to be the backbone of cloud technology. The mobile agent was embedded to both servers and switches to regulate their activities and then shut down underutilised components. Mobile agent (Java agent) is the first of its kind used on a cloud network. Our proposed approach saved a significant amount of energy and also improved the entire system performance during runtime.

Key-Words: energy efficiency, mobile agent, cloud computing, data centre

1 Introduction

Cloud computing is continuing to gain broader acceptance in the information technology world because it provides the simplest way of accessing servers, storage and other software applications through the internet. The revolution of network-based services via the internet like online shopping, social networking, internet of things and video broadcasting service has shown the significance of cloud innovation. Cloud systems have vast network architecture, interconnectivity capability and flexibility. This cloud attribute enables secure internetwork connections to multichannel through the internet. The author [1] acknowledged that cloud technology is the future and backbone for more incoming smart wireless innovations. Okonor said that cloud services had leveraged the deadlock in the IT world due to its acceptability and affordability.

Cloud communication flow can be categorised into two main parts; the cloud-to-user flow and the intra-cloud flow. Compared to the era of fewer cloud activities where local computers hold all the data and software for the user to now where correspondences are received on request from the cloud data centre. Cisco calculated that network traffic is the fastest -growing data centre component, rising to 4.3ZiB in 2016 with a combined annual growth rate of 44% [2].

All of these fantastic cloud computing attributes of the cloud system wouldn't have been valid without its unseen data centre infrastructure. Therefore, for clarity cloud operation depends on the geographical location of distributed data centre networks. For cloud system sustainability purposes, it is essential to understand the design, performance and management strategy that creates an optimal data centre.

The Data centre(DC) structure is a multitude of networks of both computing and storage devices. Data centres are designed in such a way that it must be agile, resilient, communicative, powered 24/7 and produce an acceptable throughput. A data centre can be in a centralised or distributed mode of operation. The architectural design of the data centre affects its structural performance, i.e. how the server connects to the switches and its deployment mechanism. A typical data centre structure consists of a core, switches, pods, servers and virtual machines(VM) which enable the delivery of shared software applications. The data consumes a substantial amount of energy during its activities runtime to be able to maintain the service level agreement(SLA) agreed by the cloud service providers; one of the adverse effects on cloud technology.

To evaluate the performance and efficiency of the data centre, we measured the amount of supplied electrical energy that turned into computing power during the network transmission process. Research Publication from [3] forecasted that in Europe, data centres would consume up to 93TWh by 2020. Furthermore, it is evident from Andera [4] research article that the power usage rate keeps increasing. By 2025, the data centre would amount to the most significant ICT share of global electricity production at 33%, followed by smartphones (15%). An

American researcher validated the finding of Andrea in 2019 by modelling actual figures available on power usage trend of the data centre will be up to 20% of global electricity by 2025.

Over the years, emphasis on how the use of agent technology was significantly capable of solving large scale distributed system problems. On the other hand, testing agent technology in cloud system networks will leverage some of its high power usage challenges, being that agents are flexible and easily interact with each other on a network to achieve a set goal.

Therefore, the remaining part of this paper is sectioned as follows: section 2, reviews on related works; section 3 agent activities; section 4 overview of proposed agent model and result analysis, section 5 conclusion and future work.

.

2 Related *Work*

Technical white papers and academic research articles have described methods for modelling power based on experience or data centre subsystem design. Insights from their previous works have led to constant improvement and understanding of the cloud system requirement. However, the irresistible fact from S Mittal et al.[5,6] work shows that more than 100 million videos are watched via YouTube per day. At the same time, Facebook has more than 400 million active users and 3 billion photos uploaded every month. Data centres power all these activities through their computation-intensive software programs. Based on the high-level traffic activities, the data centre process per minute and some traditional systems build-in not too agile and resilience to cope with the pressure of new inflow of traffic experience leaves the quest for an energy-efficient DC in limbo. Therefore, understanding not just the mechanism of operating the data centre but also the structural design will help minimise the power usage by the system. Related works on data centre energy efficiency have focused on different factors (from scheduling to allocation polices and then to some parts of the hardware. The authors [7] based their research findings on traditional x86 enterprise server architecture which when compared to the sophisticated transaction expected of the modern data centre can't compete because the energy-efficient tools were not the target at the time. Many studies on power usage within the data centres used the dynamic voltage frequency scaling (DVFS) management technique to predict the amount of power used at every given traffic transaction. The author [8], summarised the energy consumption of the hardware components of DC up to 2010. Fan et al. [9] focused on DC energy model based on just CPU utilisation. Work from [9] on processors in FD-SOI shown the effect setting a justification near-threshold on DC technology is capable of running a dual-core Cortex processor at 1GHz at the supply voltage of 0.6V.

3 The Proposed System

3.1 Overview of Agent Technology and its Activities

An agent is an intelligent computational entity that can act on its own or behalf of other entities (be it on stand-alone or embedded basic) to achieve a set goal. Agent technology has two perspectives the end-user agent perspective and the software perspective. Agent technology can also be stationary and Mobile. A stationary agent is an agent that just stays in the same system; it initially started its execution process. In contrast, the mobile agent is not bound to one system; it can begin its execution process in one network and then end up in another because of its unique ability to transport itself to other networks. In this work, we are considering a mobile agent called Java agent. The aim is to use an agent in an applet form to communicate with cloud complex distributed network without more induced complexity while reducing the power consumption rate.

In this paper, the agent used has the following features:

- **Autonomy**: The ability to act on its own to a certain degree and adapt to system traffic demand change. This reduces the rate of technical errors and the human struggle to keep up with the operational challenge.
- **Re-activity**: The ability of the mobile agent to react to external events, stimuli and consequently adapt to a sudden change in the system performance, then make a quick decision to suit the system state.
- **Communicative**: The mobile agent must have the ability to communicate with the network and other agents (in case of multi-agent scenario) to achieve a given goal.
- **Learning**: The java agent can learn from the network performance, then improve the performance by making adjustable decisions as it keeps actively interacting with the executable environment.

With regards to the use of mobile agent within the cloud environment, not much work has been done using this lightweight mobile agent technology; thereby making our proposed method a novel approach to tackling the challenges that face cloud technology, especially in the area of energy efficiency. According to Claudio et al-2017, cloud computing relies on the geographical distribution of a data centre network to function. Agent activities are thereby highly recommended for cloud network design due to these three agent attributes: 1, Agents are actively at their best, where the problem domain is geographically distributed. 2. The subsystem exists in a dynamic environment 3. The subsystem regularly interacts with each other flexibly [10]. Therefore, considering the domain of cloud networks and data centre activities, the agent-based system technology is well suited for use in solving the complexity of the network. Related literature research has shown that agent technology has already been deployed to many aspect ad-hoc networks, road traffic control [11] and fault detection. The framework in figure 1 shows the structure, link and advantageous effect mobile agent leverages for the cloud system.

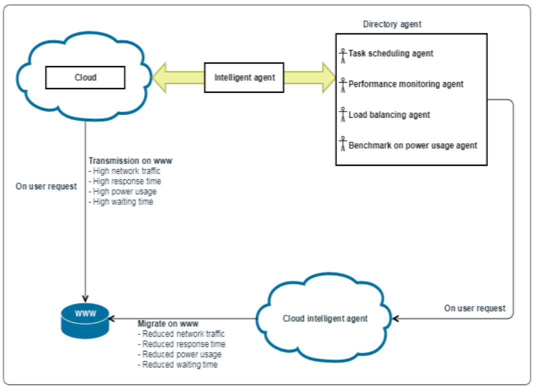


Figure1: Cloud-Java mobile agent framework

3.2 Data Centre System Model

In this section, the description of the data centre system model used for this experiment is made clear. This paper considers a three-tier data centre topology, shown in figure 2. The data centre consists of three structural layers—the core layer containing $(n/2)^2$ n-port switches. The centre layer (aggregation level) contains n/2 switches, i.e. one pod has n n-port switches, and the last layer (The Access Layer) connects to the servers also called hosts. In this work, we also study cisco power consumption of the 8-port(2960-8TC-L), 24-port(2960-24TC-L), and 48-port(2960-48TC-L) cisco switches are 12W, 27W and 39W[12] to have an insight of what to expect from stimulation environment with our synthesised topology.

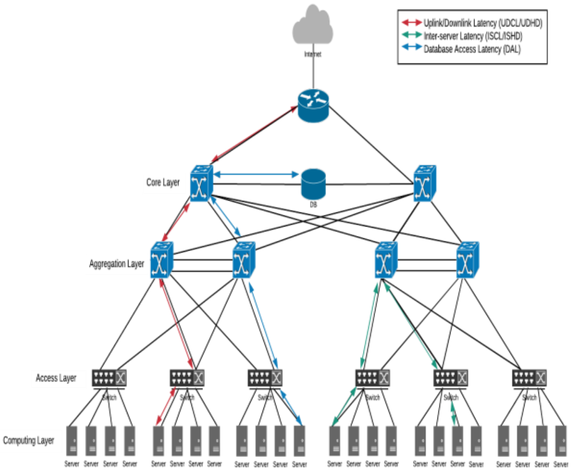
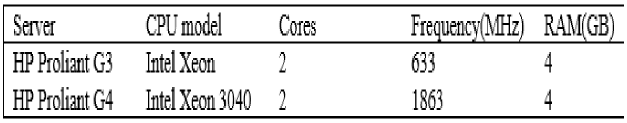

Uplink/Downlink Latency (UDCL/UDHD)
Inter-server Latency (ISCL/ISHD)
Database Access Latency (DAL)
Internet
Core Layer
DB
Aggregation Layer
Access Layer
Switch
Computing Layer
Server

Figure2: Data centre architectural design used for this Model

This system has two core switches, four aggregation switches, six access switches, 48links and N-heterogeneous physical nodes. Each server then has this attribute: CPU performance defined in millions of instructions per second(MIPS), amount of RAM and bandwidth.SLA is negotiated and agreed upon by the service provider and the user, depending on the QoS requirement. If there is any discrepancy with the SLA, the service provider bears the responsibility. This setup consists of a software layer whereby the global and local manager function with an embedded agent in between. The global manager receives a request from users, sends it to the local network manager while the local agent autonomously migrates VMs according to the specified requirement and threshold. The Java agent then sorts out the underutilised servers and switches on the aggregation and access layer then powers it off or puts it in sleep mode depending on a particular threshold requirement. In this system model, the server has a multicore CPUs. Each multicore CPU has K cores, and each core has M MIPs as a single-core which make the CPU sum up its total capacity as K*M MIPS. Table 1 below gives a clear description of the server specification and configuration used in this research work.

Table 1: Servers Specifications.



Server	CPU model	Cores	Frequency(MHz)	RAM(GB)
HP Proliant G3	Intel Xeon	2	633	4
HP Proliant G4	Intel Xeon 3040	2	1863	4

The Java agent has two deployment modes into the system model (time-based and daemon-base.). An agent is thereby sent to the server and switch nodes at a set time to check the utilisation level. The server can be shut down by the agent if it is under-utilised and switches made inactive if it is lower than the fixed performance capacity.

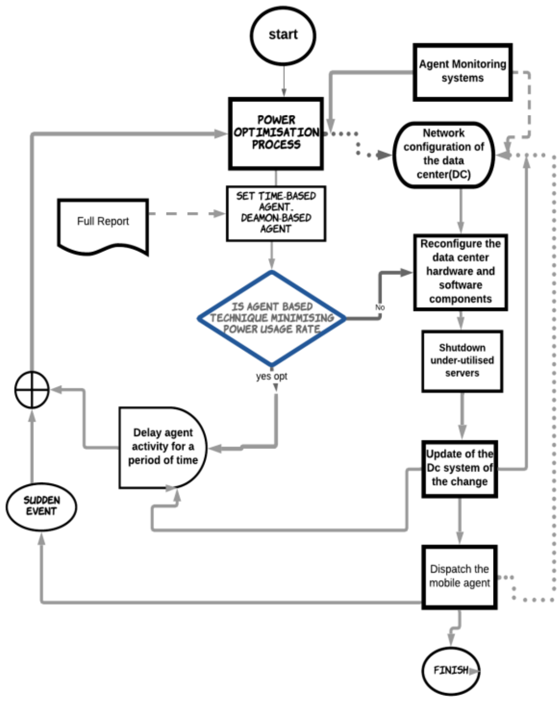

start
Agent Monitoring systems
POWER OPTIMISATION PROCESS
Network configuration of the data center(DC)
Full Report
SET TIME-BASED AGENT. DEAMON-BASED AGENT
Reconfigure the data center hardware and software components
IS AGENT BASED TECHNIQUE MINIMISING POWER USAGE RATE
No
Shutdown under-utilised servers
yes opt
Delay agent activity for a period of time
Update of the Dc system of the change
SUDDEN EVENT
Dispatch the mobile agent
FINISH

Figure 3: The agent-based flowchart used for DC system model implementation.

Building an algorithm from the flowchart in figure 3, we, therefore, present the mathematical backing for our working simulation. Given a detailed insight of the components that consume the most power rate during processing time. Following up from our previous work, Therefore, we consider the power consumption model for switches (P_{swi})as:

$$\boldsymbol{P_{swi} = P_{chi} + \sum_{j=1}^{n_{lc}} P_{lc_j} + \sum_{j=1}^{n_{pi}} P_{r_j} * U_{ij}} \qquad (\mathbf{1})$$

where
U_{ij}: Utilisation factor of an active port *j* per switch *i*
P_{chi}: Power consumed by a switch i base on hardware chassis
P_{lc} : Power consumed by the active line-cards
P_{ri} : Power consumed by the active port running at a transmission rate ‘r’
P_{aLcswi} : power consumed by the switch when the agent controls the line-card
n_{lc}: the number of linecard.
n_{pi}: number of ports

$$\boldsymbol{P_{aLcswi} = P_{chi} + \sum_{j=1}^{n_{lc}} (1-\beta_j) P_{lc_j} + \sum_{j=1}^{n_{pi}} P_{r_j} * U_{ij}} \quad (\mathbf{2})$$

$$\beta_i = \begin{cases} \quad 0 \quad \; if\ a\ linecard\ is\ enabled \\ \quad 1 \quad \; if a\ linecard\ is\ disabled \\ otherwise\ operational\ by\ the\ agent \end{cases}$$

Where β_i is between 0 & 1 and is used by the agent to control the linecard operation.
Now simplifying equation 2 further we then get:

$$\boldsymbol{P_{aLcsw_i}} = \boldsymbol{P_{swi}} - \sum_{j=1}^{\boldsymbol{n_{lc}}} \boldsymbol{\beta_j P_{lc_j}} \qquad (3)$$

Considering the agent activities on the switches, α_i is tabulated below as

$$\alpha_i = \begin{cases} 0 \quad if\ a\ switch\ is\ off \\ 1 \quad if\ \ a\ switch\ is\ on. \\ otherwise\ operational\ by\ the\ agent \end{cases}$$

Where α_i is between 0 &1 and is used by the agent to control the switch operation.
$\alpha_i = 0$ means the whole switches are powered off,
$\alpha_i = 1$ means the switches are fully active.
Therefore, the total power consumed by an agent switch follows as:

$$\boldsymbol{P_{agentsw_i}} = \alpha_i * \boldsymbol{P_{aLcsw_i}} \qquad (4)$$

To a great extent, α_i is strongly related to β_i in the case where all the networkcard of a given switch are disabled, then the switch itself should be turn-off.
Then for the server power consumption(P_{srvi}) The calculation, we used this formula also:

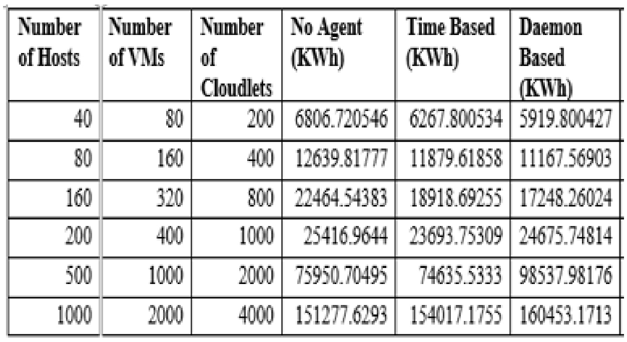
$$\boldsymbol{P_{srvi}} = \boldsymbol{P_{cpu}} + \boldsymbol{P_{memory}} + \boldsymbol{P_{fan}} + \boldsymbol{P_{io}} \qquad (5)$$

Following the same reasoning as in the switch formulation in equation 2 & 3, the power consumed by the server under the intelligent of the agent follows the same logic and parameter as the $\alpha_i and\ \beta_i$in switches, however in this case measures for $cpu, memory, fan\ and\ P_{orts}$ .The power consumed by the fan is related to the power consumed by the CPU & memory, and this result can only be obtained by simulation. The load balancing of the VM is determined by the set power threshold.

4 Result Evaluation

In this section, the mobile agent results of various simulation outputs are evaluated for both time-based and daemon based, respectively. For this simulation, the Cloudsim simulation platform was used. Cloudsim is written in Java and therefore proved to be both simple and effective to implement with the proposed mobile agent in this case called Java agent.
In the first simulation instance, a datacenter with *X* number of hosts, VMs and Cloudlets were used for the no-agent, time-based and daemon-based methods. Table 2 shows the different data centre configurations used for the number of Hosts, VMs and Cloudlets along with the total power consumption used by each agent type.

Table 2: Data configuration component and its power output on different agent scenario

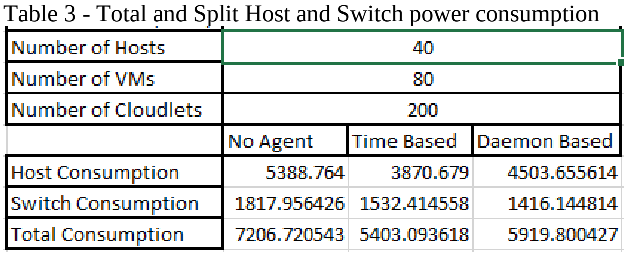


Number of Hosts	Number of VMs	Number of Cloudlets	No Agent (KWh)	Time Based (KWh)	Daemon Based (KWh)
40	80	200	6806.720546	6267.800534	5919.800427
80	160	400	12639.81777	11879.61858	11167.56903
160	320	800	22464.54383	18918.69255	17248.26024
200	400	1000	25416.9644	23693.75309	24675.74814
500	1000	2000	75950.70495	74635.5333	98537.98176
1000	2000	4000	151277.6293	154017.1755	160453.1713

From these results obtained from the set experiment, it is clear that the power consumption used by both the time-based and daemon-based agents is significantly lower than that of the no-agent approach by more than 25% efficiency. This paper has used different time frequencies ranging from 100 seconds,60 seconds,40 seconds, 20 seconds and 5 seconds to obtain a proof-of-concept dataset that both the time-based and the daemon-based agent methods significantly reduce the power consumption within a cloud data centre network.

Table 3 shows the power consumption used by all three approaches split between host and switch power consumption, further demonstrating the power savings between the two.

Table 3 - Total and Split Host and Switch power consumption



Number of Hosts	40		
Number of VMs	80		
Number of Cloudlets	200		
	No Agent	Time Based	Daemon Based
Host Consumption	5388.764	3870.679	4503.655614
Switch Consumption	1817.956426	1532.414558	1416.144814
Total Consumption	7206.720543	5403.093618	5919.800427

Furthermore, this is consistent as the data centre increases in both resource and workload size. The graphical representation in figure 4 shows the impact of our finding and when its compared to other previous research works gives a more promising result and more efficient way of operating the DC network.

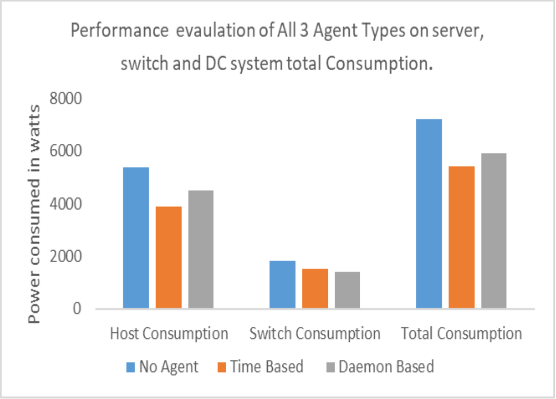


Figure 4a: Performance evaluation of all three agent types on the server

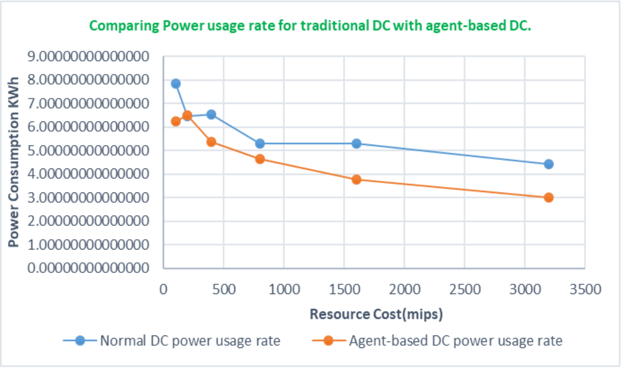


Figure 4b –Compares the performances of traditional DC power usage rate with Mobile agent-based DC

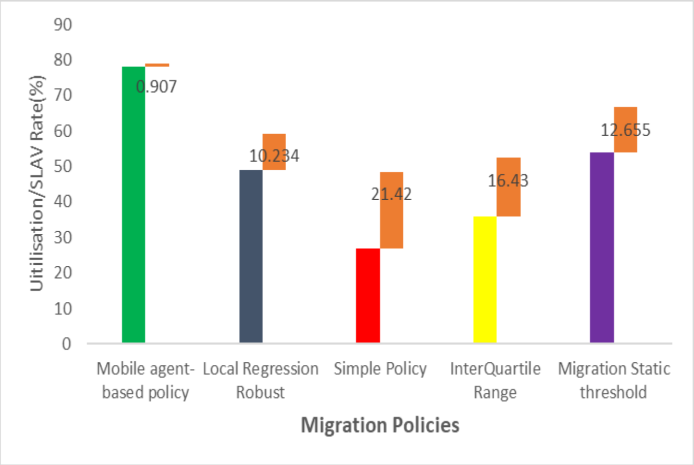


Figure 4c:-Comparing Mobile agent policy to other existing policies.

Observations from the graphical representation in figure 4(a &b) show that at all levels, the mobile agent was able to minimise energy efficiently without sabotaging the performance of the system.

From the results in figures 4, the data obtained has proven that the use of an agent increases the agility of the system, therefore reduces the operational cost of maintaining a data centre network. Figure 4c clearly shows that the performance of the mobile agent in the network was outstanding with no significant service violation of the system.

5 Conclusion and Future work

The proposed agent technology has a unique new technique of communicating efficiently with the data centre network under reduced power usage rate. This agent technique investigates the DC system efficiency from energy, performance and traffic-related perspectives. The experimental result produced showed several properties from DC components that consume a higher amount of energy and how the agent technique excellently migrates resources to reduce congestion and minimise high power usage rate. The java agent flexibly controls the DC system with no downtime and less technical error due to its adaptive nature. The mobile agent was able to asynchronously and independently monitor the system performance and power consumption rate on both the switches and servers. The introduction of time-based and demon-based agent in the cloud DC network is an entirely new novel approach to energy saving in DC. Therefore, this approach is intelligently positioned to becoming an essential tool for academy and industry researcher specialists for scaling and dynamically managing cloud DC network complexities.

Future work will continue to work on other parameters we didn’t consider in this work. This novel approach will then be used to model the complete system power consumption rate mathematically taking into consideration the service level violation rata involved during its activity.

References:

[1] Ogechukwu Okonor, Mo Adda” power optimisation model for leveraging cloud system.” IEEE Big Data Conference, Sofia, Bulgaria, 2019.

[2] Cisco.” Cisco Global Cloud Index: Forecast and Methodology,2012-2017”, White paper,2013.

[3] Claudio Fiandrino, DzmitryKliazovich, Pascal Bouvry and Albert Zomaya.”Performance and Energy Efficiency Metric for Communication Systems Of Cloud Computing Data Centers” IEEE Transactions on Cloud Vol.5 No.4,2017.

[4] Andreas Berl1,Erol Gelenbe, Marco di Girolamo, Giovanni Giuliani, Hermann de Meer1, Minh Quan Dang and Kostas Pentikousis, “Energy-Efficient Cloud Computing” Published by Oxford University Press on behalf of The British Computer Society. Advance Access publication on August 19, 2009.

[5] Okonor, Mo Adda” Intelligent Approach to Minimising Power Consumption in Cloud-Based System Collecting Sensor data and Monitoring the Status of Powered Wheelchair” Intelligent system conference 1, 2019

[6] Sharma .A, Y Yao, L Huang- 2012 Proceeding…,2012- IEEExplore.ieee.org

[7] Ali Pahlevan, Yasir Mahmood Qureshi and Marina Zapater.”Energy Proportionality in Near-Threshold Computing Server and Cloud Data Centers: Consolidating or Not?Proceeding…..,2018 - IEEExplore.ieee.org

[8] E.Faller, C. Morin, D.Leprince," State of the Art Saving in Clusters and Results from the EDF Case Study, INRIARennes-France,2010

[9] X.Fan, W.D. Weber, L. Barroso, Power provisioning for a warehouse-sized computer,in proc. International Symposium on the computer, Architecture(ISCA07),June 2007, pp13-23,DOI: 10.1109/ISPASS.2011.5762739

[10] Alder, J.L., Satapathy, G., Manikonda, V., Bowels, B., Blue, V.j., 2005. "A Multi-agent approach to cooperative traffic management and route guidance.Transportation Research Part B 39(4),297-318.

[11] Chabrol, M., Sarramia, D., Tchernev, N.,2006."Urban traffic system modelling methodology. International Journal of Production Economics 99(1-2),156-176.

[12] Cisco.http://www.cisco.com