Analyzing Component-based Energy Measurement of a Server for client-server application

Shruthi G T[1], Chetana Srinivas[2]
[1, 2] Department of CSE
[1, 2] East West Institute of Technology, Bengaluru-560 091, Karnataka, India

Abstract- *Cloud Computing provides utility oriented IT services to the users. This is based on a Pay-As-You-Go approach, thereby it enables hosting of applications from clients. Data Center hosting cloud applications consume huge amount of energy. Hence Energy Consumption by Data Center Servers is a growing issue. In this work, measurement of energy consumed by different components, such as, CPU, Disk Input-Output operations and network activity are done. Estimation of energy consumed by multiple components is based on load. And gives comparisons of energy consumed due to network connection and without network connection. Experiments done on two different scenarios: one is on Isolated Server (no network) and another one is with a network connection. Finally energy versus time graph is obtained for client-server application.*

Keywords- CPU, energy measurement, hard disk, Network.

I. INTRODUCTION

Cloud Computing delivers IaaS, PaaS, and SaaS. These services are helpful for users as a payment-based service under the Pay-As-You-Go form. Clouds aim is to design server farms by designing them as system of application, database, equipment and client interface, so clients can get to and send their applications from anyplace with respect to cost and Quality of Service. Companies like Amazon, Google, Microsoft, Yahoo! and many more have deployed data centers on cloud, which housing tens of servers.

Data centers are meant to be as a place where computer related resources and data are stored. To access the resources users do not require physical connection. There are certain challenges to maintain Data Center, they are, improving utilization of Power and Space, Maintaining availability and Uptime, managing energy usage and Costs and Reporting reduced operating expenses.

In this proposed approach, we are concentrating on measurement of energy consumed by data center server components. Here estimation of CPU, Disk and Network activity energy consumption is carried out based on load. There are three types of load generated during Put, Get and Delete operations; they are load on memory, load on disk and load on network.

Main results: (1) proposed methodology calculates energy usage of a server component used for Get, Put & Delete operations. (2) Experimental result shows the highest energy consumed component in the server system.

II. RELATED WORK

There is an expansive collection of work in the field of demonstrating server power utilization and its parts, both hypothetical and exact. The vitality utilization of servers has been expected as direct, where utilization depended mostly on CPU and straightly on its usage [1] [2].

Distinctive technique accessible to diminish vitality with lessened clock pace, for example, decreasing the voltage or utilizing reversible or adiabatic rationale. Consider three sorts of booking calculations. (1)Unbounded-delay impeccable future, (2) Bounded-delay restricted, and (3) Bounded-delay constrained. These calculations change the CPU clock speed while booking choices are made, with the objective of diminishing time squandered in the unmoving circle while holding intuitive reaction [2].

Unbounded-delay immaculate future takes the whole follow, and extends the run times to fill the unmoving times. Power reserve funds were constrained by the base velocity, accomplishing the greatest conceivable investment funds over the period. This calculation is unrealistic and undesirable. It is unrealistic in light of the fact that it requires flawless future information of the work to be done over the interim. It is undesirable since it creates expansive deferrals in runtimes of individual employments without respect to the requirement for compelling reaction to continuous occasions, for example, client keystrokes or system parcels [2].

FUTURE is like OPT, with the exception of it looks into the future just a little window and advances vitality over that window, while never postponing work past the window. It is expected that all unmoving time in the following interim can be wiped out, unless the base rate of the CPU is come to.

FUTURE is unfeasible, in light of the fact that it utilizes future information. It is attractive, in light of the fact that no ongoing reaction is ever deferred longer than the window [2].

PAST is down to earth rendition of FUTURE. It looks an altered window into the past and expects the following window will resemble the past one [2].

Consider recurrence in the examination [3]. To begin with consider CPU power usage, which relies on upon the quantity of working centers, the CPU recurrence and the CPU load. Vitality utilization with a solitary working center at consistent recurrence can be nearly approximated by a straight capacity of the CPU load. The energy utilization for a settled CPU burden is minimized by utilizing the most elevated number of centers and the least recurrence.

The vitality utilization for perusing and composing relies on upon the CPU recurrence and the I/O square sizes. While vitality utilization because of perusing is not influenced by piece estimate but rather scarcely influenced by the CPU recurrence, where as composing increments with the square size [3].

The vitality utilization and productivity of Network interface card, in transmission and gathering relies on upon the CPU recurrence, the bundle size and the transmission rate [3].

III. PROPOSED METHODOLOGY

Estimation strategies used to portray the vitality utilization of CPU action, circle access (read and compose operations), and system movement are depicted in this area. Begin estimations by profiling the CPU vitality utilization because of CPU burden. Subsequently, layout the other two parts, that is, circle and system. Benchmark and screen vitality utilization is measured by the Joulemeter device when it is begun. Here CPU estimations are of primary significance with a specific end goal to assess alternate parts, on the grounds that each time that we run a script to profile the conduct of another segment, some CPU cycles are required to execute it and additionally to utilize the segment that needs to perform the undertaking.

The estimation procedures used to portray the force utilization of CPU movement, Disk get to and Network action depend on recurrence and burden. Total energy consumption is a sum of all the components energy consumption.

$$\textbf{Total EC} = E_{CPU} + E_D + E_N$$

Aim of this proposed system is to estimate the energy consumed by Data Center Servers, by measuring the usage of different server components energy consumption.

1. System setup

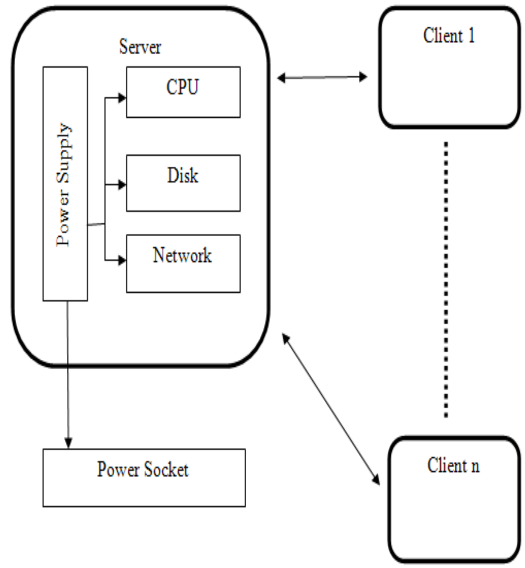

Figure 1: Proposed system architecture

One prerequisite for the experiment is to have windows machines and Joulemeter software to obtain energy values. Fix the frequency level at this tool and perform calibration before measuring energy consumption for particular operations. Because baseline and monitor energy values are important to obtain before starting application and also when system is idle. Once we have done calibration and fixed frequency, it is possible to start application to measure the energy utilization by this particular application and start saving those values. System architecture is shown in figure 1.

It includes single server with many clients. Clients are meant to generate load on server system. Components considered are CPU, Disk and Network. Server system should be connected to Joulemeter to obtain values.

To characterize energy measurement Put, Get and Delete operations are performed based on hashing technique. Figure 2, 3 and 4 shows algorithm for put, get and delete operations respectively.

Step 1: Start

Step 2: Read Key & Value pairs

Step 3: Bucket_Id = Hash (Key) mod nB

Store Key & Value pair in Bucket (Bucket_Id), file should be stored if key is associated with file

Step 4: Stop

Figure 2: Algorithm for put function

Step 1: Start

Step 2: Read Key

Step 3: Bucket_Id = Hash (Key) mod nB

Search for key in Bucket (Bucket_Id)

If Key found then

Return value as well as file if file is associated with key

Else

Return key not found

Step 4: Stop

Figure 3: Algorithm for get function

Step 1: Start

Step 2: Read Key

Step 3: Bucket_Id = Hash (Key) mod nB

Search for key in Bucket (Bucket_Id)

If Key found then

Delete Key & Value pair

Else

Return key doesn't exist

Step 4: Stop

Figure 4: Algorithm for delete function

2. Joulemeter

A software instrument that approximates the power consumption of the computers is known as Joulemeter. It trails computer resources, like CPU utilization, screen brightness and estimates the power usage.

Computation of energy costs have grow to be important, because they express blow on the power provisioning rate for computing infrastructures, the operating expense of servers and venture buildings and also battery verve for laptops and mobile devices. The Joulemeter development focuses on following features related to energy optimization:

Modeling

Joulemeter estimates the power usage of a VM, server or software application. It impacts the power organization of a variety of components, like the CPU, Screen, Memory and Storage on total power use.

Optimization

Joulemeter helps to get better power provisioning and expenditure costs in diverse scenarios ranging from data centers, enterprise computing and battery operated machines.

3. CPU

In order to assess the CPU power utilization client server script application is used to generate load on server system. Based on this load, energy measurement is done for CPU activity. Before starting this application, look at the Joulemeter tool, where it shows some watts of energy is utilizing by the system CPU. That is when system is in idle state, some part of CPU energy will be in use for the operating system operations.

4. Disk

The energy consumption of the disk was evaluated using client server script designed for reading and writing, as it permits us to read files, write files, manage the range of the blocks for writing or reading, manage the amount of blocks written or read and compel the consign of writing process after each block in order to decrease the outcome of OS caches and memory. Put, Get and Delete operations are performed using hashing technique to generate load on server Disk. This disk operation consumes less energy compare to CPU and baseline energy but it is not negligible.

5. Network

In order to assess the involvement of the network to the energy utilization, experiment is done based on a client-server script. Server program will be run on one system and client will be run on another system. Joulemeter tool is linked to server system to collect data. During this experiment load will be generated on CPU and also Disk. Then subtract the values from the data collected during without network connection. This value will be very small when compared to CPU and Disk utilization.

IV. IMPLEMENTATION

This section gives the experimental results for client server application. Experiment is carried out for two different scenarios. One is without network connection and another one is with network connection.

In the first case experiment is carried out without network connection. Both server and client will run on same system. Joulemeter tool will be turn on and at the client side operations such as get, put and delete are performed to generate load on server. CPU and Disk energy utilization will be obtained based on load on server.

In the second scenario, experiment is done with network connection. That is server program is run on one system and client program will run on another system. Joulemeter is turn on in server system. Operations like put, get and delete are performed at the client system and simultaneously load will be generated at the server system. CPU and Disk energy utilization is obtained based on load.

Figure 5 shows the results for load on server CPU and Disk without network connection.

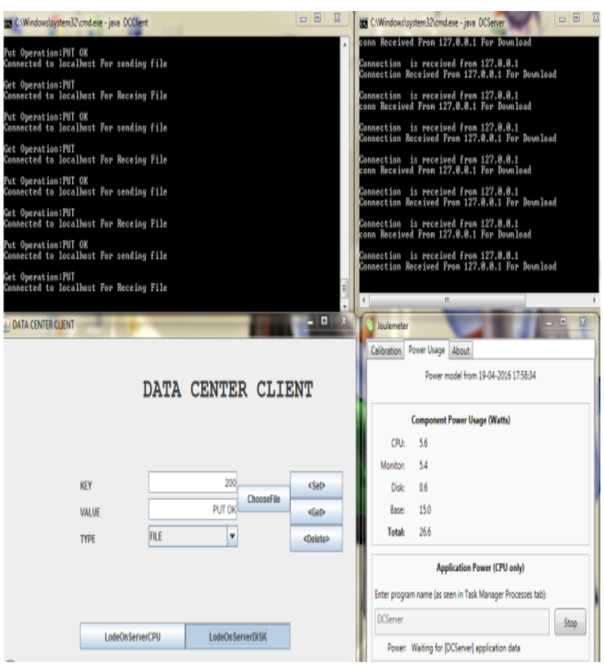

Figure 5: Load on server system without network connection

Figure 6 shows the results for load on server CPU and Disk with network connection.

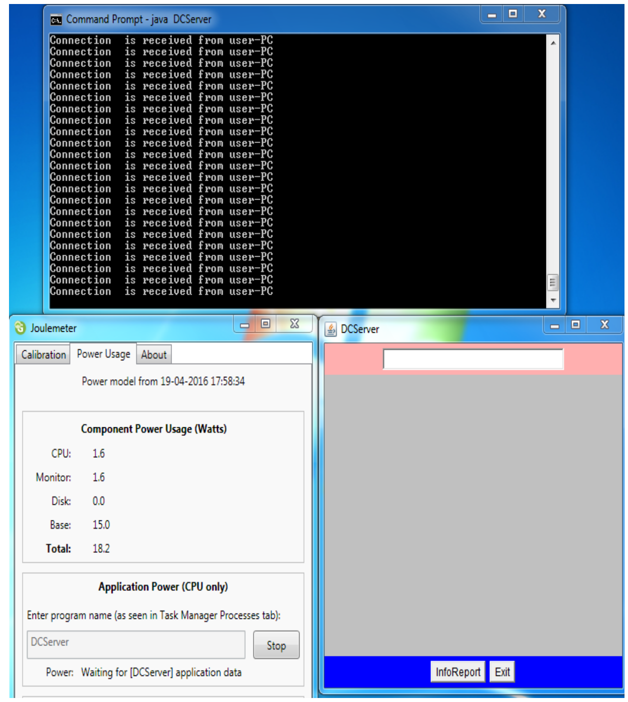

Figure 6: Load on server system during network connection.

Figure 7 shows the Graph for energy utilization obtained for CPU, Disk. And it also shows total energy consumption, Monitor and baseline utilization.

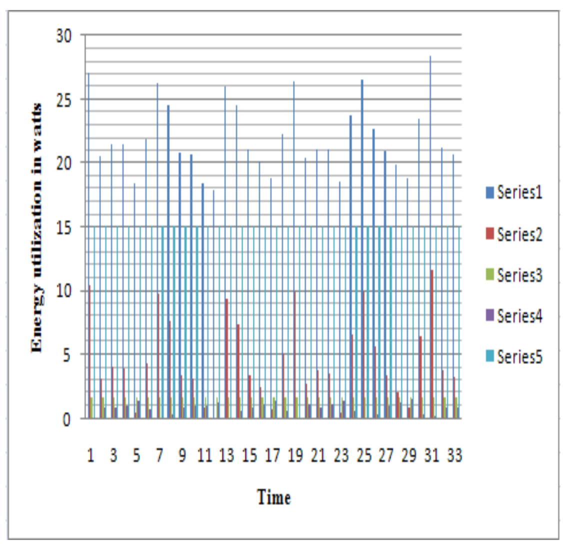

Figure 7: Graph for Energy utilization by server system.

Series 1 indicates total energy, 2 indicates CPU energy, 3 indicates monitor energy, 4 indicates Disk energy and 5 indicates baseline energy consumption.

Figure 8 shows that among CPU and Disk, CPU utilizes more energy compared to disk.

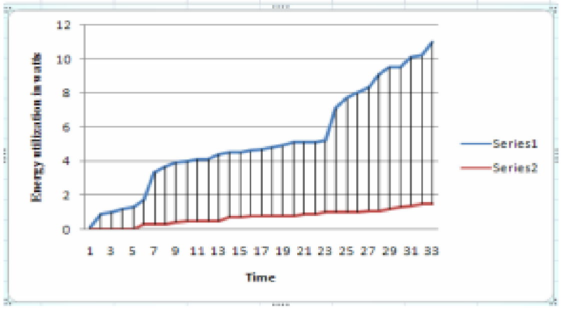

Figure 8: Graph shows the CPU and Disk energy utilization. Where CPU utilization is more compared to Disk.

Series 1 indicates CPU energy and series 2 indicates Disk energy consumption.

From the above two graphs, it shows CPU energy utilization is more compared to Disk. And Monitor and Baseline energy consumption is constant. Graph is plotted as time versus energy utilization. Time is in milliseconds, energy utilization is in watts.

V. CONCLUSION AND FUTURE WORK

This proposed method gives the study of components energy utilization for windows platform. From the experiment it shows CPU consumes more energy and Disk consumes less energy compared to CPU and it is not negligible. Monitor and baseline energy is calculated and it is constant. During Network activity also CPU energy consumption is more.

In future it is possible to obtain energy values for different platforms and if experiment is carried out for different hardware configuration, then it is easy to predict the best hardware configuration, which consumes less energy. Experiment can be carried out for Virtual machines. And also Experiment can be done for different applications, such as big data applications.

ACKNOWLEDGMENT

Sincere thanks to the respective guide Mrs. Chetana Srinivas for the constant encouragement and support.

REFERENCES

[1] Jordi Arjona Aroca, Angelos Chatzipapas, Antonio Fernandez Anta and Vincenzo Mancuso, "A Measurement-based Characterization of the Energy Consumption in Data Center Servers", IEEE Journal on selected areas in communications, 2015.

[2] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced CPU energy," in Mobile Computing, 1996, vol. 353, pp. 449–471.

[3] J. Arjona Aroca, A. Chatzipapas, A. Fern´andez Anta, and V. Mancuso, "A measurement-based analysis of the energy consumption of data center servers," in Proceedings of the 5th international conference on Future energy systems. ACM, 2014, pp. 63–74.

[4] Rajkumar Buyya, Anton Beloglazov, and Jemal Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges", 2013.

[5] Lin Wang, Fa Zhang, Jordi Arjona Aroca, Athanasios V Vasilakos, "GreenDCN: a General Framework for Achieving Energy Efficiency in Data Center Networks",IEEE Journal on selected areas in communication, January 2014.

[6] Yichao Jin and Yonggang Wen, Qinghua Chen, "Energy Efficiency and Server Virtualization in Data Centers: An Empirical Investigation",2013.

[7] W. V. Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, and P. Demeester, "Trends in worldwide ICT electricity consumption from 2007 to 2012," Computer Communications, 2014.

[8] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," Future Gener. Comp. Sy., vol. 28, no. 5, pp.755–768, 2012.

[9] L. Nonde, T. E. El-Gorashi, and J. M. Elmirghani, "Energy efficient virtual network embedding for cloud networks," Journal on Lightwave Technology, vol. 33, no. 9, pp. 1828–1849,2015.

[10] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi, "Energy consumption anatomy of 802.11 devices and its implication on modeling and design," in ACM CoNEXT, 2012, pp. 169–180.

[11] A. Brihi and W. Dargie, "Dynamic voltage and frequency scaling in multimedia servers," in IEEE AINA, 2013.

[12] R. Basmadjian and H. de Meer, "Evaluating and modeling power consumption of multi-core processors," in IEEE e-Energy, 2012, pp. 1–10.