International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 6 | ISSN : 2456-3307

Efficient Utilization in Processing Rate of Data on Occurrence of a Bursty Traffic in a Virtualization Based Cloud Data Center

Kaushalya Kanth[1], Dr Prakash S[2]
[1]Department of CSE, Dr.AIT, Nagarbhavi, Bangalore, Karnataka, India
[2]Professor, Department of CSE, Dr.AIT, Nagarbhavi, Bangalore, Karnataka, India

ABSTRACT

From past many years reducing the miss rate and energy consumption in data center has become the major concern. This can be done by dynamic programming, virtualization and also by sharing resources when there is a low traffic. However, when there is excess traffic (bursty traffic) one cannot share the resources, one need to switch on their own physical machine. In that span of time many users are lost. So, to avoid the above mentioned problem, one need to know when will the bursty traffic will arrive to do the needful. This can be done using Markov model.

Keywords : Bursty Traffic, Markov Model, Miss Rate Virtualization, Data Center, Dynamic Programming.

I. INTRODUCTION

Vitality utilization speaks to an undeniably huge thought for innovation organizations running huge information focuses. Around the world, server farms as of now represent more than one percent of aggregate vitality utilization [1]. Vitality costs in information focuses developed by over half in the vicinity of 2004and 2011, and given the persistently rising interest for handling power and cloud-base administrations comparative patterns are probably to continue . Also, as server farms keep on growing in estimate also, expend more vitality, an additional measurement of concern is their unmistakable effect on nature. The IT business is evaluated to contribute two percent of worldwide CO2 outflows, which is keeping pace with the aeronautics business [2].

The most recent years has deliberate push to lessen vitality costs in server farms through different unique power administration methods. Ideas like "power proportionality" what's more, "right-estimating" comprise coordinating framework control utilization and execution to the framework workload, specifically stopping or backing off framework parts sit still amid times of movement [3], [4], [5]. Dynamic programming and virtualization [6], [7], [8] has prompted additionally walks in managing vitality costs in server farms. Executing various machines on a solitary physical server, server farms are ready to lessen the measure of equipment being used and use its assets all the more proficiently, apportioning portions of CPUs, data transfer capacity, and circle stockpiling among various machines as their different pile of work. In any case, relentless difficulties exist in server farm asset administration that restrains the objective of accomplishing vitality proficiency. Progressing framework parts among various machines and vitality states causes setting times, particularly the servers are brought online once more [9], [10]. In interim, the task additionally devours noteworthy measure intensity, with extra vitality costes [5], [11]. The costly idea of reorganizing framework assets additionally, adjustments landing are generally difficult to foresee [12]. At point landing are occupations and are flexible, each reorganization assets turns into hazard choice: apportioning assets and vitality that procedure rashly may squander control, yet neglecting to foresee a sudden flood of movement can bring about huge postponements. This inborn strain between vitality investment funds and responsiveness in taking care of advancing workload requests remains a huge hindrance to across the board selection of advancements of current server farms [9].

Here, we present model with administration ratings, records two changes in arriving activity, exchanging contacts advances in vitality states. Changes in the entry rate are demonstrated as a limited state Markov model, yet the genuine covered up with controller, with end goal that the control choice can as it were be made in light of the watched grouping of occupation entries. Utilizing Bayesian surmising, the controller translates the entry arrangement to progressively refresh an arrangement of "convictions" about the concealed condition of the Markov model, and demands or discharges preparing assets in like manner. We clarify how our model can be communicated as a Markov choice process, and understand for the ideal control utilizing dynamic programming [13]. As far as anyone is concerned, this is the main case of an investigative model to consolidate highlights for both basic vacillations in the landing rate and change costs and postponements for exchanging among various administration rates.; Other work has utilized dynamic programming to consider versatile administration rates in a line with a Markov-regulated Poisson entry process [14]; thought to be known, and switches in the administration rate are both costless and quick.

II. BURSTY TRAFFIC USING MARKOV MODEL

There has been a requirement for techniques to conquer the excess utilization of energy in data center. Some outstanding productive systems are built to diminish this utilization rate, yet have failed to make that big appearance as the decrease is at a less rate. In spite of the fact that with change in innovation step by step, with creation of new strategies, for example, virtualization and managing power dynamically have managed the resources use components. Likewise we come over a new technique for running data centers at energy rate in the time of less traffic. However, these components and their switching are subjected to setup time required to get them online and furthermore noteworthy loss of energy for this procedure. This likewise fails on occurrence of a bursty traffic and prompts SLA violations. Here, in this project, I give the utilization of a queuing model based hypothesis by controlling the service rate, keeping in mind to overcome the above issue. A hidden state machine model is used to detect the occurrence of bursty traffic and an optimal Markov control policy is characterized with the help of dynamic programming in diminishing the average miss and energy utilization caused because of the setting up of machines. We compare its execution and the normal benchmarks and obtain satisfactory results.

Here in this project we have used morkov model to reduce the miss rate and reduce energy consumption while bursty traffic arrives.

Markov model:"A morkov model is a tool for representing probability distributions over sequence of observations" Here, first we execute bursty traffic (numerous number of instructions)without using markov mode, using cloudsim as a platform of execution.

Four hosts are used here, instructions are given to hosts for execution, after certain time limit the rest three hosts gets switched off, only one host is responsible for execution of all the instruction, since only one host is responsible for execution, many instructions do not get executed so miss rate will be high.

Now the hidden markov model would have observed the arrival time and arrival rate of bursty traffic.

next,we execute bursty traffic (numerous number of instructions)using markov mode, using cloudsim as a platform of execution.

Four hosts are used here, only one host is be switched on initially, depending upon pervious observations of arrival rate of bursty traffic rest three hosts are switched on, since all the hosts are switched on, instructions or bursty traffic gets exuted by all the hosts, by reducing the miss rate to an extent.

III. SYSTEM ARCHITECTURE

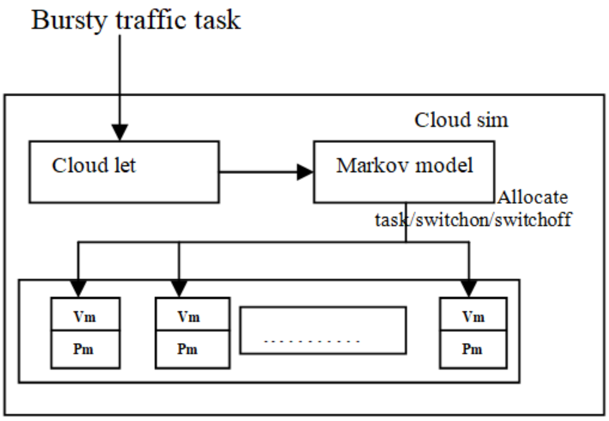

Figure 1. System architecture 1

Cloud Sim: "It's a platform/frameworkfor simulation and modeling of cloud computing infrastructure and services"
Clod sim contains cloudlet pool, markov model, datacenter which contains both virtual memory and physical memory. Bursty traffic enters the cloudlet pool this bursty traffic is given to markov model. Markov model analyses the traffic and decides whether to switch on or switch off the host and decides which task has to be allocated to which hosts

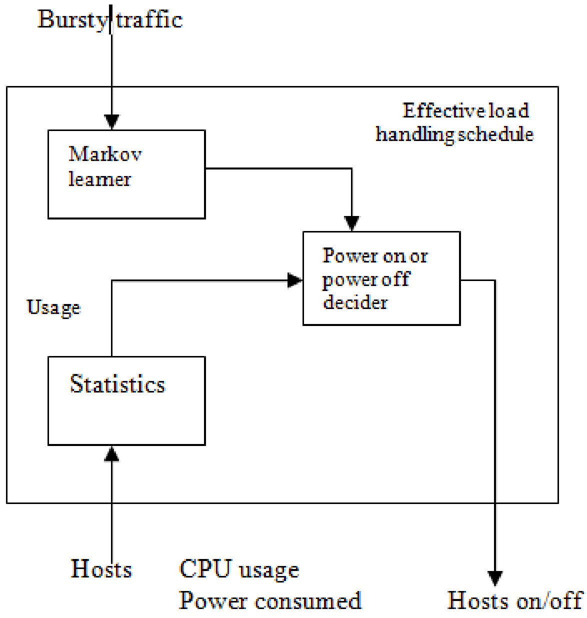

Figure 2. System architecture 2

Effective load handling schedule contains markov learner, power on or power off decider, statistics.

Bursty traffic is given to markov learner, depending on the traffic the power on/off decider decides whether to switch on/off the hosts, after switching on the hosts the CPU usage and power consumed for executing the bursty traffic is collected in a static format and given to power on/power of decider , so that next, before the bursty traffic arrives depending on the previous observation power on/off decider can switch on the hosts. By doing this one can reduce the miss rate.

IV. EVALUATION

In evaluation step we evaluate the results of both natural benchmark(without using markov model) and with using markov model. Simulation data of both the model are collected

1. Simulation data collected without using markov model

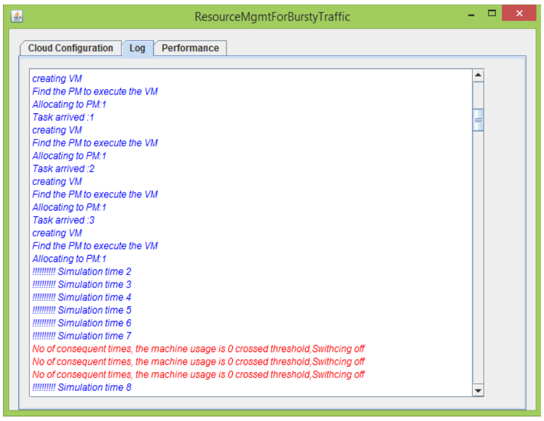

Figure 3. Blue: task execute Red: task not executed

In the above simulation data collected, one can notice that some tasks are executed and some tasks are not executed. so we can conclude that using natural benchmark policies one can encounter miss rate.

2. Simulation data collected with using markov model

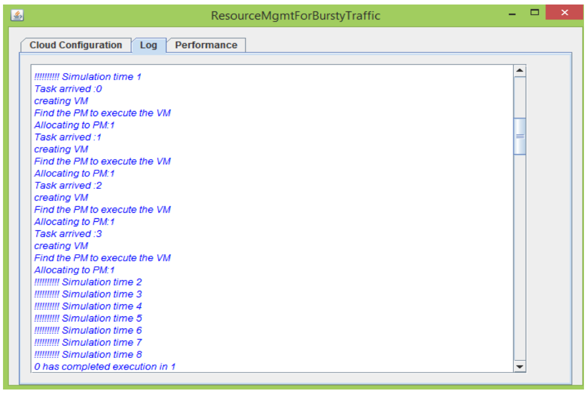

Figure 4. Blue: task execute Red: task not executed

In the above simulation data collected, one can notice that all tasks are executed. so we can conclude that using markov model one cannot encounter miss rate.

Now after collecting the simulation data one can plot the graph of miss rate and energy consumption of both natural benchmark policies and with using markov model.

3. Graph of miss rate is plotted without using markov model.

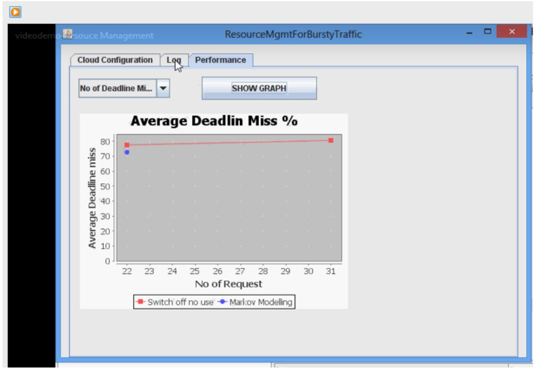

Figure 5. Miss rate

In the above graph, we can see that when we don't use Markov model the miss rate is high, i.e. red line.

4. Graph of miss rate is plotted with using markov model.

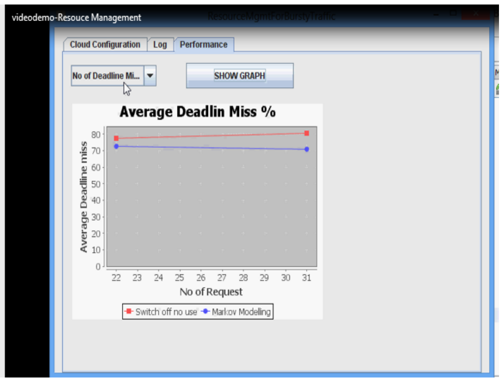

Figure 6. Miss rate

In the above graph ,we can see that when we don't use Markov model the miss rate is high. i.e red line. But when we use markov model and then execute the code the miss rate reduced i.e blue line.

5. Energy consumption graph is plotted

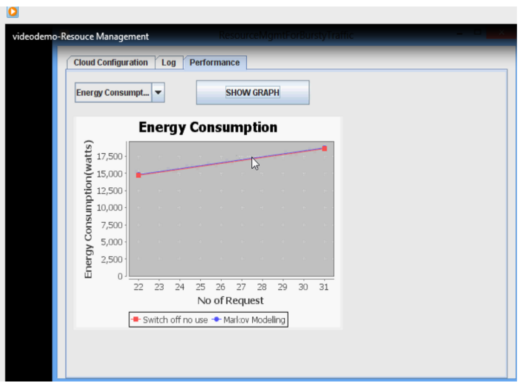

Figure 7. Energy consumption

The energy consumption done by using markov model is very minimal when compared to energy consumption of normal method(As you can see in graph the line is almost overlapped (red line: without using markove model, blue line: with using markov model)).

V. CONCLUSION AND FUTURE WORK

In this paper we have used markov model to handle the miss rate and energy consumption in data center. Using markov model we have reduced the miss rate and energy consumption to an extent while handling the bursty traffic.

Here we collect the simulation data of both the methods i.e. natural bench mark and markov model, using that we plot the graph, from that graph we infer that miss rate and energy consumption has reduced while handling bursty traffic using markov model.

Here one drawback is reduction of energy consumption is very minimal using markov model, this is because one cannot predict the time taken by CPU to handle the bursty traffic and amount of CPU cycle, storage , memory required to handle bursty traffic. So the above drawback can be avoided by building more sophisticated markov model

VI. REFERENCES

[1]. J. KOOMEY, "Growth data centre electric use 2004 to 2011".

[2]. Gatner, Inc 2007 Gatner estimates industries account For two percent of global carbon di oxide emissions.

[3]. M. LIN, A. WIERMAN, L. ANDREW, AND E. THERESKA, "Dynamic rightsizing for power-proportional data centre"

[4]. L. MASTROLEON, N. BAMBOS, C. KOZYRAKIS, AND D. ECONOMOU, "Autonmic power managements for internet server and data center"

[5]. S. ALBERS, "Energy efficient algorithm" , "Communication with ACM"

[6]. A. BELOGLAZOV AND R. BUYYA, "Energy efficient resource management in virtualized cloud data centres"

[7]. R. URGAONKAR, U. KOZAT, K. IGARASHI, AND M. NEELY, "Dynamic resource allocation , power management virtualized data centers"

[8]. U. DREPPER, "costs of virtualizations"

[9]. A. GANDHI, M. HARCHOL-BALTER, AND M. KOZUCH, "sleep state effectiveness in data centres"

[10]. A. GANDHI, M. HARCHOL-BALTER, S. DOROUDI, AND A. SCHELLER-WOLF, "Exact analysis the setup class of Markov model versus recursiveness renewal rewards"

[11]. S. IRANI, R. GUPTA, AND S. SHUKLA, "Competitive analysis dynamic power strategies system multiple power state"

[12]. G. GERONIMO, J. WERNER, R. WEINGARTNER, C. WESTPHALL, AND C. WESTPHALL "Provisioning, resource allocation, and green clouds"

[13]. D. BERTSEKAS, Dynamic Programming and virtualization,

[14]. R. KUMAR, M. LEWIS, AND H. TOPALOGLU, "Dynamic services rates control a single server with Markov module arrivals"

[15]. O. CAPPE, E. MOULINES, AND T. RYDEN, Hidden Markov Model.

[16]. E. BAYRAKTAR AND M. LUDKOVSKI, "Sequential tracking of a hidden Markov model using point observations", Stochastic Process and their Applications.