System processing using Oozie Work flow Engine in Green Cloud Computing

R.Ranjani[1], Dr.S.Babu[2]
Department of Computer Science and Engineering[1, 2]
IFET College of Engineering[1, 2], Villupuram, India
ranjanigrs@gmail.com , babubalaji2k5@gmail.com

Abstract: **Cloud Computing is an evolving area of efficient utilization of computing resources. It has changed the model of storing and managing data for scalable, real time, internet based applications and resources satisfying end users' needs. The area of Green computing is also becoming increasingly important in a world with limited energy resources and an ever-rising demand for more computing power. However, achieving an energy-efficiency control and simultaneously satisfying a performance guarantee have become critical issues for cloud providers. A cost function has been developed in which the costs of power consumption, system congestion and server startup are all taken into consideration. The effect of energy-efficiency controls on response times, operating modes and incurred costs are all demonstrated. The aim of the work is to provide power saving policy for reducing the power consumption in cloud datacenters. The power that produces in idle state of server moves to sleep state of the server. An energy efficient task scheduling algorithm will be developed that can help to minimize the scheduling cost and time and to improve the system performance. An efficient green control (EGC) algorithm is first proposed for solving constrained optimization problems. To address the conflict issue between performances and power-saving, Energy Efficient Task Scheduling algorithm is used to minimize server power consumption in cloud servers.**

***Keywords*: Green computing, Power consumption, Operating modes, Task scheduling, Energy Efficiency.**

I. INTRODUCTION

Cloud Computing is an evolving area of efficient utilization of computing resources. It has changed the model of storing and managing data for scalable, real time, internet based applications and resources satisfying end users' needs. The area of Green computing is also becoming increasingly important in a world with limited energy resources and an ever-rising demand for more computing power. However, achieving an energy-efficiency control and simultaneously satisfying a performance guarantee have become critical issues for cloud providers. In this paper, three power-saving policies are implemented in cloud systems to mitigate server idle power. The challenges of controlling service rates and applying the N-policy to optimize operational cost within a performance guarantee are first studied. A cost function has been developed in which the costs of power consumption, system congestion and server startup are all taken into consideration.

The effect of energy-efficiency controls on response times, operating modes and incurred costs are all demonstrated. Our objectives are to find the optimal service rate and mode-switching restriction, so as to minimize cost within a response time guarantee under varying arrival rates. An efficient green control (EGC) algorithm is first proposed for solving constrained optimization problems and making costs/performances tradeoffs in systems with different power-saving policies.

II.LITERATURE SURVEY

Ahmed Amokrane [1] Cloud computing gives an on-demand computing, storage, and networking resources. However, most cloud providers simply offer virtual machines (VMs) without bandwidth and delay in processing, which may hurt the performance of the deployed services. Recently, some proposals suggested remediating such limitation by offering virtual data centers (VDCs) instead of VMs only. However, they have only considered the case where VDCs are embedded within a single data center. In practice, infrastructure providers should possess the ability to provision requested VDCs across their distributed infrastructure to achieve multiple goals including revenue maximization, operational costs reduction, energy efficiency, and green IT, or to simply satisfy geographic location constraints of the VDCs. In this paper Greenhead, a holistic resource management framework for embedding VDCs across geographically distributed data centers connected through a backbone network. The main objective of Greenhead is to maximize the cloud provider's revenue while ensuring that the infrastructure is as environment-friendly as possible. To evaluate the effectiveness of our proposal, we conducted extensive simulations of four data centers connected through the NSFNet topology. As a result the Greenhead improves requests' acceptance ratio and revenue by up to 40 percent while ensuring high usage of renewable energy and minimal carbon footprint.

Jie Song , Tiantian Li [2] Due to the limited energy, environmental problem and fast growth of computer power consumption, energy-efficient computing has now become a critical and urgent issue. Therefore, an energy-consumption (energy consumed per task) evaluation model is necessary to be established. However, the existing models are almost qualitative rather than quantitative and with poor practicability because the EC measurement of cloud computing systems is rather difficult. An EC model and the corresponding calculation approach for cloud computing systems prove its correctness and accuracy through a series of experiments. Based on this model, we also analyse the EC works of cloud computing systems, list out some EC regularities and propose some EC optimization suggestions. With this model, one can calculate EC only by the values that are easily measured without special hardware, and can explore and evaluate new optimization methods.

Guazzone [3] Cloud computing is growing in popular among computing paradigms for its appealing property of considering "Everything as a Service". The goal of a Cloud infrastructure provider is to improve its profit by reducing the amount of violations of Quality-of-Service (QoS) levels agreed with service providers, and, at the same time, by lowering infrastructure costs. Among these costs, the energy consumption induced by the Cloud infrastructure, for running Cloud services, plays a primary role. Unfortunately, the minimization of QoS violations and, at the same time, the reduction of energy consumption is a challenging problem. A framework to automatically manage computing resources of Cloud infrastructures in order to simultaneously achieve suitable QoS levels and to reduce as much as possible the amount of energy used for providing services. We show, through simulation, that our approach is able to dynamically adapt to time-varying workloads (without any prior knowledge) and to significantly reduce QoS violations and energy consumption with respect to traditional static approaches.

Huang[4] Virtualization is a key technology for resource sharing in IaaS/PaaS cloud infrastructures. One primary issue in virtualization is the virtual machine placement (VMP) problem, which is to choose proper physical machine (PM) to deploy virtual machines (VMs) in runtime. VMP problem with the goal of minimizing the total energy consumption has been studied clearly were, We first present a multi-dimensional space partition model to characterize the resource usage states of PMs. Based on this model, we then propose a virtual machine placement algorithm, which can balance the utilization of multi-dimensional resources, reduces the number of running PMs and thus lowering down the energy consumption. We also evaluate our proposed balanced algorithm via extensive simulations. Simulation results show that our approach can save as much as 15% energy compared to the first fit algorithm over a long run.

III.PROPOSED SYSTEM

The aim of the work is to provide power saving policy for reducing the power consumption in cloud datacentres. The power that produces in idle state of server moves to sleep state of the server. An energy efficient task scheduling algorithm will be developed that can help to minimize the scheduling cost and time and to improve the system performance. To save the power the heartbeat of the idle nodes has been detected in the network. And schedule the work to all the nodes to avoid data traffic in the cluster. The job dispatcher in our designed system is used to identify an arrival job request and forward it to a queue of a corresponding VM manager that can satisfy its QoS levels, meet its target web application or specific requirements. When there has no job in a queue or no job is being processed, a server becomes idle and it remains until a subsequent job has been sent to its processor node. Generally, a server operates alternately between a busy mode and an idle mode for a system with random job arrivals in a cloud environment.

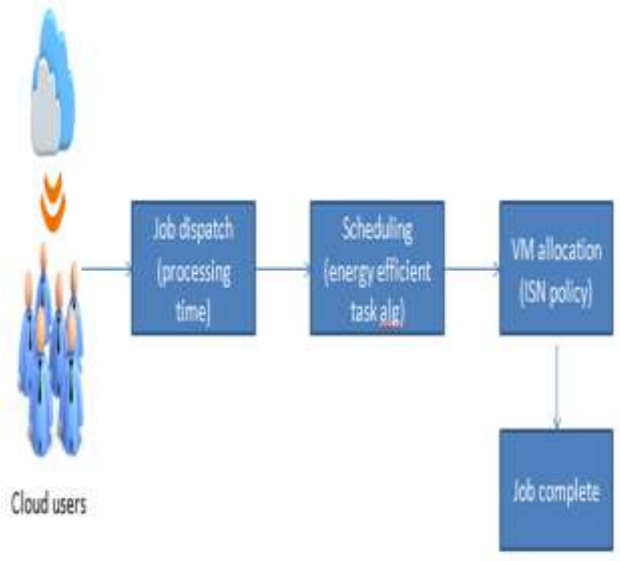
Job dispatch (processing time)
Scheduling (energy efficient task alg)
VM allocation (ISN policy)
Job complete
Cloud users

Fig1: Overall System Architecture

The VM nodes consist of Busy Server, Idle Server, Energy-efficient control in a system with three operating modes m= { Busy, Idle, Sleep}, where a sleep mode would be responsible for saving power consumption. The arrived job is completed by Using ISN, SN and SI policy.

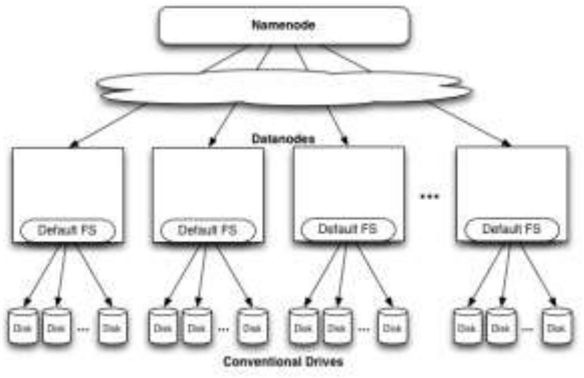

Fig 2: HDFS Architecture

IV.TASK SCHEDULING ALGORITHM

(minimize server power consumption in cloud)

Set of tasks and servers are taken as input. The scheduling of tasks to the servers and the data center server energy consumption is given as output of the algorithm.

The users will request for computing various types of tasks. Each task may fall under a particular task type like reading file contents, updating data, uploading files, downloading software, etc. Based on the type of task selected, the processing time vary. The number of instruction in each task is obtained. Energy slope is calculated for each task of different types in each server with the help of processing time. Energy consumption is calculated by using the number of instructions and the energy slope. Task allocation is done in such a way that most-efficient-server gets the tasks first. Number of active servers among the set of available servers is reduced. The algorithm follows a greedy approach.

Input: Set of tasks and servers.

Output: Scheduling of tasks to servers and data center server energy consumption. Get the number of tasks, number of available servers, task type of each task and the number of instruction of each task.

```
 a=b=1;
for each task x of type i do
for each server k do
        Calculate $\gamma_{i,k}=C_i[pt_{i,k}]^{\alpha_i -1}$
        Calculate server energy consumption
        $E(i,k)=R_{i,k}\,\gamma_{i,k}$
if E(i,k) <= E(a,b) then a=i, b=k
end
end
end
```

Output the task allocation to the servers and the total energy consumption by the servers.

Where, $pt_{i,k}$ is the processing time for task of type "i" at server "k", $R_{i,k}$ is the number of instructions of each task and $\gamma_{i,k}$ is the energy slope of task "i" in server " k".

Using ISN (Idle to Sleep with N policy), allocate the tasks to servers that minimize the energy consumption in datacenter server. To avoid switching too often, a control approach called N policy, Queuing systems with the N policy will turn a server on only when items in a queue is greater than or equal to a predetermined N threshold, instead of activating a power-off server immediately upon an item arrival time.

V.OOZIE WORK FLOW ENGINE FOR APACHE HADOOP

Oozie is a server based Workflow Engine which is used in running workflow jobs with actions that run Hadoop Map/Reduce and Pig jobs. Oozie is a Java Web-Application that runs in a Java servlet-container. For the purposes of Oozie, a workflow is a collection of actions (i.e. Hadoop Map/Reduce jobs, Pig jobs) which controls the dependency DAG (Direct Acyclic Graph). " control dependency" from one action to another means that the second action can't run until the first action has completed. Oozie workflows definitions are written in hPDL (a XML Process Definition Language similar to JBOSS JBPM jPDL). Oozie workflow actions start jobs in remote systems (i.e. Hadoop, Pig). Upon action completion, the remote systems callback Oozie to specify the action completion, at this point Oozie proceeds to the next action in the workflow.

Oozie workflows contains control flow nodes and action nodes. Control flow nodes defines the beginning and the end of a workflow (start, end and fail nodes) and provide a mechanism to control the workflow execution path (decision, fork and join nodes). Action nodes are the mechanism by which a workflow triggers the execution of a computation/processing task. Oozie provides supports for different types of actions: Hadoop map-reduce, Hadoop file system, Pig, SSH, HTTP, eMail and Oozie sub-workflow. Oozie can be extended to support additional type of actions.

Oozie v3 is a server based Bundle Engine that provides a higher-level oozie abstraction that will batch a set of coordinator applications. The user will be able to start/stop/suspend/resume/rerun a set coordinator jobs in the bundle level resulting a better and easy operational control.

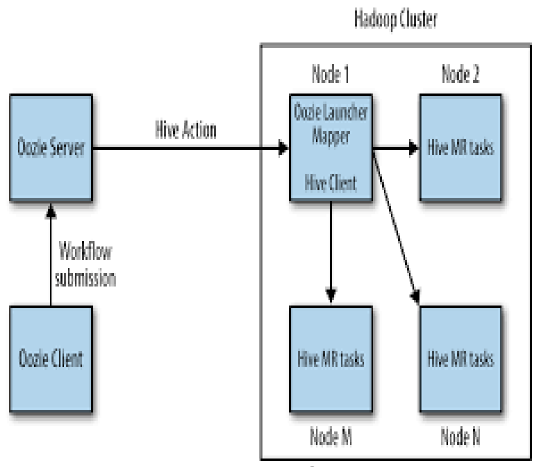

Fig3: Oozie workflow process

Oozie v2 is a server based Coordinator Engine specialized in running workflows based on time and data triggers. It can continuously run workflows based on time (e.g. run it every hour), and data availability (e.g. wait for my input data to exist before running my workflow). Oozie v1 is a server based Workflow Engine specialized in running workflow jobs with actions that execute Hadoop Map/Reduce and Pig jobs. Here we are going to implement oozie v1 to implement map reduce jobs.

VI.CONCLUSION

The growing crisis in power shortages has been brought in a concern in existing and future cloud system designs. To eliminate unnecessary idle power consumption, the power saving policies with different decision processes and mode switching controls has been carried out. In our proposed algorithm cloud providers to optimize the decision-making in service rate and mode switching restriction, has been described so as to minimize the operational cost without violating the SLA constraint. The problem of choosing a suitable policy among diverse power managements to reach a relatively high effectiveness has been examined based on the variations of arrival rates and incurred costs. Experimental results show that a system with the Oozie work flow process can significantly improve the response time in a low arrival rate situation. On the other hand, applying policies can obtain more cost benefits when the startup cost is high. As compared to it, cost efficiency and response time improvement can be improved and verfied.

REFERENCES

[1] A.Amokrane, M.Zhani, R.Langar, R. Boutaba, and G.Pujolle,"Greenhead: Virtual data center embedding across distributed infrastructures," IEEE Trans. Cloud Compute., vol.1, no.1,pp.36-49,Jan.Jun.2013.

[2] J.Song, T.Li, Z. Wang, and Z. Zhu, "Study on energy consumption regularities of cloud computing systems by a novel evaluation model," Computing, vol.95,no.4, pp.269-287,2013.

[3] M. Guazzone, C. Anglano and M. Canonico, "Energy efficient resource management for cloud computing infrastructures," in proc.IEEE Int. Conf. Cloud Compute. Technol. Sci.,2011,pp.424-431.

[4] W.Huang, X. Li, and Z. Qian, "An energy efficient virtual machine placement algorithm with balanced resource utilization," in proc. 7th Int. Conf. Innovative Mobile Internet Serv. Ubiquitous Compute., 2013, pp.313-319.

[5] G.P.Duggan and P.M.Young,"A resource allocation model for energy management systems", in Proc.IEEE Int.Syst.Conf.,2012,pp.1-3.

[6] F.Larumbe, and B.Sanso,"A tabu search algorithm for the location of data centers and software components in green cloud computing networks, "IEEE Trans. Cloud Comput.,vol.1,no.1,pp.22-35,Jan.-Jun.2013.

[7] K.H.Wang and H.M.Huang, "Optimal control of an M/Ek/1 queueing system with a removable service station, "J.Oper.Res.Soc.,"vol.46,pp.1014-1022, 1995.

[8] M.Zhang, and Z.Hou, "M/G/1 queue with single working vacation. "J. Appl. Math. Compute, vol.39,no .1-2, pp.221-234, 2012.

[9] G.Wang and T.E. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in Proc. IEEE Proc, INFO-COM, 2010, pp.1-9.

[10] R.Ranjan, L. Zhao, X. Wu, A. Quiroz, and M. Parashar, "peer-to-peer cloud provisioning: Service discovery and load-balancing," in Cloud Computing. London, U.K: Spinger, 2010, pp.195-217.