Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 78 (2016) 457 – 463

Procedia Computer Science

International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA

Power and Fault Aware Reliable Resource Allocation for Cloud Infrastructure

Punit Gupta[a] , S. P. Ghrera[b]

[a,b]*Department of Computer Science Engineering, Jaypee University of Information Technology Himachal Pradesh, India*

Abstract

Cloud computing is now trending and more popular in these days for the computation and adopted by many companies like google, amazon, Microsoft etc. As the cloud size increases with increase in number of data center power consumption over a data center increases. As number of request over the data center increase with increase in load and power consumption of the data center. So the requests need to be balanced in such manner which having more effective strategy for resources utilization, request failure and improved power consumption. Cloud computing made it more complicated with respective to requests type that may increase or decrease power consumption. A recent survey on cloud computation shows that the power consumption of a server, increasing in a linear way due to utilization of resource (processors) resulting in request failure at datacenters. Request balancing in such manner without having knowledge of load over server maximize resource utilization but also increasing power consumption at server. So to overcome these issues in cloud Infrastructure as a service (IaaS), we have proposing a fault and power aware scheduling algorithm to minimize the power consumption, request failure and cost over a data center. Proposed algorithm has proven to have better performance in term of load and power efficiency as compared to previously proposed load balancing algorithm for cloud IaaS.

Peer-review under responsibility of organizing committee of the ICISP2015

Keywords: Cloud computing; Power aware computing; Resource Utilization; Failure probability; Cloud Infrastructure as a service.

1. Introduction

Cloud computing is more popular and promising epitome for both consumer and provider in different field like engineering mathematics and highly used in business industries in homogeneous and heterogeneous way. Request from different kind of area are served by data centers in cloud environment also increase power consumption.

Peer-review under responsibility of organizing committee of the ICISP2015
doi:10.1016/j.procs.2016.02.088

However, to keep computing large scale requests on data centers required huge amount of power, which leads to high power consumption. Request type also affect the services i.e., private or public requests. As per survey in 2006, data center consumed around 4.5 billion kWh, which is equal to 1.5% of total power consumed by USA, and increasing 18% yearly[1]. In general Cloud Computing deals with various power issues listed as follows: 1) as cloud computing adopted by industry and number of user also rapidly growing with number of data centers with increasing power consumption. 2) Load distributed among data centers without having information of power consumed by them compare to usually consumed power in under loaded datacenter.3) Current load balancing algorithms focus on load balance when request load increases but not as consumed power increases. 4) High loaded data centers consume more power to compute requests and may be due to high load theses data center slow down which is not good for user as well cloud provider.5) Some data centers having less load compare to highly loaded data center and they are under maximum load but high loaded data center computing them with high energy consumption.6) Some request are need to be computed within it's time line but due to high load they may miss their deadline which is not appropriate to user and will be a critical issue. Requests having less completion time line can be compute on least or power consuming data centers. 7) As per recent study[4-7], utilization of data centers is major problem because 60% data centers are idle and most of 20% data centers are fully utilize and wastes of power respectively. This show poor utilization of resources as well power but this shows importance of new approach that have sufficient strategy to minimize wastes of power as much as. 8) Request type make big difference of services, if requests are private then service provider has to take it as high priority and low priority to public.

This paper focus on the resource allocation algorithm that emphasis hardware capability and functionality of cloud computation. Proposed algorithm lead computation in efficient energy consumption in new directions and improved utilization of all resources accordingly. There are many factors of cloud computing which are untouched and can give best results and optimize the computation schema, power consumption one of them. Power or energy consumed by a data center depends on configuration of server hardware, high configuration of hardware will defiantly consume much power as compared to less configured. Data centers are processing the requests which are scheduled by scheduler using a scheduling algorithm and at the end of data center there is one more algorithm applied to balance the load over data center and optimistically increase the data center performance, but there should be policy to make priority factor to check whether request is public or private. There is always a fixed capacity of data center to serve request beyond that data center need extra resource to do extra computation with the same hardware configuration, here is chance of task failure or some interruption during the process due to limited resource. If task is private, it has to be done without any problem and public request may server later or wait for a while. This paper proposed an approach to avoid such situation with a cost efficient manner as well continue task computing without any interruption. Data center applies an algorithm to allocate resource to complete request with some configuration which makes it computable as soon as possible to serve request and response them, but some time due to more than its request server capability request comes and queued and data center compute the task with its maximum serve capability and finish the computation and it's increase the power consumption and there are chances of system failure which is not appropriate and cause loss of money. This scenario may be coming into existence and decrease the productivity and down the system performance. There are many algorithms to find least loaded server, which can provide service efficiently and economically.

Private and public cloud computation are basis of data which is going to compute, public task can wait and can wait for some time and compute later, but the same thing cannot be applied to private, somehow it has to be done. If a private task has to be complete within the time-line, then scheduler has to be make sure it will be finished within time line, if lost its time line, then it's going to be a big problem to server provider and customer.

This paper proposes an dynamic way to minimize the computation and maximize utilization of resource by allocation resource to request by getting energy efficiency, fault tolerant and factors which make it more efficient and reliable computation over cloud environment. Power consumption based scheduling lead to efficient computation and increase computation of data center economical.

2. Related Work

Cloud Computation is a new domain and need more research to make it more reliable, in order to have a better user experience in term of task computation. We have proposed a scheduling algorithm which is based on energy consumption and reliability of task to complete task scheduling in a power efficient manner and better utilization.

Many researchers have done research and introduce us some beneficial and optical scheduling algorithm. Brown, Richard[1]. Proposed Min-min algorithm, in this choose least the completion time of task and schedule to serve accordingly. In this paper they proposed load balance Min-min algorithm which having basic properties of Min-min algorithm and consider minimizing completion of all request. In this proposed three level service model used.

1. Request manager- To take request and forward to Service managers.
2. Service manager- various manger works or task and dispatch them to respective service node.
3. Service Node- Service node provide service to request which came to request mode

They have merged two approaches (OLB Opportunistic load balancing and load balance min-min) scheduling algorithms in this model. The main focus of combined approaches is to distribute the request or dispatched task basis of their completion time to suitable service node via an agent. This approach not saying about main system, suppose if request are somehow moving or scheduled in the same server and due to lots of load sever need more power to complete these request and more physical heat will generate and to stop heating system need an external cooling system which also lead to extra power source and one more important thing is due to overheating system performance slow down The same way Shu-Ching et al.[2] proposed and another algorithm for task scheduling, this paper proposed VM resource allocation basis on genetic algorithm to avoid dynamic VM migration to completion of request. They have proposed a strategy to share or allow resource equally to VM so it can work fast and minimize response time to subscribe. They also proposed hotspot memory (virtual memory) assignment and dispose that after completion of request via remapping of VM migration. Here VMware distribution tool is used to schedule computation work in a virtual environment. As genetic algorithm characteristics is to find best, fittest VM in terms of Cloud computation. This paper check fitness of each VM and schedule task accordingly. When creating a VM a process executes to create that and increase process work that also lead to more process and increase energy consumption.

Hu, Jinhua et al.[3] Proposed another scheduling algorithm, this paper proposed an approach for collective collaborative computing on trust model. The trust value taking as a factor for task scheduling, trust value mutually took from consumers as well service provider, which make it fail free execution environment. Here they have proposed a mathematical equation to calculate the Reputation point which enhances the reputation of VM in terms of fast execution and type of task. If the reputation of VM is high them more task allocation will be happening to that VM. To calculate Reputation many factors have to consider which also reflect QoS of cloud computing. This paper also proposed a way to serve request reliability, as well trust management with a reputation of VM factor which are lead to trustworthy. Trust has calculated by a mathematical equation and schedule accordingly.

Hu, Jinhua et al.[4] proposed a live VM migration algorithm, this paper proposed a method for VM live migration with various resource reservation system. VM migration is taking place on the basis of source machine load, if the load is high then it can wear, during execution of the request it migrates the VM to another server or data centers to complete the task without interruption for better performance. Resource reservation done both sides, i.e., Source machine and target machine as well will in such manner CAP (maximum availability of CPU) allocate them and adjust memory resource dynamically. At the end of target machine, they properties time bound program which will keep monitoring for cup resource utilization. Memory Reservation done by allocating crating certain number of VM and when the migration process comes into existence these VM got shut down to evacuate the space to migrate VM. Sometime it may be possible that target machine not having enough space to migrate in such condition that physical machine should remove from candidate machine for migration and which physical machine having the capability or enough space will lead to migrate VM. This paper implemented and simulated using Xen Virtualization.

Barroso et al.[5] This paper proposed an algorithm, dynamic and integrated resource scheduling algorithm for Cloud Data center which balance load between servers in overall run time of request, here they are migrating an application from one data center to another without interruption. Here they are introducing some measurement to ensure load balancing. They have given a mathematical reputation to calculate imbalance load to calculate average utilization to its threshold value to balance load. To implement DAIRS they have used physical server with physical cluster and Virtual servers with virtual cluster. Application migration saves time instead of migrating whole VM data. Barroso et al.[6] - proposed a most known base scheduling algorithm ACO (ant colony optimization) they

proposed ant colony optimization algorithm to load balance by distributing request in a cloud computing environment. This paper proposed LBACO with dynamic load balancing strategy to distribute load among the node.

The problem with traditional ACO in cloud is that it's a schedule task to most frequent (high pheromone intensity) node, if what if node is bearing heavy load in such situation may create a problem of overhead. This paper proposed and LBACO algorithm to reduce such problem. In this algorithm decrease the time of computation and monitor load on each VM with tracking previous scheduling. Xiaobo et al.[7] proposed and Real-time VM provisioning model which is based on energy models which follow a Min-Price RT-VM Provisioning to allocate VM. Above proposed algorithms are either general scheduling algorithms or proposing a strategy to improve resource utilization based on power, but what if we talk about cloud then we have to make sure private requests must be finished within time-line and requests to server first or same time but in which manner. Existing algorithm consider system as non faulty, but in real work faults occur at every data center.

In next section we have proposed an algorithm which will deal with such problem in a faulty system and requests have to be completed within less power and with higher power efficiency.

3. Proposed Model

In above section existing proposed algorithms are either talk about task scheduling or resource utilization and some of them talk about task or VM migrating to fulfil requests but power play a vital role and can't be ignore. The problem with these algorithms is that these algorithms are proposing simple task scheduling based on power or resource utilization to complete the task without any problem and maintain quality of service at data center. Existing algorithms are not fault tolerant and only take load over the data center into consideration which is in sufficient to maintain quality of service to the user. So to overcome these issues a fault and power aware resource allocation algorithm is been proposed. Proposed algorithm uses linear power model to get power efficiency of data center. On the other hand, request failure over a data center may occur randomly due to storage or network failure. Based on power efficiency and fault over a d data center, we have proposed a VM allocation policy to minimize the power consumption of the system and reduce probability of request failure.

Then sort them based on fitness value based on power efficiency and failure probability, in such manner to distribute requests to high power efficiency data center with least probability of failure.

Parameters to evaluate fitness value:
PDi: Data center i.
PEi : power efficiency of ith host in a data center .
Ui : Current Utilization of ith host in a data center.
FRi : Fault rate that is the number of request failed due to system failure over time t.
FPi : Failure probability over a Host i.
Fi : Fitness value of ith host.

By Applying liner power utilization of PEi can be calculated.

$$PEi = LineaPower\left(\frac{(\text{Pmax - Pmin})* \text{Ui}}{100}\right) \quad (1)$$

Where Pmax & Pmin = maximum and minimum power consumed by PDi respectively.

Utilization of Data Center can be calculated by

$$Ui = \left(\frac{(\text{Total\_MIPS} - \text{Allocated\_MIPS})}{\text{Total\_MPIS}} \right) \tag{2}$$

Since faults over a data center are random in nature and follows Poisson distribution, which over a period of time t and t+ Δ T can be defined as :

$$\text{FPi}(t \le T \le t + \Delta T \mid T > t) = \frac{\exp(-\lambda t) - \exp(\lambda(t + \Delta T))}{\exp(-\lambda t)}$$

$$\text{FPi}(t) = \left(1 - \exp(-\lambda \Delta t)\right) \tag{3}$$

//Fitness value

$$\text{Fi} = \text{PEi} + \text{FPi}(t) \tag{4}$$

As in above formula of Ui is calculated by getting total utilization from total MIPS allocated by data center PDi. Once calculate utilization of data centers then calculating power consumed by these data centers and using linear power efficiency formula as above. To get power efficiency of data centers as well to allocation resources for requests is done with below steps. On the other end we need to calculate the fault rate over a data center PDi, which depend on the number of request failed on a data center over a period of time 't'. Since fault is random in nature so the probability of failure can be found using poison distribution as shown in equation 3. Equation 3 defines the probability of failure at data center PDi. Base on the above defined parameters fitness value of each datacenter is calculated, as shown in equation 4 which is sum of power efficiency and probability of failure in fraction which range from 0 to 1.

Steps for Proposed algorithm

Resource Allocation ALgorithm

Algorithm:-PFARA(DataCenters List PD and Q_{length})

Input : PD and Q_{length} ,
1. PD← DataCenterList ;
2. i ←No. of Data Centers;
3. Q_{length}← current queue size;
4. PEi← Power Efficiency of DataCenter PD_i;
5. FPi← Failure Probability of DataCenter PD_i;
6. If(Q_{length} !=0)
7. Allocate_Resource(Req); // processing the client request.
8. End

Fig. 1 Proposed Algorithm initialization

```
1.  Allocate_Resource(Request){
2.  Host_list=gethostlist();
3.  Sorted_host = Sort_Fitness(Host_list);// based on resource
4.  For(Host h: Sorted_host)
5.  {  Fi=PEi + FPi (t);
6.     If( h.isSuitable()  &&  h.fitness_Least() )
7.       selected_host= h;
8.  }
9.  allocate(request,h);
10. Else
11. printf("cannot find suitable server")
12. }
```

Fig. 2 Proposed Algorithm resource allocation

4. Experimental Results

For simulation we CloudSim 3.0 power module is used. CloudSim 3.0 provides cloud simulation and predefined power model simulation. We have simulated proposed power and fault aware VM allocation algorithm in Cloudsim power package. Proposed algorithm is being tested over various test cases with 3 servers S1-S5 and linear power model. Power model directly depend on utilization of servers.

Testing of proposed algorithm is done with basic DVFS (Dynamic voltage and frequency scaling) based scheduling which is a power management in servers. Testing is done for 600, 800, 1200, 1400 requests. Server configuration is as follows.

Table 1

Server	RAM(Mb)	MIPS	Storage (Gb)	core	PE	HOST
S1	2000	10000	100000	4	10	2
S2	2000	10000	100000	6	10	2
S3	2000	10000	100000	6	10	2
S4	2000	10000	100000	6	10	2
S5	2000	10000	100000	6	10	2

Table 2

	Power consumed in KW	
Requests	Proposed	DVFS
200	22	29
400	34	41
600	48	59
800	60	71
1200	83	96

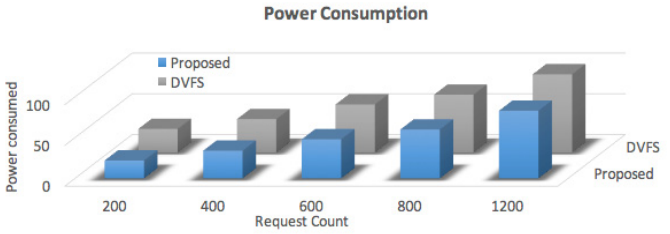

Fig. 3 Power consumption

Figure 3 shows the improvement in power consumption by proposed algorithm over DVFS.

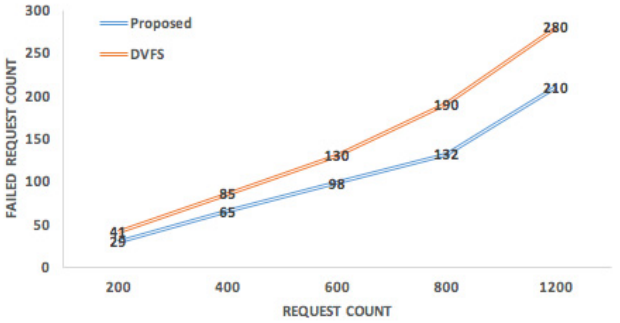

Fig. 4 Request Failed

Figure 4 shows the improvement in number of request failed by proposed algorithm over DVFS in various test cases.

5. Conclusion

From experimental result section, it is clear that PFARA is giving high performance as compare to previous proposed algorithm for both private as well as for public request. The main idea of this algorithm in cloud computing is to complete the requests as possible as minimum power and full utilization of resource, proposed algorithm shown that it can maximize throughput and minimize the computation power. This strategy has proven that both requests are completed within time with utilization of resources

References

1. Brown, Richard. "Report to congress on server and data center energy efficiency: Public law 109-431." Lawrence Berkeley National Laboratory (2008).
2. Wang, Shu-Ching, Kuo-Qin Yan, Wen-Pin Liao, and Shun-Sheng Wang. "Towards a load balancing in a three-level cloud computing network." In*Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 1, pp. 108-113. IEEE, 2010.
3. Hu, Jinhua, Jianhua Gu, Guofei Sun, and Tianhai Zhao. "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment." In *Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on*, pp. 89-96. IEEE, 2010.
4. Hu, Jinhua, Jianhua Gu, Guofei Sun, and Tianhai Zhao. "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment." In *Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on*, pp. 89-96. IEEE, 2010.
5. Barroso, Luiz André, and Urs Holzle. "The case for energy-proportional computing." *Computer* 12 (2007): 33-37.
6. Bohrer, Pat, Elmootazbellah N. Elnozahy, Tom Keller, Michael Kistler, Charles Lefurgy, Chandler McDowell, and Ram Rajamony. "The case for power management in web servers." In *Power aware computing*, pp. 261-289. Springer US, 2002.
7. Fan, Xiaobo, Wolf-Dietrich Weber, and Luiz Andre Barroso. "Power provisioning for a warehouse-sized computer." In ACM SIGARCH Computer Architecture News, vol. 35, no. 2, pp. 13-23. ACM, 2007.
8. Lefurgy, Charles, Xiaorui Wang, and Malcolm Ware. "Server-level power control." In Autonomic Computing, 2007. ICAC'07. Fourth International Conference on, pp. 4-4. IEEE, 2007.
9. Mehta, Hemant Kumar, Manohar Chandwani, and Priyesh Kanungo. "On trust management and reliability of distributed scheduling algorithms." In *Advanced Computing (ICoAC), 2010 Second International Conference on*, pp. 46-50. IEEE, 2010.
10. Ye, Kejiang, Xiaohong Jiang, Dawei Huang, Jianhai Chen, and Bei Wang. "Live migration of multiple virtual machines with resource reservation in cloud computing environments." In *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, pp. 267-274. IEEE, 2011.
11. Tian, Wenhong, Yong Zhao, Yuanliang Zhong, Minxian Xu, and Chen Jing. "A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters." In *Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on*, pp. 311-315. IEEE, 2011.
12. Li, Kun, Gaochao Xu, Guangyu Zhao, Yushuang Dong, and Dan Wang. "Cloud task scheduling based on load balancing ant colony optimization." In *Chinagrid Conference (ChinaGrid), 2011 Sixth Annual*, pp. 3-9. IEEE, 2011.
13. Kim, Kyong Hoon, Anton Beloglazov, and Rajkumar Buyya. "Power-aware provisioning of cloud resources for real-time services." In *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*, p. 1. ACM, 2009.