International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

Scheduling Algorithm with Load Balancing in Cloud Computing

Ashish Kumar Singh[1], Sandeep Sahu[2], Mangal Nath Tiwari[3], R. K. Katare[4]

[1,2]Department of Computer Science, SRIT, Jabalpur, MP, India

[3,4]Department of Computer Science, APS University, Rewa, MP, India

Abstract: ***Cloud Computing is an emerging field and prefer by many one at current but it's craze is lot more depend on its performance which in turn is too much depend on the effective scheduling algorithm and load balancing . In this paper we address this issue and propose an algorithm for private cloud which has high throughput and for public cloud which address the issue of environment consciousness also with performance. To improve the throughput in private cloud SJF is used for scheduling and to overcome form the problem of starvation we use bounded waiting. For load balancing we monitor the load and dispatch the job to the least loaded VM. To gain more benefit and to have opportunity for future enhancement in public cloud environment consciousness is the key factor and for better performance and load balancing also desired. While load balancing improve the performance, the environment consciousness increase the profit of cloud providers.***

Keywords: Private cloud, public cloud, virtualization, load balancing, bounded waiting.

1. Introduction

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing has become a popular buzzword; it has been widely used to refer to different technologies, services, and concepts.

1.1 Types of Cloud Computing

a) Public cloud: Public cloud applications, storage, and other resources are made available to the general public by a service provider. These services are free or offered on a pay-per-use model. Generally, public cloud service providers like Amazon AWS, Microsoft and Google own and operate the infrastructure and offer access only via Internet (direct connectivity is not offered).
b) Community cloud: Community cloud shares infrastructure between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally. The costs are spread over fewer users than a public cloud (but more than a private cloud), so only some of the cost savings potential of cloud computing are realized.
c) Hybrid cloud: Hybrid cloud is a composition of two or more clouds (private, community or public) that remain unique entities but are bound together, offering the benefits of multiple deployment models. By utilizing "hybrid cloud" architecture, companies and individuals are able to obtain degrees of fault tolerance combined with locally immediate usability without dependency on internet connectivity. Hybrid cloud architecture requires both on-premises resources and off-site (remote) server-based cloud infrastructure. Hybrid clouds lack the flexibility, security and certainty of in-house applications. Hybrid cloud provides the flexibility of in house applications with the fault tolerance and scalability of cloud based services.
d) Private cloud: Private cloud is cloud infrastructure operated solely for a single organization, whether managed internally or by a third-party and hosted internally or externally. Undertaking a private cloud project requires a significant level and degree of engagement to virtualize the business environment, and it will require the organization to reevaluate decisions about existing resources. When it is done right, it can have a positive impact on a business, but every one of the steps in the project raises security issues that must be addressed in order to avoid serious vulnerabilities.

1.2 Virtualization

It is a very useful concept in context of cloud systems. Virtualization means "something which isn't real", but gives all the facilities of a real. It is the software implementation of a computer which will execute different programs like a real machine. Virtualization is related to cloud, because using virtualization an end user can use different services of a cloud. The remote datacenter will provide different services in a full or partial virtualized manner.

1.3 Motivation

1. The use of cloud computing are increasing day by day because now people the use of smart phone and tablet PC are becoming common . People want same computing

capability as they got on their stationary Desktop PC are Laptop computer. But these are small and have less computing facility so they can't run heavy software. Cloud computing are used here to run the desired software on the mobile as a service over the Internet.

2. The needs of organization are very dynamic today and if they go to purchase all things like infrastructure, software and platform they have to spend a lot of money every time and also spend time for the setup. They have to do a lot of documentation for maintaining the list of vendor, license key etc. With cloud environment organizations have to pay for the resource they want to use and they have to pay for according to approximately their usage time. They not have to spend almost any time for the setup.
3. With cloud computing organization have to worry about the maintenance because it is done by the cloud provider. So businessman can think more about their business.

RGPV portal and MPOnline Portal , IRCTC all are the example of cloud computing which shows how the work become easier for both manager and customer with the use of cloud computing. But Cloud Computing performance have depend a lot more on the scheduling algorithm and proper load balancing algorithm. Scheduling algorithm will make such a sequence of process that throughputs are increased and load balancing algorithm divide the load properly between all available resources. Cloud Computing is a parallel processing model where these issue is of vital importance so they also have importance. As cloud is a pay-go-model, the business performance needs to be accelerated which is a challenging issue in the domain. There is certain important factor like;

- Job Scheduling
- Load Balancing
- Resource Allocation

So, we have chosen this topic to find a better strategy to improve the performance in cloud computing environment.

1.4 Paper Organization

In section two we review the previous work of various author of same field. We discuss about their work and also discuss about their limitations. In section three we propose our algorithm which is two parts. In First part we discuss algorithm of private cloud and in second part we discuss algorithm for public cloud. We also evaluate the performance of algorithm by calculating their time complexity. We also put the result of our work. In section four we conclude our work and discuss about the future work.

2. Review of Prior Works

Author of [1] suggests an algorithm which covers environment conscious issue for scheduling of HPC applications on distributed cloud centers. Author study shows that at present the carbon emission of ICT industry is becoming equal to the carbon emission of aviation industry. Now some of the government imposing a carbon emission limit over the ICT industry. So if cloud providers not consider this issue they are not able to extend their infrastructure in future. For environment conscious the author consider the carbon emission rate of data center. Carbon emission rate of different data center is different so scheduling algorithm schedule the job to a data center which have minimum carbon emission rate. Author also covers the issue of cost which will maximize the profit of cloud provider. For cost he considers execution price, elasticity price and data transfer price. On base of these information authors suggest algorithm which will find out data center which maximize the profit from the data center pair of minimum carbon rate.

Author's algorithm is of HPC application where applications have the need of more than one VM. After selection of a data center scheduling algorithm select the VM according to the requirement. Author's solution is good for environment conscious but he not cover the issue of load balancing to choose the VM form a data center which is also a major issue at present time.

Author of [2] has study the factor that influence the job rejection in the cloud environment. He shows the comparison of SJF and R-R scheduling algorithm in peak hour means when no of arrival of job is very high. Author suggests R-R scheduling for scheduling and SJF for load balancing and process migration to avoid deadlocks. Author shows in the result that in peak hour SJF has better performance in job rejection issue. SJF has the problem of starvation so long job has very high turnaround time. His study shows that in the case of cloud computing the number of job rejection should be less because cloud is pay-go model so if customer's job is rejected then the feature of cloud computing would not attract the customer .

Author actually shows a comparisons between SJF and RR scheduling but he not suggest any way of changing the algorithm from SJF to RR and form RR to SJF author only says about the migration of processes for load balancing but how migration will occur and what is the method of knowing overloaded VM . Author has study the factor that influence the job rejection in cloud environment and he suggest the R-R algorithm for scheduling and SJF scheduling for load balancing but R-R scheduling has a lot of overhead of preemption and SJS has starvation problem.

Author of [6] propose a solution for managing large image collections. Author presents a cloud computing service and its application for the storage and analysis of very –large image. His solution allows that an input image can be divided into different sub-images that can be stored and processed separately by different agents in the system, facilitating processing very-large images in a parallel manner. His purpose is to create a cloud computing service capable of storing and analyzing very- large image datasets.

Author develop cloud computing service prototype for storing and analyzing images. Adequate parallelism and workload balancing of our distributed system is crucial feature to ensure an improved performance. Author actually study real life application of very large image datasets and parallelism will really improves the performance such application. His solution divide large image into sub-images and improve the performance by parallelism. Author not covers issue of environment and cost and also of load balancing.

Author of [8] suggest the algebraic scheduling of the processes because he found that some different processes have different need for the execution so he suggest the algorithm which show the process resource demand in the form of utility function . He suggest that desired resource demand should be in the form soft constraint means it is not necessary that process can execute with the desired resource. As cloud resources and applications grow more heterogeneous, allocating the right resources to different tenants' activities increasingly depends upon understanding tradeoffs regarding their individual behaviors. One may require a specific amount of RAM, another may benefit from a GPU, and a third may benefit from executing on the same rack as a fourth. Author modify the existing approach where resource consumers has to specify zero or more hard constraints with each request, based on some predetermined attribute schema understood by the cluster scheduler . Such constraints could serve as a filter on the set of machines, enabling identification of the subset that is suitable for the corresponding request. But, this approach ignores an important issue: in many cases, the desired machine characteristics provide benefit but are not mandatory.

This paper proposes a specific approach for accommodating soft constraints, as well as hard constraints and general machine heterogeneity. In this model, each job submitted for processing is accompanied by a resource request, which is expressed as utility functions in the form of algebraic expressions indicating what benefit would be realized if particular resources were assigned to it. Author suggests algebraic scheduling because of heterogeneity in cloud environment. He suggests that process has to express their resource requirement in soft constraint and algebraic scheduling decide that process has to be executed on which VM. In his scheduling utility function express all option in sequence which will have a lot of overhead.

3. Proposed work

As we have read that there are four types of deployment model in cloud computing.

1) Private Cloud
2) Public Cloud
3) Hybrid Cloud
4) Community Cloud

In our paper we propose algorithm of following two types of model 1) Private Cloud, 2) Public Cloud.

3.1 Private Cloud Algorithm

The common differences between distributed computing and cloud computing are-

1. As we know that distributed system is transparent means user is working on the computer as he have all the resources on that computer but he actually have only little part of it . In cloud computing user have nothing on his machine and he has to access everything in non-transparent manner.
2. For distributed system we not have to connect ourselves to the network or interne but for cloud computing we have to connect ourselves to the network (for private cloud) and to the internet (for pubic cloud) to access the services.

3.2 Important Point of Scheduling In Private Cloud

1) Jobs are admitted only of the network so there are less number of jobs compares to public cloud so some overhead permissible in scheduling.
2) Higher throughput is primary requirement because if this is not the case then it is beneficial to switch to higher hardware and software cost means not to use cloud computing.
3) SJF scheduling has the higher throughput so this is better option of scheduling in private cloud.
4) With SJF some jobs may suffer due large burst time this is called starvation. To overcome form this problem bounded waiting must be there which introduce a tag with every arrived and decreased automatically with every new arrival of process and when it will reach to zero then that job must be executed
5) In cloud computing there is number of processor is available for the execution so a proper load balancing is also mandatory for this there should be a common queue for all jobs arrived from any user and cloud manager can allocate the job to the idle VM (Virtual Machine).
6) On summarizing we can say that :
   a) On arrival every process is kept on a common queue this queue is maintained by the cloud manager which is running on the server machine which have the control over all resources.
   b) Cloud manager check for the idle VM and any VM send a signal that it has finished the execution of current allocated job then cloud manger has to select a process for execution.
   c) Cloud manage first check that is there is any job with
   d) Zero tag which is assigned to the job on its arrival for ensuring bounded waiting if there is any process then that will be selected and dispatched to the least loaded Virtual Machine (VM).
   e) If there is no process with zero tag then cloud manager select the process with least burst time and allocate that job to the VM.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

3.3 Proposed Algorithm for Private Cloud

Database of VM is maintained at cloud manager which have 4 fields.

a) VM id
b) Num field indicate the number of process it can process.
c) p_cur Number of processes currently allocated to the VM.
d) load_per indicate load percent on the VM
On PCB an extra tag field is used for the bound waiting. Ready queue is maintained by cloud manager which contain the list of all arrived jobs. Number of process is indicating by p_num.

3.4 Algorithm

Step – I → initially all VM have the zero tag value in the database of VM and a value in Num field.
Step – II→ While [queue]! = NULL
Step – III→ if tag [process] ==0 then, step –V else step –IV
Step – IV→ Select min burst [process] form ready queue
Step – V → size [vm_table] = n
Step – V → Select min load_per[VM] (least loaded and if More than one then least hop time and VM capacity indicate by the num field also more than the request size) VM from the VM pool and p_cur[VM] = p_cur[VM] +1;
load_per[VM] = p_cur[VM]*100/num[VM]
Step – VI→ remove the selected process form queue and Dispatch it to selected VM p_num= p_num -1;
Step – VII→if new process is arrived add to the tail of the Queue and p_num= p_num + 1;
Step – IX →While i<p_num
Tag [process]=tag[process] -1;
If tag [process] =0 then move this process at the head of the queue
Repeat
Step – X → goto step – II
Step – XI →Stop

3.5 Time Complexity of Proposed Algorithm

Let n processes arrives in unit time then n time tag[process] of ready queue process is updated until it will reach to zero let bound waiting tag is m which is less than n and let number of existing process in the ready queue is p and number of process leaves the ready queue is r then . Let number of VM is s.

Time required to update tag [process] = p (Number of Existing process in the queue) + n (Number of processes in the queue) - r (Number of processes leaves the queue)

Time required to find min. burst process = [p (Existing process in ready queue) + n (process arrived in the ready queue) – r (process leave the queue)] * r

Time required to find the least loaded VM = s

Time required to update the load_per[VM] = r (Number of process leaves the ready queue)

Total time = p + n –r + (p + n – r) *r + s + r
= p + n + p*r + n*r – r^2 + s
= p (r+1) + n(r + 1) - r2 + s
= (p + n) (r+1) - r^2 + s; n & r >> s
So Total time = (p + n) (r + 1) - r^2
p is approximately equal to n
Then total time = 2n (r+1) - r^2
Case – I n >> r

It means that number of job arrived is much more than number of job leaved then

Total time = O (n^2)
Case – II n is approximately equal to r

It means that number of job arrived is approximately equal to number of job leaved

Then total time = 2r (r+1) – r^2
= 2 r^2 + 2r – r^2
Total time = O (r^2)

3.6 Proposed Algorithm for Public Cloud

These are some important point of scheduling in public cloud:

1) Cloud computing is pay-go model so number of job rejected must be less so scheduling must consider this issue.
2) Cloud provider has to gain more profit so scheduling algorithm must increase the profit of cloud provider.
3) Algorithm must be environment conscious because Carbon emission rate of ICT (Information and Communication Technology) industry is now approximately equal to the aviation industry so government impose carbon emission limit. If scheduling algorithm not cover this issue then cloud provider are not able expand their infrastructure.
4) Environment conscious also provide the benefit of point (1) and (2) because if cloud provide has more infrastructure then number of job rejection is also low and to meet the more demand of cloud user cloud provider has to increase infrastructure which in turn also increase their profit.
5) Load balancing must be there so process must be migrated form over loaded VM to less loaded VM within data centre. This thing will not done continuously mean it will done on periodic base so it will not increase more overhead. Cloud manager periodically monitors the status of the VMs for the distribution of the load, if an overloaded VM is found then the cloud manage migrates the load of the overloaded VM to the underutilized VM.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

3.7 Algorithm

A user submit his requirement for an application j in the form of a tuple (dj, eji , (DT)j) where dj is the deadline to complete application j, eji is the application execution time on the data center i. , (DT)j is the size of data to be transferred.

Here $(CO_2E)_{ij} = r_i^{CO2} \times E_{ij}$ Where r_i^{CO2} is the carbon emission rate of data center i and (Profit)ij = (ProfitExec)ij + (ProfData)ij means (ProfitExec)ij = $e_{ji}p^c - p_i^e \times E_{ij}$

Here p^c CPU execution price fot processing time p_i^e is the electricity price (ProfData) ij = (DT) j x ($p^{DTU} - p_i^{DT}$)

Here p^{DTU} is the dara transfer price for the upload/download p_i^{DT} is the data transfer price for upload/download Cloud provider has to pay data center I the energy cost and data transfer cost depending on its electricity price p_i^e and data transfer price for p_i^{DT} upload/download. Cloud provider then charges foxed price to the user for executing his application based on the CPU execution price p^c and data transfer price p_i^{DT} for the processing time and upload/download respectively.

Step – I → For each application in the list of application to be mapped find the datacenter of which the carbon emission is the minimum means minimum (CO2E) ij among all the data centers which can complete the application by its deadline.
Step – II → Among all the application – data center pairs found in Step – I find the pair that results in the maximum profit means maximum (Prof)ij.
Step – III → once a data center a selected then it will put in the queue of of request. Cloud manager in the data center maintain a data structure comprising of the Job ID, VM ID and VM Status. Cloud manager parses the data structure for allocation to identify the least utilized VM. If availability of VMs is more than one then the VM with least hop time is considered.
Step – IV → Cloud manager updates the data structure automatically after allocation.
Step – V → Cloud manager periodically monitors the status of the VMs for the distribution of the load, if an overloaded VM is found then the cloud manager migrates the load of overloaded VM to the underutilized VM. If more than one is available then VM with least hop time is considered.
Step – VI → Cloud manager updates the data structure by modifying the entries.
Step – VII → On arrival of next process go to Step – I
Step – VIII → Stop

3.8 Time Complexity of Proposed Algorithm

Let n processes arrives in unit time then there are d data center and on average there are v number of VM in data centers.
Time required to choose minimum carbon emission and max profit data center on worst case = nd^2
Time required to choose a VM is = nd * v
Time required for load balancing on an average = d*v*n
Total time = (nvd + nvd)

Normally n (Number of processes arrived) and v (average number of VM in data center) is much more than heat of number of data center means, n & v >>d

So, Total time = nv + nv
= 2nv
Total time = O (n^2).

4. Conclusion and Future Work

In this paper we developed algorithm for private cloud and public cloud to handle load balancing with effective scheduling algorithm. For private cloud we develop algorithm which use SJF with starvation and also consider the issue of load balancing. Result of private cloud shows that less loaded VM is chosen for the execution of user request which will in turn increase the throughput of private cloud for public cloud we develop algorithm which cover the issue of environment and profit maximization with load balancing. Result shows that carbon emission rate not vary too much with the increase or decrease in the arrival rate of processes. Total profit is increased with arrival rate up to a limit after that it starts decreasing which will increased with extra setup and cloud provider may go for that if they have the budget means they not have to face legal problem due to carbon emission. Private cloud work is good for the organization who wants to create its own setup to provide cloud computing to its user. Public cloud work is good for the cloud provider who want environment conscious solution and after that want to maximize their profit.

In the future we also cover the issue of soft constraint in environment conscious solution of public cloud. In public cloud our algorithm firstly a pair of data center is selected on the base of environment conscious. After that a data center is chosen on the base of profit maximization. Soft constraint specify the process resource need in a way that if resources are available they will be allocated to the process. This will further increase the performance of cloud system.

References

[1] Saurabh K Garg,Chee Shin Yeo, Arun Anandasivam, Rajkumar Buyya “Environment – Conscious Scheduling of HPC application on distributed Cloud – oriented centers”, ScienceDirect, May 2010.
[2] Rashmi K S and Suma V and Vaidehi M “Factors Influencing Job Rejection in Cloud Environment “IJC May 2012.
[3] Rashmi K S and Suma V and Vaidehi M “Enhanced Load Balancing Approach to avoid Deadlocks in Cloud” IJCA-ACCTHPCA, June 2012.
[4] Gabriel Mateescu, Wolfgan Gentzsch , Calvin J Ribbens “Hybrid Computing – Where HPC meets grid and Cloud Computing” Science Direct November 2010.

[5] Weiwei Lin, james Z. Wang, Chen Liang, Deyu Qi "A Thrseshold – Based Dynamic Resource Allocation Scheme for Cloud Computing" Science Direct 2011.
[6] Raul Alonso-Calvo, Jose Crespo, Miguel Garcia-Remesal, Alberto Aungita and Victor Maojo "On Distributing load in cloud computing: A real application for very large image datasets", Science Direct 2010.
[7] Ashraf Zia & Muhammad Naeem Ahmad Khan "Idetifying Key Challenges in Performance Issues in Cloud Computing", IJMECS 2012.
[8] Alexey Tumanov, James Cipar and Michae, A Kozuch, "Algebraic Scheduling of Mixed Workloads in Hetrogeneous Clouds", ACM 2012.
[9] Monir Abdullah , Mohamed Othman "Cost – Based Multi – QoS Job Scheduling using Divisible Load Theory in Cloud Computing", Science Direct 2013.
[10] R. Santosh and T. Ravichandran "Non-Preemptive on-Line Scheduling of Real-Time Services with Task Migration for Cloud Computing", EJSR 2012.
[11] Soumya Ray and Ajanta De Sarkar "Execution Analysis of Load Balancing Algorithm in Cloud Computing Environment in International Journal on Cloud Computing : Service and Architecture (IJCCSA)" , Vol 2, Oct 2012.
[12] K. L. Giridas, A Shajin nargunam, "CHPS in Cloud Computing Environment " in International Journal of Engineering and Technology (IJET) October Nov 2012.
[13] Zhang. Y and Zhou Y "TransOS: A Transparent computing-based operating system for the cloud".
[14] Harsora and Dr. Apurva Shah "A Modified Genetic Algorithm for Process Scheduling in Distributed System", IJCA, 2011.
[15] Indraveer Chana and Anju Bala, "A Survey of Varoous Workflow Scheduling Algorithm in Cloud Environment", NCICT 2011.
[16] Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, "Cloud Computing a Practical Approach", TATA McGRAW HILL Edition2010.

Authors Profile

Ashish Kumar Singh completed the B.E. in Computer Science from RGPV, Bhopal and M.B.A. from BU, Bhopal degrees in 2003 and 2007, respectively. Currently he is doing dissertation of M.E. (CS) final semester on same topic as of paper, From SRIT, Jabalpur, MP, India.

Sandeep Sahu completed the M.Tech. in Computer Science from IIT Guwahati, India in 2009. Currently he is working with SRIT, Jabalpur as HOD, Department of Computer Science and Applications. He is the guide of dissertation work of Ashish Singh

Mangal Nath Tiwari completed the M. Sc. (IT) and M. Phil (CS) Degrees from APS University, Rewa in 2004 and 2010, respectively. Currently he is a student of Ph.D. in APS University, Rewa, MP, India.

Dr. R.K. Katare received the M. Tech and Ph.D. in Computer Science from DAVV Indore, India. Currently he is Professor in APS University, Rewa, MP, India.