International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 5
May 2017

Academic Science

Resource Provisioning byTask Scheduling in Cloud Environment

Subhash S Lokhande
M.E.Student
Department of Electronics and Telecommunication
N.M.I.T Talegaon,Pune

Dr.RajendraD.Kanphade
Professor & Principal
Department of Electronics and Telecommunication
N.M.I.T Talegaon, Pune,

Prof. S.V.Joshi
Assistant Professor
Department of Electronics and Telecommunication
N.M.I.T Talegaon, Pune,

ABSTRACT

Today Cloud environment is in very much demand due to it dynamically allocating resources, by providing reliable services to end users. One can pay as per usage. In order to satisfy customer requirement, one has to make provision of accessible assets efficiently to requesting customers as per their need. Service Level Agreements (SLA) plays very important role during resource provisioning. Recent works consider single parameter from SLA. Considering numerous SLA parameters and asset allocation using preemption mechanism for execution high priority tasks for improving asset usage in cloud. In this paper we proposed an algorithm which considers Preemptable tasks execution and multiple parameters such as memory, CPU time, and network bandwidth. Test result appear situations contention in resources and how algorithm provides better utilization.

Keywords

Cloud computing, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Resource management, Software as a Service (SaaS), Virtual machine, Virtualization.

INTRODUCTION

Cloud computing has become an enabling paradigm for On-request provisioning of registering assets. Server combination by virtual machines (VMs), and application union by multi-tenure and pay-as-you-go utility processing model one can accomplish versatility, cost-proficiency and high asset use for dynamic workload requirements [1]. Improved asset uses for high output and high benefit motivated Cloud providers. Sometime getting high asset utilization can result in unacceptable reaction time for few client request, resulting in degrading quality of service which result is request latency Cloud providers should put more efforts for an accommodating maximum request while using minimal resources without compromising quality of service. So for achieving high efficiency and quality of service, task scheduling and resource provisioning plays critical role in cloud environment. Differing qualities of uses becoming quickly running from business, designing, science which is sent on cloud. Long running assignments, for example, Web crawling systems commonly require to be fault tolerant to keep away from high cost of move back operations in the nearness of disappointment, in spite of the fact that this sort of errands is less delicate to response (the consummation time) compared with those with continuous due dates. By ensuring adaptation to non-critical failure, the errands are allotted in a way to such an extent that the effect of a disappointment on framework execution is limited [2]. Similarity, one can likewise choose energy preservation as the objective of execution improvement, where the assets are utilized as a part of an approach to guarantee that the aggregate vitality required to execute a given workload is limited. For addressing the diversity in cloud administrations and applications, late innovative work engaged in implementing and designing task scheduler, algorithms for particular tasks, like dependent and independent tasks, fault-tolerance with energy efficiency or real time.

Task scheduling and asset provisioning plans, however enhanced with particular targets, experience the ill effects of a few inherent issues. First, the optimization objectives, once set at the design time, will be statically

incorporated with the errand planning and asset provisioning calculation and usage as solid framework part, along these lines lacking adaptability and flexibility in the proximity of changing workload portrayal, changing resource provisioning and changing cloud execution condition. Second, many assignment planning and asset provisioning methodologies and algorithms, despite the fact that outlined with changed diverse advancement goals. In any case, including new planning capacity should be accomplished for each booking calculation each one in turn, which is dreary as well as costly and mistake inclined. we propose a general Structure for errand booking and asset provisioning in distributed computing frameworks with dynamic adaptability. By using programming building system as the plan rule, we meld numerous planning destinations and various sorts of events to be handled under varied asset constraints to empower cloud applications to powerfully choose and assemble scheduling procedures and calculations as per diverse runtime QoS necessities. We check the adaptability and customization utilizing two scheduling algorithms:

EASU and *RAS*. We analyze effectiveness by doing experimental assessment of calculations utilizing simulation. Through this paper we have made some contribution. Flexibility in selecting objectives for scheduling algorithm achieved through undertaking planning and asset provisioning. Prepared scheduling structure for supporting flexible and adaptive resource provisioning. Provided example for showing effectiveness and efficiency.

II. Related work

Now a days cloud computing become more popular works carried in this context. Many parameters considered in this regard in research. While scheduling tasks different objectives and different task types are considered. Cardosa et al consider energy efficient scheduling. He highlighted problem of energy consumption and job run-time [3]. Now everywhere in IT industries green IT concept came in to picture. This leads to changes in infrastructure which is more energy efficient. Rack servers are replaced by blade servers which require less power. This kind of infrastructure reduces require work space. Also job run-time can be reduce by properly balancing load on CPU.Deng et al. proposed smart power system in datacenter for delivering sufficient power as demanded [4]. Reliability is also consider as objective. Wang et al. used conventional PB model for analyzing the limitations while scheduling and resource provisioning flexibility [5]. Priority base model help in managing and utilizing shared and available resources efficiently. Due to ideal usage of accessible resources, one can save lot of money and resources will not be underutilized. Reliability means the customer will get requested resources as per SLA without compromising SLA.Real time task scheduling is considered by Abrishami et al, who use PCP algorithm which is two phase scheduling for minimizing execution cost within predefined time [6]. Small industries generally don't spent on initial infrastructure. Due to availability of many service providers, small scale industries prefer outsourcing. So they use shared infrastructure over cloud. In this manner they save infrastcture cost. Customer need to pay only what they use. Also due to service level agreement customer get good service. Due to tough completion in market quality of service is improved. Also due to virtualization, managing data center become very easy. Through virtualization tool like vcenter administrator can manage datacenter centrally. Also one can use shared resources and provisioning can be done in real-time. This results in zero downtime. One can manage task scheduling by queuing the request which are not processed in case of peak load. In such cases one can one can create task for checking the availability of desired resources in resource pool manager or checking the task end time [7].

III. DESIGN OVERVIEW

Scheduling administration targets, event types and asset characterization are important aspects in resource management.

Fig 1 shows design overview of scheduling framework, which is basically having three important parts.

Academic Science

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 5
May 2017

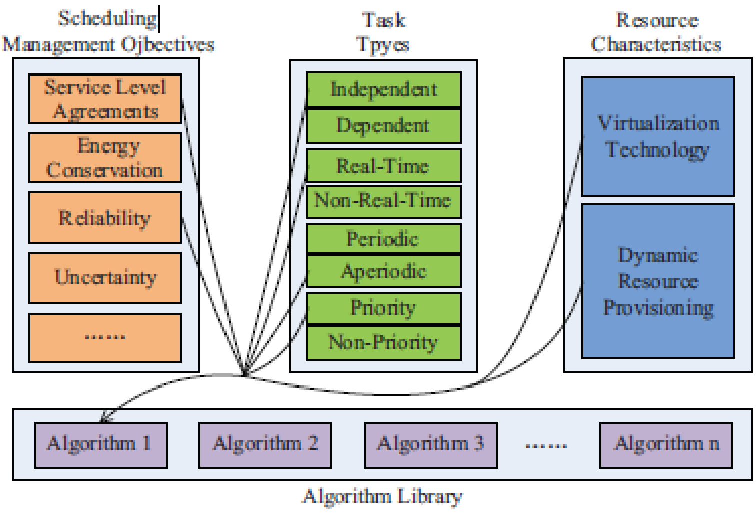

Fig 1 Design Overview of Scheduling Framework.

A. Scheduling Management Objectives.

This allocates tasks to group of resources which intern perform some tasks while providing resources to perform task.

In this there are several parameters used.

1) Service Level Agreement:

In short it is called as SLA. This agreement is between service provider and end customer. This agreement includes response time, costing, support time for any issue etc.

2) Energy Conservation:

Datacenters consumes lot of energy which results in high costing and degrades system reliability. So task scheduling and resource provision should be done in such a way that will result in minimizing energy.

3) Reliability:

It is service provided to the customer with zero downtime. Guaranteed service to the customer. It is assurance provided by service provider. Fault tolerance of system during peak load or critical condition gives reliability.

4) Uncertainty:

One has to monitor and control the uncertainty. Without controlling the parameters, tasking scheduling will not function properly and will result in negative impact.

B. Tasks Types

The tasks are divided into different categories depends on their use. These tasks are running simultaneously in cloud environment for public or private cloud. There are four types of tasks.

1) Independent and Dependent Tasks:

Independent tasks are not depends on any control or data. There is no dependency. While dependent tasks are fully or in part relies on some data output from previous stage. These tasks should be follow some sequence or protocol set by system.

2) Real-time and Non-real-time Tasks:

Real time tasks need to follow some define time limits or deadlines. If time frame did not meet, this results in failure in system. Non-real-time tasks are not depending on time frame or dead line. But these tasks should be performing as quickly as possible. There should not be much time lag.

3) Periodic and Aperiodic Tasks:

Periodic tasks are carried in specific time interval repeatedly. Time interval between two consecutive tasks is constant. In case of aperiodic tasks their entry times are not periodic, it varies as time to complete task varies.

4) Priority and Non-Priority Tasks:

As name suggested, priority based tasks are urgent and important tasks which should be carried as per request given by user. For non-priority tasks, these are not urgent and can be carried after completion of active tasks. These tasks can be stored in queue till resource get available.

C. Resource Characteristics

There are two important components in cloud computing and tasks scheduling.

1) **Virtualization:** For providing flexibility and scalability we use virtualization technique. One implement multiple VM on single physical system. So tasks are allocated to VM instead of physical system [7].

2) **Dynamic Resource Provisioning:** This is very unique feature. System can provide resources to the end user as per their request. Due to shared environment on can pay money as per their usage [8].

IV. SYSTEM MODEL

For handling different task scheduling and resource provisioning one need to devise some system architecture. Fig 2 shows scheduling architecture. The architecture consists of SMO Analyzer, Task Analyzer, Task Scheduler, Objective pool, Resource Monitor, Resource Allocator, Algorithm Library, Resource pool. Task Analyzer get tasks to be performed and it then provide required data to SMO Analyzer. SMO Analyzer decides which objective to be selected based on analysis provided by SMO, objectives are carried as per SLA.Task scheduler select the particular algorithm for performing the task Resource monitor check the available resource and resource allocator provides the resource accordingly. Resource allocator check the resource pool and accordingly assign the task to available VM. If resource not available, then the required tasks are kept in queue.

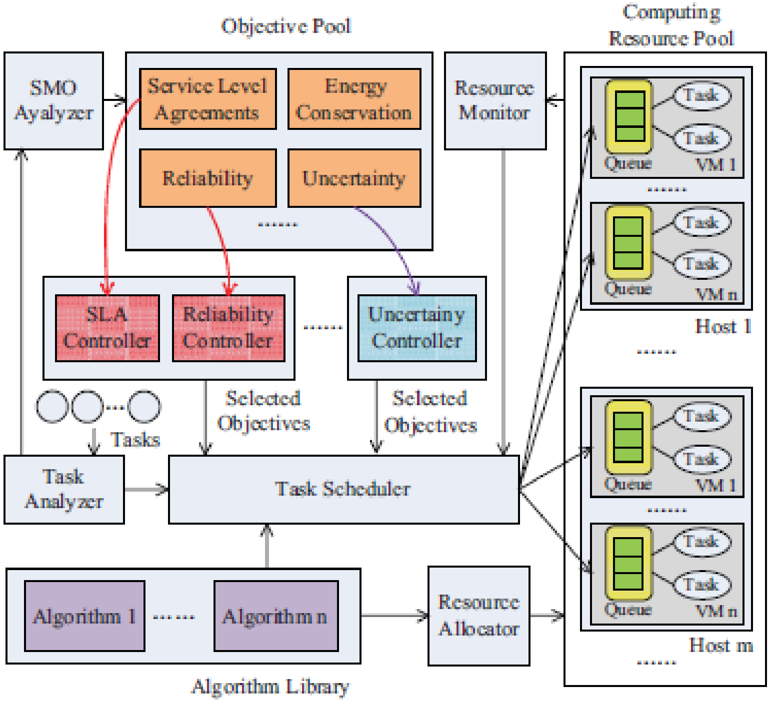

Fig 2 Scheduling Architecture.

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 5
May 2017

V. ALGORITHM DESIGN

Algorithm 1: Pseudocode of *EASU*

$WQ \leftarrow \emptyset; UQ \leftarrow \emptyset;$
foreach *new task* t_i **do**
  $ft_{MAX} \leftarrow 0; ft_{MIN} \leftarrow +\infty;$
  foreach *VM* v_{jk} **do**
    Calculate its estimated minimal finish time ft^-_{ijk} and maximal finish time ft^+_{ijk} and energy ec_{ijk};
    if $ft^-_{ijk} < ft_{MIN}$ **then**
      $ft_{MIN} \leftarrow ft^-_{ijk}$;
    if $ft^+_{ijk} > ft_{MAX}$ **then**
      $ft_{MAX} \leftarrow ft^+_{ijk}$;
  if $ft_{MAX} \leq d_i$ **then**
    Allocate t_i to the VM v_{jk} with minimal ec_{ijk};
  else if $ft_{MAX} > d_i$ & $ft_{MIN} \leq d_i$ **then**
    Allocate t_i to the VM v_{jk} with minimal ft^+_{ijk};
  else
    if $L_i < L_d$ **then**
      Reject task t_i;
    else
      Call scaleUpResources() and Allocate t_i on a new VM;

Fig 3 EASU Algorithm

In RAS PB model is used. It provides resources for task as per the priorities. The VM which is having earliest finish time are selected and provided to the task as per priority.

Algorithm 2: Pseudocode of Primaries Scheduling in *RAS*

$H_{candidate} \leftarrow$ top $\alpha\%$ hosts in H_a;
$eft \leftarrow +\infty$; $v \leftarrow NULL$;
while *!all hosts in* H_a *have been scanned* **do**
  foreach h_k *in* $H_{candidate}$ **do**
    if h_k *satisfies* t_i*'s scheduling dependent constraints* **then**
      foreach v_{kl} *in* $h_k.VmList$ **do**
        Calculate the earliest start time est_i;
        $eft^P_i \leftarrow est^P_i + e^P_{ikl}$;
        if $eft^P_i < eft$ **then**
          $eft \leftarrow eft^P_i$;
          $v \leftarrow v_{kl}$;
  if $eft > d_i$ **then**
    $H_{candidate} \leftarrow$next top $\alpha\%$ hosts in H_a;
  else
    break;
if $eft > d_i$ **then**
  if *scaleUpResources*(t_i) **then**
    return *true*;
  else
    Allocate t_i to v_{kl};
else
  Allocate t^P_i to v_{kl};

Fig 4 RAS Algorithm

Academic Science

International Journal of Electronics, Electrical and Computational System
IJEECS
ISSN 2348-117X
Volume 6, Issue 5
May 2017

VI. IMPLEMENTATION

We have created data center with cloud simulator. We have created 5 hosts with 512Mb Ram with data storage capacity.

Then we have created number of VM, provided some storage and define VM scheduling. We have created task list which should be carried out.

Fig 5 (a) shows the EASU performance.

Fig5 (b) shows task scheduling performance based on priority.

Fig5(c) Shows comparison of EASU and RAS.

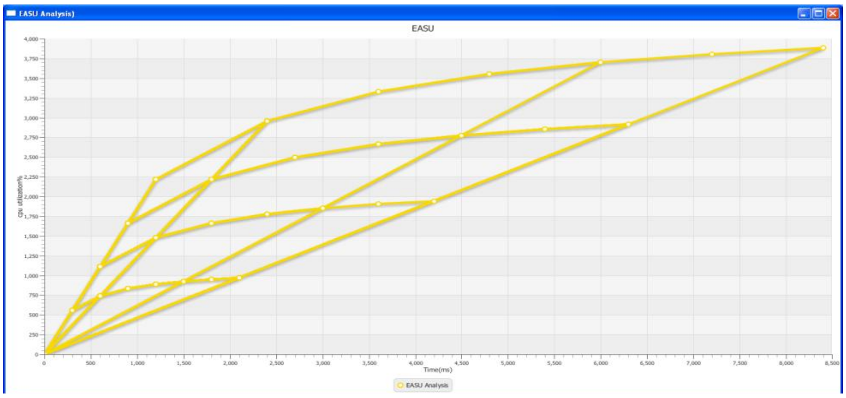

Fig 5 (a) shows the EASU performance.

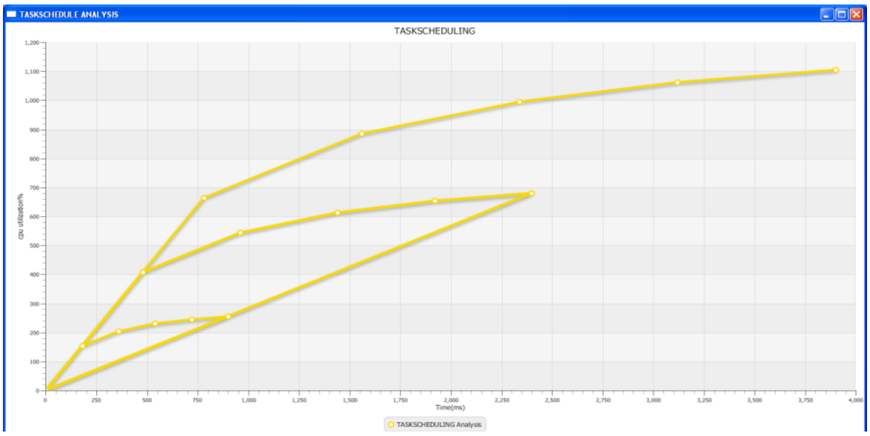

Fig 5 (b) shows task scheduling performance based on priority.

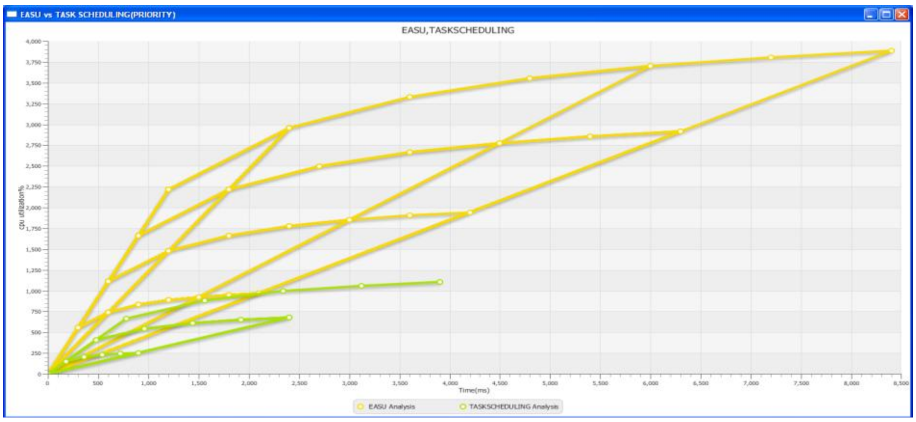

Fig5(c) Shows comparison of EASU and RAS.

VII. CONCLUSIONS

We have considered assignment planning and asset provisioning in cloud processing condition. We actualized EASU and RAS calculations for making general engineering for assignment planning and asset provisioning. By using scheduling. We have accomplished adaptability in running assignments and asset allotment based on needs which is characterized according to SLA. We have carried different experiments with number of parameters and analyze the time required to complete the task.

REFERENCES

[1] J. Rao, Y. Wei, J. Gong, and C. Xu, "QoS guarantees and serviceDifferentiation for dynamic cloud applications," *IEEE Trans. Network andService Management*, vol. 10, no. 1, pp. 43-55, 2013.
[2] G. Mehta, E. Deelman, J. A. Knowles, T. Chen, Y. Wang, J. V¨ockler,S. Buyske, T. Matise, "Enabling data and compute intensive workflowsin bioinformatics," *Euro-Par: Parallel Processing Workshops*, pp.23-32,2011.
[3] M. Cardosa, A. Singh, H. Pucha, and A. Chandra, "Exploiting spatiotemporaltradeoffs for energy-aware mapreduce in the cloud," *IEEE Trans.Computers*, vol. 61, no. 12, pp. 1737-1751, 2012.
[4] W. Deng, F. Liu, H. Jin, and C. Wu, "Smartdpss: cost-minimizing multisource power supply for datacenters with arbitrary demand," *Proc. Int'lConf. Distributed Computing Systems (ICDCS '13)*, pp. 420-429, 2013.
[5] J. Wang, W. Bao, X. Zhu, L. T. Yang, Y. Xiang, "FESTAL: fault-tolerantelastic scheduling algorithm for real-time tasks in virtualized clouds,"*IEEE Trans. Computers*, vol. 64, no. 9, pp. 2545-2558, 2015.
[6] S. Abrishami, M. Naghibzadeh, and D. H. Epema, "Deadline-constrainedworkflow scheduling algorithms for infrastructure as a service clouds,"*Future Generation Computer Systems*, vol. 29, no. 1, pp. 158-169, 2013.
[7] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski,G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloudComputing," *Communications of the ACM*, vol. 53, no. 4, pp. 50-58, 2010.
[8] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, "CloudComputing and emerging IT platforms: vision, hype, and reality fordelivering computing as the 5th utility," *Future Generation ComputerSystems*, vol. 57, no. 3, pp. 599-616, 2009.