Volume 8, No. 3, March – April 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Advance Resource Reservation based on Context Aware Workload

Parimal Gajre, Prof. Vivek Prasad and Dr. Madhuri Bhavsar
Institute of Technology
Nirma University
Ahmedabad, Gujarat, India

***Abstract*:** Cloud computing gives the facility of provisioning resources on rent in pay as-you-go fashion, due to which resource demand changes dynamically over time. Such type of dynamic resource demand leads to mainly two types of problems such as Over-Provisioning and Under-Provisioning. The former leads to violations of Service Level Objectives (SLOs) whereas latter leads to wastage of resources, as the system is not being used to full capacity all the time. Also some cloud having limited number of resources cannot satisfy all the requests at a time.
To handle such scenario advance reservation techniques are used, so that the resources available could be used efficiently with minimum possible provisioning cost and at the same time satisfying service level objectives. In the proposed technique, history of resource usage profile of tasks is maintained. For each task submitted to cloud, pattern finding technique is used a task with similar resource usage requirement from history of resource usage profile. After that check-pointing mechanism is used to monitor completion of new task based on resource usage profile of task found in history. Such type of monitoring helps in order to estimate the amount resources that will be released in short duration and based on that resources can be reserved in advance as per user's request. Hence problem of under as well as over provisioning could be solved up to great extent at the same time meeting the Service Level Objectives.

***Keywords*:** Resource Provisioning, Check-pointing, Advance Reservation, Profiling, K-means clustering.

I. INTRODUCTION

In recent years cloud computing has gained an immense growth, especially in the use of public clouds. Giant companies like Microsoft, Google, Amazon and Rackspace have released their public cloud infrastructures such as Microsoft Azure, Google App Engine, Amazon Web Services and Rackspace Cloud Servers. For such huge enterprise software systems provisioning high assurance in terms of Quality of Service (QoS) metrics such as service availability, high throughput and response time is necessary. Customers signs Service Level Agreements (SLA) with Cloud Service Providers (CSP), which species the QoS metrics agreed by CSP to satisfy. In case if it fails in fulfilling the QoS mentioned in SLA, it will result in a great loss in income as it will tend to lose its customers base. While at same time maintaining SLA and also keeping costs low is dicult task to achieve for cloud service providers, because number customers to cloud computing system are not constant.

Also the applications that customers need to migrate to cloud have varying resource requirements, so it is necessary tohave dynamic mechanism for resource provisioning, in order to satisfy such varying requirements of the applications. This raises the difficult challenge to predict behaviour of applications at run-time in order adjust provisioning of resources dynamically.[5] [7]

A. Provisioning Issues

Mainly there two types of problems in provisioning of cloud resources as:

- Under Provisioning: Leads to the violations of the service level objectives, often associated with financial penalties.

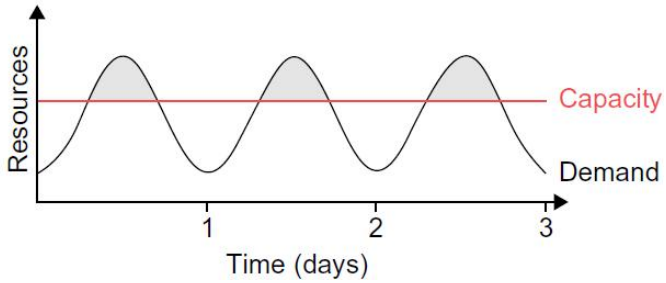

Fig. 1: Under Provisioning of Resources[3]

Figure 1 shows the case of under-provisioning when capacity of resources is planned such that SLAs are broken most of the time leading to customer's dis-satisfaction. So it would lead to loss of customer's base thereby adversely affecting revenue generated by Cloud Service Provider (CSP).

- Over Provisioning: Leads to wastage of the resources, as the system is not being used to full capacity all the time.

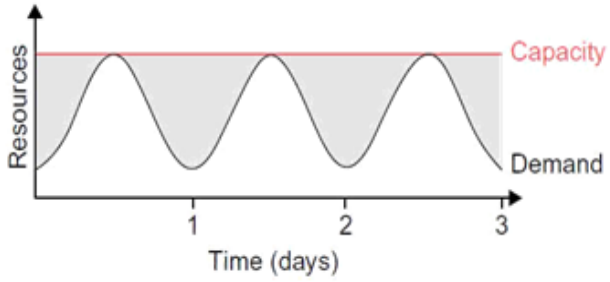

Fig. 2: Over Provisioning of Resources[3]

Figure 2 shows the case over-provisioning when capacity of resources is planned to meet peak load condition, so resources will remain idle most of the time. Here, SLA with customers will be satisfied but wastage of resources will increase as resources will remain in idle condition when the system is not being used to its full capacity.

B. Provisioning Strategies

For provisioning of cloud resources mainly two strategies are used as:

- On Demand Provisioning: In this method resources are allocated, if available at the time of request, else wait till enough resources becomes available. Hence latency for resource availability can be much high. Also as resource demand is dynamic, so problem of thrashing may occur due frequent allocation and de-allocation of resources.

- Reservation of Resources: In this method resource requirement is predicted based on previous resource usage history and accordingly resources could be kept ready. So that resources would be available when resource request is actually encountered. Also release time of allocated resources could be estimated and accordingly reservation of resources for resource re-quests could be made in near future. Hence using reservation method it can be ensured resources will be available when required and so latency for resource availability is reduced. Also availability of resource can be planned, thereby reducing frequent allocation and de-allocation of resources, so problem of thrashing can be mitigated.

II. RELATED WORK

Survey done regarding related work for prediction of workload and reservation of resources is as follows,

Rostyslav, Leitner and others discussed two methods of profiling used for provisioning of resources and task scheduling. The two methods discussed for profiling are passive profiling which corresponds coarse grain profiling whereas active profiling which corresponds fine grain profiling.[10]

Ren, Gang and Tune and others specifies profiling done in google data centres, which captures events such as resource usages, kernel events, heap profiles, lock contention profiles, hardware events and so on. Such profile is used to manage workload on data centres.[6]

Sheng, Robert, Yves and Vivien defines mechanism for checkpoint in which transient memory of running tasks is saved to disk instead of storing entire VM state, so the overhead of check-pointing can be reduced.[2]

Sangho, Kondo, Derrick and Andrzejak discusses about Amazon Elastic Cloud Compute (EC2), which uses checkpointing mechanism to offer high reliability at low cost and volatility of resource provisioning.[9]

Lemarinier, Pierre, Krawezik and others compares two checkpointing methods namely, Coordinated Checkpoint and Message Logging associated with Uncoordinated Checkpoint. Based on evaluation results of both method it is discovered that message logging has higher performance over coordinated checkpointing even with a frequency of one fault per hour.[1]

Sotomayor, Borja, Keahey and others discusses about Haizea, a lease management architecture that implements leases as provisioning of virtual machines (VM), taking advantage of VMs to be suspended, migrated and resumed. Also applications can be provided customized environment using VMs. Here resource requests are categorized as preemptible best effort and non-preemptible advance reservation.[8]

Hui, Groep, David and Templon discusses about prediction of start time of jobs based on statistical analysis of historical job traces and simulation of schedulers at different sites. Using such prediction of job start time middle-ware component such as resource broker can balance workload distribution. Also it can be used to compute price of resources in a grid accounting system. [4]

III. PROBLEM STATEMENT

Provisioning of resources in cloud is crucial task as demand of resources varies continuously. Preparing for peak load would lead to wastage of resources as the resources will remain idle when system is not used to its full capacity, whereas preparing for average load would lead to violations of SLAs with customers. So in-order to provision resources incurring low cost for CSP and at the same time maintaining the SLAs with customers, resource provisioning should be done judiciously. To solve this problem of resource provisioning, advance reservation of resources can be used as mechanism. In this technique resources are provisioned in such a manner that it benefits customer by providing guaranty of resource avail-ability and the same time benefiting Cloud Service Provider(CSP) by providing power to service all customers utilizing the available resources efficiently.

IV. PROPOSED SYSTEM

A. Proposed Working Model

As shown in figure 3 is proposed working model of the system is presented. Functionality of various modules included in proposed working model are as follows,

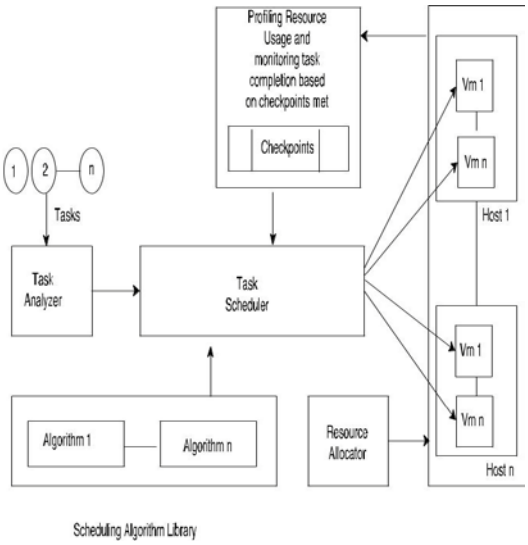

Fig. 3: Proposed Working Model

- **Profiling and Resource Monitoring:** Functionality of this module is to compare new task submitted to cloud with tasks in historical data set of resource usage profile and to find appropriate resource usage pattern. Then monitor the completion of tasks based on checkpoints met according to the matched resource usage profile.

- **Task Analyser and Scheduler:** Task Analyser's functionality is to gather information of new tasks submit-ted to cloud. It checks the requirements of the task like arrival time of the task, deadline of the task and execution time of the task and so on. Then it passes this information to the scheduler, which schedules the tasks on VM's based on availability of resources.

- **Scheduling Algorithm Library:** Scheduling Algorithm Library consists of multiple scheduling algorithm from which a scheduling algorithm is chosen according to resource availability.

- **Resource Allocator:** Functionality of Resource Allocator is to allocate VM's to the task based on resource usage requirement of tasks and availability of tasks. Ifsome task is unable to complete its execution within expected amount of time, then new VM will be created as per the tasks requirement by resource allocator. Also if resource allocator find some of the VM's are lightly loaded or may be idle then it consolidates tasks of such VM's thereby efficiently utilizing the available resources.

B. Flow Model

As shown in Figure 4, flow model of system is presented. Following are steps involved in flow of system,

- Take input from user interface for newly arriving tasks.
- Compare resource usage requirement of task submit-ted to cloud with historical data of resource usage profile to find tasks with similar resource usage requirement(Profiling).
- Determine estimated time required by task submitted to cloud from the tasks in history having similar resource usage profile.

- Divide the new task submitted to cloud based on time and place checkpoints at time when a smallest unit of divided tasks is completed.

- Profile the execution of the newly arrived tasks and also simultaneously monitor the tasks for its partial completion and meeting checkpoints.

- Reserve the resources for next arriving task after the currently executing tasks reaches certain threshold of its execution.

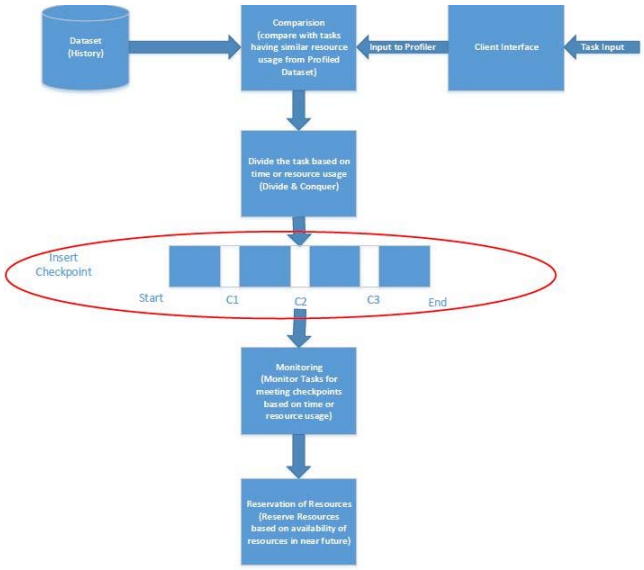

Fig. 4: Flow Model

C. Checkpoint Mechanism

As shown in Figure 5 checkpoints are placed in profiled data, so that it can be used to monitor the completion of newly arrived task of similar resource usage requirement. Comparison is being done between profiled task and new task submitted to cloud, to get status of completion of newly arrived task, based on the checkpoints met by gradual completion of the newly arrived task. And if a delay is measured in meeting the checkpoints than it can be predicted that completion of task will suffer a delay. Also if checkpoints are met in time then accuracy of prediction about the completion of task gets higher gradually.

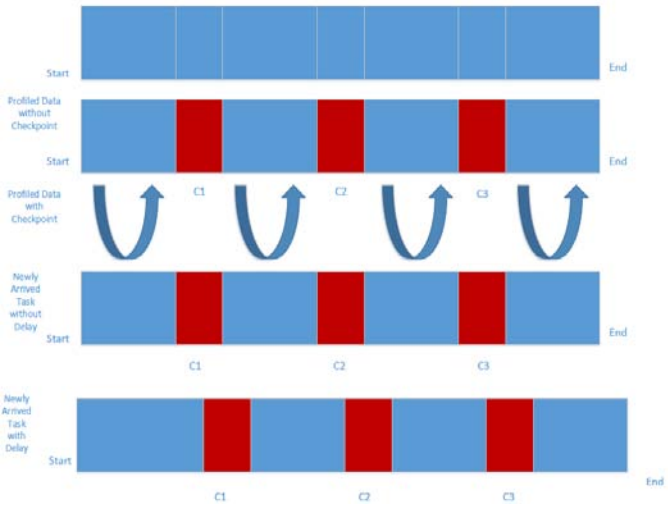

Fig. 5: Checkpoint Mechanism

D. Algorithm

Here, algorithms related to proposed system are described. Task Mapper algorithm is related to allocation of resources to task, if available, else call Advance Reservation algorithm. Advance Reservation algorithm reserves resources for tasks by estimating time to release of resources allocated to executing task.

Assumption

1) Resource usage profile is done in ideal environment (i.e. in average loaded condition).
2) Assumption all Tasks are belong to some restricted domain and are in available Resource Usage Profile.

Algorithm 1: Task Mapper

```
1: Initialization
2: for each (Task[i] in NewlyArrivedTasks[])
3: if (There is a VM[j] available in AvailableVMCapacity[] then) then
4:    Assign Task[i] to VM[j] for execution and update status of VM[j]
5:    Delete Task[i] from NewlyArrivedTasks[]
6: else if (No VM available in AvailableVMCapacity[]) then
7:    Append NewlyArrivedTasks[] to TaskQueue[]
8:    AdvanceReservation(TaskQueue[])
9: end if
```

Algorithm 2 Advance Reservation

```
1: Initialization
2: for each (Task[i] in RunningTaskList[]) do
   if ( look for task to similar Task[i] found in Re-
3. Source_Usage Profile) then
4. Do profiling of Task[i] and append to
   Resource Usage Profile
5:    insert Checkpoints in Task[i] using Logs
6: else
7:    RemainingTime=updateCompletion(Task[i])
8:    if (RemainingTime < (estimatedTask-Time(Task[i])*0.20)) then
9:       80% of Task completed
10:      makeCheckpoint()
11:      notify Task[j] next in queue for TaskQueue[] for Advance Reservation
12:      for each (Task[j] in TaskQueue[])
13:      if (Resource Requirement of next Task[j] in TaskQueue[] <= CheckResourceUtilization(Task[i])) then
14:         //Reserve Resources for Task[j] in TaskQueue[]
15:         Assign Task[j] to VM on which Task[i] is executing
16:         Delete Task[j] from TaskQueue[]
17:         Append Task[j] to ReservedTask[]
18:         break;
19:      else
20:         wait() //wait for some more resources to be
21:      end if
22:   else if (RemainingTime < (EstimatedTask-Time(Task[i])*0.40)) then
23:      60% of Task completed
24:      makeCheckpoint()
25:      notify Task[j] next in queue for TaskQueue[] for Advance Reservation
26:   else if (RemainingTime < (EstimatedTask-Time(Task[i])*0.60)) then
27:      40% of Task completed
28:      makeCheckpoint()
29:      notify Task[j] next in queue for TaskQueue[] for Advance Reservation
30:   else if (RemainingTime < (EstimatedTask-Time(Task[i])*0.80)) then
31:      20% of Task completed
32:      makeCheckpoint()
33:      notify Task[j] next in queue for TaskQueue[] for Advance Reservation
34:   end if
```

V. EXPERIMENTATION

ForexperimentationpurposedatasetusedisSanDiegoSupercomputerCenter(SDSC) SP2 log, which is available at URL: “http://www.cs.huji.ac.il/labs/parallel/workload/logs.html”. Using the specified data set simulation is done in CloudSim simulator, which provides cloud based environment for execution of tasks. Initially as new incoming tasks are executed, a profile of resource usage is generated by monitoring resource utilization at different instances in time. As shown in figure 6, graph of resource usage profileis presented, which shows resource utilization of tasks at different instances of time, namely Time Instance 1 to Time Instance 5.

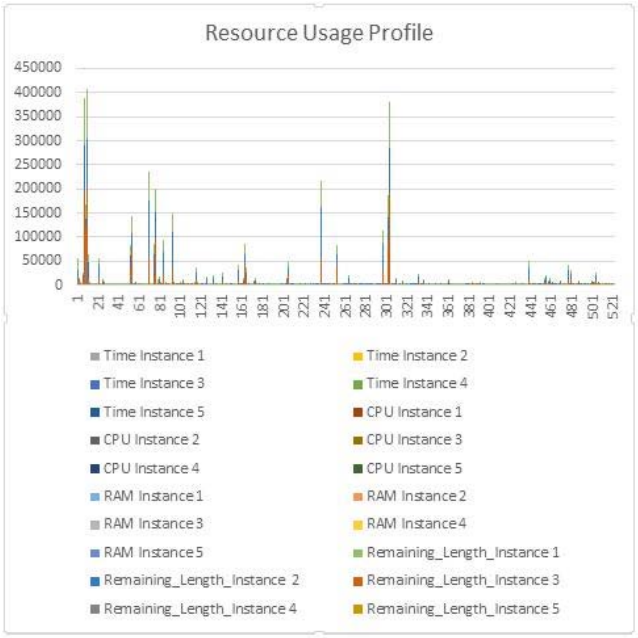

Fig. 6: Profile of Resource Usage

Here resources considered for resource usage profile are CPU and RAM, which are represented by CPU Instance 1 to CPU Instance 5 and RAM Instance 1 to Ram Instance 5 respectively, corresponding to measures of CPU and RAM at different instances of time from Time Instance 1 to Time Instance 5. Remaining Length Instance 1 to Remaining Length Instance 5 corresponds to amount of tasks completed at different instances of time from time Time Instance 1 to Time Instance 5.

Hence, when a new tasks is submitted to the cloud, a search is made for category of tasks with similar resource requirement. For finding category of newly submitted tasks, K-means clustering algorithm is used. Clusters are formed using historical data of resource usage profile and when a new tasks arrives the model is used to find appropriate cluster to which the tasks belong. As shown in figure 7, weka tool is used for

the purpose of K-means clustering to form clusters based on resource usage profile.

Based on cluster determined an approximate execution time of tasks is found and then tasks are monitored for partial completion based on checkpoints met while executing the tasks.

Fig. 7: Clustering of Tasks using K-means

VI. CONCLUSION

Provisioning of resources is a difficult task, as planning for peak load leads to over-provisioning and planning for average load leads to under provisioning. Advance Reservation discussed here mitigates these problems of under-provisioning and over-provisioning by reserving resources for tasks. In technique discussed here advance reservation is done by estimating time required for release of resources that are currently executing the ongoing tasks and based on it reserve resources for new incoming tasks. Hence using such a reservation technique for resource provisioning, latency of tasks waiting for resources to become available is also reduced.

VII. FUTURE WORK

In future, focus will be on developing a system that is capable of incorporating tasks of new type which are currently not present in resource usage profile. Hence, when similar type of tasks arrives in future resource usage profile of this task incorporated into history can be used to monitor completion of new tasks submitted to cloud as per the method discussed in this paper.

VIII. ACKNOWLEDGMENT

We are extremely thankful to our guide Professor Vivek Kumar Prasad and Dr. Madhuri Bhavsar(Head of Department) who have provided us a lot of guidance in doing research and opportunity to do such a wonderful project on Provisioning of Resources using Profiling and Dynamic Scheduling Policy. We are glad to work under their expertise.

IX REFERENCES

[1] Aurelien´ Bouteiller et al. “Coordinated checkpoint ver-sus message log for fault tolerant MPI”. In: Cluster Computing, 2003. Proceedings. 2003 IEEE Interna-tional Conference on. IEEE. 2003, pp. 242–250.

[2] Sheng Di et al. “Optimization of cloud task process-ing with checkpoint-restart mechanism”. In: 2013 SC-International Conference for High Performance Com-puting, Networking, Storage and Analysis (SC). IEEE. 2013, pp. 1–12.

[3] K. Hwang, J. Dongarra, and G.C. Fox. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Elsevier Science, 2013. ISBN: 9780128002049. URL: https : / / books . google . co . in / books?id=IjgVAgAAQBAJ.

[4] Hui Li et al. “Predicting job start times on clusters”. In: Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on. IEEE. 2004, pp. 301–308.

[5] Vivek Kumar Prasad. “Load Balancing and Scheduling of Tasks in Parallel Processing Environment”. In: ().

[6] Gang Ren et al. “Google-wide profiling: A continuous profiling infrastructure for data centers”. In: (2010).

[7] Ashish G Revar and Madhuri D Bhavsar. “Securing user authentication using single sign-on in Cloud Comput-ing”. In: Engineering (NUiCONE), 2011 Nirma Univer-sity International Conference on. IEEE. 2011, pp. 1–4.

[8] Borja Sotomayor, Kate Keahey, and Ian Foster. “Com-bining batch execution and leasing using virtual ma-chines”. In: Proceedings of the 17th international sym-posium on High performance distributed computing. ACM. 2008, pp. 87–96.

[9] Sangho Yi, Derrick Kondo, and Artur Andrzejak. “Re-ducing costs of spot instances via checkpointing in the amazon elastic compute cloud”. In: 2010 IEEE 3rd International Conference on Cloud Computing. IEEE.2010, pp. 236–243.

[10] Rostyslav Zabolotnyi, Philipp Leitner, and Schahram Dustdar. “Profiling-Based Task Scheduling for Factory-Worker Applications in Infrastructure-as-a-Service Clouds”. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications. IEEE. 2014, pp. 119–126.