Capacity Planning Through Monitoring of Context Aware Tasks at IaaS Level of Cloud Computing

Vivek Kumar Prasad(✉), Harshil Mehta, Parimal Gajre, Vidhi Sutaria, and Madhuri Bhavsar

Nirma University, Ahmedabad 382481, Gujarat, India
{vivek.prasad,15mcen12,15mcec12,15mcei28,madhuri.bhavsar}@nirmauni.ac.in
http://www.nirmauni.ac.in/

Abstract. Cloud Computing is the exercise of using a network of remote servers held on the Internet to store, manage, and process data which have the characteristics as an elasticity, scalability or scalable resource sharing managed by the resource management. Even the growing demand of cloud computing has radically increased the energy consumption of the data centres, which is a critical scenario in the era of cloud computing, hence the resources has to be used efficiently, which ultimately will minimise the energy. Resource management itself will get the data from resource monitoring and resource prediction for the smooth conduction of the tasks and its allocated resources. In this paper the monitoring mechanism in the cloud has been discussed and its results are used to trigger the prediction rule engine which provides the cloud service provider (CSP) to start allocating the resources in the efficient manner, even the concept of failure handling has been mentioned based upon the certain parameter which will also inform the CSP to handle the failure task and try to mitigate this and again re schedule the failed task.

Keywords: Cloud computing · Monitoring · Resource management
Resource prediction · Scheduling · Error handling

1 Introduction

Cloud computing [9] is a major force, which is changing the Information technology landscape and moving it entire data to the cloud for putting it into the remote location to store, manage and process the data using Internet.

Cloud Service Model: The services are classified based upon their functionality, i.e. Software as a service (SaaS) which is used to deliver the web applications, Platform as a Service (PaaS), to create or deploy application and services for user, Application test, development, integration and deployment type of services. Infrastructure as a service, provides services as rent storage, processing and communication using the concepts of virtual machine.

Z. Patel and S. Gupta (Eds.): ICFITT 2017, LNICST 220, pp. 66–74, 2018.
https://doi.org/10.1007/978-3-319-73712-6_7

In this paper we have proposed the monitoring of the cloud computing environment, in such a way so that the resources utilisation state will be observed continuously and the threshold value of the resources has to be identified and if the value reaches at the threshold, then for the efficient usage of the resources prediction (and its profiling) mechanism will be invoked, which will allow the CSP to make efficient usage of resources, so that the cloud can handle maximum number of requests, thus ultimately will lead to increase in the revenue. The context aware tasks profiling will make us sure that exactly how much resources the particular task will consume and its behaviour for the future request also. Any deviation from its normal behaviour will make the CSP aware of something wrong has happened in the cloud, which in turn will trigger the failure handling and its mitigation approach towards the handling of the failed tasks and again reschedule them.

2 Monitoring the Cloud Environment

2.1 Monitoring as an Essential Tool in Cloud Computing Environment

Monitoring [1] is important for both provider and consumers for managing and controlling the hardware and software infrastructure, it also provides the key performance indicators and information for both platforms and applications. It is also useful for capacity planning [12] where the estimation of the correct resources will improve the efficiency of the resource utility, which will help to meet the criteria of SLA management [14].

Proposed Algorithm. The above Table 1 indicates the execution time taken by various categories of scheduling algorithms based upon the varied availability (the number of Virtual Machines availability) of the resources at the cloud computing environment, Which indicates the scheduling algorithms performs better in different availability of resources, as cloud is dynamic in nature [18]. The resources are in terms of Virtual Machine.

Table 1. Execution time of 25 task

Algorithm	VM 10	VM 25	VM 50	VM 80
FCFS	2541.95	1782.58	677.45	401.01
MCT	430.20	314.6	252.8	236.8
MINMIN	339.2	270.8	240.3	244
MAXMIN	274.3	259.40	248.4	238.2
RR	1449.4	1158.5	930.95	924.57
DATA AWARE	2435.04	1221.65	1009.09	518.25

Algorithm 1. Dynamic Scheduling Mechanisum

```
Initialization
Let the current CPU, RAM is assumed as resources are available on cloud.
{
Initialize threshold value =50%
When new task arrives check threshold value(resources).
if Threshold value >current resource utilization then
    {
    below threshold value;
    schedule the task;
    }
else
    (Threshold values[i]<current resource utilization)
    {
    above threshold value;
    Redirect to check-pointing mechanism();
    }
end if
}
```

The Table 2 above indicates the resources (CPU and RAM) usage values at different interval of time while executing tasks in cloud computing environment. Figure 1 shows the result of the Table 2 where x axis indicates the time intervals and y axis indicates the utility of the resources.

While considering the CPU usage of Table 2 the observations indicates that one observation has value above 50%, and rest are below 50%, on the flipped side in case if number of readings are more than 50% utilization of CPU indicates that the rest of the available resources should be utilised wisely. Likewise same observations has to be noted down for the memory utilization also. Now resource

Table 2. Data observed during experimental

Date and Time	CPU Usage	RAM Usage
April 13 2017 10:14 AM IST	16.02	54
April 13 2017 10:20 AM IST	40.87	67
April 13 2017 10:25 AM IST	34.38	74
April 13 2017 10:30 AM IST	28.2	69
April 13 2017 10:35 AM IST	13.88	68
April 13 2017 10:40 AM IST	78.99	74
April 13 2017 10:45 AM IST	72.3	71
April 13 2017 10:50 AM IST	83.26	73
April 13 2017 10:55 AM IST	65.3	52
April 13 2017 11:00 AM IST	68.80	44

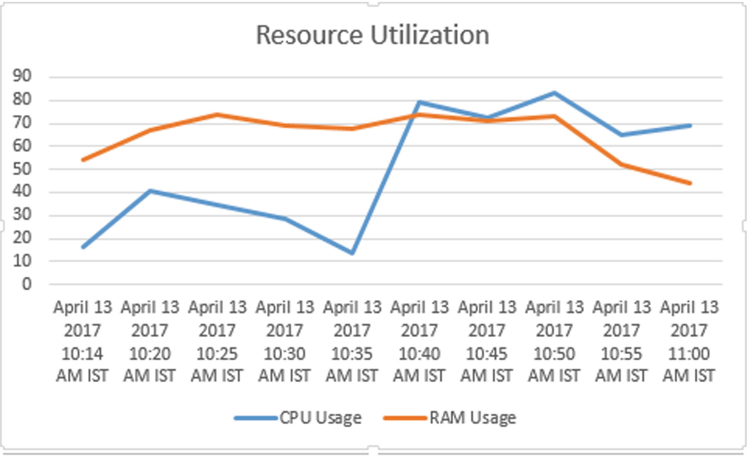

Fig. 1. Resources utilization

prediction for context aware workload module will be invoked which is explained in Sect. 2, So that the SLA will be maintained.

3 Resource Prediction for Context Aware Workload

3.1 Profiling in Cloud Computing for Context Aware Workload

Profiling is a mechanism through which the behaviours of the task execution can be recorded and can be used for the audit purpose. Profiling can be done in two ways, active profiling and passive profiling, where the active profiling is fine grain and the passive profiling is a coarse grain [13].

Here in this research paper for profiling we are considering only the context aware tasks, so that their total execution time and resource usage are determined [16], then these execution time has been divided into certain check points [4] and in every check points whatever resources has been consumed by the tasks are noted and a metric has been prepared, so that if the same task comes next time in future, the same metrics can be used to evaluate the performance of the task [16].

3.2 Proposed Algorithm

Figure 2 indicates the clustering of tasks based on their resource utilisation patterns and the experimentation has been done using weka. If any discrepancy occur due to the pattern mismatch (i.e. the stored pattern and the current pattern), then the error handling mechanism will be invoked, which is mentioned in Sect. 4 below.

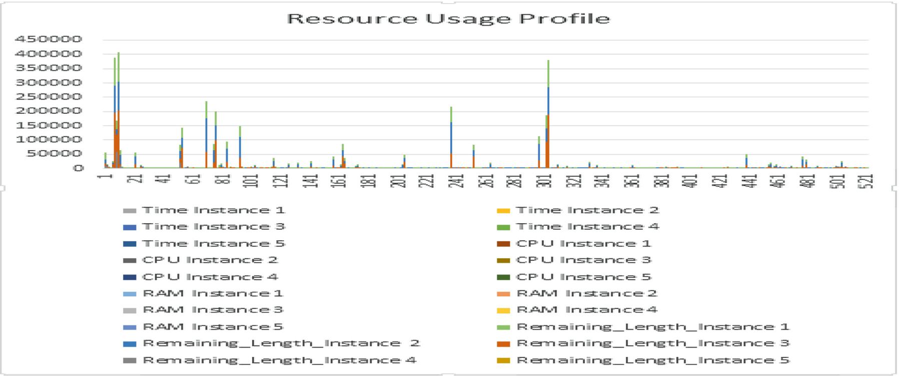

Fig. 2. Profiling of resources for checkpoints

Algorithm 2. Check pointing Mechanism

```
Divided whole task into 5 frame, each of 0.20.
compare with the previous data set by context aware mechanism
if Task match with previous data set task then
    {
    Scheduled the task to server;
    }
else {Task not match with previous data set task}
    {
    check for error handling;
    Redirect to error handling mechanism();
    }
end if
```

4 Error Handling and Mitigation

In this section we have highlighted the mechanism of the error handling and its mitigation techniques [15]. Error handling is the procedure of finding errors in the system. Error should be handle in dynamic way in cloud computing [8,11]. Error handling will also provide robustness and system availability against hardware and software errors in cloud [2]. The proactive method deals with recovery of fault in advance, whereas reactive method deals with recovery after the occurrence of error [3,5,6,10,17] Reactive mitigation techniques: Check-pointing, Restart, Replication, Job migration, Sguard, Retry, Task resubmission and Recue workflow [7].

4.1 Proposed Algorithm

In the given algorithm it classifies hardware and software errors and invoke appropriate mitigation technique for reduce the adverse effect of the error.

Algorithm 3. Error Handling Mechanism

```
Let assumed it will be hardware error or software error.
{
predict the error with error prediction;
if error code <500 then
    {
    software error;
    Mitigate software error by mitigation technique;
    }
else
    (error code>500)
    {
    Hardware error;
    Mitigate software error by mitigation technique;
    }
end if
Dynamic schedule task to the server;
go to next task;
}
```

Figure 3 mention about the execution time verses density (error), X- axis shown the execution time and Y- axis shown the Density, by this we can conclude that which type of error will occur (it has been mentioned in top right corner of figure as error names).

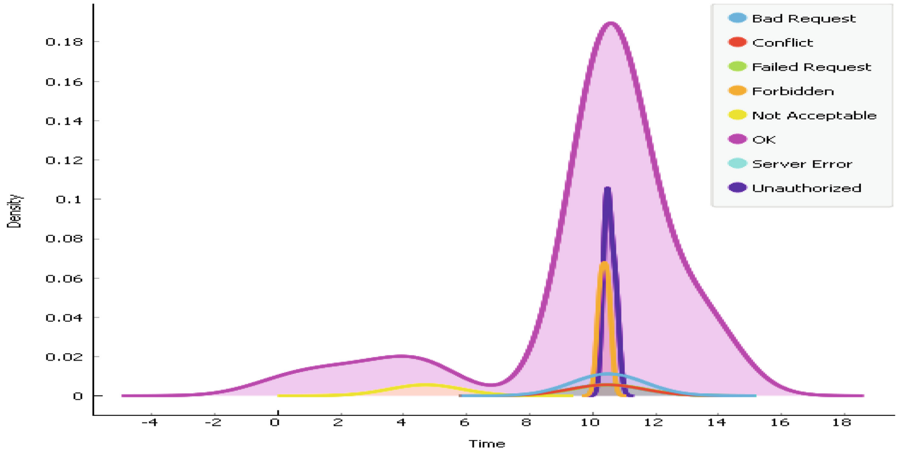

Fig. 3. Execution time Vs Density

5 Proposed Model

Proposed model is as shown in Fig. 4. Key items of the proposed model are as follow:

5.1 Guaranteed SLA Management

The agreement between the client and Cloud Service Provider (CSP) based on certain QoS parameter will be established and monitoring will be done based upon agreed SLA.

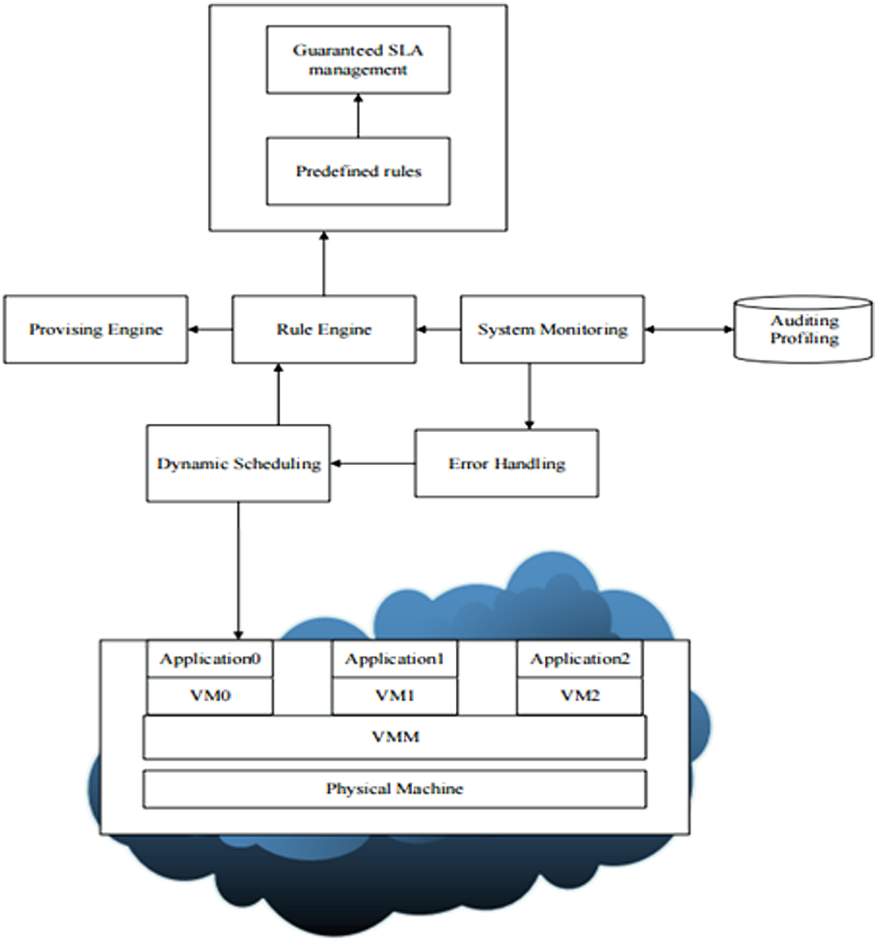

Fig. 4. Proposed model

5.2 Provisioning Engine

Functionality of provisioning engine is to enact according to set of steps known as provisioning plan. It is responsible for various requests from users and applications like to start an application, stop an application, halt an application, request for more resources and so on.

5.3 Rules Engine

Rules Engines functionality is to evaluate data captured by monitoring system based on operational policy. Operational policy defines different action sequence that should be triggered in incase of occurrence of an event. So Rules Engine and operational policy together provide the key to guaranteeing SLA under a self-healing system.

5.4 System Monitoring

Functionality of System Monitoring module is to collect information of different metrics which affect the performance of system and are defined in SLAs.

5.5 Auditing/Profiling

The attachment to the predefined SLA needs to be recorded and monitored. It is indispensable to monitor the compliance with SLA.

5.6 Dynamic Scheduling

As cloud is elastic in nature, the demand of the resources will always be fluctuating, so we require mechanism that will adapt to changing resources scenario.

5.7 Error Handling

As cloud is a on-demand network access to resources and software application, so there is chance that we can encounter the scenario where hardware or software may fail because of uncertainty, so we require some mechanism through which this errors can be handled and mitigated automatically without intervention of human being and after mitigation it should be reschedule back to the dynamic scheduler.

6 Conclusion and Future Work

In this research paper we have proposed an efficient capacity planning at IaaS level of cloud computing for context aware workload. To achieve this cloud monitoring concepts has been used and it has been incorporated to Hidden Markov model (HMM) to categorise the usage of resources at cloud. In critical state of cloud resource usage profiling/auditing mechanism has been triggered. To handle the faulty scenarios where the resources are in peak demand has also been covered.

In this study only CPU and RAM are considered. However still there are other resources which are to be considered such as network, IO and so on. In our future works we will take these factors into consideration. We also will develop and built better energy efficient resource provisioning.

References

1. Aceto, G., Botta, A., De Donato, W., Pescapè, A.: Cloud monitoring: a survey. Comput. Netw. **57**(9), 2093–2115 (2013)
2. Agarwal, H., Sharma, A.: A comprehensive survey of fault tolerance techniques in cloud computing, pp. 408–413 (2015)
3. Bala, A., Chana, I.: Fault tolerance-challenges, techniques and implementation in cloud computing. IJCSI Int. J. Comput. Sci. Issues **9**(1), 1694–0814 (2012)
4. Bouteiller, A., Lemarinier, P., Krawezik, K., Capello, F.: Coordinated checkpoint versus message log for fault tolerant mpi. In: Proceedings of the 2003 IEEE International Conference on Cluster Computing, pp. 242–250. IEEE (2003)
5. Cheraghlou, M.N., Khadem-Zadeh, A., Haghparast, M.: A survey of fault tolerance architecture in cloud computing. J. Netw. Comput. Appl. **61**, 81–92 (2016)
6. Ganesh, A., Sandhya, M., Shankar, S.: A study on fault tolerance methods in cloud computing, pp. 844–849 (2014)
7. Jhawar, R., Piuri, V., Santambrogio, M.: Fault tolerance management in cloud computing: a system-level perspective. IEEE Syst. J. **7**(2), 288–297 (2013)
8. Kaur, P.D., Priya, K.: Fault tolerance techniques and architectures in cloud computing-a comparative analysis, pp. 1090–1095 (2015)
9. Mell, P., Grance, T., et al.: The nist definition of cloud computing (2011)
10. Mittal, D., Agarwal, N.: A review paper on fault tolerance in cloud computing, pp. 31–34 (2015)
11. Patra, P.K., Singh, H., Singh, G.: Fault tolerance techniques and comparative implementation in cloud computing. Int. J. Comput. Appl. **64**(14), 37–41 (2013)
12. Psoroulas, I., Anagnostopoulos, I., Loumos, V., Kayafas, E.: A study of the parameters concerning load balancing algorithms. IJCSNS Int. J. Comput. Sci. Netw. Secur. **7**(4), 202–214 (2007)
13. Ren, G., Tune, E., Moseley, T., Shi, Y., Rus, S., Hundt, R.: A continuous profiling infrastructure for data centers, Google-wide profiling (2010)
14. Shin, S., Kim, Y., Lee, S.: Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing. In: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), pp. 814–819. IEEE (2015)
15. Singla, N., Bawa, S.: Priority scheduling algorithm with fault tolerance in cloud computing. Int. J. 3(12) (2013)
16. Sotomayor, B., Keahey, K., Foster, I.: Combining batch execution and leasing using virtual machines. In: Proceedings of the 17th International Symposium on High Performance Distributed Computing, pp. 87–96. ACM (2008)
17. Tchana, A., Broto, L., Hagimont, D.: Approaches to cloud computing fault tolerance, pp. 1–6 (2012)
18. Zhong, H., Tao, K., Zhang, X.: An approach to optimized resource scheduling algorithm for open-source cloud systems. In: 2010 Fifth Annual ChinaGrid Conference, pp. 124–129. IEEE (2010)