- **Measured Service:**
  Although computing resources are pooled and shared by multiple consumers (i.e. multi-tenancy), the cloud infrastructure is able to use appropriate mechanisms to measure the usage of these resources for each individual consumer through its metering capabilities.

- **Collaborative Environment:**
  Cloud environment provide collaborative working environment for team members located at geographically different location.

"Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service level agreements established through negotiation between the service provider and consumers."

## Service Model

Cloud computing provide virtualized environment in four major cloud service forms as shown in the Figure 1. Services are as explained below.

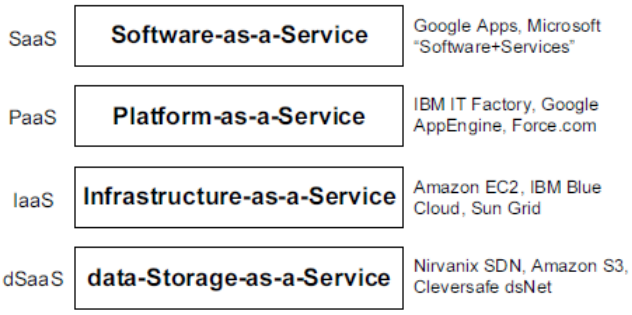


**Figure 1**: Cloud computing service layers with examples

## Objectives

Our objective is to select best VM for optimal CPU Utilization. Virtual machine selection criteria depends on available knowledgebase of similar task executed in past. Knowledgebase contains task execution time, % of CPU utilization for that task. Select appropriate virtual machine to execute same type of task by analyzing available knowledgebase information and generate quick output for user.

The list of objectives that need to be attained includes following:

- Study of Literatures on Cloud computing
- Design of proposed system architecture
- Design proposed algorithm
- Testing different Scenarios
- Results analysis of the proposed algorithm

## II. METHODS AND MATERIAL

### 2.1 System Architecture

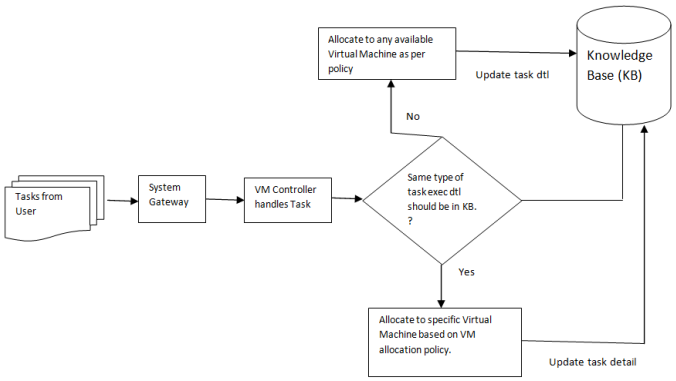


**Figure 2:** Proposed System Design for Resource Allocation

**Working of System:**

AS shown in above figure 2 when user submit the task, controller handle the incoming task and check whether same type of task has been be executed in past as per the available knowledge base. If controller found the task details, it can estimate the CPU requirement and execution time for incoming task. Based on this data, controller will select Virtual Machine from the available list of Virtual Machine as per the algorithm policy.

### 2.2 Proposed Algorithm

1. User task execution request
   Find out task characteristics Task-Name.
2. Find out similar type of task which was executed in past?
   If No go to step 3 else go to step 5
3. Store incoming task details and go to step 4
   Task-ID, Task-Name, Task-Arrival-Time
4. Allocate available Virtual Machine on basis of Max VM-CPU-Available Criteria and go to step - 9.
5. Find out task execution summary from available record.
6. Retrieve available Virtual Machine details in descending sorted order on basis of current CPU utilization.
7. Match these VMs status with required CPU utilization of task and allocate VM using Following criteria :
   a. If Task_CPU_Requirement < VM_CPU_Available[] then
      Find nearest %CPU availability value with combination of executed task estimated time on that

machine to select VM and *schedule task* on that Virtual Machine.

b. If Task_CPU_Requirement > VM_CPU_Available[] then
Calculate average time to complete assigned task based on estimated task time for CPU availability of each VMs and *schedule task* on VM which has minimum value of average time.

8. At end of task executionStore Task-End-Time, Average-CPU-Utilization, Required-Exec-Time.
9. Exit.

**Note:** VM-CPU-Usage for every available VMs is being update at regular interval.

## Assumption:

1. Consider SAAS scenario provides specific services with private cloud environment.
2. Number of Virtual Machine has same type of hardware configuration.
3. All Virtual Machines are always up.

## III. RESULT AND DISCUSSION

### 3.1 Result Analysis

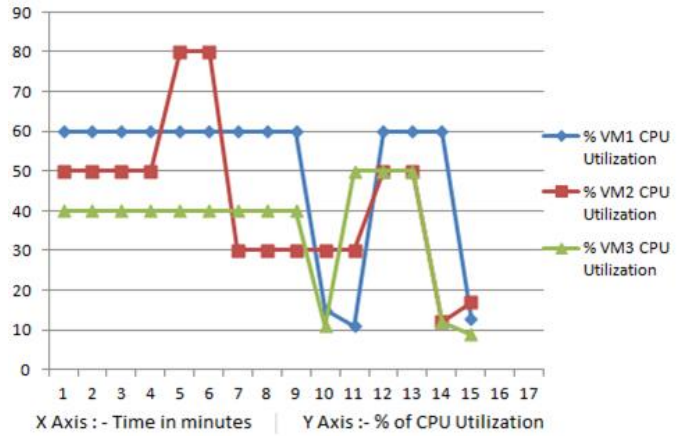


**Figure 3:** Analytical Graph: Case (i) : Task CPU Requirement < VM-CPU-Availability

**Table1:** Task Execution Scenario

| Time | % VM1 CPU Utilization | % VM2 CPU Utilization | % VM3 CPU Utilization |
|---|---|---|---|
| 1 | 60 | 50 | 40 |
| 2 | 60 | 50 | 40 |
| 3 | 60 | 50 | 40 |
| 4 | 60 | 50 | 40 |
| 5 | 60 | 80 | 40 |
| 6 | 60 | 80 | 40 |
| 7 | 60 | 30 | 40 |
| 8 | 60 | 30 | 40 |
| 9 | 60 | 30 | 40 |
| 10 | 15 | 30 | 11 |
| 11 | 11 | 30 | 50 |
| 12 | 60 | 50 | 50 |
| 13 | 60 | 50 | 50 |
| 14 | 60 | 12 | 12 |
| 15 | 13 | 17 | 9 |

**Table2:** User Requested Task History

| Allot VM NO | TASK | %CPU REQUIRED | EXPECTED EXEC TIME | ARRIVAL TIME | %CPU UTILIZED | ACTUAL EXECTIME |
|---|---|---|---|---|---|---|
| V1 | T1 | 60 | 9 | 1 | 60 | 9 |
| V2 | T2 | 50 | 6 | 1 | 50 | 6 |
| V3 | T3 | 40 | 8 | 1 | 40 | 8 |
| V2 | T4 | 30 | 7 | 4 | 80 | 7 |
| V3 | T5 | 50 | 3 | 11 | 50 | 3 |
| V1 | T6 | 60 | 3 | 12 | 60 | 3 |

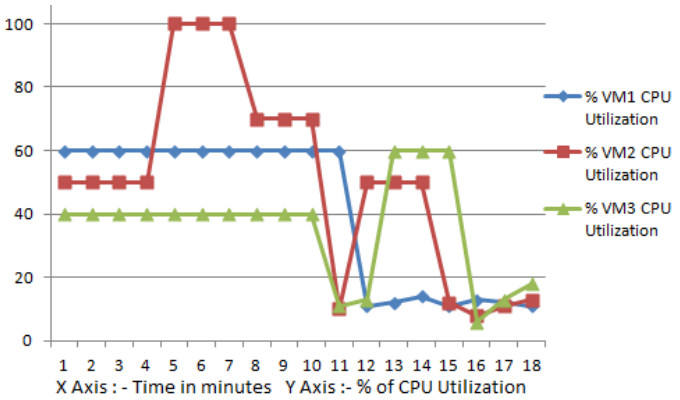


**Figure 3:** Case (ii): Task CPU Requirement > VM-CPU-Availability

**Table 1:**Task Execution Scenario

| Time | % VM1 CPU Utilization | % VM2 CPU Utilization | % VM3 CPU Utilization |
|---|---|---|---|
| 1 | 60 | 50 | 40 |
| 2 | 60 | 50 | 40 |
| 3 | 60 | 50 | 40 |
| 4 | 60 | 50 | 40 |
| 5 | 60 | 100 | 40 |
| 6 | 60 | 100 | 40 |
| 6.7 | 60 | 100 | 40 |
| 7 | 60 | 70 | 40 |
| 8 | 60 | 70 | 40 |
| 9 | 60 | 70 | 40 |
| 10 | 60 | 10 | 11 |
| 11 | 11 | 50 | 13 |
| 12 | 12 | 50 | 60 |
| 13 | 14 | 50 | 60 |
| 14 | 11 | 12 | 60 |
| 15 | 13 | 8 | 6 |
| 16 | 12 | 11 | 13 |
| 17 | 11 | 13 | 18 |

| VM NO | TASK | %CPU REQUIRED | EXPECTED EXEC TIME | ARRIVAL TIME | %CPU UTILIZD | ACTUAL EXEC TIME |
|---|---|---|---|---|---|---|
| V1 | T1 | 60 | 10 | 1 | 60 | 10 |
| V2 | T2 | 50 | 6 | 1 | 50 | 6.7 |
| V3 | T3 | 40 | 9 | 1 | 40 | 9 |
| V2 | T4 | 70 | 5 | 6 | 100 | 6 |
| V2 | T5 | 50 | 3 | 11 | 50 | 3 |
| V3 | T6 | 60 | 3 | 12 | 60 | 3 |

**Table 2:**User Requested Task History

### 3.2 Comparison with Existing Resource Allocation Policy

The effect of the service broker and load balancing techniques are analysed defining

### 3.3 Observations

The effect of the service broker and load balancing techniques are analysed defining the simulation parameters as follows:

I have described that we can analyze the effect of various service brokers and load balancer in the Cloud computing using Cloud Analyst. The output of various service brokers and load balancer is analyzed to evaluate the performance of algorithm.

We can observe from the analysing test cases results that

- Threshold based load balancer provides efficient load balancing.
- Threshold based load balancer results in less response time with proper selection of threshold value.
- Threshold based load balancer requires less processing time.
- Threshold based load balancer is less costly.

- Throttled load balancer with optimized response time service broker results in less response time while both data centers having equal capacity.
- Throttled load balancer with Closest data center service broker results in less response time while closest data centers having different/less capacity then other.
- In both of the above stated cases Throttled load balancer gives better results.

We can also configure different scenarios using more number of Data Centers and VMs to test results.

## IV. CONCLUSION AND FUTURE THOUGHTS

Cloud computing is a new and promising paradigm delivering IT services as computing utilities. As Clouds are designed to provide services to external users, providers need to be compensated for sharing their resources and capabilities. In this paper, they have proposed architecture for market-oriented allocation of resources within Clouds. They have also presented a vision for the creation of global Cloud exchange for trading services. Moreover, they have discussed some representative platforms for Cloud computing covering the state-of-the-art.

Data Centres are known to be expensive to operate and they consume huge amounts of electric power. For example, the Google data centre consumes power as much as a city such as San Francisco.

As Clouds are emerging as next-generation data centres and aim to support ubiquitous service-oriented applications, it is important that they are designed to be energy efficient to reduce both their power bill and carbon footprint on the environment.

To achieve this at software systems level, we need to investigate new techniques for allocation of resources to applications depending on quality of service expectations of users and service contracts established between consumers and providers.

As Cloud platforms become ubiquitous, we expect the need for internetworking them to create market-oriented global Cloud exchanges for trading services. Several challenges need to be addressed to realize this vision.

They include: market-maker for bringing service providers and consumers; market registry for publishing and discovering Cloud service providers and their services; clearing houses and brokers for mapping service requests to providers who can meet QoS expectations; and payment management and accounting infrastructure for trading services.

Some of these issues are explored in related paradigms such as Grids and service-oriented computing systems. Hence, rather than competing, these past developments need to be leveraged for advancing Cloud computing. Also, Cloud computing and other related paradigms need to converge so as to produce unified and interoperable platforms for delivering IT services as the 5th utility to individuals, organizations, and corporations.

## V. REFERENCES


[1] [1] Eran Chinthaka Withana and Beth Plale, "Usage Patterns to Provision for Scientific Experimentation in Clouds". 2nd IEEE International Conference on Cloud Computing Technology and Science. Pages 226-233.

[2] David Candeia, Ricardo Ara´ujo, Raquel Lopes, Francisco Brasileiro, "Investigating Business-Driven Cloudburst Schedulers for e-Science Bag-of-Tasks Applications". 2nd IEEE International Conference on Cloud Computing Technology and Science. Pages 343-350.

[3] D. P. da Silva, W. Cirne, and F. V. Brasileiro. Trading cycles for information: Using replication to schedule bag-of -tasks applications on computational grids. In Euro-Par, pages 169–180, 2003.

[4] M. Armbrust et al., "Above the clouds: A berkeley view of cloud computing," EECS Department, University of California, Berkeley, Tech.

[5] W. Smith, I. Foster, and V. Taylor, "Predicting application run times using historical information," in Job Scheduling Strategies for Parallel Processing. Springer, p. 122.

[6] H. Li, D. Groep, J. Templon, and L. Wolters, "Predicting job start times on clusters," in ccgrid. IEEE, 2004, pp. 301–308.

[7] D. Nurmi, J. Brevik, and R. Wolski, "QBETS: Queue bounds estimation from time series," in Job Scheduling Strategies for Parallel Processing. Springer, pp. 76–101.

[8] A. A. Julian, J. Bunn, R. Cavanaugh, F. V. Lingen, M. A. Mehmood, H. Newman, C. Steenberg, and I. Willers, "Predicting the resource requirements of a job submission arshadali," in In Proceedings of the Conference on Computing in High Energy and Nuclear Physics (CHEP 2004, 2004, p. 273.

[9] F. Berman et al., "Adaptive computing on the grid using apples," IEEE Transactions on Parallel and Distributed Systems, vol. 14, no. 4, pp. 369–382, 2003.

[10] F. Berman, A. Chien, K. Cooper, J. Dongarra, I. Foster, D. Gannon, L. Johnsson, K. Kennedy, C. Kesselman, J. Mellor-Crumme et al.,

[11] "The GrADS project: Software support for high-level grid application development," International Journal of High Performance Computing Applications, vol. 15, no. 4, p. 327, 2001.

[12] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson, "Statistics-Driven Workload Modeling for the Cloud," Technical Report

[13] UCB/EECS-2009-160, EECS Department, University of California, Berkeley, Tech. Rep., 2009.

[14] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113,2008.

[15] F. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao, "Applicationlevel scheduling on distributed heterogeneous networks," in Proceedings of the 1996 ACM/IEEE conference on Supercomputing (CDROM). IEEE Computer Society, 1996, p. 39.