American Journal of Computer Science and Engineering Survey
a Pubicon Open Journal

RESEARCH OPEN ACCESS

A Comparative Analysis of Resource Scheduling Algorithms in Cloud Computing

Aarti Singh[1], Manisha Malhotra*[2]

[1]Associate Prof., MMICT & BM, MMU, Mullana, Haryana, India.
[2]Research Scholar MMICT & BM, MMU, Mullana, Haryana, India.
Email Id: mmanishamalhotra@gmail.com

Abstract

The research in cloud computing is gaining momentum; it has been accepted more and more widely by enterprises. This business model offers dynamic flexible resources to its users on pay-as-you-use basis. At the time of resource allocation, user may send request for multiple resources simultaneously, thus a provision is required for optimal allocation of resources. The aim here is that the provider should render the desired services to user and the user should have reliable and guaranteed services as per the service level agreement (SLA). This paper focuses on resource allocation problem which addresses the optimum use and assignment of resources for particular task. This work explores the current resource scheduling algorithms employed by cloud providers. In this review, the algorithms are divided according to their nature and categorized as dynamic scheduling algorithms, agent based scheduling algorithms and cost optimization based scheduling. Various algorithms falling in each category have been discussed and a comparison among them is being performed.

Keywords- Cloud computing; Optimization; Resource allocation; Service provisioning; Scheduling algorithms.

INTRODUCTION

Nowadays the virtualization is becoming the need of industry. Cloud computing (CC) is the best way to avail the benefit of virtualization. Though CC is one of the latest domain of research interests, but at core it is not something new really. CC may be compared with post-paid mobile phones where user may avail the subscribed services and pays based on the usage. User is unaware of the technical details of establishing network and service provisioning. All those details are handled by the service provider. Similar concept is core of CC, where user now enjoys unlimited services on computer system without actually owning them[33]. Thus an important concern here is service provisioning at the service provider's end. Being commercial, optimal allocation of resources is of utmost interest for the provider[11,12]. Whereas assurance of QoS and sticking to SLA are major concern for the consumer. The term resource specified in this paper is described as utilization of virtual machines (services) which is allocated by service providers. Resource scheduling[13] is the mechanism of mapping between on-demand request of cloud user and the resources available at the data center. Many researchers have proposed different scheduling algorithms[14] for service provisioning in this domain. However traditional job scheduling algorithms might also be applied in CC, researchers have presented strategies incorporating parameters related to cloud environment.

Main objective of this paper is to explore various service provisioning algorithms[18,28,30] and draw a contrast among them so as to arrive at a conclusion about best available strategy. Rest of the paper is organized as follows: section 2 presents the existing scheduling algorithms in cloud environment. Section 3 depicts the role of those algorithms in present scenario of cloud computing. Section 4 provides a comparative analysis of various existing scheduling algorithms. Section 5 concludes the paper.

SCHEDULING ALGORITHMS IN CLOUD COMPUTING

The need of scheduling was felt in technological transition of traditional computer system into modern system of CC. Due to inherent multitasking in present user requests, their scheduling becomes essential for CC. Scheduling aims to balance the load along with efficient execution of requests leading to improved QoS. Also it minimizes resource starvation and ensures fair resource allocation. It focuses on the problem of decision making about the outstanding requests and which request has to be allocated resources next. Resource allocation[5,6,14] is used to assign the resources in an optimal way. CC also needs similar mechanism to ensure optimal resource allocation and minimum wait time for the users. Some advances have already been made in this direction. Upcoming subsections elaborate existing resource allocation algorithms that are currently being deployed in cloud computing. This paper divides the existing scheduling algorithms in following three categories:

Category I: Dynamic Scheduling Algorithms.
Category II: Agent Based Scheduling Algorithms.
Category III: Cost Optimization Based Scheduling Algorithms.
All of these categories are discussed in detail one by one.

Category I: Dynamic Scheduling Algorithms

A. Linear Scheduling for Tasks and Resources (LSTR)

This technique[1] focuses on distribution of resources among the requesting users so as to maximize the QoS (Quality of Service) parameters. In this algorithm the requests are collected and then sorted in different queues according to their threshold value (Th). Here threshold value is taken as Th= $\sum R_i$ (Resources) i.e. summation of all requested resources. If the requested (RQ) is less than the threshold value, the request is placed in low array A [RQ] else, in high array B[RQ]. Resources are first allocated to requests in low array A[RQ], and again the process is repeated by continuously calculating the threshold value. The same mechanism is applied in B [RQ]. This strategy improves the resource utilization results as compared to FIFO.

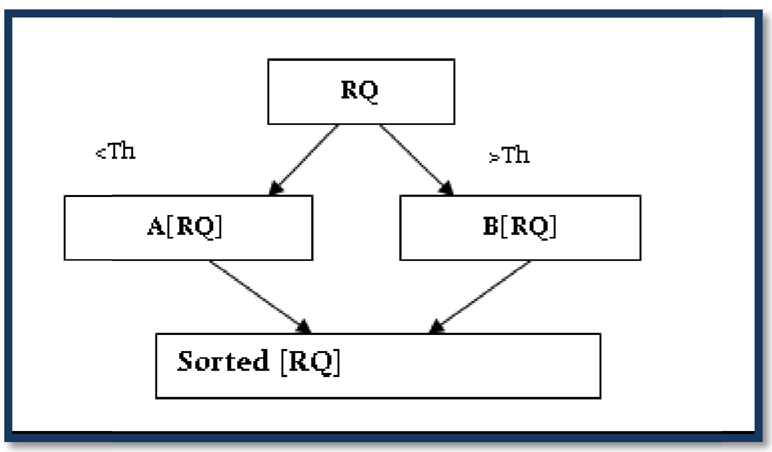

Fig. 1 LSTR Scheduling Strategy[1]

B. ProfminVm Max Avai Space and ProfminVm Min Avai Space

Two resource scheduling algorithms had been presented in[29], first algorithm ProfminVm Max Avai Space is based on maximizing the profit by minimizing the cost. For this purpose VM space is being reused. When a request arrives and all VMs are already been used, the one having largest free space is used for accommodating the incoming request. Fig. 2 (a) given below indicates that all VMs are being used.

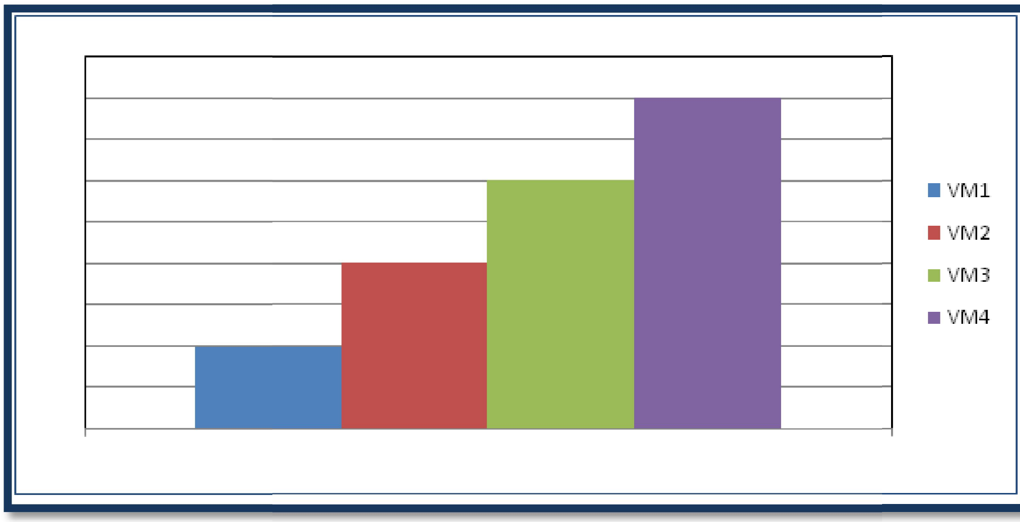

Fig. 2(a) Example Profmin Vm Max Avai Space[29]

On arrival of new request it has been placed in VM 1 having the maximum free space available shown in figure 2(b).

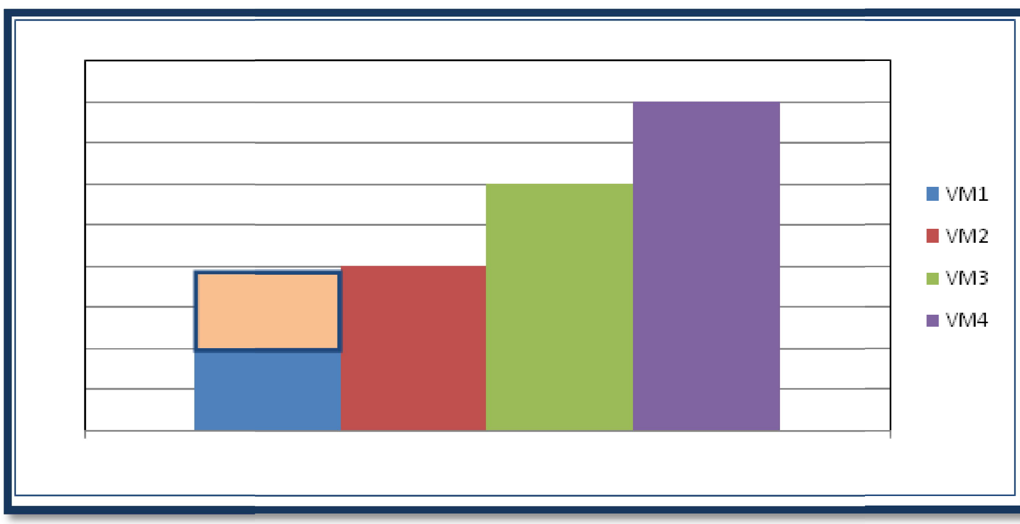

Fig. 2(b) Example Profmin Vm Max Avai Space[29]

The above algorithm has the drawback of reducing the free space available at maximum point, so if in future request for VM space more than smallest unit arrives there are less chances of being able to serve it.

ProfminVm Min Avai Space maximizes the profit by minimizing the cost by reusing VM. For this purpose it choose the VM having the smallest free space available to meet the request. This strategy provides better space utilization than the former algorithm.

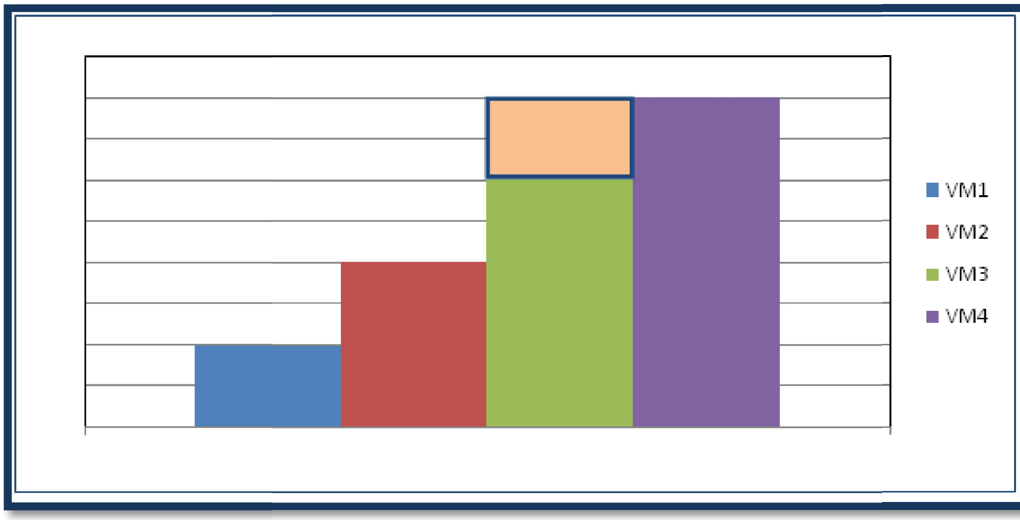

Fig. 2(c) Example Profmin Vm Min Avai Space[29]

C. Round Robin Based Resource Selection

The algorithm in[8] uses Round Robin algorithm as resource selection mechanism.

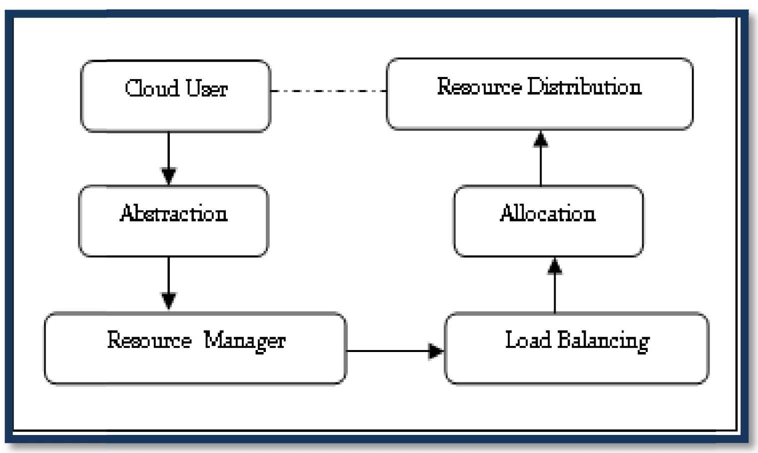

Fig. 3 Workflow for service based selection framework[8]

In this mechanism a weight has been assigned to every service request and requests leave the waiting queue based on their burst time for future execution. The request is sent by end user using cloud service abstraction to resource manager. The resources are allocated by balancing the load using load sharing algorithm

D. MCDA using PROMETHEE Method

This method[2] is based on distributed architecture where resource management is divided into independent tasks, each of the tasks being performed by autonomous node agents. These agents perform configurations as a change in resource requirements in parallel through multiple Criteria Decision Analysis. These changes would involve redistribution of resources and provide resources for other requests if it is not being utilized by anyone. This algorithm was implemented in C-language and results were compared with simple distributed strategy in terms of CPU allocation and physical machines.

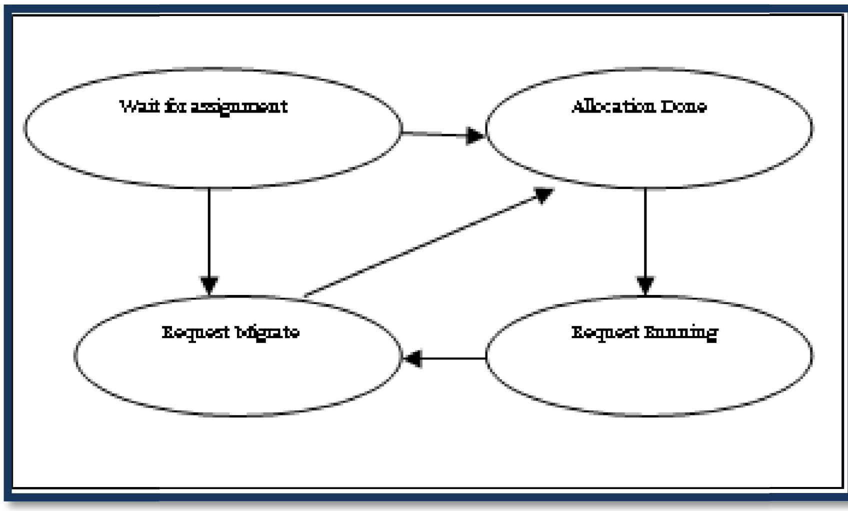

Fig. 4 Task Life cycle[2]

The above figure is indicating the life cycle of a task being received from end user for execution. There are different phases of a task life cycle such as unassigned at the initial stage, allocated after resource allocation, running while in execution, migrating & terminating.

E. Dynamic Resource Allocation Using Virtual Machines (DRM-VM)

This work presents[30] a system architecture in which virtualization is used for dynamic allocation of resources in a data center. The concept of skewness has and overcome the load. Figure 5 shows the proposed system architecture:

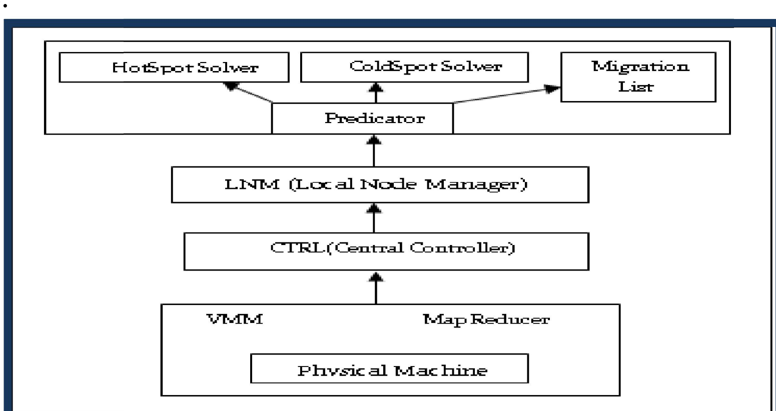

Fig. 5 Proposed System Architecture[30]

The proposed system architecture consists of a physical machine, central controller (CTRL), local node manager (LNM) and scheduler. It firstly measures the configurations of physical machine and forwards it to CTRL. Predicator scheduler invokes and receives the resource demands and capacity of VM from LNM. Hotspot solver is automatically invoked if the resource utilization rate becomes more than the hot spot threshold. Then some running VMs are migrated to reduce the load. Cold spot solver is automatically invoked in vice-versa condition then unused VM will shut down for some time to save energy.

Category II: Agent Based Scheduling Algorithms

A. Negotiation Strategy for Buyer and Seller

This strategy[4] proposed the automated negotiation mechanism where agents negotiate on both a contract price and de-commitment penalty which allows an agent to withdraw from the contract price. In this mechanism the buyer sends the offer to seller. If the seller accepts the offer from buyer then firstly seller establishes a tentative contract and after finalization the contract is established between both. This work compared their mechanism with the fixed price model of Amazon's Elastic Compute Cloud.

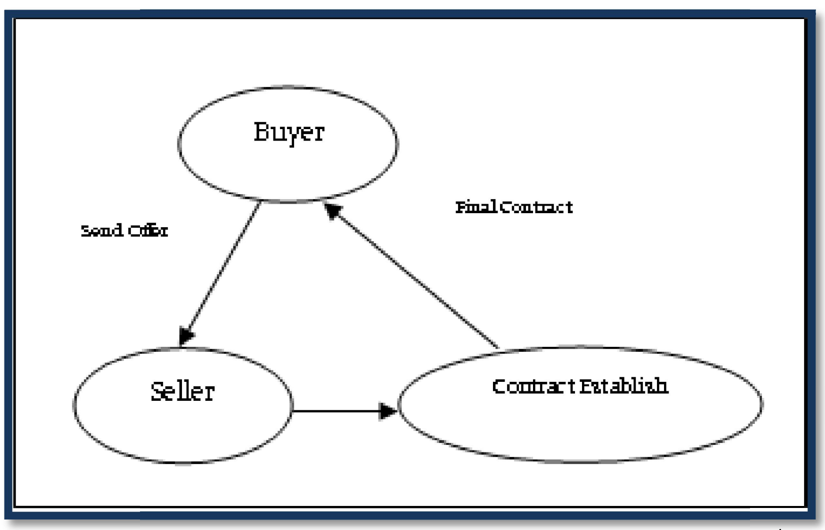

Fig. 6 Finite State Machine for the Negotiation Protocol[4]

B. Intelligent Multi Agent for Virtualization (IMAV)

The proposed model is responsible for distribution of resources in real time and adjusts them according to the behavior of cloud users[22]. Figure 7 describes the proposed IMAV platform.

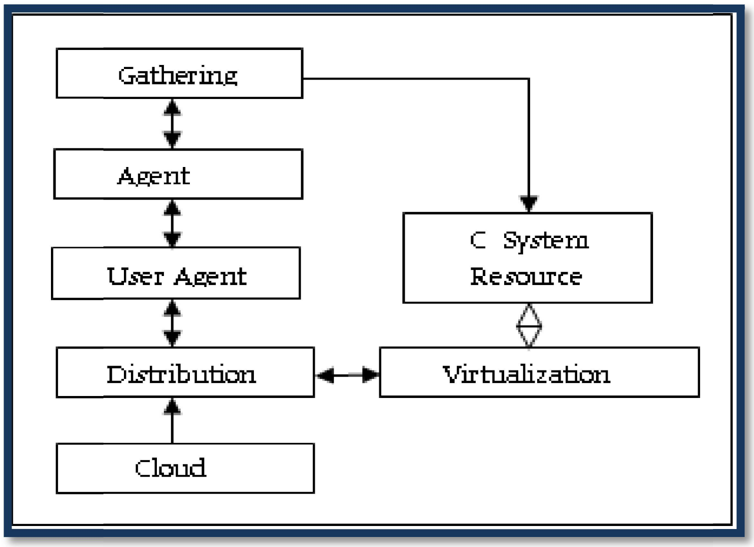

Fig. 7 Intelligent Agent for Virtualization Platform[22]

The proposed IMAV platform works as follows: gathering agent is responsible to collect service information for system. It uses the search engine for gathering information. Agent manger manages the creation and registration of all agents. It is responsible for managing their activities. The user agent receives the information from agent manager and analyzes service types and access type desired in services. Distribution agent allocates cloud resources and virtual machines to the user. Virtualization register manages and registers allocation information of all resources being distributed by distribution agent. It regularly analyzes the log information to analyze system performance and usage.

C. Agent Based Scheme Architecture

This architecture[37] is used to discover service, provides selection method and allocates resource in wireless platforms. The main objective of this technique is to provide optimization in resource scheduling and ensuring quality of service through intelligent agents.

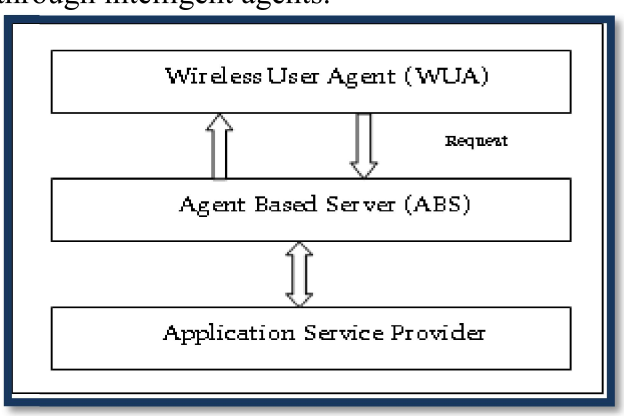

Fig. 8 Agent Based Scheme Architecture for Wireless Cloud[37]

Wireless user agent (WUA) manages all application requirements and delivers services in wireless environments. It also handles the issue of optimization of resources. Agent based server (ABS) collects the resource information from application service providers and maintains updated information of all resources being listed. Application service provider deals with the request coming from ABS and WUA and is responsible for actual delivery of desired resources.

D. Multi Agent System Architecture

This paper[15] presents a self organizing agent based service composition framework (shown in figure 9) n which each cloud participant and resources were presented by an agent. This composition is supported by two techniques: one is an acquaintance network which depicts all agents and their capabilities and another is contract net protocol which is used for dynamically selecting services.

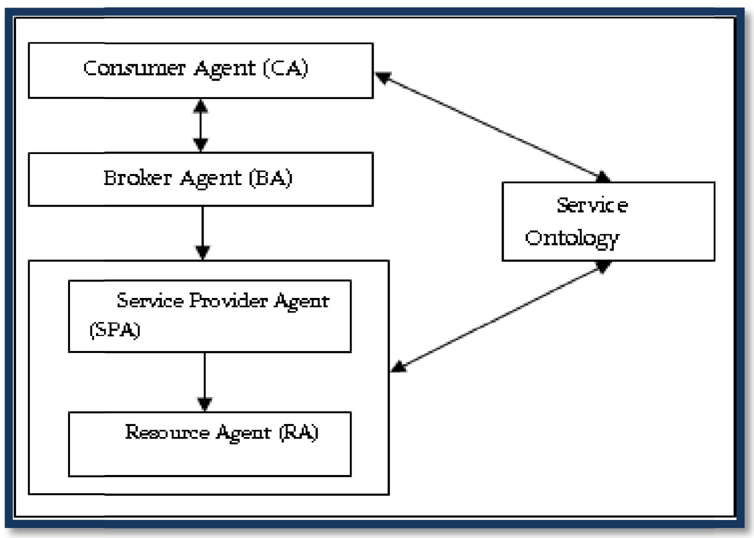

Fig. 9 Multi Agent system Architecture[15]

Consumer Agent (CA) is responsible for formalizing consumer requirements. Broker Agent (BA) is the mediator between cloud consumer and service provider agent (SPA), BA provides estimated quotations to consumer for their demands. If the quotation is accepted by consumer then it forward to SPA. SPA is the combination of resource agents (RA). RA manages the web services and accepts the request from SPA, once requirements may be satisfied it communicates the same to the end user.

E. Agent based Resource Allocation Model(ARAM)

Manvi *et.al* in[23] has presented an agent based model which involves migration of agents based on topology, analysis of different topologies, agent overheads, dynamic pricing & negotiation. The ARAM system model works as follows: The local cluster server send the resource status information to their master cluster server. The master cluster servers exchange this information with the grid information server and create a database in the server, which contains the information about the status and cost of resources. This information arrives at resource broker with complete detail of requirements. If present broker is unable to fulfill the demand then it migrates to another resource broker. The allocation information is send to response master cluster server, further it sends that information to local cluster server.

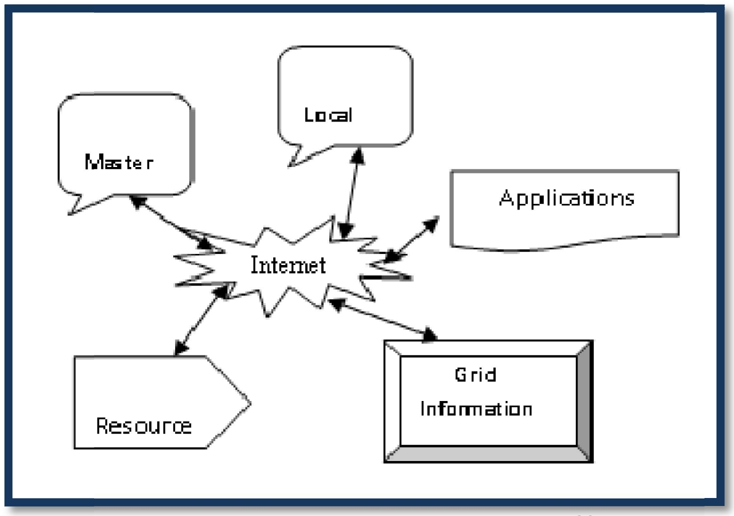

Fig. 10 System Model of ARAM[23]

F. Adaptive Resource Allocation Model

This work[20] presented an adaptive mechanism for resource provisioning based on network delay i.e. geographical distance between data center, customer and workload of each data center. The result of adaptive model provided better response time as compared to linear and random models.

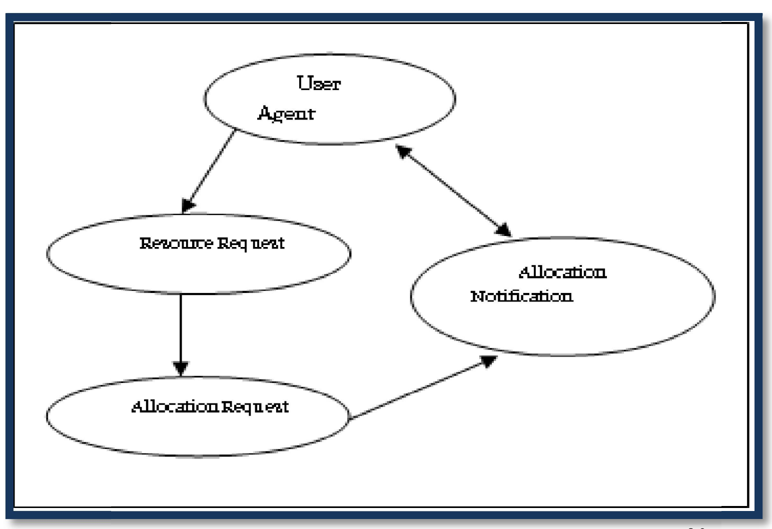

Fig. 11 The Overview of Agent Based Test Bed[20]

G. Market Based Model

In[34] allocation and pricing mechanism is being described as a market based model for resource. Service providers use k-pricing and buyers submit their bids correspondingly. Buyers and service providers interact through the trade exchange mechanism. However this model doesn't provide scalability.

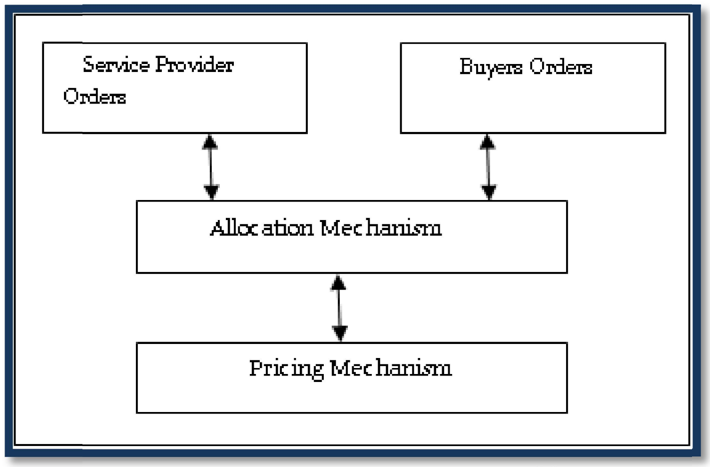

Fig. 12 Market Based Model[34]

Category III: Cost Optimization Based Scheduling Algorithms

A. Lightweight Scaling Algorithm

In this algorithm[25] the Iaas providers specify the running cost of a single server, which represent the resource consumption and operating cost. The cost is decided on the basis of resource usage per unit. The lightweight scaling algorithm has two fold meaning. Firstly the proposed algorithm presents scaling by modifying resource configuration of each virtual machine. Secondly the algorithm applies an automatic reactive scaling mechanism. When an application is scaled up or down then the load balancing server automatically redistributes the incoming request and updates response time.

B. Optimal Resource Allocation Technique (ORAT)

This paper proposed[17] an optimal resource allocation method which considered two types of resources: bandwidth and processing capabilities. The working of ORAT is given below:

There are number of servers present in the proposed architecture involving optimization techniques for resource allocation.

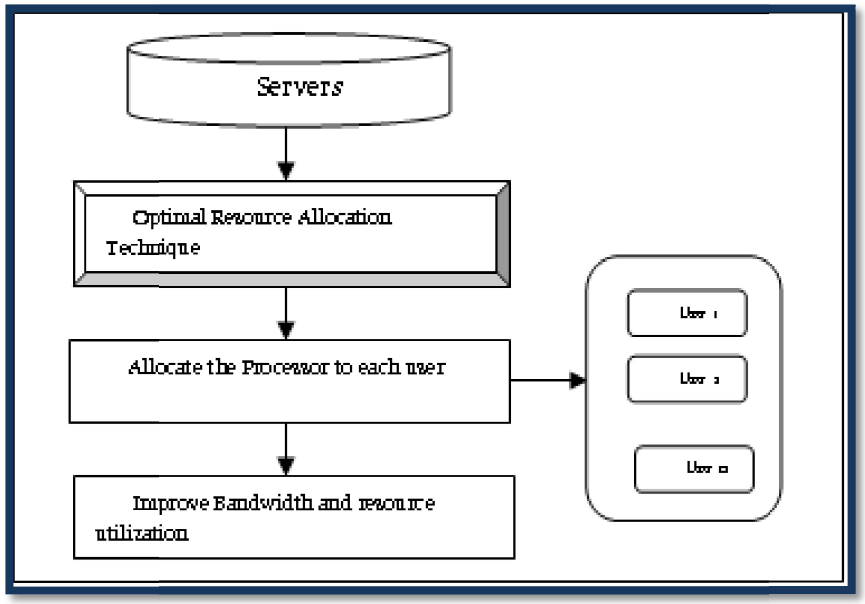

Fig. 13 ORAT Architecture[17]

These servers apply their techniques on each user request and allocate resources to them. After every allocation the server update the status of bandwidth and resource utilization. If there is need to improve them then it improves automatically at every step.

C. Compromised–Time-Cost Scheduling Algorithm (CTC)

CTC algorithm[21] focuses on two main factors: execution cost and execution time. This algorithm takes into consideration sharing, conflicting and competition services used by multiple concurrent instances which are running on highly dynamic cloud computing platform. The pre step of CTC algorithm is to check for incomplete tasks and to schedule them.

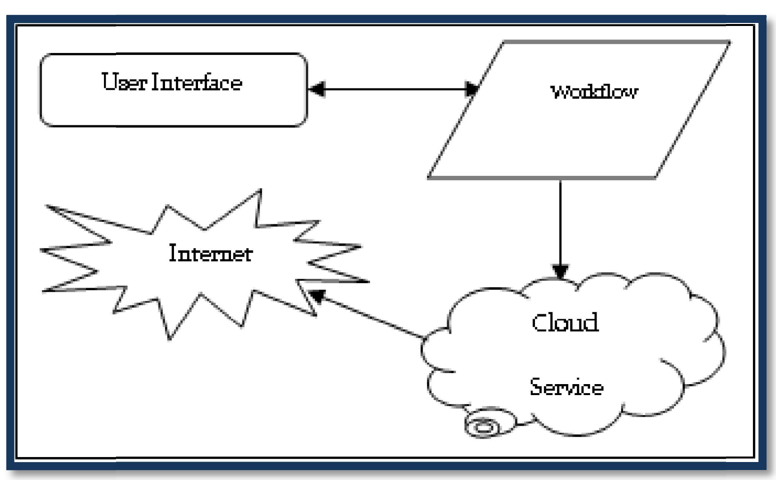

Fig. 14 Architect of Swine DeW-C[21]

It then calculate the deadlines for every task and estimate the execution time and execution cost of each service. Depending on the lowest cost and execution time each task is allocated to a service. CTC algorithm is based on SwineDeW-C (Swinburne Decentralized Workflow for Cloud) architecture.

Next section presents the role of these algorithms in present scenario of cloud computing.

ROLE OF SCHEDULING ALGORITHMS IN CLOUD COMPUTING

Above discussed resource scheduling algorithms have been categorized based on their dimension of focus in CC. Here an attempt is being made to highlight scope of these algorithms.

Dynamic scheduling algorithms focus on incorporating dynamic change in number of users and types of resources requested by them. LSTR aims to reduce the starvation and deadlock conditions which improves the performance of cloud resources and yields high system throughput. ProfminVmMaxAvaiSpace and ProfminVmMinAvaiSpace techniques minimize the cost of resources and ensure that providers were able to manage the dynamic change in number of customers. Round Robin scheduling algorithm enables the user to access the services in a workflow manner increasing the efficiency of resource usage and avoiding starvation of resources.

Agent based scheduling algorithms focus on scalability of services that can be achieved with the help of mobile agents. Literature review highlighted that researchers are inclined towards deploying intelligent agents in CC. MCDA method deploys autonomous node agents to provide scalability, flexibility and feasibility. IMAV architecture focuses on real time distribution of resources adapting to cloud user behavior. Agent based architectures are focusing on providing QoS, adapting user behavior and dynamic service compositions. They have potential to manage load balancing with reduced migration cost.

Algorithms in the third category are focusing on cost optimization with minimum response time.

Next section provides comparison of all the above discussed algorithms based on their parameters.

COMPARATIVE ANALYSIS OF EXISTING ALGORITHMS IN CLOUD COMPUTING

Above sections have laid down foundation for existing resource scheduling algorithms and discussed their working. Next we would draw a comparative analysis of these algorithms.

Since most of above discussed algorithms are deploying diverse parameters. For the purpose of comparison we have chosen resource utilization, virtual machine (VM) cost and load balancing features where:

- ***Resource Utilization:*** A good resource scheduling algorithm should optimize resource utilization.
- ***Virtual Machine Cost:*** Per user cost of VM should be minimized through proper sharing strategy.
- ***Load Balancing:*** Balancing load among available VMs reduces chances of deadlocks and improves system throughput.

Table 1 given below provides a summarized analysis of resource scheduling algorithms based on above parameters.

From Table 1 it is clear that none of the existing algorithm satisfies all three parameters taken into consideration. Thus existing resource scheduling algorithms have scope for further improvement so as to provide optimized resource scheduling.

Abbreviations used in Table 1 are as follows:

IMAV: Intelligence Multi-Agent for Virtualization
ARAM: Agent based Resource Allocation Model
MCDA: Multiple Criteria Decision Analysis
LSTR: Linear Scheduling for Tasks and Resources
DRM-VM: Dynamic Resource Allocation Using Virtual Machines
ORAT: Optimal Resource Allocation Technique
CTC: Compromised–Time-Cost Scheduling Algorithm

Table 1. Summarized Analysis of Resource Scheduling algorithm in CC

Resource Schduling Algorithm	Algorithm Parameters	Tool Used	Resource Utilization	VM Cost?	Load Balancing?
LSTR[1]	Time, Resource Utilization	Nimbus, Cumulus	√	×	×
Profminvmmax Avaispace, ProfminvmMin Avaispace[29]	Response time, Service Initiation, Number of Initiation	CloudSim	√	√	×
Round Robin Based Algorithm[8]	Response Time, No. of services, CPU Usage	Resource Monitor	√	×	√
MCDA using PROMETHEE Method[2]	No. of iterations, No. of Physical Machines	C-programming	√	×	×
DRM-VM[30]	CPU Load, Time, Number of VM's	Xen Hypervisor	√	×	√
Negotiation Strategy of buyer & seller[4]	Success rate of buyer, No. of resource to acquire	GENI	√	×	×
IMAV[22]	User Context, System Context	MovieLens Dataset	×	×	×
Agent Based Scheme Architecture [37]	No. of Users, No. of successful request	Cloud Sim	×	×	√
Multi Agent System Architecture[15]	Probability of Failure, No. of successful completion	JADE	√	×	×
ARAM[23]	Resource utilization, No. of JA's	C++	√	√	×
Agent – based Adaptive Resource Allocation[20]	Number of Consumer, Success Rate, Average Allocation Time	Agent Based TestBed	√	×	×
Market Based model[34]	Trading Between buyer and Seller	Software Agent Simulation	×	×	√
Lightweight Scaling	Number of Server, Bandwidth, Time,	DataCenter on IC-Cloud Platform	×	√	√
ORAT[17]	Resource Utilization, Data Transfer Rate, Number of Requests, Request Loss Probability	CloudSim	√	×	×
CTC[21]	Execution Cost, Execution Time	DAG (Database Availability Group)	×	√	√

CONCLUSION AND FUTURE WORK

This work explored the resource scheduling algorithms presently existing and been employed in cloud computing applications. Increasing popularity of CC needs effective resource allocation strategies. From this survey it is clear that research community is paying attention on this aspect and lot of work is being done in this direction. We have tried to categorize existing algorithms based on point of focus and their nature. Also a comparison of existing resource scheduling algorithms is being made on the basis of three parameters significant in resource scheduling. From this analysis it has been observed that there is no one algorithm satisfying all three basic parameters chosen for analysis. Thus CC needs improvement in resource scheduling algorithms. There is need for a single, scalable, adaptive and optimized technique for resource allocation which could satisfy diverse user demands. Agent based techniques seems to be promising solution. Future work aims towards developing an optimized resource allocation technique for cloud computing environments.

References

1. Abirami SP, Ramanathan S(2012). Linear Scheduling Strategy for Resource Allocation in Cloud Environment. Published in *International Journal of Cloud Computing: Services and Architecture*, Volume 2, Number 1, pp. 9-17.
2. Akana CM, Kumar S, Divakar C, Satyanarayana Ch (2011). Dynamic Resource Allocation in Computing Clouds through Distributed Multiple Criteria Decision Analysis using PROMETHEE Method. Published in *International Journal of Advanced Networking and Applications*, Volume 03, Issue 02, pp. 1060-1069.
3. Alexandru I, Simon O, Nezih M, Radu P, Thomas F, Epema H (2011). Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing. Published in IEEE Transactions On Parallel And Distributed Systems, Volume 22, Number. 6, June, pp 931-945.
4. An B, Lesser V, Irwin D, Zink M (2010). Automated Negotiation With Decommitment for Dynamic Resource Allocation in Cloud Computing. Published in proceddings of 9th *International Conference on Autonomous Agents and MultiAgent Systems* (AAMAS-2010), pp 981-988.
5. Andr´e R, Luiz S, Haroldo G (2008). New effective algorithm for Dynamic Resource Constrained Project Scheduling Problem. Published in proceedings of *International Conference on Engineering Optimization*, Brazil, pp 641-653.
6. Arfeen MA, Pawlikowski K, Willing A(2011). A Framework for Resource allocation Strategies in Cloud Computing Environment. Published in

proceedings in 35th *IEEE Conference on Computer Software and Application* , July, pp. 261-266.

7. Bacigalupo DA, Hemert J, Chen X, Usmani A, He L, Donna N, Wills G, Gilbert L, Jarvis S (2011). Managing dynamic enterprise and urgent workloads on clouds using layered queuing and historical performance models. Published in proceedings of Simulation Modelling Practice and Theory, Volume 19, pp 14 79-1495.
8. Banerjee C, kundu A, Bhaumik S, Babu RS, Gupta RD(2012). Framework on Service based Resource Selection in Cloud Computing. Published in *International Journal of Information Processing and Management*, Volume 3, Number 1, pp. 17-25.
9. Baomin X, Chunyan Z, Enzhao H, Bin H(2011). Job scheduling algorithm based on Berger model in cloud environment. Published in Elseivier Advances in Engineering Software, Volume 42, pp 419–425.
10. Battr´e D, Ewen S, Hueske F, Kao O, Markl V, Warneke D(2010). A Programming Model and Execution Framework for Web-Scale Analytical Processing. Published in proceedings of the ACM Symposium on Cloud Computing, New York, pp 119–130.
11. Chen K, Shen M, Zheng W (2005). Resources Allocation Schemas for Web Information Monitoring". Published in *Tsinghua Science and Technology*. Published in Volume 10, Number 3, pp 309-315.
12. Daniel W, Odej K (2011). Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud. Published in *IEEE Transaction on Parallel and Distributed System*, January, pp 459-464.
13. Dinesh K, Poornima G, Kiruthika K (2012). Efficient Resource Allocation for Different Jobs in Cloud. Published in *International Journal of Computer Applications*, Volume 56, Number 10, pp. 30-35.
14. Fan CT, Wang WJ, Chang YS (2011). Agent – based Service Migration in Hybrid Cloud. Published in *IEEE International Conference on High Performance Computing and Communications*, pp. 887-892.
15. Garcia JO, Sim, KM (2010). Self-Organizing Agents for Service Composition in Cloud Computing. Published in 2nd *IEEE International Conference on Cloud Computing Technology and Science*, pp. 59-66.
16. Garg SK, Buyya R, Siegel H(2010). Time and Cost Trade-off Management for Scheduling Parallel Applications on Utility Grids. Publshed in *Future Generation Computer Systems*, Volume 26, Number 8, pp. 1344-1355.

17. Giridas K L, Shajin N (2010). Optimal Resource Allocation Technique (ORAT) for Green Cloud Computing. Published in *Journal of Computer Applications*, Volume 55, Number 5, pp 20-26.
18. Iqbal W, Dailey MM, Carrera D, Janecek P (2011). Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Published in *Future Generation Computer Systems*, Volume 27, pp. 871-879.
19. Jianhua G, Jinhua H, Tianhai Z, Guofei S (2012). A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment. Published in *Journal of Computers* Vol. 7. No. 1, January, pp 42-52.
20. Jung G, Sim KM(2011). Agent based Adaptive Resource Allocation on the Cloud Computing Environment. Published in *IEEE conference on Parallel Processing Workshop*, 2011, pp. 345-351.
21. Ke L, Hai J, Jinjun C, Xiao L, Dong Y, Yun Y(2010). A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on Cloud Computing Platform. Published in *International Journal of High Performance Computing Applications*, Volume 24, Issue 4, pp 445-456.
22. Kim M, Lee H, Nam W (2011). A Model of Multi-Agent Design for Virtualization Resource Configuration in Cloud Computing. Published in 1st *IEEE ACIS International Conference on Computers, Network, Systems, and Industrial Engineering*, pp. 234-239.
23. Manvi SS, Birje MM, Prasad B(2005). An Agent – based Resource Allocation Model for Computational grids. Published in Multiagent and Grid System- An International Journal, pp. 17-27.
24. Mihailescu M, Teo YM(2010). Dynamic Resource Pricing on Federated Clouds. Published in proceedings of 10th *IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne*, Australia, pp. 513-517.
25. Rui H, Li G, Moustafa MG, Yike G(2012). Lightweight Resource Scaling for Cloud Applications. Published in proceeding of IEEE/ACM *International Symposium on Cluster Cloud and Grid Computing*, pp 644-651.
26. Streitberger W, Eymann T, Veit D, Catalano M., Giulioni G., Joita L, Rana O,F(2007). Evaluation of Economic Resource Allocation in Application Layer Networks a Metrics Framework. Published in eOrganisation: Service-, Prozess-, Market-Engineering, pp 447- 494.

27. Subramoniam K, Maheswaran M, Toulouse M(2002). Towards a micro-economic model for resource allocation in grid computing systems. Published in proceedings of the *IEEE Canadian conference on electrical and computer engineering*. Manitob, pp 782–785.
28. Weiwei L, James Z, Chen L, Deyu Q(2011). A threshold-based dynamic resource allocation scheme for cloud computing. Published in *Sci., Journal Procedia Engineering*, Volume. 23, pp 695-703.
29. Wu L, Garg SK, Buyya R(2011). SLA-based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments. Published in 11th *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing,* pp 195-204.
30. Xiao Z, Song W, Chen Q(2013). Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment. Published in *IEEE TRANSACTION ON PARALLEL AND DISTRIBUTED SYSTEMS (TPDS)*, June, Volume 24 Number 6, pp. 1107-1117.
31. Xin S, Sen S, Peng X, Shuang K, Yan L(2011). Multi-dimensional resource integrated scheduling in a shared data center. Published in *IEEE conference on Distributed Computing Systems Workshops* (ICDCSW), pp 7-13.
32. Xindong Y, Xianghua X, Jian W, Dongjin Y(2009). RAS-M: Resource allocation strategy based on market mechanism in cloud computing. Published in *chinagrid, Fourth China Grid Annual Conference*, pp 256-263.
33. Yang H, Tate M(2009). Where are we at with Cloud Computing?: A Descriptive Literarure Review. Published in proceedings of 20th *Australasian Conference on Information Systems*, pp 807-819.
34. Yee-Ming, C, Hsin-mie, Y(2010). Autonomous Adaptive Agents for Market-Based Resource allocation of Cloud Computing. Published in proceedings of the 9th *International Conference on Machine Learning and Cybernetics*, Qingdao, 11-14 July, pp. 2760-2764.
35. Zhang Q, Cheng L(2010). Cloud computing: State-of-the-art and Research Challenge. *Raouf Boutaba J Internet* Serv Appl, pp 7-18.
36. Zheng Y, Vasilakos VA, Wei G, Xiaong, N(2009). A Game –Theoretic Method of Fair Resource Allocation for Cloud Computing Services. Published in Springer Science + Business Media, LLc, pp 252-269.
37. Zhou Y, Yang Y, Liang L, Sun Z(2010). An Agent-Based Scheme for Supporting Service and Resource Management in Wireless Cloud. Published in proceedings of 9th *IEEE International Conference on Grid and Cloud Computing*, UK, pp 34-39.