ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

A Survey on Resource Allocation and Task Scheduling Algorithms in Cloud Environment

M.Gokilavani, S.Selvi, C.Udhayakumar

***Abstract*— Cloud computing is a computing service paradigm that charges under the basis of the amount of resources consumed i.e. pay per use constraint .One key characteristic that differentiate cloud computing from the other enterprises computing is that the infrastructure itself is programmable. An important issue faced by Infrastructure as a service (IaaS) in task scheduling and resource allocation is a NP-Complete problem. Although, there are various algorithms and methods were existing to solve the problem of resource allocation but none of these algorithms can be extended. The present study involves surveying the different task scheduling and resource allocation algorithms for cloud. Comparing the various algorithms, we conclude that various scheduling algorithms depend on the type of the task to be scheduled.**

Index Terms—**cloud computing, Infrastructure as a Service (IaaS), Task Scheduling, Resource allocation.**

I. INTRODUCTION

Cloud computing is everywhere. The term "cloud", appears to have its origins in network diagrams that represented the internet [18]. Recently, "Cloud computing" means the applications and services are moved into the internet "cloud" is not something that when computer systems remotely shared resources in timely manner [1]. The devices used to access these resources do not have any specific limitations. Cloud provides provide various types of services such as SaaS, IaaS, and PaaS according to their user requirements. SaaS allows Consumers to purchase an authorization to access and use a resource that is hosted in the cloud. A benchmark example is that information interaction between the consumer and the service provider is hosted as one of the resources in the cloud. Microsoft is expanding its involvement of the area of cloud computing option for Microsoft Office 2010. Its Office web Apps is available to Office volume licensing customers and Office Web App subscriptions through its cloud-based online services. PaaS consumers purchase access to the platforms, enabling them to develop their own software and applications. The operating systems and network access are not managed by the users, and there should be certain parameters as to which applications can be developed. IaaS controls the user and manages the system in terms of bandwidth, response time resource expenses, and network connectivity, but do not concentrate on infrastructure. Scheduling is a very important problem in cloud computing environment. Scheduling becomes difficult only when the number of users in cloud gets increased. Therefore, there is a need to go for a better scheduling algorithm than existing one. This can be done by comparing and evaluating the various existing algorithms, thereby identifying the loop holes in the existing algorithms. There are different algorithms for Resource Allocation System (RAS) and Task Scheduling in cloud differs according to the task accessed in Environment. Various algorithms are Survey in following sections.

II. LITERATURE REVIEW

A. Cloud Environment Infrastructure Architecture

Cloud users combine virtualization, automated software, and internet connectivity [13] to provide their services. A basic element of the cloud environment is client, server, and network connectivity [11]. A hybrid computing model allows customer to leverage both public and private computing services [3] to create a more flexible and cost-effective computing utility. The public cloud environment involves Web based application, Data as a service (DaaS), Infrastructure as a Service (IaaS), Software as a service (SaaS), and Email as a service (EaaS). A private cloud accesses the resources from the public cloud organization to provide services to its customers. In a hybrid cloud environment, an organization combines various services and data model from various cloud environments to create an automated cloud computing environment.

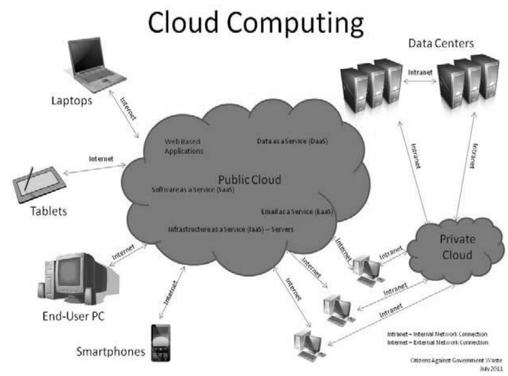

Fig 1: Cloud Environment Infrastructure Architecture

B. Infrastructure as a service (IaaS)

Infrastructure as a service (IaaS) controls user and manage the systems in terms of the bandwidth, response time, resource expenses, and network connectivity, but they need not control the cloud infrastructure. Some of the key features of IaaS such as cloud bursting, resource gathering etc. differ according to the cloud environment. The greatest value of IaaS is mainly through a key element known as *cloud bursting*. The process of off-loading tasks to the cloud during times when the most compute resources are needed. However, for business IaaS takes an advantage in its capacity. IT companies [1], [2] able to develop its own software and implements that can able to handles the ability to re-schedule resources in an IaaS cloud. Elasticity is the first critical facet

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

of IaaS. IaaS is easy to spot, because it is typically platform-independent. IaaS consists of a combination of internal and external resources. IaaS is low-level resource that runs independent of an operating system called a *hypervisor* and is responsible for taking rent of hardware resources based on pay as you go basics. This process is referred to as *resource gathering [2]*. Resource gathering by the hypervisor makes virtualization possible, and virtualization makes *multiprocessing computing* that leads to an infrastructure shared by several users with similar resources in regard to their requirements.

C. Task Scheduling and Resource Allocation

To increase the flexibility, cloud allocates the resources according to their demands [4]. Major problems in task scheduling environment are load balancing, scalability, reliability, performance, and re-allocation of resources to the computing nodes dynamically. Resources such as internal and external requirements are maintained only in the cloud environment not on the primary environment. For the efficient use of such resources the users' needs the efficient scheduling algorithms to deal with such inefficiency. In past days, there are various methods and algorithms to solve the problem of scheduling a resource in Preempt able Job [16] in cloud environment. A task is an action that takes resources as an input to produce the efficient output in computation nodes. In cloud environment, resources are allocated to the customers under the basics of pay per use on demand. Algorithms used in the allocation of the resources in cloud computing environment differ according to schedule of task in different environment under different circumstances. Dynamic load balancing [5] in cloud allocates resource to computational node dynamically. Task Scheduling algorithms aim at minimizing the execution of tasks with maximizing resource usage efficiently. Rescheduling is need only when the customer's request the same type of resources. Each and every task is different and autonomous their requirement of more bandwidth, response time, resource expenses, and memory storage also differs. Efficient scheduling algorithms maintain load balancing of task in efficient manner. Efficiency of cloud environment only depends on the type of scheduling algorithm used for task scheduling.

D. Scheduling and Resource allocation algorithms

a. Gossip protocol [7]

Cloud environment differs in terms of services and applications provided by cloud. The problem of resource management for large systems is focused and general Gossip protocol is proposed for fair resource allocation of CPU resources to customers. A gossip-based protocol for resource allocation in large- cloud environments is proposed in [7]. It performs a major function within distributed middleware architecture for large clouds. This paper, considered the system as a dynamic set of computational nodes that represents the system of cloud environment. Each node has an independent internal capacity. This algorithm implements a distributed scheme that distributes cloud resources to a set of computational nodes that are time-dependent with memory requirements and maximized with the global resource usage. The simulation results show that the protocol produces optimal resource allocation when memory requirement is smaller than the memory in the cloud environment. In paper [8], proposed protocol for resource allocation in a cloud environment, called P*. It is based on a heuristic algorithm for solving OP (2) and is implemented under the basics of a gossip protocol. P* has the construction of a round based distributed algorithm. When executing this gossip protocol, each node selects its own set of other nodes interact each other through selection function .Interaction taken place through messages, which are executed and changes taken places locally. Node communication with P* follows under push-pull prototype to exchange information and updates in local states under the structure of round based distributed systems. Compared with other protocols gossip protocol is scalable and robust. P* is executed in all systems of cloud because systems are in distributed environment. During initialization, the resource manager implements a feasible cloud configuration A, then it invokes P* to compute for global utility. If the protocol introduce new configuration called matrix A, resource manager check whether the new configuration gain the utility. Compare the new configuration with computed configuration weight of cost causes small change. If this is the case, then the resource manager implements the new configuration. The protocol P* takes as input from cloud resources, under new configuration and demand and get interaction with other set of system in distributed environment. The basic gossip protocol follows the concept that each system has their incomplete view of the cloud environment at any time. A system is aware of the resources it executes, and it maintains a row of the new matrix A that relates to the allocation of its own resources. Similarly, the new matrix A is distributed across the systems of the cloud. P* is designed to run continuously in an asynchronous environment where a system does not synchronize the start time of a protocol round with any other system. Further, a system coordinates an update of the new matrix A only with one additional system at a time, namely its current interaction system in the gossip protocol. Therefore, during the evolution of the system, the implemented cloud new matrix A changes dynamically and asynchronously, as each system maintains its part of the configuration. P* is based on a gossip protocol that computes the global average of local node variables. Every iteration of this protocol, a node average sits local value with that of another node selected uniformly at random. The authors show that each local value converges exponentially fast to the global average which has been extended by presents a distributed middleware for application placement in data centers. The goal of that work is to maximize a cluster utility under changing demand, although a different concept of utility is used. In contrast, our approach guarantees that every system receives its fair share of the CPU resources from cloud environment under user demands. The proposed design

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

in this paper scales with the number of systems, but it does not scale in the number of applications.

b. Bee's algorithm [6]

Bee's algorithm in nature tracks the actions of bee to get their foodstuff. Initially they pick scout bee to a search a food areas, if that bee find the area with large foodstuffs informs the place and direction to the other bees to find the area .Some other selected bee's and scout bee's collected honey as a foodstuff from different places. Identically some other set of scout bees inform the location of foodstuffs from different direction. By applying this method to scheduling consider set of scout bees as (α) ,no of location informed to other bees as (β) , large foodstuff location out of all location as (n) , bees need for the large location as (Ω) , bees need for the other sites(β-n). Bee's algorithm for resource scheduling is as given below. The proposed algorithm for resource scheduling based on bees concept. In this paper [6] scheduler is used, which sends autonomous task to various nodes present in it group. Initially, the task is submitted to scheduler.

1) ***Select:*** The scheduler using a selection function to find a task which has smallest memory, I/O requirement and processor required to complete their task which will act as scout bees to find a location.

$$F(\alpha) = \min\{U^{\alpha} i=1 t(i)\} \qquad (1)$$

Where t(i) denotes the t^{th} task and function F (α) to determines the minimum resource requirement task.

2) ***Fitness:*** The minimum resource requirement task which acts as a scout bee is sent to the location at which task requires the resource at present .A scout identifies the location by using a fitness function which runs that task in a particular resource and if the evolution is made it is necessary to determine the resource requirement is either memory or processor dependent. Conceptually, fitness refers to how much progress each task is making with their allocated resources compared to the same task executing on the entire group. Therefore, it is between 0 and 1 the computing rate (CR) is used to calculate the fitness of a request to the task. Specifically, given that CR (q; t) is the computing rate of slot q for task t, the progress share where t_j is a set of slots running tasks t and q is the set of all slots.

$$F(t) = CR(q;t)/CR(q';t) \qquad (2)$$

3) ***Waggle:*** By identifying the location resources the scout's returns to scheduler and does the waggle function. Waggle function segregates the task present in scheduler based upon scout's information such as cost, memory and processor requirements. The combination takes place in such a way that the memory oriented information is passed to the memory oriented task for execution with fixed capacity. If a task exceeds capacity then the task has to wait until the scout task finds another resource available adjacent location.

$$W(n) = \{U_{\beta} i=1q(t) \in q\} \qquad (3)$$

Where t task's resources which is an integral multiple of scout task i.e. q(t).After waggle function the subsets of tasks are rendered to the desired location by scheduler and scout task sets course for another location . Hence after the fitness function have to execute only for the scout tasks in algorithm and the time is reduced significantly. By reducing time resources are efficiently allocated to computational nodes.

c. Ant colony optimization algorithm [11]

Naturally ants are behaviorally innocent insects with limited memory. Temporary they form a group to perform a task of collecting food with reliability and consistency. Like ants cloud computing performs a complicated task [11] [20] providing resource optimally to customers to solve the problem of in an efficient a manner.

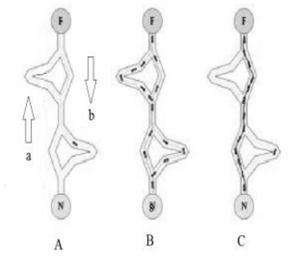

Fig 2: Ant colony optimization algorithm in cloud

Laterally this method in cloud computing become the field of *ant colony optimization* (ACO) act like an ant in cloud environment. Basic principles of ACO algorithm depending on the species ant behavior when ant moving from food to nest or from nest to food or in both direction. Cloud Resources Scheduling Based Ant Colony Optimization algorithm is a random search algorithm [10] [19], in *Travelling Sales Man* (TSP) problem given in n Place and the person starts from the point again reach the same point in shortest path. Given that m is the number of the ants' in ant colony, τxy (t) show that the place x and y the path concentration at time equal to the information, Set A=C (Constant), S express that at t time ant k from the position p transferred to q location. Simulation software Cloud Sim simulation experiment, says that, the algorithm for calculating node distribution and load balancing has good performance, but there is an increase in execution time. In paper [11] [20] Cloud Computing and Load balancing based on the algorithm ant colony optimization(ACO) algorithm .In addition to that, the load balancing technique that is based on Swarm intelligence tells about how the mobile agents can balance the load in cloud environment. The main objective of this paper is to develop an effective load balancing algorithm using Ant colony optimization technique to maximize or minimize different performance parameters like CPU load, Memory bandwidth, and Delay or network load including various features such as scalability, reliability and autonomy [12] for the different cloud environment. Similarly, a heuristic algorithm based on ant colony optimization has been proposed to initiate the service load distribution under cloud computing architecture and also update mechanism has been proved as an efficient and effective tool to balance the load.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

This paper supports to minimize the cloud computing based services and portability of servicing the request also has been converged using the ant colony optimization technique. Limitation is that the fault tolerance problems [21].

d. Bin-Packing algorithm [13]

Bin packing problems involve the packing of objects of given sizes into bins of given capacity. In the case of one-dimensional bin packing the size of each object is a real number between 0 and 1, and each bin is of same capacity. It is required that the sum of the objects packed into any given bin may not exceed 1. The problem of finding a packing using a minimum number of bins is known to be NP-hard. Bin packing problem (BPP) is, in a sense, complementary to the Minimum Make span Scheduling problem. In this problem, the goal is to schedule jobs of various lengths on a fixed number of system whereas minimizing the make span, or equivalently to pack items of various sizes into a fixed number of bins whereas minimizing the largest bin size. We now consider the problem [13] where we swap the roles of constraint and objective: all bins have a fixed size, and we wish to minimize the number of bins needed. The Bin Packing Problem (BPP) can be described using the terminology Knapsack problem, as follows. Given n items and n knapsack (or bins) with

$\mathbf{W_j}$ =Weight of the item j

C=capacity of each bin

Assign each item to one bin so that the total weight of the items in each bin does not exceed C and the number of bin used is a minimum. The three most common and simplest bin packing algorithms are

(1) *Next Fit* is a bounded space online algorithm in which the only partially filled bin that is open is the most recent one to be started. It uses one active bin into which it packs the resource. Once the zero space in this bin becomes too small to accommodate the further resources, a fresh dynamic bin is created and the previous active bin is closed. This process continues until the resource acquired.

(2)First Fit manages all Non-empty bins active and tries to pack every item in these bins before opening a new one. If there is a bin absent, it opens a new one and puts the resource in the new one.

(3)*Best Fit* picks the bin with the least amount of free space in which it can still hold the current resources. All non-empty bins are kept open until the resources filled. Let *SA* (*n*) denote the maximum number of storage locations (for active bins) needed by some algorithm A during the processing of the list L whose size is *n*, and refer to algorithm A as an *SA* (*n*) space complexity algorithm. Next Fit is an *O* (1) space complexity algorithm, since it involves only one active bin at all times; both First Fit and Best Fit are an *O*(*n*) space complexity algorithms, since they keep all non-empty bins active until the end of the processing .

The problem of achieving HA systems has gathered a lot of attention with work ranging from cluster and grid to utility computing. The overall problem can be reduced to the problem of placing virtual machines on a limited physical node so that the number of physical resources is minimized. This is also known as the bin packing problem which is known to be NP-hard. In paper [13], Bin-Packing algorithms used best fit algorithm of resources in cloud environment. BPP is a mathematical way to deal with efficiently fitting resources into Bins. A formal definition of the BPP can be defined as given a list of objects and their weights, bins size, find the least number of bins so that all of the resources are assigned to that bin. Now, a Bin can hold inside itself a certain amount of resources according to its Bin Height. Every Resource is of certain, nonzero and positive value (Resource Height).A Multi-dimensional Resource Allocation Algorithm in Cloud Computing [14] based on Multidimensional-Bin-Packing (MDBP) is defined as the problem, where the bins are hosts, the items are the VMs to be placed on the resource and the goal is to use as few resources as possible. A description of the MDBP-Problem can be found that multi-dimensional resource allocation problem for cloud computing that dynamically allocates the virtual resources among the cloud computing applications to reduce cost by reducing the number of work nodes to acquire better time complexity of MDRA algorithm using binary integer programming problem.

e. Non-preemptive and preemptive scheduling Algorithm [15]

Dynamic Resource Allocation for Efficient Parallel data processing introduces a new processing framework explicitly designed for cloud environments called Nephele. Most notably, Nephele is the first data processing framework to include the possibility of dynamically allocating/reallocating different compute resources from a cloud in its Scheduling and during job execution. Particular tasks of a processing job can be assigned to different types of virtual machines which are automatically instantiated and terminated during the job execution.

1) ***Architecture***: Nephele's architecture follows a classic master-worker pattern. Before submitting a Nephele compute task to process a job, a client must start a Virtual Machine (VM) in the cloud environment which execute and so called Job Manager (JM). The Job Manager receives the customer's set of task, to scheduling them, and their execution. It is communicating through the interface in cloud environment provides to control the virtual machine (VMs) infrastructure called as interface Cloud Controller. Cloud Controller contains Job Manager (JMs) can allocate or de-allocate VMs according to the current phase job execution. The current execution of tasks in Nephele's architecture with set of resource's as input. A Task Manager (TMs) receives set of tasks from the Job Manager at a time-driven, executes them and error is intimate to the Job Manager (JMs).

2) ***Job Description***: Jobs in Nephele are expressed as a directed acyclic graph (DAG). Each vertex in the graph represents a task of the overall processing job; the graph's

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

edges represent communication flow between tasks Job parameters are based on the following criteria:

- Number of subtasks
- Resource sharing between orders of task
- Input type
- Number of subtasks per Input

3) ***Job Graph:*** Once the Job Graph is specified, the customer submits it to the Job Manager, together with the identifications he has obtained from his cloud operator. The identifications are required since the Job Manager must allocate/reallocate input during the job execution on behalf of the customer.

f. Priority algorithm [17]

The basic idea of dynamic resource allocation mechanism for Preempt able jobs in cloud [17] environment is equal allocation to user according to their demands. Propose priority based algorithm, in which considering multiple SLA objectives of job, for dynamic resource allocation to AR job by preempting [25] best-effort job. PBSA perform better than Cloud min-min scheduling (CMMS) in resource contention situation. In proposed priority based scheduling algorithm is modified by the scheduling heuristic or executing highest priority task with advance reservation by preempting best-effort task as done in [16]. Algorithm shows the pseudo codes of priority based scheduling algorithm (PBSA) [30].

Algorithm:

1. **Input:** UserServiceRequest
2. //call Algorithm to form the list of task based on Priorities
3. Get global Available VM t and User List and also available Resource List from each cloud scheduler
4. // find the appropriate VM List from each cloud Scheduler
5. **If** AP(R, AR)! = ϕ **then**
6. // call the algorithm to load balancer
7. Deployable=load-balancer (AP(R, AR))
8. Deploy service on deployable VM
9. Deploy=true
10. **Else if** R has advance reservation and best-effort Task is running on any cloud **then**
11. // Call algorithm for executing R With advance reservation
12. Deployed=true
13. **Else if** global Resource Able to Host Extra VM **then**
14. Start new VM Instance
15. Add VM to Available VM List
16. Deploy service on new VM
17. Deployed=true
18. **Else**
19. Queue service Request until
20. Queue Time > waiting Time
21. Deployed=false
22. **End if**
23. If deployed then
24. Return successful
25. Terminate
26. **Else**
27. Return failure
28. Terminate

As the above Algorithm shows that the customers' service utilization requests (R), which is composed of the SLA terms (S) and the application data (A) to be provisioned, is provided as input to scheduler. When service request (i.e. job) arrive at cloud scheduler, scheduler divide it in tasks as per there dependencies then the Algorithm is called to form the list of tasks based to their priority. In the first step, it extracts the SLA terms, which forms the basis for finding the VM with the appropriate resources for deploying the application. In next step, it collects the information about the number of running VMs in each cloud and the total available resources (AR). According to SLA terms appropriate VMs (AP) list is collected, which are capable of provisioning the requested service (R). Once the list of appropriate VMs is form, the Algorithm - load-balancer decides which particular VM is allocated to service request in order to balance the load in the data center of each cloud. When there is no VM with the appropriate resources running in the data center of any cloud, the scheduler checks if service request (R) has advance reservation then check it that task is executing in any user environment in cloud, if it identified task of user then calls Algorithm for executing advance reservation request by preempting task best-effort. If no best-effort task is found on any cloud then scheduler checks whether the global resources consisting of physical resources can host new VMs, if yes then, it automatically starts new VMs with predefined resource capacities to provision service requests. But when global resources are not sufficient to host extra VMs, the provisioning of service request is place in queue by the scheduler until a VM with appropriate resources is available .If after a certain period of time, the service requests can be scheduled and installed, and then scheduler instruct success to the cloud admin, otherwise it needs repeat action.

III. IMPROVED PRIORITY BASED TASK-ORIENTED RESOURCE ALLOCATION RESCHEDULING (IPTRA):

Primary advantage with the cloud computing is that the business enterprises need not concentrate with new IT infrastructure that includes client, servers or database administrators and new licensed software. Major problems in task scheduling environment are load balancing, Scalability, reliability, performance, and re-allocation of resources to the computing nodes. Cloud users combine virtualization, automated software, and internet connectivity to provide their services. To access this type of services user does not have any limitations. Another way to utilized multiple resources in single machine is to analyze a database. Main concept of cloud computing is to accesses resources form various servers those are remotes both stored user data in remote server and accessed via the Internet. The resource allocation scheme decides to provide priority among different customers request to resources in cloud environment. Using priority scheduling algorithm, there could be several possible ways for discarding jobs in cloud. One of the important things is that breaking up task according to time and space which could be solved by considering user reservation in order to form a large request one. Another important issue is that unpredictability with respect to resource availability. These problems could be solved by changing the reservation client's request of already admitted jobs. Thus, our algorithm performs Bag of Task Rescheduling (BoT-R) [22] [23] [24] changes time to time. This type of rescheduling algorithm is regularly applied to reduce job allocation failures. Hence, only the allocated jobs are executed within the intervals should take part into the rescheduling process without preemption of jobs. The proposed algorithm includes two steps namely, evaluating if there is need on rescheduling task and performing the task rescheduling for selected resources according to their customer demands in given time driven intervals. To performs rescheduling process during breaking up of task to be measured within user intervals. To find solution, several variables parameter such as bandwidth, response time resource expenses, and network connectivity are checked for each interval, such that parameters should attend consistency ratio (CR). Also manage the load of the system and maintain the status of the gap management between client and servers. Once the service provider identified which task needs rescheduling, the BoT rescheduling process is called. Initially, a filtering process is carried over the resources to filter out the resources without usable breaking up process. Then, jobs belonging to those resources are executed within defined intervals is stored in remote servers with start time restriction. At last breaking up resources are collected and sorted according to the priority with the aim of allocating as many jobs as possible in each resource. This way of mapping resources to job reduce breaking up of resources and time intervals in between allocations. This algorithm also increases the efficiency. When it is not possible to, allocate any more jobs to resources the next resources is used for allocating the jobs which are not allocated yet, and so on.

IV. CONCLUSION

Cloud computing is a computing service paradigm that charges under the basis of the amount of resources consumed i.e. pay per use constraint. Primary advantage in cloud environment is that IaaS controls the user and manages the systems in terms of bandwidth, response time resource expenses, and network connectivity, but do not concentrate on infrastructure. This paper discuss about the various types of resources allocation and task scheduling algorithm. Although, there are various algorithms and methods were existing to solve the problem of resource allocation but none of these algorithms could be extended. Efficiency of cloud depends on the type scheduling algorithm used in environment. All above discussed algorithm used for resource allocation completely depends on types of task to be scheduled. Time driven based resource allocation gives better response time and increase resource utilization. Depending on surveying the various algorithm it can be concluded that, make span can be reduced by grouping the task. Since cloud computing systems have a high degree of unpredictability with respect to resource availability. In future as the cloud size increases, there is a need for better task scheduling algorithm.

REFERENCES

[1] W. Zhao, K.Ramamritham, and J.A.Stankovic, "Preemptive scheduling under time and resource constraints", In: IEEE Transactions on Computers C-36 (8) (1987)949–960.

[2] Alex King Yeung Cheung and Hans-Arno Jacobsen, "Green Resource Allocation Algorithms for Publish/Subscribe Systems", In: the 31th IEEE International Conference on Distributed Computing Systems (ICDCS), 2011.

[3] Chandrashekhar S. Pawar and R.B.Wagh, "A review of resource allocation policies in cloud computing", IN: World Journal of Science and Technology (WJST) Vol 3, PP 165-167 (2012).

[4] Mrs.S.Selvarani1; Dr.G.Sudha Sadhasivam, "Improved Cost - Based Algorithm For Task Scheduling In Cloud Computing", IEEE 2010.

[5] S. Mohana Priya, B. Subramani, "A New Approach For Load Balancing In Cloud Computing", In: International Journal Of Engineering And Computer Science (IJECS-2013) Vol 2, PP 1636-1640(2013).

[6] Pradeep.R, Kavinya.R, "Resource Scheduling In Cloud Using Bee Algorithm for Heterogeneous Environment", In: IOSR Journal of Computer Engineering (IOSRJCE) Vol 2, PP 15-16 (2012).

[7] V.Vinothina, Dr.R.Sridaran, and Dr.PadmavathiGanapathi, "A Survey on Resource Allocation Strategies in Cloud Computing", In: International Journal of Advanced Computer Science and Applications (IJACSA-2012) Vol 3, PP 97-104(2012).

[8] Rerngvit Yanggratoke, Fetahi Wuhib and Rolf Stadler,"Gossip-based resource allocation for green computing in Large Clouds", 7th International conference on network and service management, Paris, France, 24-28 October, 2011.

[9] J. Gu, J. Hu, T. Zhao, G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment", J. Comput. 7 (2012) 42–52.

[10] Linan Zhu, Qingshui Li， and Lingna He, "Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm", In: International Journal of Computer Science (IJCSI-2012) Vol 9, PP 1694-0814(2012).

[11] Ratan Mishra and Anant Jaiswal, "Ant colony Optimization: A Solution of Load balancing in Cloud", in: International Journal of Web & Semantic Technology (IJWesT-2012) Vol 3, PP 33-50 (2012). DOI: 10.5121/ijwest.2012.3203.

[12] Hao Yuan, Changbing Li and Mankang Du, "Resource Scheduling of Cloud Computing for Node of Wireless Sensor Network Based on Ant colony Algorithm", In: Information Technology journal (ITJ-2012) Vol 11, PP 1638-1643 (2012) DOI: 10.3923/itj.2012.1638.1643.

[13] K C Gouda, Radhika T V, Akshatha M, "Priority based resource allocation model for cloud computing", Volume 2, Issue 1, January 2013, International Journal of Science, Engineering and Technology Research (IJSETR).

[14] Bo Yin, Ying Wang, Luoming Meng, Xuesong Qiu, "A Multi-dimensional Resource Allocation Algorithm in Cloud

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 4, October 2013

Computing ",Journal of Information & Computational Science 9: 11 (2012) 3021–3028 .

[15] Venkatesa Kumar, V. and S. Palaniswami," A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing", Journal of Computer Science 8 (5): 780-788, 2012.

[16] Jiayin Li, Meikang Qiu, Jian-Wei Niu, Yu Chen, Zhong Ming,"Adaptive Resource Allocation for Preempt able Jobs in Cloud Systems", in 10th International Conference on Intelligent System Design and Application, Jan. 2011, pp. 31-36.

[17] Chandrashekhar S. Pawar, Rajnikant B. Wagh," Priority Based Dynamic resource allocation in Cloud Computing".

[18] "What is Cloud computing?" http://www.zeus.comlcloud_computinglc1oud.html, January 2010.

[19] Xin Lu, Zilong Gu, "A load-adaptive cloud resource Scheduling model based on ant colony algorithm", IEEE,2011.

[20] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Wang, D., "Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization", IEEE, 2011.

[21] Xiangqian Song, Lin Gao, Jieping Wang, "Job Scheduling based on ant colony optimization in cloud Computing", IEEE, 2011.

[22] Gutierrez-Garcia, J.O., Kwang Mong Sim, "A Family of Heuristics for Agent-Based Cloud Bag-of-Tasks Scheduling", IEEE, 2011.

[23] Oprescu, A., Kielmann, T., "Bag-of-Tasks Scheduling under Budget Constraints", IEEE, 2011.

[24] Netto, M.A.S., Buyya, R., "Offer-based scheduling of Deadline - constrained Bag-of-Tasks applications for utility Computing systems", IEEE, 2009.

[25] Shuo Liu, Gang Quan, Shangping Ren, "On-line Preemptive scheduling of real-time services with profit And penalty", IEEE, 2011.

[26] J. Yu and R. Buyya, "Workflow Scheduling Algorithms for Grid Computing", Technical Report, GRIDS-TR-2007-10, Grid Computing and Distributed Systems Laboratory, the University of Melbourne, Australia, May 2007.

[27] Anjou Bala, Dr.Inderveer Chana,"A Survey of Various Workflow Scheduling Algorithms in Cloud Environment", 2nd National Conference on Information And Communication Technology (NCICT) 2011.

[28] Proceedings published in International Journal of Computer Applications® (IJCA).

[29] Navjot Kaur, Taranjit Singh Aulakh, Rajbir Singh Cheema, "Comparison of Workflow Scheduling Algorithms in Cloud Computing", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 10, 2011.

[30] Ke Liu, "Scheduling Algorithms for Instance- Intensive Cloud Workflows", Swinburne University of Technology, 2009.

[31] Zhongyuan Lee, Ying Wang, Wen Zhou, "A Dynamic priority scheduling algorithm on service request Scheduling in cloud computing", IEEE, 2011.

AUTHOR BIOGRAPHY

M.Gokilavani is currently pursuing her M.E. Computer Science and Engineering in Angel College of Engineering, Tirupur, and Anna University of Chennai. She had completed UG degree in Sri Subramaniya College of Engineering and Technology, palani, Anna University of Tirucharapalli. She has presented 3 papers in National Conferences and publishes 1 paper in International Conference. She also attended various workshops and seminars in the area of cloud computing. She is the member of MISTE.

Mrs.S.Selvi is currently pursuing her Ph.D. Degree in the area of Cloud Computing in Anna University of Chennai. She had completed PG degree in Velalar College of Engineering and Technology, Erode and UG degree in Kongu Engineering College, Perundurai.She has published more 7 papers in National Conferences and 3 papers in International Conference. She is currently working as an Assistant Professor in Angel College of Engineering, Tirupur. She has coordinate various workshops, seminars and guest Lectures. She also attended various workshops and seminars.

C.Udhayakumar received his BE degree from V.S.B. Engineering College 2012. He presented 2 national conferences in various colleges. He participate various symposiums, seminars and workshop in various college. His Area of interest is Cloud Computing and Theory of computation. Currently he is doing his ME Computer Science and Engineering in Angel college of Engineering and Technology. He is a member of the MISTE.