Scholars Journal of Engineering and Technology (SJET) **ISSN 2321-435X (Online)**
Sch. J. Eng. Tech., 2014; 2(3C):467-471 **ISSN 2347-9523 (Print)**

(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

Research Article

A Study on Resource Allocation in Cloud

Rajesh Kumar D[1], Bhupathiraja D K*[2]
[1]Assistant Professor, [2]PG Scholar, Department of Information Technology, Bannari Amman Institute of Technology, Sathyamangalam, Erode, Tamil Nadu, India

***Corresponding author**
Bhupathiraja D K
Email: dpmbhupathi@gmail.com

Abstract: Cloud computing ia a type of internet based computing that is relies on sharing of computer resource from anywhere and anytime. It is similar to utility computing. In this paper a review of various policies for dynamic resource allocation in cloud computing is shown based on Topology Aware Resource Allocation (TARA),Linear Scheduling Strategy for Resource Allocation and Dynamic Resource Allocation for Parallel Data Processing. Moreover, significance, advantages and limitations of using Resource Allocation in Cloud computing systems is also discussed.
Keywords: Dynamic Resource Allocation, Cloud Computing, Resource Management, Resource Scheduling

INTRODUCTION

Cloud computing is the next generation in computation. Possibly people can have everything they need on the cloud. Cloud computing is the next natural step in the evolution of on-demand information technology services and products. Cloud Computing is an emerging computing technology that is rapidly consolidating itself as the next big step in the development and deployment of an increasing number of distributed applications.

Cloud computing nowadays becomes quite popular among a community of cloud users by offering a variety of resources. Cloud computing platforms, such as those provided by Microsoft, Amazon, Google, IBM, and Hewlett-Packard, let developers deploy applications across computers hosted by a central organization. These applications can access a large network of computing resources that are deployed and managed by a cloud computing provider. Developers obtain the advantages of a managed computing platform, without having to commit resources to design, build and maintain the network. Yet, an important problem that must be addressed effectively in the cloud is how to manage QoS and maintain SLA for cloud users that share cloud resources.

The cloud computing technology makes the resource as a single point of access to the client and is implemented as pay per usage. Though there are various advantages in cloud computing such as prescribed and abstracted infrastructure, completely virtualized environment, equipped with dynamic infrastructure, pay per consumption, free of software and hardware installations, the major concern is the order in which the requests are satisfied. This evolves the scheduling of the resources. This allocation of resources must be made efficiently that maximizes the system utilization and overall performance. Cloud computing is sold on demand on the basis of time constrains basically specified in minutes or hours. Thus scheduling should be made in such a way that the resource should be utilized efficiently.

In cloud platforms, resource allocation (or load balancing) takes place at two levels. First, when an application is uploaded to the cloud, the load balancer assigns the requested instances to physical computers, attempting to balance the computational load of multiple applications across physical computers. Second, when an application receives multiple incoming requests, these requests should be each assigned to a specific application instance to balance the computational load across a set of instances of the same application. For example, Amazon EC2 uses elastic load balancing (ELB) to control how incoming requests are handled. Application designers can direct requests to instances in specific availability zones, to specific instances, or to instances demonstrating the shortest response times. In the following sections a review of existing resource allocation techniques like Topology Aware Resource Allocation, Linear Scheduling and Resource Allocation for parallel data processing is described briefly.

SIGNIFICANCE OF RESOURCE ALLOCATION

In cloud computing, Resource Allocation (RA) is the[1]process of assigning available resources to the

needed cloud applications over the internet. Resource allocation starves services if the allocation is not managed precisely. Resource provisioning solves that problem by allowing the service providers to manage the resources for each individual module. Resource Allocation Strategy (RAS) is all about integrating cloud provider activities for utilizing and allocating scarce resources within the limit of cloud environment so as to meet the needs of the cloud application. It requires the type and amount of resources needed by each application in order to complete a user job. The order and time of allocation of resources are also an input for an optimal RAS. An optimal RAS should avoid the following criteria as follows: [1]

a) Resource contention situation arises when two applications try to access the same resource at the same time.
b) Scarcity of resources arises when there are limited resources.
c) Resource fragmentation situation arises when the resources are isolated. [There will be enough resources but not able to allocate to the needed application.]
d) Over-provisioning of resources arises when the application gets surplus resources than the demanded one.
e) Under-provisioning of resources occurs when the application is assigned with fewer numbers of resources than the demand.

RESOURCE ALLOCATION STRATEGIES &ALGORITHMS

Recently many resource allocation schemes have come up in the literature of cloud computing as this technology has started maturing. Researchers around the world have proposed and / or implemented several types of resource allocation. Few of the strategies for resource allocation in cloud computing are covered here briefly.

A. Topology Aware Resource Allocation (TARA)

Different kinds of resource allocation mechanisms are proposed in cloud. The one mentioned in proposes architecture for optimized resource allocation in Infrastructure-as-a-Service (IaaS) based cloud systems. Current IaaS systems are usually unaware of the hosted[2] application's requirements and therefore allocate resources independently of its needs, which can significantly impact performance for distributed data-intensive applications.

To speak to this resource allocation problem, an architecture that adopts a "what if" methodology to guide allocation decisions taken by the IaaS is proposed. Results showed that TARA reduced the job completion time of these applications by up to 59% when compared to application-independent allocation policies.

1) Architecture of TARA: TARA is composed of two major components: **a prediction engine and a fast genetic algorithm-based search technique.[2]**

The prediction engine is then tity responsible for optimizing resource allocation. When it receives a resource request, the prediction engine iterates through the possible subsets of available resources (each distinct subset is known as a candidate) and identifies an allocation that optimizes estimated job completion time. However, even with a lightweight prediction engine, exhaustively iterating through all possible candidates is infeasible due to the scale of IaaS systems. Therefore a genetic algorithm-based search technique that allows TARA to guide the prediction engine through the search space intelligently is used.

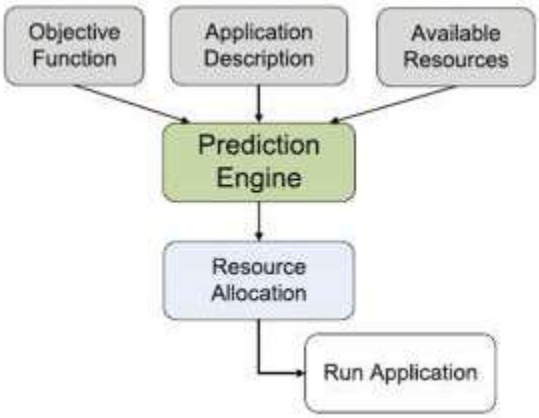

Fig No:A.1 Architecture of TARA[2]

2) Prediction Engine: The prediction engine maps resource allocation candidates to scores that measures their "fitness" with respect to a given objective function, so that TARA can compare and rank different candidates. The inputs used in the scoring process can be seen in Architecture of TARA.

3) Objective Function: The objective function defines the metric that TARA should optimize. For example, given the increasing cost and scarcity of power in the data center, an objective function might measure the increase in power usage due to a particular allocation.

4) Application Description: The application description consists of three parts: i) the framework type that identifies the framework model to use, ii) workload specific parameters that describe the particular application's resource usage and iii) a request for resources including the number of VMs, storage, etc.

5) Available Resources: The final input required by the prediction engine is a resource snapshot of the IaaS data centre. This includes information derived from both the virtualization layer and the IaaS monitoring service. The information gathered ranges from a list of available servers, current load and available capacity on

individual servers to data centre topology and a recent measurement of available bandwidth on each network link.

B. Linear Scheduling Strategy for Resource Allocation

Considering the processing time, resource utilization based on CPU usage, memory usage and throughput, the cloud environment with the service node to control all clients request, could provide maximum service to all clients. Scheduling the resource and tasks separately involves more waiting time and response time. A scheduling algorithm named as Linear Scheduling for Tasks and Resources (LSTR) is designed, which performs tasks and resources scheduling respectively. Here, a server node is used to establish the IaaS cloud environment and KVM/Xen virtualization along with LSTR scheduling to allocate resources which maximize the system throughput and resource utilization.

Resource consumption and resource allocation have to be integrated so as to improve the resource utilization. The preparation algorithms mostly focal point on the distribution of the resources among the requestors that will maximize the selected QoS parameters. The QoS parameter selected in our evaluation is the cost function. The scheduling algorithm is designed considering the tasks and the available virtual machines together and named LSTR scheduling strategy. This is designed to maximize the resource utilization.

Algorithm [3]:

1) The requests are collected between every predetermined
interval of time
2) Resources Ri => {R1,R2, R3,...,Rn}
3) Requests RQi => {RQ1, RQ2, RQ3,..., RQn}
4) Calculate Threshold (static at initial)
5) Th = Σ Ri
6) for every unsorted array A and B
7) sort A and B
8) for every RQi
9) if RQ*i* < Th then
10) add RQ*i* in low array, A[RQ*i*]
11) else if RQ*i* > Th then
12) add RQ*i* in high array B[RQ*i*]
13) for every B[RQ*i*]
14) allocate resource for RQ*i* of B
15) R*i* = R*i* - RQ*i*; Th = Σ R*i*
16) satisfy the resource of A[RQ*i*]
17) for every A[RQ*i*]
18) allocate resource for RQ*i* of A
19) R*i* = R*i* - RQ*i*; Th = Σ R*i*
20) satisfy the resource of B[RQ*i*]

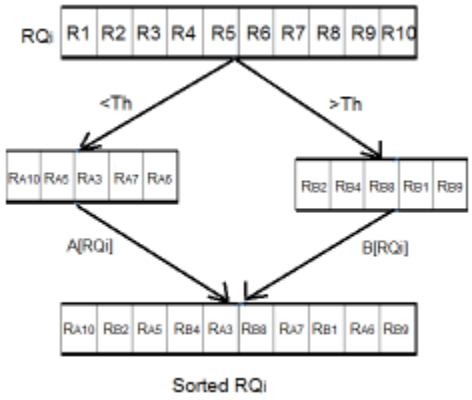

Fig No:B.1 Linear Scheduling Strategies

C. Dynamic Resource Allocation for Parallel Data Processing

Dynamic Resource Allocation[4] 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/de-allocating 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[4] follows a classic master-worker pattern as illustrated in Figure.Before submitting a Nephele compute job, a user must start a VM in the cloud which runs the so called Job Manager (JM). The Job Manager receives the client's jobs, is responsible for scheduling them, and coordinates their execution. It is capable of communicating with the interface the cloud operator provides to control the instantiation of VMs. We call thisinterface the Cloud Controller. By means of the Cloud Controller the Job Manager can allocate or de-allocate VMs according to the current job execution phase.

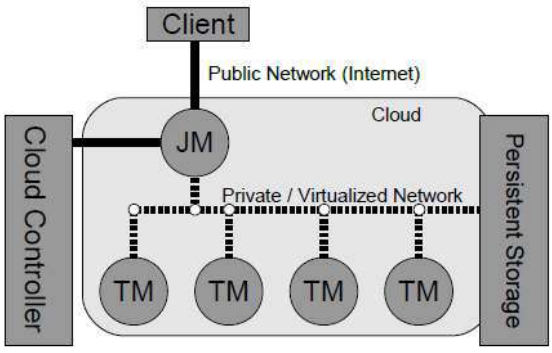

Fig No:C.1 Nephele Architecture[4]

The actual execution of tasks which a Nephele job consists of is carried out by a set of instances. Each

instance runs a so called Task Manager (TM). A Task Manager receives one or more tasks from the Job Manager at a time, executes them, and after that informs the Job Manager about their completion or possible errors

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 edges define the communication flow between these tasks Job description parameters are based on the following criterias:

• Number of subtasks
• Data sharing between instances of task
• Instance type
• Number of subtasks per instance.

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

IV.*Advantages and Limitations of Resource Allocation*

There are many benefits in resource allocation while using cloud computing irrespective of size of the organization and business markets.But there are some limitations as well, since it is an evolving technology. Let's have a comparative look at the advantages and limitations of resource allocation in cloud.[1]

Advantages:

• The biggest benefit of resource allocation is that user neither has to install software nor hardware to access the applications, to develop the application and to host the application over the internet.
• The next major benefit is that there is no limitation of place and medium. We can reach our applications and data anywhere in the world, on any system.
• The user does not need to expend on hardware and software systems.
• Cloud providers can share their resources over theinternet during resource scarcity.

Limitations:

• Since users rent resources from remote servers for their purpose, they don't have control over their resources.
• Migration problem occurs, when the users wants to switch to some other provider for the better storage of their data. It's not simple to move enormous information from one supplier to the other.
• In public cloud, the clients data can be susceptible to hacking or phishing attacks. Since the servers on cloud are interconnected, it is easy for malware to spread.
• Peripheral devices like printers or scanners might not work with cloud. Many of them require software to be installed locally. Networked peripherals have lesser problems.
• More and deeper knowledge is required for allocating and managing resources in cloud, since all knowledge about the working of the cloud mainly depends upon the cloud service provider.

Table-1:Resource Allocation and Algorithms

Algorithms	TARA	Linear Scheduling Strategies	Dynamic Resource Allocation for parallel data processing
Security Policies	✓	✓	✓
Time Management	Partially	Partially	✓
Virtual Machine	Partially	✓	✓
Optimized Allocation	✓	✓	✓
Auction Mechanism	Partially	Partially	Partially
Workflow Representation	✓	✓	✓
Machine Learning Techniques	✓	✓	✓

CONCLUSION

Cloud computing technology is increasingly being used in enterprises and business markets. A review shows that dynamic resource allocation is growing need of cloud providers for more number of users and with the less response time. In cloud paradigm, an effective resource allocation strategy is required for achieving user satisfaction and maximizing the profit for cloud service providers. This paper summarizes the main types of RAS and its impacts in cloud system. Some of the strategies discussed above mainly focus on memory resources but are lacking in other factors. Hence this survey paper will hopefully motivate future researchers to come up with smarter and secured optimal resource allocation algorithms and framework to strengthen the cloud computing.

.

REFERENCES

1. Vinothina V, Shridaran R, Ganpathi P; A survey on resource allocation strategies in cloud computing, International Journal of Advanced Computer Science and Applications, 2012; 3(6):97—104.
2. Lee G, Tolia N; Ranganathan P, Katz RH; Topology aware resorce allocation for data-intensive workloads, ACM SIGCOMM Computer Communication Review, 2011; 41(1):120—124.

3. Abirami SP, Ramanathan S; Linear scheduling strategy for resource allocation in cloud environment, International Journal on Cloud Computing: Services and Architecture(IJCCSA), 2012; 2(1):9—17.
4. Warneke D, Kao O; Exploiting dynamic resource allocation for efficient parallel data processing in the cloud, IEEE TransactionsOn Parallel And Distributed Systems, 2011.
5. Inomata A, Morikawa T, Ikebe M, Rahman M; Proposal and Evaluation of Dynamic Resource Allocation Method Based on the Load Of VMs on IaaS, IEEE, 2010.
6. Minarolli D, Freisleben B; Uitlity–based Resource Allocations for virtual machines in cloud computing, IEEE, 2011.
7. Jiyani; Adaptive resource allocation for preemptable jobs in cloud systems, IEEE, 2010.
8. Jung G, Sim KM; Location-Aware Dynamic Resource Allocation Model for Cloud Computing Environment, International Conference on Information and Computer Applications (ICICA), IACSIT Press, Singapore, 2012.
9. Chandrashekhar PS, Wagh RB; A review of resource allocation policies in cloud computing, World Journal of Science and Technology, 2012; 2(3):165-167.
10. Tayal S; Tasks Scheduling Optimization for the Cloud Computing systems, International Journal of Advanced Engineering Sciences and Technologies (IJAEST), 2011; 5(2): 111 – 115.
11. Van HN, Tran FD, Menaud JM; Autonomic virtual resource management for service hosting platforms. In Software Engineering Challenges of Cloud Computing, ICSE Workshop on IEEE, 2009; 1-8.
12. Popovici FI, Wilkes J; Profitable services in an uncertain world. In Proceedings of the 2005 ACM/IEEE conference on Supercomputing (p. 36). IEEE Computer Society. 2005.
13. Jiayin Li, Meikang Qiu, Jian-Wei Niu, Yu Chen; Adaptive resource allocation for preemptable jobs in cloud systems (IEEE, 2010), 31-36.
14. Melendez JO, Majumdar S; Matchmaking with Limited knowledge of Resources on Clouds and Grids. Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010; 102 – 110.