Challenges in Dynamic Resource Allocation and Task Scheduling in Heterogeneous Clouds

Ms. Ragini Karwayun
Researcher
Mewar University
Rajasthan
raginik22 @gmail.com

Dr. H. S. Sharma
Pro Vice Chancellor
Mewar University
Rajasthan
profhssharma@yahoo.com

Dr. K. P. Yadav
Director
IIMT
Greater Noida
karunesh732@gmail.com

Abstract— Resource Allocation and Task scheduling are the most important key words in today's dynamic cloud based applications. Task scheduling involves assigning tasks to available processors with the aim of producing minimum execution time, whereas resource allocation involves deciding on an allocation policy to allocate resources to various tasks so as to have maximum resource utilization. Algorithms used for scheduling resources for virtual machines are designed for both homogeneous and heterogeneous environments. Majority of the algorithms focus on processing ability often neglecting other features such as network bandwidth and actual resource requirements. One of the major pitfalls in cloud computing is related to optimizing the resources being allocated. Because of the uniqueness of the model, resource allocation is performed with the objective of minimizing the costs associated with it. The other challenges of resource allocation are meeting customer demands and application requirements. In this paper we will focus on the challenges faced in task scheduling and resource allocation in dynamic heterogeneous clouds.

Keywords*-Resource Allocation, Task Scheduling, distributed Heterogeneous Clouds*

*****

I. INTRODUCTION

Internet based environment of cloud computing framework provides opportunities for provisioning, consumption and delivery of services such as information, software and computing resources. In today's world there are numerous vendors that rent computing resources on-demand, bill on a pay-as-you-go basis, and multiplex many users on the same physical infrastructure. These cloud computing environments provide an illusion of infinite computing resources to cloud users so that they can increase or decrease their resource consumption rate according to the demands.

In cloud computing environments, the two important stakeholders: cloud providers and cloud users have different set of goals and objectives often conflicting with each other. There are providers who hold massive computing resources in their large datacenters and rent resources out to users on a per-usage basis. Then there are users who have applications with fluctuating loads and lease resources from providers to run their applications. Providers on one hand wish to maximize revenues by achieving high resource utilization, while users wish to have minimum expenses while meeting their performance requirements[6].

The two aspects of cloud computing applications are based on tradeoff between various characteristics depending on the domain and for providing the service by the cloud service provider and using the services by the cloud user, an initial agreement called the Service Level Agreement (SLA) has to be made between the cloud users and the cloud service provider. [2] While resource allocation is made to the cloud user, SLA violation should be avoided as much as possible or SLA violation should be minimal without compromising Quality of Service (QoS) parameters like performance, availability, response time, security, reliability, throughput etc.

Performance: For some application demands, performance is an important criteria. The system should perform well to provide service to the cloud user.

Response Time: For interactive applications, response time is an important factor. The system must respond well for these kinds of applications.

Reliability: The system used should be reliable so that the cloud user has no head ache of changing the system.

Availability: Whenever cloud resources are requested the cloud service provider must be able to allocate within short span.

Security: For critical applications like online transaction applications, system us ed has to be secure. Otherwise it is not safe to use such a kind of system.

Throughput: No. of applications run per unit time should be more.

II. CHALLENGES IN RESOURCE ALLOCATION IN CLOUDS

In cloud computing, Resource Allocation is the process of assigning available resources to the needed cloud applications over the internet. Cloud services are starved if the resource allocation is not managed properly. Resource provisioning solves that problem by allowing the service providers to properly manage the resources for each individual module.

Resource Allocation Strategy is all about integrating cloud provider activities for utilizing and allocating available 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 strategy. An optimal Resource allocation scheme should avoid the following criteria as follows:

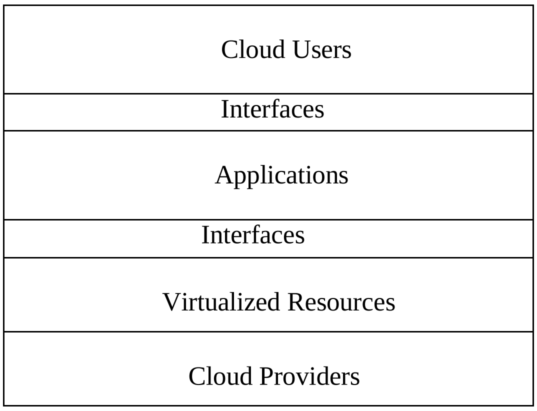
Resource contention situation arises when two applications try to access the same resource at the same time.

Scarcity of resources arises when there are limited resources.

Resource fragmentation situation arises when the resources are isolated. [There will be enough resources but not able to allocate to the needed application.]

Over-provisioning of resources arises when the application gets surplus resources than the demanded one.

Under-provisioning of resources occurs when the application is assigned with fewer numbers of resources than the demand.



Cloud users estimation of resources demands to complete a job before the estimated time may lead to an over-provisioning of resources. Cloud providers' allocation of resources may lead to an under-provisioning of resources. From the cloud user's angle, the application requirement and Service Level Agreement (SLA) are major inputs to allocation algorithms. The offerings, resource status and available resources are the inputs required from the provider's side to manage and allocate resources to host applications [1]. The outcome of any optimal Resource allocation scheme must satisfy the QoS parameters specified in SLA. Cloud providers while providing reliable resources, faces a crucial problem in allocating and managing resources dynamically across the applications.

Resource Allocation algorithms are available in two categories – static and dynamic.

Static Allocation: The cloud user declares the required resources beforehand. By doing so user is aware of the details of required resources – types of resources and how many instances of each type of resources are needed ahead of using the system. But the drawback in this case is it leads to underutilization or overutilization of resources depending on the time the application is run. The striking feature of these types of algorithms is that they use mostly fair allocation strategies.

Dynamic Allocation: Cloud resources are not declared prior to starting the application, but are requested by the cloud user on the run or as and when the application needs. These algorithms provide best utilization of resources avoiding underutilization and overutilization of resources, as much as possible. But nearly all dynamic resource allocation policies are unfair. The requested resources might not be available when requested on the fly. The service provider has to make allocation from other participating cloud data center.

In most cases, the interaction between providers and users is as follows:

1. A user sends a request for resources to a provider.
2. When the provider receives the request, it looks for resources to satisfy the request and assigns the resources to the requesting user, typically as a form of virtual machines(VMs).
3. Then the user uses the assigned resources to run applications and pays for the resources that are used.
4. When the user is done with the resources, they are returned to the provider.

The most interesting aspect of the cloud computing environment is that the major players in the field are often different parties with their own interests. Typically, the goal of providers is to generate as much revenue as possible with minimum investment[6]. To achieve this they might want to squash their computing resources; for example, by hosting as many VMs as possible on each machine, thus wishing to maximize resource utilization. However, placing too many VMs on a single machine will cause VMs to interfere with each other, resulting in degraded and/or unpredictable performance which, in turn, affects the users. Thus, the providers may evict existing VMs or reject resource requests to maintain service quality, but it could make the environment even more unpredictable. On the other hand, users want their jobs done at minimal expense or, in other words, they wish to have maximum performance at minimal cost. This involves having proper resources that suit the workload requirements of users' applications and utilize resources effectively. They also have to take unpredictable resources into account when they request resources and schedule applications.

Resource allocation in distributed heterogeneous clouds in dynamic computing environment focuses on four fundamental issues :

1. Resource modelling.
2. Resource allocation and treatment.
3. Resource offering and treatment.
4. Resource discovery and monitoring selection.

Resource allocation algorithms must have the complete knowledge of the status of each resource in the computing environment and use this knowledge to apply it intelligently to allocate physical or virtual resources to applications as per their requirements. [1] So the information that is required by the resource allocation algorithms can be summed up as: cloud resources, resource modelling, application requirements and specifications and providers requirements.

Resource Modelling : Infinite computing resources are available to be used on demand by the cloud users. The cloud environment thru the use of resource allocation algorithms tries to meet the demands of the users in a dynamic way, allowing the multiplexing of physical resources, avoiding both under provisioning and over provisioning of resources. Moreover as it is a heterogeneous environment it is essential to develop a proper resource model. All optimized allocation algorithms are very strongly dependent on the resource modelling scheme used. Computing and network resources can be represented by using RDF (Resource Description Framework) and NDL (Network Description Language). It must be kept in mind that these tools must be capable of representing resources, networks and applications virtually in terms of properties and functionalities[7].

Another important point of consideration is the granularity of resource description. More the details used in describing the resources better will be the allocation being more flexible in usage of resources , on the other hand more details will make the resource selection and allocation module complex and harder to handle.

One additional point of challenge in resource allocation in heterogeneous environment is the interoperability in clouds inorder to provide seamless flow of data between clouds, across clouds and their local applications. Layered architectural design and use of open APIs are practical options for interoperability.

There are two types of heterogeneities faced by interoperability in clouds [8]. Vertical is intra-cloud whereas Horizontal is inter-cloud(different vendors) heterogeneities. Vertical heterogeneity can be addressed with the help of middleware and standardization. OVF (Open Virtualization Format) can be used for managing virtual machines across the system. Horizontal heterogeneity, as clouds from different vendors are involved, is very difficult to obtain. As each vendor may use different level of abstraction to represent the resources, it is challenging to define a method of interaction between clouds. This issue can be addressed with the help of high level of granularity in the modelling but has a serious drawback of losing information. But it is advantageous to have horizontal interoperability as there are numerous instances when a user's request cannot be addressed by a cloud provider due to unavailability of resources locally. In such cases, the provider can borrow resources from another vendor by dynamically negotiating.

Resource Offering and Treatment: After modelling of resources, the providers offer interfaces (middleware) that on one hand handles resources at a lower end and on the other hand deal with the requirements of the applications at a higher level. It is important to emphasize on the fact that there should be a clear level of abstraction between the resource modelling and the way they are offered to the end user. For example, a provider can model the resources independently such as speed of processor or size of memory but offer them as a collection of items , such as a VM.

Since it is important to have a generic solution for clouds so that they may support as many applications as possible, resource offering becomes a very complex process. Providers must handle major issues like – how to have a good trade off between the granularity of resource modelling and an easy generic solution, and how can one define the generic level of the cloud?

The Resource Allocation Scheme must ensure that all requirements must be met with the set of available resources as much as possible. Moreover the requirements of the cloud users must be specified properly in the SLA and it is essential that both parties must adhere to it. The list of the requirements specified in the SLA may include but not limited to the following:

i. Network bandwidth requirement.
ii. Computational capability (processor speed and memory) requirement.
iii. Topology of the nodes which help the cloud users to define inter-node relationships and restrictions on uplink and downlink communications.
iv. Jurisdiction of physical location and methods of handling data by applications. Location of data may be restricted within a country or a continent by copyright laws.
v. The node proximity is another requirement which may be seen as a constraint imposed on the physical distance or delay value (maximum or minimum) between nodes.
vi. The application interaction defines the types of configuration applications will use to exchange information with each other.
vii. The cloud user should also be able to define scalability rules which gives the specifications for the growth of applications and how will and how many resources will they consume.

Authors in [9] describes the methods how the VMs will be deployed by the cloud users based on the thresholds of the above specified metrics.

Resource Discovery and Monitoring: Providers need to find appropriate resources that comply with the user's request. The most daunting task is to fine suitable resources within the boundaries (physical/geographical) in a heterogeneous cloud. In addition to this the costing factor of inter domain traffic also need to be take care of.

In [8] a very simple implementation of the resource discovery service is described for the network virtualization scenario that uses a discovery framework with an advertisement process. Cloud users (brokers) use it to discover and match available resources from different providers. It consists of distributed repositories responsible for storing resource descriptions and states. Considering that one of the key features of cloud computing is its capability of acquiring and releasing resources on demand [10], *resource monitoring* should be continuous, and should help with allocation and reallocation decisions as part of overall resource usage optimization. It should be carefully analyzed to find an acceptable trade-off between the amount of control overhead and the frequency of resource information refreshing.

The resource monitoring may be passive or active. It is *passive* when one or more entities collect information by continuously sending polling messages to nodes asking for information or do this on demand when necessary. On the other hand, the monitoring is *active* when nodes are autonomous and may decide when to send asynchronously state information to some central entity. Both the alternatives can be simultaneously be used to improve the monitoring solution. To achieve this it is necessary to synchronize updates in repositories to maintain consistency and validity of state information.

Resource Selection and Optimization: Once the information regarding cloud resource availability is collected, a set of appropriate candidates may be highlighted. In the next phase the resource selection process needs to find a configuration that fulfils all requirements and optimizes the usage of the infrastructure. In virtual networks, for example, the resource selection mechanisms main objective is to find the best mapping of the virtual networks on the substrate network with respect to the constraints [11]. Selection of a suitable solution from a set is a very complex task due to the dynamically changing environment , very complex algorithm, and all the other different requirements relevant to the provider. Optimization algorithms may be used for resource selection. Optimization strategies used may vary , from simple and well-known techniques such as simple heuristics with thresholds or linear programming to newer, more complex ones, such as Lyapunov optimization [12]. In addition, artificial intelligence algorithms, biologically inspired ones (e.g., ant colony behaviour), and game theory may also be applied in this scenario. Authors in [13] define a system called Volley to automatically migrate data across geo-distributed datacentres.

This solution uses an iterative optimization algorithm based on weighted spherical means [13]. Resource selection strategies fall into a priori and *a posterior* classes. In the a priori case, the first allocation solution is an optimal one. To achieve this goal, the strategy should consider all variables influencing the allocation. For example, considering VM instances being allocated, the optimization strategy should figure out the problem, presenting a solution (or a set of possibilities) that satisfies all constraints and meets the goals (e.g., minimization of reallocations) in an optimal manner. In an *a posteriori* case, once an initial allocation that can be a suboptimal solution is made, the provider should manage its resources in a continuous way in order to improve this solution. If necessary, decisions such as to add or reallocate resources should be made in order to optimize the system utilization or comply with cloud users' requirements. Since resource utilization and provisioning are dynamic and changing all the time, it is important that any *a posteriori* optimization strategy quickly reach an optimal allocation level, as a result of a few configuration trials. Furthermore, it should also be able to optimize the old ones, readjusting them according to new demand. In this case, the optimization strategy may also fit with the definition of a priori and dynamic classification

Thus we have described the four challenges that should be faced for defining any resource allocation scheme. The first two challenges need to be taken care of in conception phase and the next two in operational phase. When designing a distributed heterogeneous cloud the provider should choose the nature of its offering such as service, infrastructure and platform as a service (SaaS, IaaS and PaaS).

III. Challenges in Task Scheduling in Clouds

The main objective of task scheduling is to schedule the arriving tasks onto available processors with the aim of producing minimum schedule length and without violating the precedence constraints. It is often seen that the different sections of the application task need different types of computations. So for single machine architecture it is impossible to satisfy all the computational requirements of such applications. So it is essential to schedule different tasks of such applications to different suitable processors across the distributed heterogeneous computing resources. This system helps in executing computationally intensive applications with different computational requirements. But the performance of such a system is highly dependent on the scheduling algorithm used.

We assume that the physical locations of VMs are determined by the provider. It is assumed that, the responsibility to determine when to request how many resources of which type, and how to schedule the user's application on the allocated resources, lies with the user[6].

In a heterogeneous environment, it is important to schedule a job on its preferred resources to achieve high performance. In addition, it is not straightforward to provide fairness among jobs when there are multiple jobs. Moreover, the capacity of different machine types needs to be considered when a user makes a resource request.

Task Scheduling algorithms are static as well as dynamic in nature. As static algorithms are more fair, stable and easy to use, they are more commonly used. Hadoop also uses static algorithm for task scheduling [3].

Performance metrics for task scheduling algorithms are schedule length, speedup, efficiency and time complexity :

Schedule length : is the maximum finish time of the exit task in scheduled DAG (Directed Acyclic Graph).

Speedup: of a schedule is defined as the ratio of the schedule length obtained by assigning all tasks to the fastest processor, to the scheduled length of application.

Efficiency: is the speedup divided by the number of processors used. It is the measure of the effective percentage of processor time is doing useful computation.

Time complexity: is the amount of time taken to assign every task to specific processor according to specific priority.

Once a set of resources such as virtual machines are allocated by the cloud driver ,the analytics engine uses the resources that are heterogeneous and shared among multiple jobs. In this section, we consider challenges in job scheduling on a shared, heterogeneous cluster, to provide good performance while guaranteeing fairness.

Share and Fairness

In a shared cluster, providing fairness is one of the most important features that the analytics engine should support. There are different ways to define fairness, but one method might be having each job receive equal (or weighted) share of computing resources at any given moment. In that sense, Hadoop Fair Scheduler [4] takes the number of slots assigned to a job as a metric of share, and fairness is provided by assigning same number of slots to each job.

In a heterogeneous cluster, as all the slots are different, the number of slots might not be an appropriate metric of the share. Moreover, even on the same slot, the computation speed varies depending on jobs. The performance variance on different resources is also important to consider to improve overall performance of the data analytics cluster; if jobs are assigned to slots irrespective of their preferences will not only make the job run slow, but also may prevent other jobs that prefer the slot from utilizing it.

Progress Share

Progress share refers to how much progress each job is making with assigned resources (or slots in Hadoop) compared to the case of running the job on the entire cluster without sharing; therefore, it is between 0 (no progress at all) and 1 (all available resources are occupied).

Scheduler

To calculate the Progress share of each job, the analytics engine scheduler should be aware of the per-slot computing rate(CR). To that end, each job should go through two phases: calibration and normal. Calibration phase starts when a task is submitted. In this phase, by measuring the completion time of these tasks, the scheduler can determine the CR. Once the scheduler knows the CR, the task enters the normal phase. During this phase, when a slot becomes available, a job of which the share is less than its minimum or fair share is selected. However, if there is another job with a significantly higher computing rate on the slot, the scheduler chooses that job to improve overall performance. This is similar to the "delay scheduling" [14] mechanism in Hadoop fair scheduler. By using progress share, the scheduler can make an appropriate decision. As a result, the cluster is better utilized. In addition, each job receives a fair amount of computing resources.

IV. CONCLUSION

The cloud environment provides heterogeneous hardware and resource demands; therefore, it is important to exploit these features to make a data analytics cluster in the cloud efficient. In this paper, we presented the issues concerning the cloud users and providers to allocate resources in a cost-effective manner, and discussed a scheduling issues that should be taken care of to provide good performance and fairness simultaneously in heterogeneous cluster.

Beyond data analytics systems, many systems running in the cloud involve multilevel scheduling. Resource allocation is done at infrastructure level and job scheduling is performed at application level. Mesos [15] takes this approach to provide resource sharing and isolation across distributed applications. It adopts Dominant Resource Fairness(DRF) [6] as an example of resource allocation policy and leaves application level scheduling to each application.

RFERENCES

[1] Patricia Takako Endo et al. :Resource allocation for distributed cloud :Concept and Research challenges(IEEE,2011),pp.42-46 .

[2] Lizhewang,JieTao,Kunze Castellanos ,A.C,Kramer ,D.,Karl,w, "High Performance Computing and Communications",IEEE International Conference HPCC,2008,pp.825-830.

[3] Hadoop. http://hadoop.apache.org.

[4] Hadoop fair scheduler. http://hadoop.apache.org/ common/docs/r0.20.1/fair_scheduler.html.

[5] I. Houidi et al., "Virtual Resource Description and Clustering for Virtual Network Discovery," Proc. IEEE ICC Wksp. Network of the Future, Dresden,bGermany, June 2009.

[6] Gunho Lee et al., " Heterogeneity aware Resource Allocation and Scheduling in Clouds", Yahoo Research.

[7] N. M. K. Chowdhury and R. Boutaba, "A Survey of Network Virtualization," Computer Networks: Int'l. J. Comp. and Telecommun. Networking, Apr.2010, pp. 862–76.

[8] A. Sheth, and A. Ranabahu, "Semantic Modeling for Cloud Computing, Part I," IEEE Computer Society — Semantics & Services, 2010.

[9] C. Chapman et al., "Software Architecture Definition for On-Demand Cloud Provisioning," Proc. 19th ACM Int'l. Symp. High Performance Distrib. Computing, 2010, pp. 61–72.

[10] Q. Zhang, L. Cheng, and R. Boutaba. "Cloud Computing: State-of-the-Art and Research Challenges," J. Internet Services and Apps., Springer, 2010, pp. 7–18.

[11] I. Houidi, W. Louati, and D. Zeghlache, "A Distributed Virtual Network Mapping Algorithm," Proc. IEEE ICC, 2008, pp. 5634–40.

[12] R. Urgaonkar et al., "Dynamic Resource Allocation and Power Managementin Virtualized Data Centers," IEEE NOMS), 2010.

[13] S. Agarwal et al., "Volley: Automated Data Placement for Geo-Distributed Cloud Services," Proc. 7th USENIX Conf. Networked Sys. Design and Implementation, 2010.

[14] M. Zaharia et al. Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In EuroSys, 2010.

[15] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, D. Joseph, R. Katz, S. Shenker, and I. Stoica.Mesos: A platform for fine-grained resource sharingin the data center. In NSDI, 2011

[16] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. Dominant resource fairness: Fair allocation of multiple resource types. In NSDI, 2011.