ISSN:0975-9646

Applying Parallel Genetic and Considering Proximity for Resource Allocation in Cloud

Romil Chauhan, Yash Jain, Harsh Agarwal, Aruna Gawde

Computer Engineering Department,
D J. Sanghvi College of Engineering,
Mumbai, India.

***Abstract*—This era of virtualization of services has been adopted by majority of users all over the world. Virtualization of services reduces the amount of efforts a company puts in order to design their product as service providers takes care of the background work. This increases the responsibility of the cloud providers. The service provider's goal is to provide the user with a facility with minimum response time and minimum cost. This was already achieved by individually considering proximity and genetic algorithm for resource allocation. In this paper, we propose a method to help the service providers achieve the same. We do so by considering various factors including the proximity factor and apply genetic algorithm to analyze and find the best combinations by considering their fitness.**

Keywords— IaaS; PaaS; SaaS; resource allocation; proximity; parallel genetic algorithm; distributed cloud computing.

I. INTRODUCTION

A cloud storage infrastructure is the hardware and software components -- such as servers, storage, virtualization software and operating systems -- needed to meet the computing requirements of a cloud storage model.

Cloud computing is the new era in the field of computing. Computing, storing, accessing and various other services are required everywhere. But not all the resources are available at each site to full fill them. This is where cloud computing comes into picture. The clouds provides three kinds of services SaaS, PaaS and IaaS.

In this paper, we study services provided by the cloud and also shed light on the concept of distributed cloud and the issues associated with it. We discuss the concept of proximity in cloud, how it affects the cloud service in general. Then genetic algorithm is understood and applied for allocating resources considering various factors. The best combination is found by calculating the fitness and the one with maximum fitness is considered as the solution.

A. SaaS: Software as a Service

In SaaS, third-party vendors make their applications accessible via the web browser so that their clients can use them on their PC's without any worry of managing them.
Most SaaS applications run directly on the browser, there are installation or downloads are required, although some plugins are required.

The web delivery model has made the life of the people and the enterprises very easy because everything is managed by the vendors: Data, Storage, Processors, Network, Servers, Operating Systems, Virtualization, Runtime, and Middleware. Thus, individual or enterprises are carefree about maintenance and support as it is managed by the vendors. SaaS examples: Google Apps, Salesforce, Workday, etc.

B. PaaS: Platform as a Service

PaaS is a framework where developers can develop their own applications. PaaS typically provide a platform where software can be developed and deployed. PaaS leads to a simple, quick and cost-effective development and testing of applications. This technology has helped businesses focus on their actual work and increase their profits rather than worrying about the complicated things like maintaining servers and other things. Thus PaaS increases developer's productivity and utilization rates while also decreasing their time-to-market. PaaS examples: Heroku, Google App Engine, and Red Hat's open shift.

C. IaaS: Infrastructure as a Service

Moving down the stack, we get the fundamental building blocks of cloud services. IaaS, can be seen as self-service models which are used to access, monitor and control remote data center infrastructure such as servers, memory storages, network services, etc. So rather than buying and setting up servers and other elements, users can buy IaaS based on consumption similar to electricity and other utility billing. By using this service, the user need not worry about maintenance and other costs.

This model relieves users of the complex management of data centers. There is a lot of cost-benefit which the client enjoy as, now being able to use the resources according to the requirement, need not spend so much of time, money and man-force towards the management of it. Thus, this service avails current infrastructure of data center for handling ephemeral workloads. IaaS ex: Amazon Web Services, Microsoft Azure, Google Compute Engine (GCE).

II. CONCEPT OF DISTRIBUTED CLOUD

Usually cloud computing providers rely mainly on large and centralized data centers rather than many distributed data centers. But nowadays providers are trying the concept of distributed data centers, but there are few issues which may arise.

As mentioned in [2], using this methodology, one may take advantage of geo-diversity and even reduce the cost and improve the performance but it has a drawback, it may transfer the traffic control to Internet service providers if it uses public infrastructure for communication between data centers. By distributed data centers, the goal of building more dispersed clouds with low costs of deployment may be achieved. However, the absence of a central management system is an issue when it comes to reliability and maintenance in case of failures [2].

To do away with these limitations, a generic concept of distributed cloud is used wherein infrastructure is hired on demand by cloud providers. Infrastructure includes router and hyperlinks to even servers and databases. The important area of application this concept finds is that of network virtualization (NV) [5]. A NV is a system where there exists multiple heterogeneous network architectures from different service providers yet exist together and share a physical substrate (called as Substrate Network (SN)) which is common for all.

III. PROXIMITY

In the above section we have discussed the concept of distributed cloud and the problems associated with it. The virtualized availability of resources have increased the role of data center services in the supply chain. This have altogether produced new challenges like meeting the growing demands for capacity to accommodate the large amounts of data travelling over the network and has resulted in a bottleneck. Though virtualizing of resources has solved previous problems like having storage space for keeping racks, but has led to new problems altogether.

The main challenge is making the application developer aware of service delivery issues. There is sometimes a communication gap which exist between the developers of the cloud applications and those who handle the delivery of those applications. The developers sometimes just take the delivery of resources for granted which reduces the chances of the product to perform optimally. To solve this, the service nodes can be brought closer to users. The CSP, can overcome the network congestion and latency issues that affect application performance.

Proximity, the distance between infrastructure elements and users have a direct impact on the performance of any cloud system. Proximity not only focuses on the current location of the user, but also takes into consideration the future location of users and how close the service nodes will be to those future users.

Consider the examples of few courier services, who are famous for their fast, and reliable delivery. They tend to place their warehouses in areas where many packets are delivered. Thus, it reduces the cost and at the same time increases the speed. Thus, in the same way the proximity reduces the latency of cloud service delivery [3]. We imply genetic algorithm to find an optimized solution by analyzing the paths and the manual commands, which were given in order to optimize the cost and response time when the scheduling algorithm failed to provide an efficient solution.

IV. PARALLEL GENETIC ALGORITHM

Genetic Algorithms is the technological implementation of the biological Genetic evolution. Many traits are taken into account, usually for a large population and the fitness is calculated based on the desired output. The one with the maximum fitness is the best option which survives and reproduces in the next generation. Genetic algorithms are good search optimization techniques, which are used to solve difficult problems. But the issue with this technique is that it increases computational load and has high memory requirements. Parallel Genetic Algorithm (PGA) solves this problem to a great extent. PGA can be implemented on parallel mainframes which, in a way distributes the computational load and memory and increase the efficiency of the system. Following is the scheduling model, wherein we use PGA for optimization.

A. Scheduling Model

There are three main steps in the scheduling:

a) The system set an idle resource list and VM request list, update them at the initiate time and each time new VM requests come or VMs are shutdown or physical resources change are being detected.
b) Use the PGA to find the optimal allocation sequence.
c) Launch the specified physical machines to the VM requests.

There are various nodes located at various sites. Each node has CPU, Memory, and Hard disks. The super scheduler in PGA combines all the resources and then allocate the memory using mathematical fitness function as shown in [4]. The one with maximum fitness is considered. First, we apply the genetic algorithm based on the factors mentioned above. The assumption in this case is that all the nodes are equidistant from the client. During calculations we consider the distance variable as a constant which is the average of distances considered.

For example, consider cloud service provider have six nodes A, B, C, D, E and F each having three traits. The service provider needs to provide service to user X. The Table I. below shows the traits at each nodes and their distance from the user X (the distance here is a constant same for all nodes).

In order to provide best service to the user the combination of all the traits should be done in a way that it

is best fitted. This best fitted combination of traits are found by iterating all over the table by considering fitness of combinations of each traits of all the nodes. The measure of memory and processor should be high while on the other hand the CMT (Cost per unit Memory per unit Time) should be low.

TABLE I. RESOURCES AVAILABLE AT NODES

Nodes (N)	Memory (m)	Processor (p)	C|M|T (cmt)	Distance from users (d)
A	100 TB	3.6 GHz	35 units	70
B	150 TB	4.7 GHz	40 units	50
C	500 TB	5 GHz	20 units	60
D	800 TB	2 GHz	30 units	45
E	1000 TB	2.6 GHz	45 units	40
F	200 TB	2 GHz	15 units	55

The total fitness for various combinations can be considered using the equation,

Total fitness combination (i,j)= ∀i(m/(cmt*d)) + ∀j(p/d) (1)

where i, j belongs to all nodes (N) and the constant 'd' here will be average of all distances which is 53.33.

After calculating m/(cmt*d) and (p/d) for each node using (1) values obtained are shown in TABLE II:

TABLE II. FITNESS OF EACH RESOURCE AT THEIR NODES BY CONSIDERING SAME AVERAGE DISTANCE

Nodes (N)	m/(cmt*d)	(p/d)
A	0.0535	0.0675
B	0.0703	0.0881
C	0.4688	0.0937
D	0.5000	0.0375
E	0.4170	0.04875
F	0.2499	0.0375

These values help us decide which node to consider, the one with the maximum fitness is selected.

V. APPLYING PARALLEL GENETIC ALONGWITH PROXIMITY

The proximity factor is also considered while finding an optimized solution using PGA which is useful for allocating resources in a more efficient way. In this section we are planning to propose an idea of combining these two concepts for resource allocation. Allocation of resources will become more efficient with the combined used of proximity and genetic algorithm.

The concept of proximity in resource allocation basically means to allocate the resources from the data center in a way which keeps it closer to the client. This will decrease the network latency. Hence, the scheduling of requests to resources will be more efficient as the communication of request will take a considerably less time as the number of hops will be less.

However, the tasks may not depend only on the time taken to process the request but also on the efficiency of the resources which operate the tasks. Therefore in this case it is equally important to consider the efficiency of resources along with the proximity by taking into account the need of the tasks operated from client side. Genetic algorithm, which is basically a technique where the efficient traits of the population are combined together to form altogether new set of populations by crossover and mutation. Applying PGA in the case of resource allocation, the population can be considered as data centers and their location, the resources available at those location can be taken as their traits. In the following example, the distance, the processor and the memory are considered as traits. Efficient traits (i.e. efficient resources) from different nodes can be combined to form a whole new set of resources to be provided to users.

Considering the same example discussed in the previous section. This time we consider the impact of the distance factor. We recalculate the fitness table using the same formula discussed above. The values are shown in Table III.

TABLE III. FITNESS OF EACH RESOURCES AT THEIR NODES BY CONSIDERING DIFFERENT DISTANCES

Nodes (N)	m/(cmt*d)	(p/d)
A	0.0408	0.0514
B	0.075	0.094
C	0.4167	0.083
D	0.5926	0.044
E	0.556	0.065
F	0.2424	0.03636

On comparing the fitness calculated in both the cases, one giving importance to distance and other neglecting the impact of distance. We can see that when distance is considered as a significant factor, the fitness of those with greater distance from the client is less compared to those with lesser distance from the client. The change in the respective fitness values from Table II. and Table III. reflects that very well.

One may notice that node E is supplying memory service, though the best fitted memory service is provided by node D. Also, node C is providing processor with maximum frequency but the efficient processing service is provided by node B. Hence, getting memory from node D and processor from node B there will be less network latency and the service provided will be maximum.

VI. CONCLUSION

Thus, based on the theoretical experiment conducted in this paper where proximity and parallel genetic algorithm where applied simultaneously we propose that parallel genetic algorithm can be applied in resource allocation in cloud to find the best possible combination of resources to meet the requirement of the user in a cost efficient manner. Service providers can consider the concept proximity and the concept of proximity we present along with parallel genetic algorithm to reduce their efforts in terms of cost.

REFERENCES

[1] Kapil Kumar, Abhinav Hans, Navdeep Singh, Ashish Sharma "Comparative analysis of various cloud based scheduling algorithms"IJARCSSE, Volume 4, Issue 8, August 2014.

[2] Library of www.apprenda.com

[3] http://www.comtelcommunications.com/blog/the-role-of-the- network-in-cloud-service-delivery52015

[4] "An approach for cloud resource scheduling based on parallel genetic algorithm", Zhongni Zheng ,Rui Wang, Hai Zhong, Xuejie Zhang.

[5] Haider, R. Potter, and A. Nakao, "Challenges in resource allocation in network virtualization," 20th ITC Specialist Seminar, 18–20 May 2009, Hoi An, Vietnam.

[6] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization", Computer Networks: Int'l. J. Comp. and Telecommunication Networking, Apr 2010, pp. 862–76.