Impact of Resource Allocation Schemes on Cost and Performance in Federation of Clouds

Bhawna Taneja
Research Scholar, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra-136119, Haryana, India
Email: tanejabhawna@gmail.com
Rajender Nath
Professor and Chairman, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra-136119, Haryana
Email: rnath2k3@rediffmail.com

Abstract

In a Federated Cloud Environment, binding of cloudlets to a Datacenter is a complex task as compared to non-federated environment and this greatly affects the performance of a federation. A number of techniques in this context are proposed in the literature but not evaluated in the context of cloud federation. This paper intends to study the impact of Round Robin, Greedy and Random resource allocation techniques on the usage cost in a federation of clouds. These techniques have been compared by evaluating them using SmartFed Simulator.

Keywords - Cloud Federation, Datacenter, SmartFed.

IJSER

1. Introduction

Participating cloud providers may have different compelling factors for being a part of a cloud federation. Although a variety of cloud federation platforms [1][6][7] are available, the decision of outsourcing the local requests to other participants of the cloud federation is a challenging task. The outsourcing has been already dealt with in hybrid clouds [8] but it is a more complex issue in case of a cloud federation.

A cloud can have one or more data centers, each having a number of hosts and VMs. Here the jobs, also known as cloudlets, need binding with VMs for execution. A number of techniques can be used for this purpose. Some of these techniques have been evaluated in our previous papers using CloudSim toolkit. These techniques are relatively simpler to implement due to inherent simplicity with a single Cloud Provider as compared to multiple Cloud Providers in a Federated Cloud Environment. A cloud Federation has multiple Cloud Providers as its members and each of these Cloud Providers has one or more datacenters. Cloudlets are submitted to the federation for execution. A number of parameters such as resource availability, response time, cost etc. can be considered before assigning cloudlets to a datacenter of a particular Cloud Provider. This paper investigates four techniques of assigning datacenter to cloudlets namely Round Robin, Greedy, Random and Genetic Algorithm based techniques. Further these techniques have been compared on the cost parameter by evaluating them using SmartFed simulator.

2. Related Work

Authors in [9] propose resource provisioning policies in a Federated Cloud Environment that help to increase provider's profit, resource utilization and user satisfaction. For this purpose, authors have proposed three different policies namely Non-Federated Totally in House Policy, Federation-Aware Outsourcing Oriented Policy and Federation-Aware Profit Oriented Policy.

Authors [10] used a genetic algorithm for adjustment of price of resources. This cloud resource allocation is based on market mechanism. The allocation is assumed to be optimal provided the resource allocation state is in equilibrium under the price of each resource. Thus, the cloud resource allocation method actually exploits the benefit of consumer and provider to the maximum level.

For distributing application service traffic across servers, GoGrid Cloud Hosting presented the F5 Load Balancers [4]. Round Robin algorithm is used by load balancer to route the application service requests server by server in an equally distributed fashion with each server taking turns and repeating this cycle. In case of server crash, this algorithm redirects future application service requests to other accessible servers.

Authors in [2] propose the use of satellite communication for VM Provisioning in Federated Cloud Environment. They have focused on use case of WEB TV Company that performs live streaming and uses a distributed IaaS. They have shown the utility of satellite communication to provide efficient transmission links to carry out opportunistic delivery of huge amount of data or VMs with size more than 10 GB.

Authors in [3] propose the use of agent-based resource- allocation method to improve resource allocation. Authors in this paper suggest a broker based multi-agent system.

Authors in [5] proposed a greedy heuristic to maximize the number of scheduled applications. This method is used to receive the applications arriving at input and allocates the application to a particular cloud. They also presented multi-objective genetic algorithm in geographically distributed cloud federation to optimize CO2 emission and energy consumption in order to increase the providers' profits.

3. Problem Formulation

In the context of Federated Cloud computing, the problem of resource allocation can be stated as to allocate tasks from users to datacenters with an overall objective of optimizing performance and cost. The problem of resource allocation is a non-trivial one and the performance of cloud federation greatly depends upon it. Selection of certain algorithm for resource allocation greatly impacts the cost to the user. While these three resource allocation algorithms namely random, round robin and greedy algorithm have been investigated in the past but no experimental research has been done to compare these algorithms in federation of clouds. This paper intends to experimentally compare these algorithms on the cost parameter in the context of Cloud Federation.

4. Experimental Setup

To simulate random, greedy and round robin algorithms, SmartFed simulator was used. A federation has been simulated with three geographically separated Cloud Providers having one datacenter each. The cost of using these datacenters varies from provider to provider and is taken from real life examples such as Amazon, Rackspace and Google. Table 1 depicts the details of cost for each datacenter in terms of US dollar.

Table 1: Cost for usage of resources for each DataCenter

	First Data Center (cost in $)	Second Data Center (cost in $)	Third Data Center (cost in $)
Bandwidth cost (per GB)	0.12	0.12	0.12
Small VM (per Hour)	1.70	1.92	1.008
Large VM (per Hour)	3.14	3.84	2.016
Memory cost(per GB per hour)	0.05	0.05	0.05
Storage cost (per GB per hour)	0.001	0.001	0.003

Host machines of these providers are connected to each other via high-speed links having a bandwidth of 64Mbps. Storage capacity of each host is 10 TB and RAM is 8 GB and each host is equipped with 8 PEs. The processing capability of resources was confined to be same for evaluating all three algorithms. The number of cloudlets submitted to the federation was confined to be 1000 in a single batch.

IJSER

Other underlying assumptions in these experiments are that in each of the datacenter, number of hosts, number of PEs and provisioning policies for VM, RAM and bandwidth are kept uniform. After input jobs are submitted, the federation allocator returns a mapping solution allocating each job to a particular datacenter. The decision of binding cloudlets to respective datacenter is made by implementing abovementioned allocation schemes. In this paper, allocation schemes have been implemented using SmartFed federation simulator.

5. Results and Discussion

Table 2 depicts the details of output achieved from the simulation for Round Robin, Random and Greedy algorithms. The table lists the cost due to each datacenter and total cost of execution of applications on the federation of clouds.

Table 2: Cost of executing Application in a federation of 3 Cloud Providers

Algorithm	First Data Center Cost (in $)	Second Data Center Cost (in $)	Third Data Center Cost (in $)	Total Cost (in $)
Random	554.12	625.92	351.79	1531.83
Round Robin	569.47	639.36	335.66	1544.49
Greedy	--	--	1008.00	1008.00

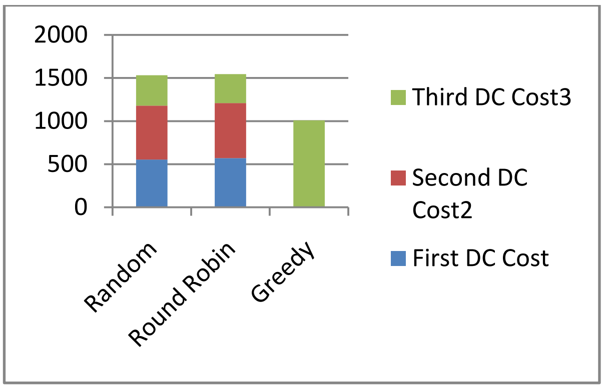

Figure 1: Cost of executing application with 1000 Cloudlets

Fig. 1 depicts the cost for all three resource allocation algorithms where a single application with 1000 cloudlets is executed in federation of clouds. The figure shows that Greedy algorithm outperforms Round Robin and Random algorithm by a huge margin. As the cost of executing the application is closer to $1535 for Random and Round Robin algorithms, it is only $1008 for Greedy algorithm. The total cost of executing the application is decreased by 34.34% with the use of greedy algorithm. Round Robin algorithm resulted in a cost value that was almost as good as Random algorithm and deviated by only 0.82%. These two algorithms namely Round Robin and Random algorithm achieve a total cost of $1531 and $1544 respectively. The increase in cost is 51.96% for Random resource allocation algorithm and by 53.22% for Round Robin algorithm. The benefit of using greedy resource allocation algorithm was quite evident due to its only focus of cost factor.

The important issue which needs to be considered is that although greedy algorithm gives the lowest possible cost as compared to other peer algorithms, it may actually degrade the performance of application in terms of makespan due to other factors viz. bandwidth of the link, response time of datacenter, weak storage strategy etc. even though, the experiment revealed that Greedy algorithm has proved to attain least cost but its selection must be made only after careful examination of the other properties of the datacenter like hardware characteristics of the hosts, network properties etc.

6. Conclusion

The resource allocation scheduling algorithms Round Robin, Random and Greedy algorithm evaluated experimentally in this paper have shown that Greedy algorithm gives the lowest cost for execution of applications since this algorithm exploits the cheapest datacenter first. As cheapest datacenter runs out of resources, greedy algorithm selects the next cheapest datacenter to execute remaining applications. As the resources are scaled up in the cheapest datacenter, the total cost of applications' execution decreases more steeply as compared to Random and Round Robin algorithm. Round Robin algorithm and Random algorithm achieve a comparable average cost. Although Greedy algorithm clearly outperforms in terms of cost but this is due to the underlying assumption that this algorithm does not apply methods of consolidation to increase the capacity of datacenter i.e. resource manager will not instantiate additional

VM on the same host, even if it is underutilized.

References

[1] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan, “The Reservoir model and architecture for open federated Cloud computing,” IBM Journal of Research and Development, vol. 53, no. 4, pp. 1–11, Jul. 2009.

[2] Celesti, Antonio, et al. "Virtual machine provisioning through satellite communications in federated cloud environments." Future Generation Computer Systems 28.1 (2012): 85-93.

[3] Haresh, M. V., Saidalavi Kalady, and V. K. Govindan. "Agent based dynamic resource allocation on federated clouds." Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE. IEEE, 2011.

[4] GoGrid Cloud Hosting (2009) F5 Load Balancer. GoGrid Wiki. http://wiki.gogrid.com/wiki/ index.php/(F5)_Load_Balancer. Accessed 21 September 2009.

[5] Kessaci, Yacine, Nouredine Melab, and El-Ghazali Talbi. "A Pareto-based metaheuristic for scheduling HPC applications on a geographically distributed cloud federation." Cluster Computing 16.3 (2013): 451-468.

[6] L. Rodero-Merino, L. M. Vaquero, V. Gil, F. Gal´an, J. Font´an, R. S. Montero, and I. M. Llorente, “From infrastructure delivery to service management in Clouds,” Future Generation Computer Systems, vol. 26, no. 8, pp. 1226–1240, Oct. 2010.

[7] R. Buyya, R. Ranjan, and R. N. Calheiros, “InterCloud: Utility-oriented federation of Cloud computing environments for scaling of application services,” in Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP’10), ser. Lecture Notes in Computer Science, vol. 6081. Busan: Springer, May 2010, pp. 13–31.

[8] R. V. den Bossche, K. Vanmechelen, and J. Broeckhove, “Cost-optimal scheduling in Hybrid IaaS Clouds for deadline constrained workloads,” in IEEE International Conference on Cloud Computing. Los Alamitos, USA: IEEE Computer Society, 2010, pp. 228–235.

[9] Toosi, Adel Nadjaran, et al. "Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment." High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on. IEEE, 2011.

[10] You, Xindong, et al. "RAS-M: Resource allocation strategy based on market mechanism in cloud computing." 2009 Fourth ChinaGrid Annual Conference. IEEE, 2009.

IJSER