Hosting Virtual Machines on Distributed Datacenters

Chuan Pham
Department of Computer Science and Engineering, KyungHee University, Korea
pchuan@khu.ac.kr

Nguyen H.Tran
Department of Computer Science and Engineering, KyungHee University, Korea
nguyenth@khu.ac.kr

Minh N.H. Nguyen
Department of Computer Science and Engineering, KyungHee University, Korea
minhnhn@khu.ac.kr

Jae Hyeok Son
Department of Computer Science and Engineering, KyungHee University, Korea
sonjaehyeok@khu.ac.kr

Choong Seon Hong*
Department of Computer Engineering, KyungHee University, Korea
cshong@khu.ac.kr

ABSTRACT

Almost of cloud services nowadays are built at top geographically distributed infrastructure for better reliability and performance. These cloud providers need an efficient method to control and direct user workload to suitable datacenter, depending on many factors such as: network traffic cost, operation cost budget, energy consumption, etc. In the virtual machine placement problem, current works mainly focus on the efficiency of packing virtual machines into servers and ignore the distributed scenario of datacenters. In this paper, we consider the problem of placing virtual machines to host applications on a shared resource pool based on distributed cloud platforms. We formulate the problem of hosting virtual machines on distributed datacenters as an optimization problem, and propose a distributed framework DHC that can dynamically direct workload between datacenters to maximize total utility of entire datacenters. We also conduct many case studies to validate our method, and evaluate its effectiveness and practicality, using real-workload traffic traces. The simulation results show that our algorithm can dynamically optimize the total utility, depending on various workload of users.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Resoure management.

Keywords

Optimization theory, Virtual machine placement problem, server, queuing theory, network traffic, delay.

1. INTRODUCTION

*Corresponding author.

IMCOM '16, January 04-06, 2016, Danang, Viet Nam

DOI: http://dx.doi.org/10.1145/2857546.2857633

Cloud computing is becoming a rising paradigm in the information and communications technology industry today. Many cloud services are deployed on multiple datacenters that are located in different region for improving performance and reliability. The term distributed systems (such as distributed computing, distributed databases, etc.) generally refer to that scenario. Fig.1 presents the architecture of distributed cloud computing, where many datacenters cooperate to share the resource pool.

Distributed cloud solutions are very useful to cloud providers to enable virtual network services that span a wide geographical area. It supports more securely. Because all data are not in the same place. It will not be damageable too much in any disaster and will easily recover in small amount of time. This technique also does not need a large resource pool such as network capacity, CPU, memory, and storage at datacenters. The cloud service is spread out everywhere and will be more likely to be close from where users are accessing it. Furthermore, with small capacity, distributed cloud computing requires less electricity almost cuts off power consumption from air conditioning

Moreover, in serving workload, when the number of requests from users in cloud services increases significantly, distributed clouds not only mitigate the suffering from massive connection but also improve load balancing and fault tolerant in cloud services. Secondly, client requests come from wide area. Current approaches often distribute requests to closest datacenters to reduce the latency and bandwidth of network.

However, considering on entire network, requests should balance between datacenters to reduce the operation cost in some specific datacenters, having high workloads. In addition, each datacenter has different resource management policies, such as revenue, difference amount of available resources, power consumption, etc. For example, geographical load balancing and energy cost policy are managed independently by each datacenter. It causes poor performance and high costs in many cases [1].

Working on power consumption of datacenters, the authors of [5] proposed a method that dynamically scales the number of active servers in datacenter depending on workloads to reduce the power consumption in datacenters. However, they did not propose for the distributed cloud datacenters where many providers implement on large-scale geo-

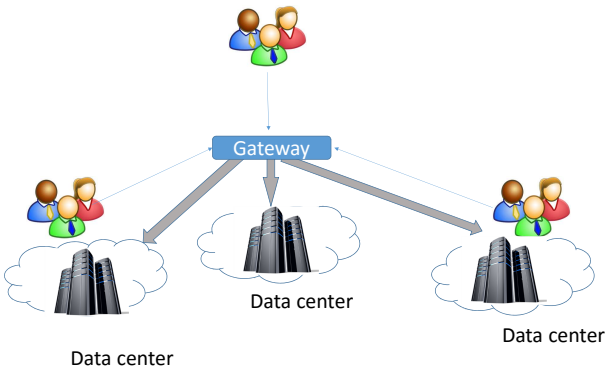


Figure 1: The VM allocation on distributed datacenters.

graphically distributed infrastructure nowadays. Designing an efficient cooperative manner that can share the pool of resources and obtain maximum revenue is still a potential problem in distributed datacenters.

In this paper, we formulate the virtual machine (VM) placement problem in distributed datacenters and propose a framework that can maximize the social-welfare of all datacenters. By using the theory of dual decomposition method, we divide the master problem into multiple sub-problems, which can easily solve at each datacenter. The rest of the paper is organized as follows. Section II represents the related work. The system model and problem formulation is represented in Section III. In Section IV, we solve the optimal problem by decomposition method. In Section V, we describe the experiment environment and simulation results. Finally, we conclude our work in Section VI.

2. RELATED WORKS

Interest in VM placement in datacenters, there are several papers focus on optimizing the total utility of all servers in a datacenter [2, 3, 4]. Using different methods, they successfully pack VMs into servers at one datacenter, but in the distributed scenario, these methods should be considered more factors that really affect to allocation scheme, such as power consumption at each datacenter, network latency, etc. Moreover, the users often submit a bundle of resource demand that includes a list of VMs. These VMs have inside relationships that cannot arbitrarily host in different servers or datacenters.

In another trend, accounting the operation cost of datacenters, many proposals [5, 6, 7, 8] focus on the centralize method to optimal resource management, reducing carbon footprint problem, or load balancing in the centralize datacenter. The authors in [9] proposed an autonomic method that provides resources for service hosting platforms. That method considered with two components: global decision module and local decision module. This architecture can automatically distribute the tasks into appropriate servers and optimize the provisioning resource. However, this is a method that applies into the centralized datacenter. It is also the NP-hard problem in VM placement, which is difficult to find out the optimal solution. Minghong Lin *et al* [5] proposed a method that dynamically scales the number of active servers in datacenters depending on workloads. This method is a good way to reduce the power consumption in datacenters. But they did not propose for the distributed cloud. We consider similar approach but extend into distributed scenario of datacenters, where the complexity of user demand is complicated and there are many factors impact to the efficiency of the controlling scheme of datacenters.

3. SYSTEM MODEL AND PROBLEM FORMULATION

3.1 System modelling

In this section, we show an architecture of our method that illustrates how to partial execution into each datacenter and how to control provisioning resource in distributed datacenters.

Each datacenter consists of a set of active physical servers. Each server can host multiple virtual machines through a hypervisor. To reduce the power consumption, the number of active servers at each datacenter can be changed depending on workloads. We define $X = (x_1, x_2, ..., x_j, ..., x_N)$ is a vector of active servers, which x_j is the number of active servers at datacenter j. We consider a discrete time model, where in each slot the cloud service provider (as a gateway in Fig. 1) receives amount of tasks $A = (a_1, a_2, ..., a_i, ..., a_M)$. Each task a_i is associated with specific performance goals specified in a service-level agreement (SLA) contract. An a_i can embed an arbitrary application topology, which requires a list of VMs, which is called a cluster. To reduce network transfer, a cluster is often hosted at one datacenter.

To model the provision resource in cloud, we consider each VM belong to class k. For example, Amazone EC2 defines a list of instants that clients can choose properly to their application. We define that a_i is the VM allocation vector of task $a_i = (n_{i1}, n_{i2}, ..., n_{ik}, ..., n_{ic})$, where n_{ik} is the number of VMs of class k attributed to task a_i, and c is maximum number of class. Each task a_i requires amount of resource R_i (such as: memory, CPU unit, storage, etc.) that can be calculated as follows:

$$R_i^s = \sum_k^c n_{ik} \cdot I_k^s, \forall s \in \mathcal{S}. \quad (1)$$

where I_k^s is amount of resource type s of instance k, and $\mathcal{S}$ is a set of considered resources.

Resource constraint. To host task a_i, a datacenter has to turn on a number of servers that have sufficient resource. We consider the datacenter has homogeneous servers that can calculate the total available resource based on the number of active server x_j as follows

$$C(x_j) = \beta x_j, \forall j = 1, ..., N, \quad (2)$$

where β is the vector resource of one server (including CPU, memory, storage, etc.). Therefore, the total resource hosting at datacenter j cannot exceed the capacity $C(x_j)$ as follows

$$\sum_i^M H_{ij} R_i^s \leq C_j^s(x_j), \forall j = 1, ..., N, \forall s \in \mathcal{S}, \quad (3)$$

where H_{ij} is an indicating binary variable, $H_{ij} = 1$ if the task a_i is hosted at datacenter j, otherwise, $H_{ij} = 0$.

Further, each task a_i can host totally at one datacenter to guarantee the quality of services of task i. The constraint is represented as follows

$$\sum_j^N H_{ij} = 1, \forall i = 1, ..., M. \quad (4)$$

Delay constraint. While improving the quality of services (QoS), we consider that the allocation of datacenters and demand assignment must satisfy a set of constraints, such as: demand constraint and Service Level Agreement (SLA) performance constraint. We define σ_j be the demand arrival rate from the gateway assigned to datacenter j ($\sigma_j = \sum_{i=1}^{N} H_{ij}$), the demand constraint ensures that all demands are satisfied:

$$\sum_{j}^{N} \sigma_j = D, \tag{5}$$

where D is the average demand arrival rate originated from the gateway. In addition, we define d_j as the network latency between the gateway and datacenter j. At datacenter j, we assume that the arriving demand σ_j is equally split among x_j active servers. From the queueing model M/M/1 [11], the queueing delay between the gateway to a server can be computed as:

$$L_j = \frac{1}{\mu - \frac{\sigma_j}{x_j}}, \forall k = 1, ..., N, \tag{6}$$

where μ is the service rate of each server. The constraint of delay is defined as:

$$\frac{1}{\mu - \frac{\sigma_j}{x_j}} + d_j \leq d_j^{\max}, \forall j = 1, ..., N, \tag{7}$$

where $d_j^{\max}$ is the maximum delay that the service provider tries to achieve.

By defining the constant

$$\alpha_j = \mu - \frac{1}{d_j^{\max} - d_j}, \forall j = 1, ..., N. \tag{8}$$

We can rewrite the constrain (7) as:

$$x_j \geq \frac{\sigma_j}{\alpha_j}, \forall j = 1, ..., N. \tag{9}$$

Budget constraint. In some special cases, the total operation cost at one datacenter is quite high. Especially, when the emergency demand response [12] comes in, the datacenters have to cut-off the power consumption by server turning on/off or workload migration, etc. We define the budget B_j for each datacenter j as the highest operation cost at datacenter j. This constraint depends on the number of active servers at datacenter j as follows

$$\gamma x_j \leq B_j, \forall j = 1, ..., N, \tag{10}$$

where γ is the price that converts the operation cost to a monetary term.

3.2 The distributed datacenter optimization problem

We aim at finding the allocation scheme for each task a_i while maximizing a global utility value U, which is expressed as a weighted sum of the application-provided resource-level utility functions and operating cost function [9]. The resource-level utility function u_i for task a_i is defined as $u_i(R_i)$ that presents the user satisfaction based on the amount of allocated resource (user satisfaction is used interchangeably in this paper).

The distributed cloud model can be formulated as:

$$\max U = \sum_{j}^{N} \sum_{i}^{M} H_{ij} u_i - \sum_{j}^{N} \gamma x_j \tag{11}$$

$$\text{s.t.} \quad \sum_{i}^{M} H_{ij} R_i^s \leq C_j^s(x_j), \forall j = 1, ..., N, \forall s \in \mathcal{S}, \tag{12}$$

$$\sum_{j}^{N} H_{ij} = 1, \forall i = 1, ..., M, \tag{13}$$

$$x_j \geq \frac{\sigma_j}{\alpha_j}, \forall j = 1, ..., N, \tag{14}$$

$$\gamma x_j \leq B_j, \forall j = 1, ..., N, \tag{15}$$

$$H_{ij} = [0, 1], \forall i = 1, ..., M, \forall j = 1, ..., N, \tag{16}$$

$$\text{var. } \{H, x\}. \tag{17}$$

Problem (11)-(17) is an mix-integer linear program (MILP), which is solved by a high complexity algorithm. With thousands of servers in a datacenter, we can further relax the integer H_{ij} as continuous variables such that this problem is tractable. For ease of notation, we name the problem (11)-(17) as DHC (Distributed Hosting datataCenter).

4. DISTRIBUTED CLOUD COMPUTING EMBEDDING DECOMPOSITION ARCHITECTURE

By relaxing the integer constraint, the optimization problem DHC is essentially a linear programming (LP), and can be solved in polynomial time. However, this requires a central coordinator which gather information from all datacenters. Further, the complexity of solving the LP also increases significantly when the problem size scales up. Thus, we are motivated to develop distributed solutions in which the mapping nodes iteratively solve DHC problem.

Particularly, the relaxed problem DHC has follows

$$\max \sum_{j}^{N} \sum_{i}^{M} H_{ij} u_i - \sum_{j}^{N} \gamma x_j + \sum_{i}^{M} \upsilon_i (\sum_{j}^{N} H_{ij} - 1) \tag{18}$$

$$\text{s.t.} \quad \sum_{i}^{M} H_{ij} R_i^s \leq C_j^s(x_j), \forall j = 1, ..., N, \forall s \in \mathcal{S}, \tag{19}$$

$$x_j \geq \frac{\sigma_j}{\alpha_j}, \forall j = 1, ..., N, \tag{20}$$

$$\gamma x_j \leq B_j, \forall j = 1, ..., N, \tag{21}$$

where υ is Lagrange multiplier associated with constraint (13).

The idea of decomposition problem above is to separate the master problem into many sub-problem, in which each datacenter tries to optimal the resource allocation independently. The gateway will collect information from all datacenters and control resource price for all the subproblems. Each datacenter, the subproblem is as follows:

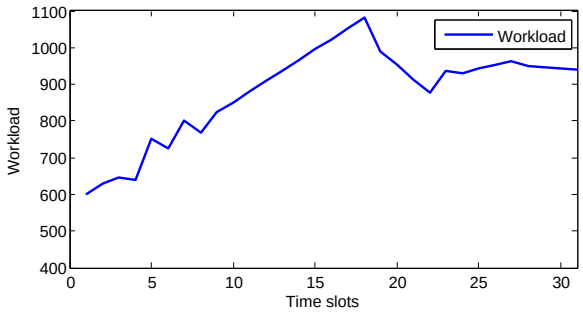


Figure 2: Sample of workload for 31 observed timeslots.

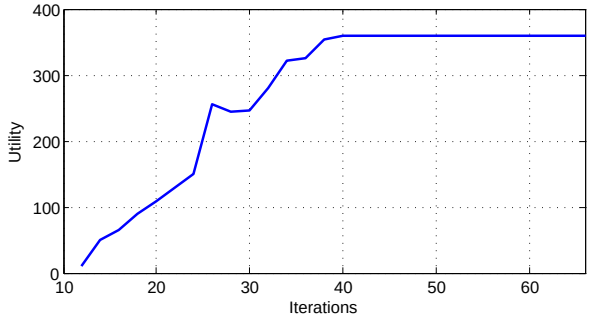


Figure 3: Evaluation of the fast convergence of DHC.

$$\max \sum_{i}^{M} H_{ij}u_i - \gamma x_j + \sum_{i}^{M} \upsilon_i H_{ij} \tag{22}$$

$$\text{s.t.} \quad \sum_{i}^{M} H_{ij}R_i^s \leq C_j^s(x_j), \forall s \in \mathcal{S}, \tag{23}$$

$$x_j \geq \frac{\sigma_j}{\alpha_j}, \tag{24}$$

$$\gamma x_j \leq B_j. \tag{25}$$

At each iteration, given variable H_{ij} and υ, N subproblems are solved simultaneously to obtain the optimal variable x. We apply Dantzig-Wolfe decomposition technique to find the optimal assignment variable H_{ij} and υ in each iteration by solving the master problem as described in [9, chap.12]. The objective of the master problem monotonically increases in each iteration and converges to optimal solution of the original LP.

In general, a gateway can instantiate a set of policies at the master problem, after receiving requests, dictating the order in which the variables need to be optimized and obtain the mapping matrix H and vector X. The gateway receives all information from datacenters and updates H (based on the formula in [9, chap.12]) to calculate for the next step. This process runs until convergence.

5. SIMULATION AND NUMERICAL

In this section, our goal is in two-folds: First, we implement our model to evaluate the performance of DHC. Second, we seek to illustrate the efficient of our model to increase revenue in the context of realistic workloads of Google cluster-usage traces [13] as shown in Fig. 2.

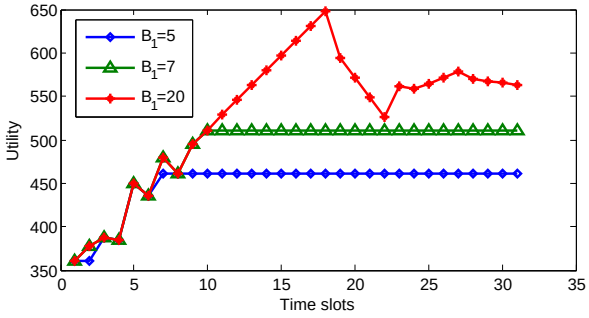


Figure 4: Evaluation of utility at datacenter 1.

5.1 Settings

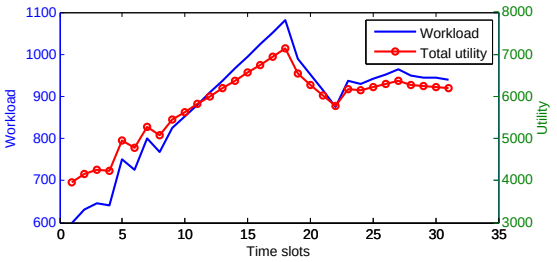
To capture the location diversity of the cloud infrastructure, we set the number of datacenter $N = 10$. For each datacenter, we consider three types of VM instances and one resource type ($c = 3, s = 1$). Each datacenter consists of homogeneous servers which have equally service rate $\mu = 0.2$. In term of the monetary weights, we set $\gamma = 0.01$. The delay d_j is set randomly in range 2 to 5ms and d_j^{max} is set randomly in range 10 to 20ms.

5.2 Results

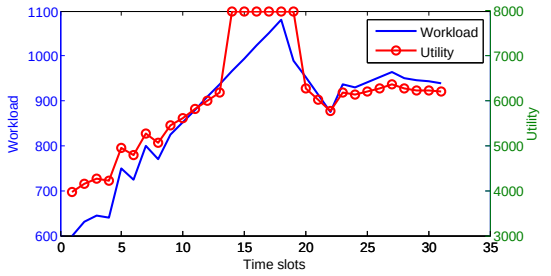
Convergence. According to settings above, we execute DHC to evaluate the convergence of our method. The result in Fig. 3 matches with the characteristics of distributed algorithm DHC, where the convergence occurs after 40 iterations. Using distributed method, the complexity of DHC is reduced significantly in large-scale of practical environment. The huge number of servers in distributed datacenters is broken down and separately solved at each datacenter.

Total utility. To evaluate the efficiency of DHC, we measure the total utility at a specific datacenter (Datacenter 1). By setting different budged B_1, we measure the alternative of utility at datacenter 1 as shown in Fig. 4. Depending on the given budget, datacenter 1 tries to maximize its utility. However, when the cost to serve the arrival workload exceeds the budget, the datacenter 1 will forward the workload to other datacenters. For example, at time slot 10, with budget $B_1 = 7$, the datacenter 1 does not receive the arrival workload. Meanwhile, with higher budget $B_1 = 20$, the fluctuation of utility reflects the dynamic workload at this datacenter.

In another measurement, we take into account total utility of entire network, as shown in Fig. 5. The total utility of distribute datacenters follows workload traffic that dynamically controls the assignment workload at each datacenter and balances between operation cost and budget. However, in Fig. 5b, network traffic cost is so high, each datacenter tries to serve workload at its local datacenter and does not forward to others. In duration time slot 14 to 19, since total operation costs of all distributed datacenter come to the limitation of the budged B_j, all datacenters are frozen. This shows the flexible in serving workload of each datacenter to guarantee its benefit. However, in practical environment, at that time the controller of datacenters will run the backup services, or special policies to guarantee the workload be



(a) Evaluation of total utility of distribute datacenters with normal traffic cost.



(b) Evaluation of total utility of distribute datacenters with high traffic cost.

Figure 5: Comparison between normal network traffic and high network traffic.

executed as least one datacenter, as mentioned in [14].

6. CONCLUSION

In this paper, we study hosting virtual machines in distributed datacenter. In view that existing studies on VM placement problem have been largely isolated to date and resulted in uncoordinated management in distributed scenario. We propose a coordinated approach to enabling sharing workload between datacenter to maximize total utility of entire distributed datacenter. We present a distribute solution, called DHC, that can optimally control workload, delay and budget at each datacenter. We also conduct many case studies to validate our method, showing that DCH can dynamically control workload at all distributed datacenter. The analysis and simulation show that our proposed system is adaptable and can be applied to the scheduling function of data centers. As a future work, we want to apply widely our model in more practical environment, focus deeply about resource constraints of VMs that increases the complexity of DHC problem.

7. ACKNOWLEDGMENTS

This research was supported by the MSIP, Korea, under the G-ITRC support program (IITP-2015-R6812-15-0001) supervised by the IITP. Dr. CS Hong is the corresponding author.

8. REFERENCES

[1] Majid Hajibaba and Saeid Gorgin. A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing. *CIT. Journal of Computing and Information Technology*, 22(2):69–84, 2014.

[2] Huandong Wang, Yong Li, Ying Zhang, and Depeng Jin. Virtual machine migration planning in software-defined networks. *arXiv preprint arXiv:1412.4980*, 2014.

[3] Fabien Hermenier, Xavier Lorca, Jean-Marc Menaud, Gilles Muller, and Julia Lawall. Entropy: a consolidation manager for clusters. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pages 41–50. ACM, 2009.

[4] F Ma, F Liu, and Z Liu. Multi-objective optimization for initial virtual machine placement in cloud data center. *J. Infor. and Computational Science*, 9(16), 2012.

[5] Minghong Lin, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking (TON)*, 21(5):1378–1391, 2013.

[6] Gong Chen, Wenbo He, Jie Liu, Suman Nath, Leonidas Rigas, Lin Xiao, and Feng Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services.

[7] Hrishikesh Amur, James Cipar, Varun Gupta, Gregory R Ganger, Michael A Kozuch, and Karsten Schwan. Robust and flexible power-proportional storage. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 217–228. ACM, 2010.

[8] Peter Bodik, Michael Paul Armbrust, Kevin Canini, Armando Fox, Michael Jordan, and David A Patterson. A case for adaptive datacenters to conserve energy and improve reliability. *University of California at Berkeley, Tech. Rep. UCB/EECS-2008-127*, 2008.

[9] Hien Nguyen Van, Frederic Dang Tran, and Jean-Marc Menaud. Autonomic virtual resource management for service hosting platforms. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, pages 1–8. IEEE Computer Society, 2009.

[10] Pelle Jakovits, Satish Narayana Srirama, and Ilja Kromonov. Stratus: A distributed computing framework for scientific simulations on the cloud. In *High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on*, pages 1053–1059. IEEE, 2012.

[11] Sheldon M Ross. *Introduction to probability models*. Access Online via Elsevier, 2006.

[12] Mohamed H Albadi and EF El-Saadany. Demand response in electricity markets: An overview. In *IEEE power engineering society general meeting*, volume 2007, pages 1–5, 2007.

[13] Google cluster-usage traces. URL http://code.google.com/p/googleclusterdata/.

[14] Shaolei Ren Nguyen Tran, Chuan Pham and Choong Seon Hong. Coordinated energy management for emergency demand response in mixed-use buildings. 2015.