Modeling and Optimization of Resource Allocation in Distributed Clouds

Atakan Aral
Department of Computer Engineering
Istanbul Technical University
Maslak, Istanbul, Turkey
e-mail: aralat@itu.edu.tr
Thesis Advisor: Tolga Ovatman

***Abstract*—This paper is a broad introduction to the resource allocation problems in cloud systems including Inter-Clouds and Mobile Clouds as well as proposed solutions to these problems. Allocation of computing and network resources to cloud tasks requires innovative approaches in each case of cloud data centers, Inter-Clouds and geographically distributed clouds in order to optimize various performance criteria, e.g., latency, throughput and cost.**

***Keywords*-cloud computing, resource allocation, inter-cloud**

I. Introduction

The magnitude of data being stored and processed in the cloud is quickly increasing due to advancements in areas that rely on cloud computing, e.g., Big Data, Internet of Things and computation offloading. Efficient management of limited computing and network resources is necessary to handle such an increase in cloud workload. Some of the critical issues in resource management for cloud computing are modeling resources / requirements and allocating resources to users [1]. Potential benefits of tackling these issues include increases in utilization, scalability, Quality of Service (QoS) and throughput as well as decreases in latency and costs. Such benefits will most certainly accelerate the adoption of cloud computing.

II. Research Areas

A. Data Center Resource Management

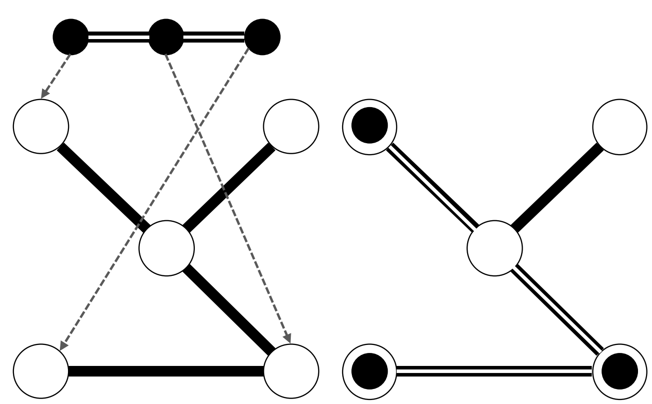
Resources within a cloud data center are usually (and sometimes intentionally) heterogeneous, as are the user demands for those resources. Therefore, arbitrarily and short-sightedly allocating resource entities (e.g., physical or virtual machines) to incoming user demands (e.g., applications) leads to fragmentation and underutilization. In such cases, one solution is to apply a postprocessing technique such as Integer Programming to find an optimum remapping between the resource and demand entities. However, postprocessing would incur migrations between resources which would harm QoS.

In contrast, we proposed a fast heuristic that runs whenever a user demand for resources is received [2]. Among feasible allocations, it chooses the one that yields the maximum evenness in the utilization of resource types (e.g., CPU, bandwidth, memory, storage). Experimental results indicate that keeping utilization rates close to each other indeed significantly increases utilization (four times more optimal placements), puts off postprocessing (up to 12.1%), and decreases the number of migrations in postprocessing (up to 34.5%). Here, selection of the evenness metric is a critical decision. Our evaluation highlights the minimum span metric which focuses solely on the outlier instances of utilization rates.

B. Inter-Cloud Resource Management

Resource modeling and allocation problems get more complicated when a distributed scenario such as Inter-Cloud is considered instead of a single data center. Inter-Cloud (or Cloud Federation) is a distributed model for cloud computing where coordinated cloud providers share their resources and dispatch workload to each other [3]. The most significant benefits of the model to the provider are better scalability, geographical coverage, and resilience.

We proposed a peer-to-peer resource allocation framework for Inter-Cloud called RalloCloud [5], which supports scaling workload across multiple clouds (see Figure 1), thus providing easy migration, high QoS for geo-distributed demand, and the possibility to exploit vendor pricing policies. However, it also requires the consideration of network



(a) Matching (b) Allocation

Figure 1. An example for the Inter-Cloud resource mapping [4].

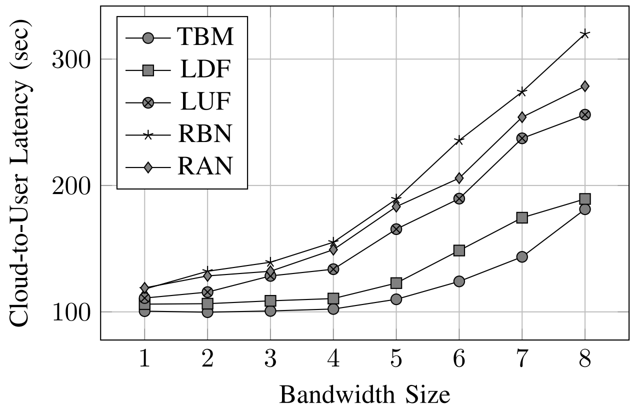


Figure 2. Evaluation results for TBM in terms of user latency with varying Inter-Cloud bandwidth capacity.

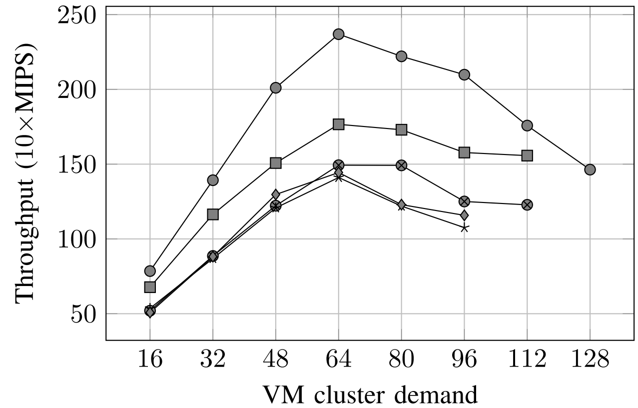


Figure 3. Evaluation results for TBM in terms of throughput with varying user demand for VM clusters.

topology for resource allocation to realize abovementioned benefits. Consequently, we also proposed a novel virtual machine cluster embedding algorithm called Topology Based Mapping (TBM) [4], [5] that aims to find an efficient mapping between the physical Inter-Cloud topology and user demands in the form of virtual topologies. It employs a graph theoretical approach (i.e., subgraph isomorphism) in combination with greedy heuristics.

The main objectives of the algorithm are to reduce network delay and optimize bandwidth utilization. Comprehensive evaluation demonstrated the efficiency of the resulting resource allocation as it achieved better job execution time (makespan), throughput, rejection rate, average network delay and average resource cost in comparison to the outputs of the baseline methods under various experimental configurations. Two selected results are provided in Figures 2 and 3. Here, baseline methods are Least-Delay-First (LDF), Least-Utilized-First (LUF), Round-Robin (RBN) and Random (RAN) mapping heuristics.

C. Geo-Distributed Resource Management

As the volume and velocity of data in the cloud is increasing, the geographical distribution of where it is produced, processed and consumed is also gaining more significance. It is getting less feasible to move data to a distant data center for processing and move output again to the consumer location. Several promising approaches including Cloudlets [6] and Fog Computing [7] are instead suggesting to bring processing entities to the edge of the cloud network to reduce latency. This is especially useful in code offloading for mobile cloud applications [8].

One issue we have identified in this scenario regarding resource management is the latency between the processing entity and the data. Although the above-mentioned approaches reduce the latency between the user and the processing entity, the data required for the cloud application is usually stored in a centralized SaaS provider. It is not feasible to replicate entire data in large number of geo-distributed locations due to economical factors. In addition, edge processing entities (e.g., cloudlets, nano data centers) have extremely limited storage capacity in comparison to SaaS infrastructure. That is why we propose creating caches of individual data objects on multiple locations based on the magnitude and location of user demand as well as storage pricing in attempt to reduce data access latency.

Optimal selection of the number and location of the caches is a challenging problem due to the varying/mobile nature of user demand and the trade-off between cost (number of caches) and latency. Moreover, knowledge of the complete topology including capacities, latencies and prices in such a fine-granular infrastructure is not realistic. Thus, a distributed and context-aware cache placement algorithm is required.

Suggested algorithm, which originates from the classical facility location problem, may carry out one of the four operations at each iteration based on a heuristic decision. Conditions for these operations are provided below where:

D_{ij} = Demand for a data object i from a neighbour j
L_{ij} = Avg. latency for a data object i from a neighbour j
N_{jk} = Latency from a node k to a neighbour j
C_{ij} = Cost of storing a data object i at a neighbour j
A = User provided level of expansionism

1) Cache Creation: A cache creation decision can be taken only at the central data storage and the cache can be created in one of its neighbours. A cache of object i at neighbour j is created if and only if Equation 1 holds true.

$$L_{ij} D_{ij} A > C_{ij} \tag{1}$$

2) Cache Elimination: A cache elimination decision (and the other following decisions) can only be taken at a existing cache location. Cache of the object i at k is removed if and only if Equation 2 holds true.

$$\sum_{\forall j}(L_{ij}D_{ij}A) < C_{ik} \quad (2)$$

3) Cache Duplication: Cache of the object i at location k is duplicated to its neighbour l if and only if Equations 3 and 4 both hold true.

$$L_{il}D_{il}A > C_{il} \quad (3)$$

$$\sum_{\forall j \neq l}(L_{ij}D_{ij}A) > C_{ik} \quad (4)$$

4) Cache Migration: Cache of the object i is migrated from location k to its neighbour l if and only if and only if Equation 5 holds true.

$$\begin{aligned} &\sum_{\forall j}(L_{ij}D_{ij}A) \\ &- (L_{il} - N_{kl})D_{il}A \\ &- \sum_{\forall j \neq l}\big((L_{ij} + N_{kl})D_{ij}A\big) > C_{il} - C_{ik} \end{aligned} \quad (5)$$

When the conditions for multiple operations hold true, elimination will be given the least priority and the decision between other operations will be made based on the amount of gain, i.e., the difference between the two sides of the inequality.

III. Conclusion and Future Work

One of the major issues in cloud research is the development of efficient resource allocation strategies. The problem is highly challenging especially in the cases of distributed and federated clouds. Our aim is to suggest solutions for this issue in multiple levels (i.e., within a data center, a federated cloud and a geographically distributed cloud) to benefit both cloud providers and users in terms of higher quality of service, scalability, availability and adaptability. Proposed algorithms address resource utilization, network factor, cost-performance tradeoff, geographical coverage and user mobility.

We present a timeline of the studies in Figure 4. As of January 2016, we have completed and published data center and Inter-Cloud resource management solutions. However, we have only recently begin working on the distributed resource management. Future work includes the implementation of the algorithm, its evaluation on simulation environment and modification based on the experimental results. Current version it is merely a draft which will take its final form after several iterations.

One of our early observations on the algorithm is the risk of getting stuck in local optimum. This may occur when duplication and migration conditions do not hold due to a high-cost node on the path to the user demand while it is

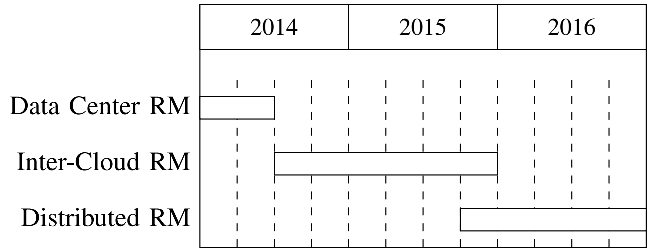


Figure 4. Timeline of the thesis studies.

actually beneficial to store a cache in the forthcoming nodes on the path. Since the algorithm distributes the caches one hop at a time, it will not be able to surpass the costly node. Another interesting research question is about the size of data objects. Increasing size would incur higher cost but may also result in better hit ratio, thus less cache creation.

Acknowledgment

The Ph.D. thesis study summarized in this paper is partially supported by *NETAŞ*, *TÜBİTAK – The Scientific and Technological Research Council of Turkey* and *Istanbul Technical University Unit for Scientific Research Projects*.

References

[1] S. S. Manvi and G. K. Shyam, “Resource Management for Infrastructure as a Service (IaaS) in Cloud Computing: A Survey,” *Journal of Network and Computer Applications*, vol. 41, pp. 424–440, 2014.

[2] A. Aral and T. Ovatman, “Improving Resource Utilization in Cloud Environments using Application Placement Heuristics,” in *4th International Conference on Cloud Computing and Services Science, CLOSER 2014*, 2014, pp. 527–534.

[3] R. Buyya, R. Ranjan, and R. N. Calheiros, “Intercloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services,” in *Algorithms and Architectures for Parallel Processing*. Springer, 2010, pp. 13–31.

[4] A. Aral and T. Ovatman, “Subgraph Matching for Resource Allocation in the Federated Cloud Environment,” in *IEEE 8th International Conference on Cloud Computing, CLOUD 2015*, 2015, pp. 1033–1036.

[5] ——, “Network-Aware Embedding of Virtual Machine Clusters onto Federated Cloud Infrastructure,” *[submitted for journal publication]*, 2016.

[6] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The Case for VM-Based Cloudlets in Mobile Computing,” *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog Computing and Its Role in the Internet of Things,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[8] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading,” in *INFOCOM 2012 Proceedings*. IEEE, 2012, pp. 945–953.