

# Chapter 1

## Introduction

Data centers are large-scale computing infrastructures which consume huge amount of energy each year: a typical data center consumes as much energy as 25,000 households [5]. Thus, reducing the energy consumption becomes the major concern of Cloud providers. In addition, data centers and computation powers support the modern Cloud computing industry, software industry and etc. Therefore, reducing the cost of data centers will lead to a reduction of cost of softwares which consequently be beneficial to most people who access the Internet on a daily basis. Among several components that consume energy such as cooling system, physical machines (PMs) (e.g servers), and network devices, PMs accounts for 40% and have a huge improvement space, since they are always in low utilization (e.g on average, from 10% to 50% of required resources) [1, 16]. This low utilization of resource problem can be solved by fine granularity management of Cloud resources (e.g CPUs and RAMs) using a new virtuzliation technology: containers [6, 7, 17] and a new service model: Container as a Service (CaaS) [15]. CaaS is a mixture of traditional IaaS (Infrastructure as a Service) [12] and PaaS (Platform as a Service); it utilizes both containers and virtual machines (VMs) as the fundamental resource management units. In CaaS, applications that were used to deployed in VMs (e.g in IaaS) are now deployed in containers. Container is an operating system (OS) level of virtualization; multiple containers can run on a VM and share OS. Therefore, server consolidation [18] can be applied in a joint of containers and VMs environment to achieve better energy reduction.

Server consolidation is an important strategy in improving the utilization throughout the Cloud resource management processes as shown in Figure 1.1 including new application allocation [8], periodic optimization [13], overloading and under-loading adjustments [13]. According to the characteristic of each process, server consolidation can be roughly classified into two categories: static problems [13] and dynamic problems [2]. Accordingly, server

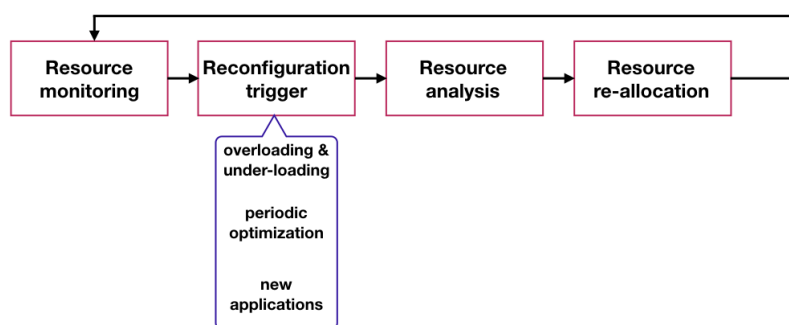


Figure 1.1: A workflow of resource management [13]

consolidation approaches also have corresponding categories. Static approaches use historical average resource utilization data as input to map applications to PMs. Static consolidation normally involves large amount of applications and PMs, therefore, the optimization is quite time-consuming and often conducted in a off-line fashion. Periodic optimization belongs to this category. It takes a number of existing applications and re-allocate them into a number of PMs. Dynamic approaches take one application each time, allocates it into one of the PMs. The operation is conducted in an online fashion, therefore, it requires fast reaction. Overloading and under-loading can be categories to dynamic consolidation problem [3]. New application allocation can be seen as either static: allocate a batch of new applications, or dynamic: allocate a new application each time. In this proposal, we consider it as a static problem.

Examples of static server consolidation are shown in the following in VM-based and container-based Cloud. In traditional VM-based Cloud, Server consolidation can be described as, given a number of Physical Machines (PMs) which can be represented as the resources(e.g CPU cores and RAM); a number of requests for fixed configurations of VMs (assume applications have been deployed in VMs), each configuration can also be represented as aforementioned resources; The objective is to allocate these requested VMs into a minimum number of PMs. The decision variable is the location of each requested VM. In container-based Cloud, instead of allocating requested VMs in PMs, a set of containers (assume applications have been deployed into containers) represented as resources is first allocated to a number of fixed type VMs, then, these VMs are allocated to PMs. The decision variables are the allocation of containers (upper level) and VMs (lower level). For the upper level of allocation, the objective is to maximize the utilization of resources (e.g a balanced utilization among several resrouces), while the lower level's objective is to minimize the number of PMs.

Traditional VM-based server consolidation are modeled as bin-packing problems [10]. This is because VMs and PMs are naturally modeled as items and bins and server consolidation and bin-packing have the same optimization objective: minimize the number of bins/PMs. The complexity of bin-packing problem is NP-hard which means it is extreme time-consuming to find its optimal solution when the number of decision variables are large. Container-based server consolidation can be categorized as a bilevel optimization problem [4] where in lower-level: Containers to VM and in upper-level: VMs to PMs. These two levels optimization are connected through decision variables. In this case, two levels of optimization are both bin packing problems and they are cooperating [9].

Currently, most research focus on VM-based server consolidation and these methods can not be directly applied on container-based consolidation because of the different structure. Only few research focus on container-based server consolidation problem. One of the state-of-the-art research is from Piraghaj and et al [15]. They first propose a VM-resizing technique that defines the types of VM based on analyzing the historical data from Google. Then they propose a two-step allocation: first allocate containers to VMs and then allocate VMs to PMs. Their major contribution is the method of defining types of VM. The allocation of containers does not optimize the energy consumption and the allocation of VMs are traditional First Fit algorithm. In addition, they propose a dynamic consolidation [14] using a series simple heuristics such as Random Host Selection Algorithm or First Fit Host Selection. Their resource allocation system completely relies on dynamic consolidation without using static methods. Although their system can execute allocation fast, the energy efficiency cannot be guaranteed. The reasons are mainly from two aspects, firstly, they mainly rely on simple bin-packing algorithms to allocate containers to VMs. As Mann's research [10] showed, server consolidation is a lot more harder than bin-packing problem because of the multi-dimensional of resources, many constraints. Therefore, general bin-packing algorithms do

not perform well. Secondly, they use a two-step allocation. Because of the interaction of two allocations, separated optimization approach will lead to local optima [11]. Therefore, these two allocations should be considered simultaneously.

The overall goal of this thesis is to develop new container-based server consolidation approaches to solve three problems: joint allocation of containers and VMs, periodic global optimization and dynamic consolidation.



# Bibliography

- [1] BARROSO, L. A., AND HÖLZLE, U. The Case for Energy-Proportional Computing. *IEEE Computer* 40, 12 (2007), 33–37.
- [2] BELOGLAZOV, A., AND BUYYA, R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurrency and Computation - Practice and Experience* 24, 13 (2012), 1397–1420.
- [3] BELOGLAZOV, A., AND BUYYA, R. Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints. *IEEE Transactions on Parallel and Distributed Systems* 24, 7 (2013), 1366–1379.
- [4] COLSON, B., MARCOTTE, P., AND SAVARD, G. An overview of bilevel optimization. *Annals OR* 153, 1 (2007), 235–256.
- [5] DAYARATHNA, M., WEN, Y., AND FAN, R. Data Center Energy Consumption Modeling - A Survey. ... *Surveys & Tutorials* (2016).
- [6] FELTER, W., FERREIRA, A., RAJAMONY, R., AND RUBIO, J. An updated performance comparison of virtual machines and Linux containers. *ISPASS* (2015), 171–172.
- [7] HE, S., GUO, L., GUO, Y., WU, C., GHANEM, M., AND HAN, R. Elastic Application Container: A Lightweight Approach for Cloud Resource Provisioning. In *2012 IEEE 26th International Conference on Advanced Information Networking and Applications (AINA)* (Feb. 2012), IEEE, pp. 15–22.
- [8] JENNINGS, B., AND STADLER, R. Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management* 23, 3 (2015), 567–619.
- [9] LEGILLON, F., LIEFOOGHE, A., AND TALBI, E.-G. CoBRA - A cooperative coevolutionary algorithm for bi-level optimization. *IEEE Congress on Evolutionary Computation* (2012), 1–8.
- [10] MANN, Z. Á. Approximability of virtual machine allocation: much harder than bin packing.
- [11] MANN, Z. Á. Interplay of Virtual Machine Selection and Virtual Machine Placement. In *Service-Oriented and Cloud Computing*. Springer International Publishing, Cham, Aug. 2016, pp. 137–151.
- [12] MELL, P. M., AND GRANCE, T. The NIST definition of cloud computing. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD, Gaithersburg, MD, 2011.

- [13] MISHRA, M., DAS, A., KULKARNI, P., AND SAHOO, A. Dynamic resource management using virtual machine migrations. *IEEE Communications ...* 50, 9 (2012), 34–40.
- [14] PIRAGHAJ, S. F., CALHEIROS, R. N., CHAN, J., DASTJERDI, A. V., AND BUYYA, R. Virtual Machine Customization and Task Mapping Architecture for Efficient Allocation of Cloud Data Center Resources. *Comput. J.* 59, 2 (2016), 208–224.
- [15] PIRAGHAJ, S. F., DASTJERDI, A. V., CALHEIROS, R. N., AND BUYYA, R. Efficient Virtual Machine Sizing for Hosting Containers as a Service (SERVICES 2015). *SERVICES* (2015).
- [16] SHEN, S., VAN BEEK, V., AND IOSUP, A. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. *CCGRID* (2015), 465–474.
- [17] SOLTESZ, S., PÖTZL, H., FIUCZYNSKI, M. E., BAVIER, A. C., AND PETERSON, L. L. Container-based operating system virtualization - a scalable, high-performance alternative to hypervisors. *EuroSys* 41, 3 (2007), 275–287.
- [18] VARASTEH, A., AND GOUDARZI, M. Server Consolidation Techniques in Virtualized Data Centers: A Survey. *IEEE Systems Journal* (2015), 1–12.