Chapter 1

Introduction

1.1 Problem Statement

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 [13]. 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) [5, 42]. 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 [15,21,45] and a new service model: Container as a Service (CaaS) [38]. CaaS is a mixture of traditional IaaS (Infrastructure as a Service) [33] 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 [50] can be applied in a joint of containers and VMs environment to achieve better energy reduction.

Server consolidation is an important stategy in improving the utilization throughout the Cloud resource management processes as shown in Figure 1.1 including new application allocation [24], periodic optimization [34], overloading and under-loading adjustments [34].

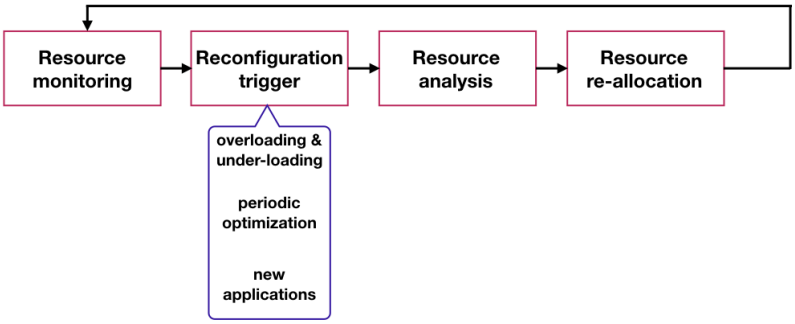

Figure 1.1: A workflow of resource management [34]

According to the characteristic of each process, server consolidation can be roughly classified into two categories: static problems [34] and dynamic problems [7]. Accordingly, server 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 [8]. 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 [31]. 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 [12]. Bilevel problems are typically non-convex and strongly NP-hard [52]. In this case, two levels are lower-level: Containers to VM and 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 [28].

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 [38]. 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 [37] 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 sim-

ple bin-packing algorithms to allocate containers to VMs. As Mann's research [31] 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 [32]. 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.

1.2 Motivation

In this thesis, we aim at providing a series of approaches to continuously optimize the joint allocation of VMs and containers. A continuous optimization procedure mainly involves with three types of server consolidation: initialization, global consolidation, and dynamic consolidation. Different stages have distinctive goals, therefore, they are considered as separated research questions. In addition, a scalability problem of static optimization is considered as an optional objective.

1. Joint allocation of containers and VMs (new applications initialization),
   In this research, we take Joint allocation as a static problem which is fundamental for server consolidation problem. At this stage, a set of containers is allocated to a set of VMs and these VMs are allocated to a set of PMs. This task is challenging because the problem is a bilevel optimization where each level is a bin packing problem. Exhaustive search of entire solution space is practically impossible, for the number of possible permutation of solution is huge. Current approaches [22, 37] use simple heuristics such as First Fit to solve the problem. These greedy-based heuristics do not consider the complex structure of the problem, therefore, often reach a local optimal solution.

2. Global consolidation,
   A Global consolidation is conducted to improve the global energy efficiency in a periodical fashion. Data center constantly receives new allocations, releasing of old resources. These changing degrades the compact structure of a data center. Therefore, the data center needs a global optimization to improve the overall energy efficiency.

   The challenges are three folds, firstly, similar with initialization problem, the problem has two level of allocations and they interact with each other. Secondly, like VM-based consolidation, Container-based consolidation is considered as a multi-objective problem with minimization of migration cost as well as keeping a good energy efficiency. In bilevel optimization, multi-objective can be defined in either or both level, therefore, it further increases the complexity. Thirdly, consolidation is a time-dependent process which means the previous solution affects the current decision. Previous VM-based research only consider each consolidation as an independent process. As a consequence, although in one consolidation, the migration is minimized, It may lead to more migrations in the future consolidation. We will consider the robustness of consolidation and propose a novel time-aware server consolidation which takes the previous immediate consolidation and the future consolidation into consideration.

3. Dynamic consolidation,
   It takes one container and allocates it to VMs. Since the size of container can be dynamically adjusted, when the an application is under-provision or over-provision, the

original container is halted, resized and re-allocated. Hence, there is a need to allocate this new container in real time.

To solve a dynamic consolidation, heuristics and dispatching rules are often used [6, 17, 40, 43]. In this scenario, a dispatching rule is considered as a function that determines the priorities of VMs that a container can be placed. However, dynamic placement is much complex than bin-packing problem [31]. Because of its dynamic nature, human designed heuristics are ill-equipped in approximating solutions when the environment has changed [47].

Hyper-heuristic methods, sepcifically, Genetic Programming (GP) technique [3] can learn from the best previous allocation and automatically evolves dispatching rules to solve this problem. GP has been applied in generating dispatching rules for bin-packing problem [10, 47] and other scheduling problems [36]. The results have shown promising results.

There are mainly two challenges, first, it is difficult to identify the related factors that construct the heuristic. Factors or features are the building blocks of heuristics. It is a difficult task because the relationship between a good heuristic and features are not obvious. Second, representations provide different patterns to construct dispatching rules. It is also unclear what representation is the most suitable for the consolidation problem.

4. Large-scale of static server consolidation problem,
   In this case, initialization and global consolidation are belonged to this category. Since Cloud data center typically has hundreds of thousands PMs and more, static server consolidation is always very challenging. Many approaches have been proposed in the literature to resolve the problem. There are mainly two ways, both relied on distributed methods, hierarchical-based [25, 35] and agent-based management systems [57]. The major problem in agent-based systems is that agents rely on heavy communication to maintain a high-level utilization. Therefore, it causes heavy load in the networking. Hierarchical-based approaches are the predominate methods. In essence, these approaches are centralized methods where all the states of PMs within its region are collected and analyzed. The major disadvantage of hierarchical-based approaches is that it only provides local solutions. In fact, it is infeasible and unnecessary to check all the states of PMs since the search space is too large and most PMs do not need a change. This idea motivates a way to improving the effectiveness is to reduce the number of variables so that the search space is narrowed. In this thesis, we are going to investigate the way to eliminate the redundant information.

1.3 Research Goals

1.3.1 Objective One: Develop EC-based approaches for the single objective joint allocation of containers and VMs

Currently, most research focus on VM-based server consolidation technique. They often modeled this problem as a vector bin-packing problem [58]. Container adds an extra layer of abstraction on top of VM. The placement problem has become a two-step procedure, in the first step, containers are packed into VMs and then VMs are consolidated into physical machines. These two steps are inter-related to each other. Previous research [38] solve this problem in separated steps where the first step allocate containers to VMs and the second step allocate VMs to PMs with simple bin-packing heuristics. Therefore, this is the first research that trying to solve the problem.

1. First, our first sub objective is to propose a descriptive single objective model for the bilevel optimization problem of joint allocation of container and VM. The reason to establish this model is because current server consolidation models are mostly VM-based, they cannot be directly applied on bilevel problems. Therefore, variables, constraints and objective functions need to be clarified before applying any optimization algorithm. Each level of the problem will be formulated to a multi-dimensional vector bin packing problem. It is still unclear that which objective function is the best to capture the relationship between container and VM so that the overall energy is low. We will investigate several resource wastage models [16, 18, 56] and select a suitable one. In addition, several models have to be considered, including energy model [13], price model [1], and workload model [30].

2. Second, we will first develop a baseline approach that solve the problem using nested Evolutionary algorithms [44]. We will start from the simplest form: one dimensional bin-packing in each level to more complex multi-dimensional bin-packing.

   Nested methods have been used in solving bilevel problem for years, they are reported as effective approaches. We will investigate several approaches such as Nested Particle Swarm Optimization [29], Differential evolution (DE) based approach [2, 60] and Co-evolutioanry approach [28]. In order to adapt our problem to these existing approaches, we will develop suitable representations and genetic operators.

3. Third, although nested approaches have been reported effective, they are often very time consuming. Therefore, our third sub-objective will focus on developing more efficient algorithms. There are several possible directions to be explored such as metamodeling-based methods [54] and single-level reduction.

1.3.2 Objective Two: Develop EC-based approaches for the multi-objective joint allocation problem

As previous section (see 1.2) mentioned, the task is multi-objective since the number of VM migration has to be minimized while keep the overall energy low. In addition, periodic optimization is a time-dependent problem which means the optimal consolidation in previous operation might lead to more migrations in the current consolidation. The robustness of a data center is particularly important. The robustness measures the stableness of result of consolidation.

1. First, we will develop EC-based approaches to solve the multi-objective joint allocation problem. In this problem, multiple objectives may involve at both of the levels. We will start from a simple case considering multi-objective in lower level: Minimizing VM migration and energy consumption. Currently, there are few the studies using EC methods [**?**, **?**] for multiobjective bilevel optimization. We will investigate which one is more suitable for this binary problem. Furthermore, like the case in single objective problem, we need to develop new representations, genetic operators to apply the algorithms to solve the problem.

2. Second, we will design a robustness measure. Previous studies only use simple measurement which counts the migration number between two static consolidation. This measurement aims at minimizing the number of migration between two static placement processes. It may cause more migration in the next consolidation. Therefore, it needs a time-aware measure of the robustness of system. Therefore, in this objective, the first sub-problem we are going to solve is to propose a robustness measure.

Currently, only a few research propose robustness aware server consolidation techniques [**?**, **?**] have been proposed. They are either static threshold or probability-based threshold to measure the robustness of PMs. We will investigate an adaptive measure based on the historical data and current status.

3. Third, we will design a proactive server consolidation approach. Based on a prediction of future server consolidation and the robustness measure,we will first design an approach which maximize the robustness and also minimize the current energy consumption. Proactive consolidation [**?**, **?**] has been studied extensively. Their experience in analyzing the workload patterns can be useful in designing new algorithms.

1.3.3 Objective Three: Develop a hyper-heuristic Genetic Programming (GP) approach for automatically generating dispatching rules for dynamic consolidation

Previously, dynamic consolidation methods,including both VM-based and container-based, are mostly based on bin-packing algorithm such as First Fit Descending and human designed heuristics. As Mann's research [31] showed, server consolidation is more harder than bin-packing problem because of multi-dimensional of resources and many constraints. Therefore, general bin-packing algorithms do not perform well with many constraints and specific designed heuristics only perform well in very narrow scope. Genetic programming has been used in automatically generating dispatching rules in many areas such as job shop scheduling [36]. GP also has been successfully applied in bin-packing problems [10]. Therefore, we will investigate GP approaches for solving the dynamic consolidation problem. We will start from considering one-level of problem: migrate one VM each time to a PM.

1. First, we will investigate which features and attributes are important when dealing with energy efficiency problem. As the basic component of a dispatching rule, primitive set contains the states of environment including: status of VMs (e.g. utilization, wastage), features of workloads (e.g. resource consumption). Although there is no research has investigate how to use them to construct dispatching rules, there are extensive statistical analysis on workload [51]. The effectiveness of functional set and primitive set will be tested by applying the constructed dispatching rules on dynamic consolidation problem.

2. Develop GP-based methods for evolving dispatching rules. This sub-objective explores suitable representations for GP to construct useful dispatching rules. It also proposes new genetic operators as well as search mechanisms.

1.3.4 Objective Four (Optional) Large-scale Static Consolidation Problem

Propose a preprocessing method to eliminate redundant variables Current static consolidation takes all servers into consider which will lead to a scalability problem. In this objective, we will investigate two branches of methods, first one categorizes a number of containers into fewer groups so that the granularity will decrease [38]. Second method categorizes PMs so that only a small number of PMs are considered. This approach will dramatically reduce the search space. The potential approaches that can be applied in this task are various clustering methods.

1.4 Published Papers

During the initial stage of this research, some investigation was carried out on the model of container-based server consolidation [**?**].

1. Tan, B., Ma, H., Mei, Y. and Zhang, M., "A NSGA-II-based Approach for Web Service Resource Allocation On Cloud". *Proceedings of 2017 IEEE Congress on Evolutioanry Computation (CEC2017).* Donostia, Spain. 5-8 June, 2017.pp.2574-2581

1.5 Organisation of Proposal

The remainder of the proposal is organised as follows: Chapter **??** provides a fundamental definition of the Container-based server consolidation problem and performs a literature review covering a range of works in this field; Chapter **??** discusses the preliminary work carried out to explore the techniques and EC-based techniques for the initialization problem; Chapter **??** presents a plan detailing this projects intended contributions, a project timeline, and a thesis outline.

Chapter 2

Literature Review

2.1 Background

This chapter begins by providing an overall understanding of Cloud computing and its related research field.

2.1.1 Cloud computing

Cloud computing is a computing model offers a network of servers to their clients in a on-demand fashion. From NIST's definition [33], *"cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*

To illustrate how it works, considering the case: a Cloud provider builds a data center which contains thousands of servers connected with network. These servers are virtualized which means they are partitioned into smaller units of resources called *Virtual Machines (VMs)* or *Containers* [15]. A web-based application provider can access and deploy their applications (e.g Endnote, Google Drive and etc.) in these resource units from anywhere in the world. Once the applications start serving, application users can use them without installing on their local computers.

Cloud computing involves three stakeholders (see Figure 2.1): Cloud providers, Cloud users (applications providers), and End (application) users [24]. Cloud providers build data centers, provide maintenance and resource management on hardware infrastructures such as servers. Cloud users develop and deploy applications on Cloud infrastructures. End users consumes applications developed by Cloud users and hosted by Cloud providers.

The detailed goal and objectives of stakeholders are described below.

- *Cloud providers*' goal is to increase the profit by boosting the income and reducing the expense. Their income comes from Cloud users' rental of servers or *Physical Machines (PMs)* in terms of resource quality (e.g 3.5GHz dual-core CPU), quantity (e.g 3 PMs), and time (e.g 1 year). Therefore, Cloud providers objective is to maximize utilization of computing resources. A high utilization brings two benefits, firstly, it increases income by accommodating more applications in limited resources. Secondly, it cuts the expense of energy consumption by packing applications in a minimum number of PMs so that idle PMs can be turned off.

- *Cloud users*' goal is also to increase the profit mainly through two objectives, attracting more End users and reduce the expense of resources. The first objective can be

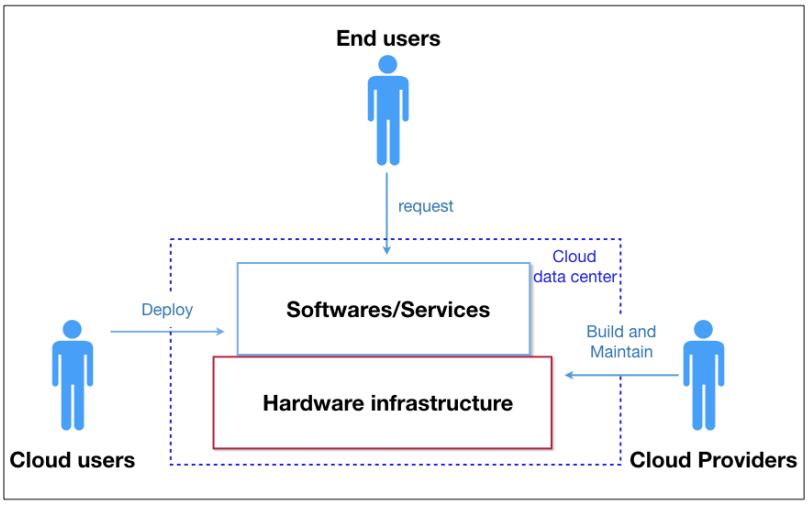

Figure 2.1: Stakeholders of Cloud computing

achieved by improving the quality of service as well as lower the fee for End users. Either way depends not only on the software entities but also the quality of service (QoS) offered by Cloud provider. The second objective can be achieved by a good estimation of the reserved resources, so that they do not rent insufficient or too much resources which cause performance degradation or wastage.

- *End Users*' goal is to obtain a satisfactory service. It is achieved by signing a Service Level Agreement (SLA) with Cloud users which constrains the performance of the services.

Cloud computing has five chacteristics:

1. On-demand self-service: A Cloud user can require computing resources (e.g CPU time, storage, software use) without the interaction with Cloud provider.
2. Broad network access: Computing resources are connected and delivered over the network.
3. Resource pool: a Cloud provider has a "pool" of resources which are normally virtualized servers. In IaaS, it provides predefined sizes of VMs. In PaaS, the resources are 'invisible' to Cloud users who have no knowledge or ability to control.
4. Rapid elasticity: From the perspective of Cloud users, computing resources are assigned and released in real time. In addition, the resources assign to their software is "infinite". Therefore, Cloud users do not need to worried about the scalability of their applications.
5. Measured Service: Cloud provides an accurate measure of the usage of computing resources. It is fundamental to the pay-as-you-go policy.

Cloud computing has three traditional service models [33]: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The relationship among three service models is showed in Figure 2.2. Service models of Cloud computing are critical in solving energy consumption problem because their distinct ways of managing resources have sever effect on the problem. These distinct ways of resource management mainly result from the responsibilities among stakeholders.

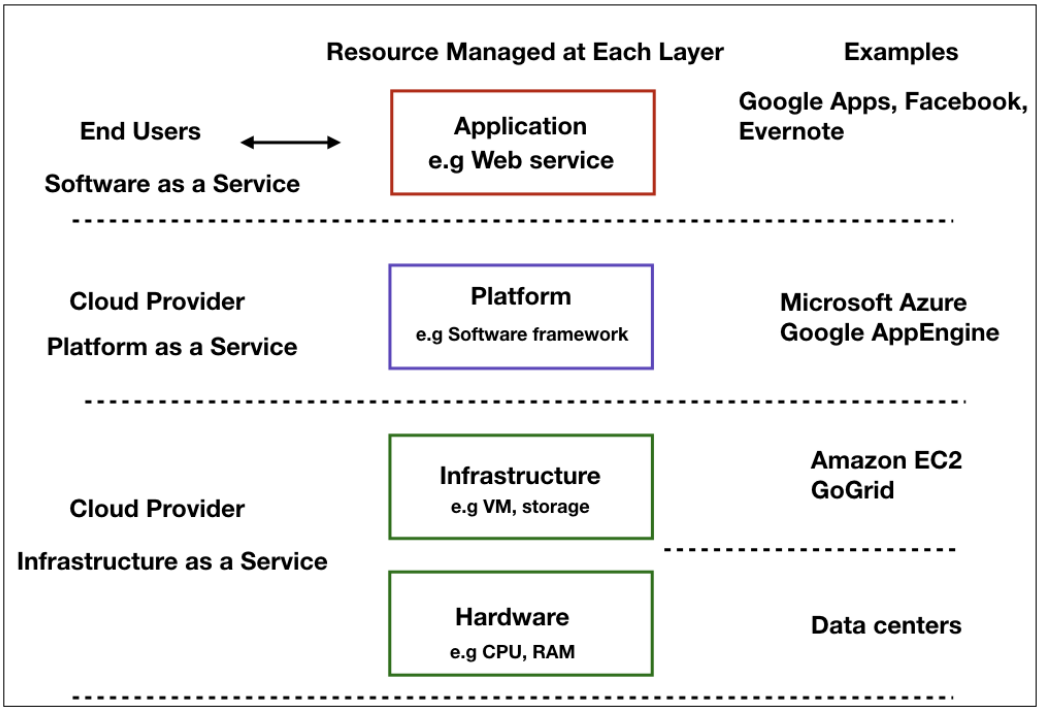

Figure 2.2: Cloud computing architecture [59]. IaaS provides the fundamental resources such as CPU cores, RAM. The resources are usually wrapped with various types of virtual machine. PaaS provides a software platform sitting on top of IaaS for Cloud users to develop applications. SaaS is the application that serves End Users.

- IaaS, a Cloud provider hosts hardwares such as PMs and cooling infrastructure on behalf of Cloud users. Computational resources are often encapsulated in virtualized computing units called virtual machines (VMs). Cloud providers establish a number of types of VM for simplifying the management. The 'type' means a VM contains a certain quantity of resources such as 2-cores and 1 GB RAM. *Traditional* IaaS and PaaS use VM as the fundamental unit of resources.

  A typical procedure of a Cloud user deploying their applications in an IaaS cloud includes several steps. Initially, Cloud users estimate the resources that their applications might consume and select a type of VM which can satisfy the requirement. After Cloud users have made the decisions, they send requests to Cloud providers for a number of VMs. Finally, Cloud providers received the request, provisioned and allocated these VMs to PMs.

- PaaS, a Cloud provider offers a platform which allows Cloud users to develop, test and deploy their applications on it.

  From resource management perspective, PaaS is sitting above the IaaS which means the underlying resource is still based on IaaS VM types. Different from IaaS, PaaS takes the responsibility of selecting VMs and allows Cloud users to focus on software development.

- SaaS, Cloud users develop applications and deploy them on Cloud so that End users can access them via the Internet. Although this service model does not directly related to the resource management, it provides the fundamental reasons for resource management and optimization: applications receive fluctuated requests from End users.

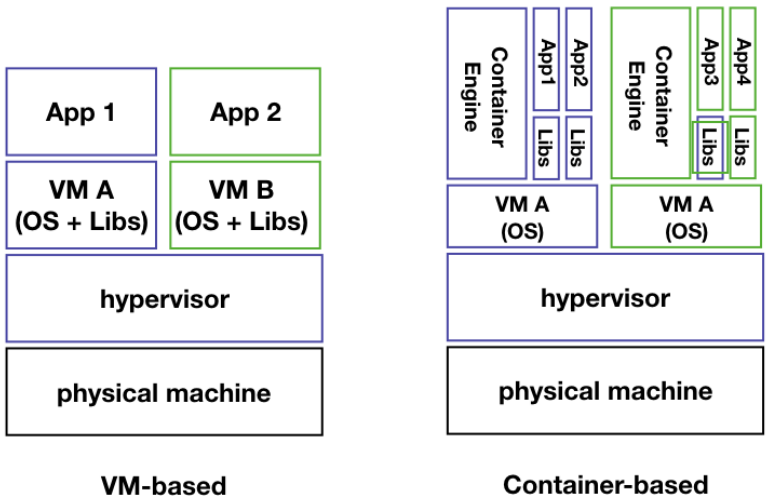

Figure 2.3: A comparison between VM-based and Container-based virtualization [37]

Because of the dynamic nature of workloads, the underlying resources must also be dynamic adjusted to meet the requirement.

2.1.2 Virtualization

Virtualization [48] is the fundamental technology that enables Cloud computing. It partitions a physical machine's resources (e.g. CPU, memory and disk) into several independent units called virtual machines (VMs) or containers. This technology rooted back in the 1960s' and was originally invented to enable isolated software testing, because each virtualized unit can provide good isolation which means multiple applications can run in separated VMs within the same PM without interfering each other [46].

Soon, people realized that it can be a way to improve the utilization of hardware resources: With each application deployed in a VM, a PM can run multiple applications. Later, multiple ways of dynamic migration (e.g pre-copy [11] and post-copy [23]) of VM were invented, which compresses and transfers a VM from one PM to another. This technique allows resource management in real time which inspires the strategy of server consolidation.

There are two classes of virtualization (see Figure 2.3): Hypervisor-based or VM-based and container-based virtualization.

1. VM-based virtualization: A virtualized system includes a new layer of software, the hypervisor or the vrirtual machine monitor (VMM). The VMM's role is to arbitrate accesses to the PM's resources so that guests' OS can share them [49].

   is the mainstream in the previous decades and producing many popular hypervisors: Xen [4], KVM [27], and VMware ESX [53].

2.

The recent development of container: application containers such as Docker, Linux Container, and Kubernetes [9] have attract the attention from both academia and industry. It provides a finer granularity of resource management by enabling an application level of operations including deployment, scaling, and migration.

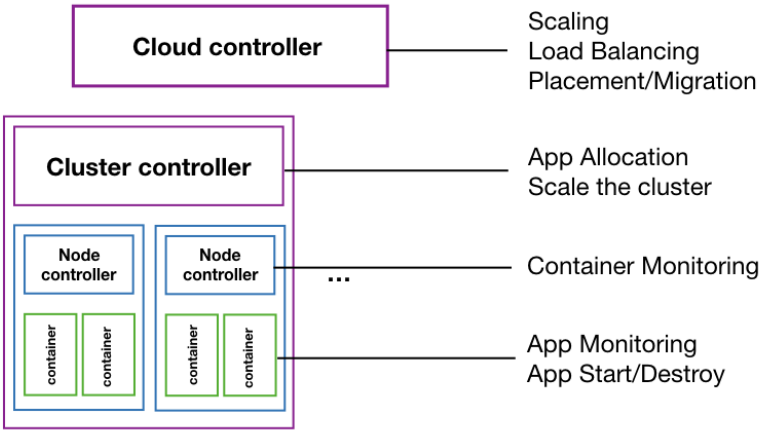

Figure 2.4: A resource management architecture of container-based Cloud and the functionalities of each layer.

Virtual machine and VM-based resource management

Container and container-based resource management

Container-based virtualization is also often addressed as operating-system-level virtualization. It includes two types of container: OS container and application container [**?**]. OS container (as shown in Figure **??**) can run in both PM and VM. Each container provides an isolated environment. There are mainly three implementations of OS-level of containers: OpenVZ, Google's control groups, and namespace [39]. Google and Facebook have been using OS container for years and being beneficial for its lightweight and fast communication among applications.

In contrast of OS containers, an application container, such as Docker and Rocket, runs a single process. It allows to separate an applications into many components. With application container, it is easy to achieve auto-scaling on a single process. In comparison between OS and application container, the former can be seen as a VM runs multiple applications where each application can has its separated environment (e.g. libraries), while an application container act like a single unit in OS container and it can be allocated in both OS container and VM.

He et al [21] propose an hierarchical resource management architecture for application container (see Figure 2.4). The major functionality of Cloud controller is to provide a global optimization.

Several research [14, 15, 21, 55] have compared VM-based and container-based virtualization. The advantages of container-based virtualization are mainly from three aspects: 1. Low overhead, containers have a lightweight management. which generate much less overhead than a hypervisor. On the other hand, shared OS system also reduces the overhead on running multiple OSs. 2. Containers have a near-native performance of CPU, memory, disk and network. While VM has a poor I/O performance [41]: up to 50% reduction of bandwidth (e.g hard disk and network). This defect also has a negative effect on the migration performance, since a VM-image is ranging from hundreds of MB to several GBs. Another advantage of containers is that, it naturally support vertical scaling while VM does not. Vertical scaling means a container can dynamically adjust its resources under the host's resource contraint. This feature offers a fine granularity management of resources. On the other hand, the disadvantages of containers are categorized in two aspects. Currently, con-

tainers have rather poor performance in isolation for memory, disk, and network. Security is immature in containers [9], therefore, high security required applications are not recommended to be deployed in such systems.

2.1.3 Server consolidation

Server consolidation is an approach to reduce the total number of PM or PM locations. It is often applied to solve the problem of physical server sprawl [26]: a situation that more PMs are used in a low-utilized way.

Physical server sprawl not only wastes valuable computing resources but leads to a waste of energy, as Hameed et al [19] addressed, even at a low level of 10% of CPU utilization, PMs still consume more than 50% of its peak time. In addition, Han et al [20] state that average resource utilization of PMs is usually between 10% to 50% of its capacity. Server consolidation is often used to solve the disproportionate between utilization and energy consumption [5].

From a broad technologies perspective, there are generally two technologies can be used to achieve server consolidation: virtualization and clustering. Clustering is used in a situation that the applications running in PMs are I/O intensive. This is because current virtualization technologies such as KVM [27] or Xen [4] have a 20% to 55% of reduction of I/O bandwidth (e.g disk reads and writes, network bandwidth) in comparison with non-virtualized PM [41]. Virtualization is more suitable for applications which require little CPU utilization (e.g 15%) and low I/O needs. Web services are mostly categorized into this group. Three major benefits of virtualization make it be the first choice for web-based application consolidation: No reliance on hardware, easy to provision and live migration.

As 2.1.2 mentioned, there are two types of virtualization: VM-based and container-based. Current consolidation are mostly VM-based since there is few data center has adopt the container-based technique.

Bibliography

[1] AL-ROOMI, M., AND AL-EBRAHIM, S. Cloud computing pricing models: a survey. *Computing* (2013).

[2] ANGELO, J. S., KREMPSER, E., AND BARBOSA, H. J. C. Differential evolution for bilevel programming. In *2013 IEEE Congress on Evolutionary Computation (CEC)* (2013), IEEE, pp. 470–477.

[3] BANZHAF, W., NORDIN, P., KELLER, R. E., AND FRANCONE, F. D. Genetic programming: an introduction, 1998.

[4] BARHAM, P., DRAGOVIC, B., FRASER, K., HAND, S., HARRIS, T. L., HO, A., NEUGEBAUER, R., PRATT, I., AND WARFIELD, A. Xen and the art of virtualization. *SOSP* (2003), 164.

[5] BARROSO, L. A., AND HÖLZLE, U. The Case for Energy-Proportional Computing. *IEEE Computer 40*, 12 (2007), 33–37.

[6] BELOGLAZOV, A., ABAWAJY, J. H., AND BUYYA, R. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Comp. Syst. 28*, 5 (2012), 755–768.

[7] 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.

[8] 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.

[9] BERNSTEIN, D. Containers and Cloud - From LXC to Docker to Kubernetes. *IEEE Cloud Computing* (2014).

[10] BURKE, E. K., HYDE, M. R., AND KENDALL, G. Evolving Bin Packing Heuristics with Genetic Programming. *PPSN 4193*, Chapter 87 (2006), 860–869.

[11] CLARK, C., FRASER, K., HAND, S., HANSEN, J. G., JUL, E., LIMPACH, C., PRATT, I., AND WARFIELD, A. Live migration of virtual machines. 273–286.

[12] COLSON, B., MARCOTTE, P., AND SAVARD, G. An overview of bilevel optimization. *Annals OR 153*, 1 (2007), 235–256.

[13] DAYARATHNA, M., WEN, Y., AND FAN, R. Data Center Energy Consumption Modeling - A Survey. *... Surveys & Tutorials* (2016).

[14] DUA, R., RAJA, A. R., AND KAKADIA, D. Virtualization vs Containerization to Support PaaS. In *2014 IEEE International Conference on Cloud Engineering (IC2E)* (2014), IEEE, pp. 610–614.

[15] FELTER, W., FERREIRA, A., RAJAMONY, R., AND RUBIO, J. An updated performance comparison of virtual machines and Linux containers. *ISPASS* (2015), 171–172.

[16] FERDAUS, M. H., MURSHED, M. M., CALHEIROS, R. N., AND BUYYA, R. Virtual Machine Consolidation in Cloud Data Centers Using ACO Metaheuristic. *Euro-Par 8632*, Chapter 26 (2014), 306–317.

[17] FORSMAN, M., GLAD, A., LUNDBERG, L., AND ILIE, D. Algorithms for automated live migration of virtual machines. *Journal of Systems and Software 101* (2015), 110–126.

[18] GAO, Y., GUAN, H., QI, Z., HOU, Y., AND LIU, L. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *J. Comput. Syst. Sci. 79*, 8 (2013), 1230–1242.

[19] HAMEED, A., KHOSHKBARFOROUSHHA, A., RANJAN, R., JAYARAMAN, P. P., KOLODZIEJ, J., BALAJI, P., ZEADALLY, S., MALLUHI, Q. M., TZIRITAS, N., VISHNU, A., KHAN, S. U., AND ZOMAYA, A. Y. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing 98*, 7 (2016), 751–774.

[20] HAN, G., QUE, W., JIA, G., AND ZHANG, W. Resource-utilization-aware energy efficient server consolidation algorithm for green computing in IIOT. *Journal of Network and Computer Applications* (July 2017).

[21] 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.

[22] HINDMAN, B., KONWINSKI, A., ZAHARIA, M., GHODSI, A., JOSEPH, A. D., KATZ, R. H., SHENKER, S., AND STOICA, I. Mesos - A Platform for Fine-Grained Resource Sharing in the Data Center. *NSDI* (2011).

[23] HINES, M. R., DESHPANDE, U., AND GOPALAN, K. Post-copy live migration of virtual machines. *SIGOPS Oper. Syst. Rev. 43*, 3 (July 2009), 14–26.

[24] JENNINGS, B., AND STADLER, R. Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management 23*, 3 (2015), 567–619.

[25] JUNG, G., HILTUNEN, M. A., JOSHI, K. R., SCHLICHTING, R. D., AND PU, C. Mistral - Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures. *ICDCS* (2010), 62–73.

[26] KHANNA, G., BEATY, K. A., KAR, G., AND KOCHUT, A. Application Performance Management in Virtualized Server Environments. *NOMS* (2006).

[27] KIVITY, A., KAMAY, Y., LAOR, D., AND LUBLIN, U. kvm: the Linux virtual machine monitor. *... of the Linux ...* (2007).

[28] 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.

[29] LI, X., TIAN, P., AND MIN, X. A Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems. In *Service-Oriented and Cloud Computing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 1169–1178.

[30] MAGALHÃES, D., CALHEIROS, R. N., BUYYA, R., AND GOMES, D. G. Workload modeling for resource usage analysis and simulation in cloud computing. *Computers & Electrical Engineering 47* (2015), 69–81.

[31] MANN, Z. Á. Approximability of virtual machine allocation: much harder than bin packing.

[32] 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.

[33] 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.

[34] MISHRA, M., DAS, A., KULKARNI, P., AND SAHOO, A. Dynamic resource management using virtual machine migrations. *IEEE Communications ... 50*, 9 (2012), 34–40.

[35] MOENS, H., FAMAEY, J., LATRÉ, S., DHOEDT, B., AND DE TURCK, F. Design and evaluation of a hierarchical application placement algorithm in large scale clouds. *Integrated Network Management* (2011), 137–144.

[36] NGUYEN, S., ZHANG, M., JOHNSTON, M., AND TAN, K. C. Automatic Design of Scheduling Policies for Dynamic Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming. *IEEE Transactions on Evolutionary Computation 18*, 2 (2014), 193–208.

[37] 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.

[38] 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).

[39] ROSEN, R. Resource management: Linux kernel namespaces and cgroups. *Haifux* (2013).

[40] SARIN, S. C., VARADARAJAN, A., AND WANG, L. A survey of dispatching rules for operational control in wafer fabrication. *Production Planning and ... 22*, 1 (Jan. 2011), 4–24.

[41] SHAFER, J. I/O virtualization bottlenecks in cloud computing today. In *Proceedings of the 2nd conference on I/O virtualization* (2010).

[42] SHEN, S., VAN BEEK, V., AND IOSUP, A. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. *CCGRID* (2015), 465–474.

[43] SHI, W., AND HONG, B. Towards Profitable Virtual Machine Placement in the Data Center. *UCC* (2011), 138–145.

[44] SINHA, A., MALO, P., AND DEB, K. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. *IEEE Transactions on Evolutionary Computation* (2017), 1–1.

[45] 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.

[46] SOMANI, G., AND CHAUDHARY, S. Application Performance Isolation in Virtualization. *IEEE CLOUD* (2009), 41–48.

[47] SOTELO-FIGUEROA, M. A., SOBERANES, H. J. P., CARPIO, J. M., HUACUJA, H. J. F., REYES, L. C., AND SORIA-ALCARAZ, J. A. Evolving Bin Packing Heuristic Using Micro-Differential Evolution with Indirect Representation. *Recent Advances on Hybrid Intelligent Systems 451*, Chapter 28 (2013), 349–359.

[48] UHLIG, R., NEIGER, G., RODGERS, D., SANTONI, A. L., MARTINS, F. C. M., ANDERSON, A. V., BENNETT, S. M., KÄGI, A., LEUNG, F. H., AND SMITH, L. Intel Virtualization Technology. *IEEE Computer 38*, 5 (2005), 48–56.

[49] UHLIG, R., NEIGER, G., RODGERS, D., SANTONI, A. L., MARTINS, F. C. M., ANDERSON, A. V., BENNETT, S. M., KAGI, A., LEUNG, F. H., AND SMITH, L. Intel virtualization technology. *Computer 38*, 5 (May 2005), 48–56.

[50] VARASTEH, A., AND GOUDARZI, M. Server Consolidation Techniques in Virtualized Data Centers: A Survey. *IEEE Systems Journal* (2015), 1–12.

[51] VERMA, A., AND DASGUPTA, G. Server Workload Analysis for Power Minimization using Consolidation. *USENIX Annual Technical Conference* (2009), 28–28.

[52] VICENTE, L., SAVARD, G., AND JÚDICE, J. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications 81*, 2 (May 1994), 379–399.

[53] WALDSPURGER, C. A. Memory resource management in VMware ESX server. *ACM SIGOPS Operating Systems Review 36*, SI (Dec. 2002), 181–194.

[54] WANG, G. G., AND SHAN, S. Review of Metamodeling Techniques in Support of Engineering Design Optimization. *Journal of Mechanical Design 129*, 4 (Apr. 2007), 370–380.

[55] XAVIER, M. G., NEVES, M. V., ROSSI, F. D., FERRETO, T. C., LANGE, T., AND DE ROSE, C. A. F. Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments. In *2013 21st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP 2013)* (2013), IEEE, pp. 233–240.

[56] XU, J., AND FORTES, J. A. B. Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. In *Int'l Conference on Cyber, Physical and Social Computing (CPSCom)* (2010), IEEE, pp. 179–188.

[57] YAZIR, Y. O., MATTHEWS, C., FARAHBOD, R., NEVILLE, S. W., GUITOUNI, A., GANTI, S., AND COADY, Y. Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis. *IEEE CLOUD* (2010), 91–98.

[58] ZHANG, J., HUANG, H., AND WANG, X. Resource provision algorithms in cloud computing - A survey. *J. Network and Computer Applications 64* (2016), 23–42.

[59] ZHANG, Q., CHENG, L., AND BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications 1*, 1 (2010), 7–18.

[60] ZHU, X., YU, Q., AND WANG, X. A Hybrid Differential Evolution Algorithm for Solving Nonlinear Bilevel Programming with Linear Constraints. In *2006 5th IEEE International Conference on Cognitive Informatics* (2006), IEEE, pp. 126–131.