Chapter 1

Introduction

1.1 Problem Statement

Cloud computing is a computing model offering a network of servers to their clients in a on-demand fashion. From NIST's definition [15], *"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 a smaller unit of resources called *Virtual Machines (VMs)*. A web-based application provider can access and deploy their applications (e.g Endnote, Google Drive and etc.) in these VMs 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: Cloud providers, Cloud users (applications providers), and End (application) users [10]. Cloud providers build data centers, provide maintenance and resource management on hardware infrastructures. Cloud users develop and deploy applications on Cloud infrastructures. End users consumes applications developed by Cloud users and hosted by Cloud providers. Our focus is on the Cloud providers' perspective to increase their profit.

Cloud providers have two ways to improve the profit, 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). The expense mainly comes from two sources: upfront investment and energy consumption. Upfront investment is the expense on infrastructure including purchasing of hardwares and etc. In a long run, energy consumption become an even critical concern.

Energy consumption [12] is derived from several parts including cooling system (account for 40% of total consumption), PM consumption (40%) and others (20%). Cooling system and servers or PMs account for a majority of the consumption. A recent survey [6] shows that the recent development of cooling techniques have reduced its energy consumption and now server consumption has become the dominate energy consumption component.

The energy consumption of PM is far from efficient [8] because the energy consumption remains high even when their actual utilization are low. Utilization refers to the fraction of real-time resource consumption (e.g 1 core CPU consumed by applications) to the overall resource (e.g. 8 cores CPU) in a PM. A concept of *energy proportional computing* [2] is raised to address the disproportionate between utilization and energy consumption. There are two

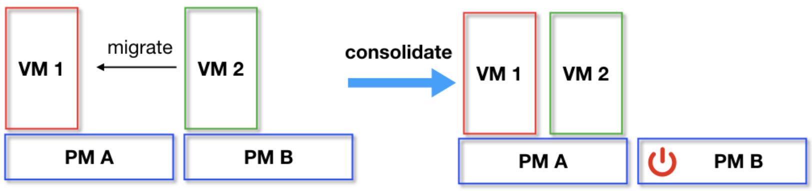

Figure 1.1: A Server Consolidation example: Initially, each PM runs an application wrapped with a VM in a low resource utilization state. After the consolidation, both VMs are running on PM A, so that PM B is turned off to save energy [2].

ways to solve the problem, Dynamic voltage and frequency scaling (DVFS) which reduces the energy consumption by dynamically adjusting the voltage of hardwares. And a *Server consolidation* [30] strategy. It resolves the low utilization problem by migrating applications (e.g. deployed in a VM) into a fewer number of PMs (see Figure 1.1), so that the resource utilization of PMs are maintained at a high level. In the meanwhile, idle PMs can be turned off to save energy. We will focus on server consolidation instead of DVFS.

Server consolidation is achieved by manipulating the allocation of applications in the PMs so that the resource utilization of PMs achieve a high level. An effective resource allocation is affected by multiple factors such as virtualization technologies and distinct approaches in different scenarios or stages.

Virtualization [26] partitions a physical machine's resources (e.g. CPU, memory and disk) into several isolated units such as virtual machines (VMs) or containers. Traditional resource allocation is based on VMs, where Cloud providers offer a number fixed types of VM; Cloud users can select VMs by themselves - in Infrastructure as a Service (IaaS) or automatically select by a software platform provided by Cloud provider - in Platform as a Service (PaaS).

However, there are mainly three inherent disadvantages of VM-based resource allocation which cause low utilization in data centers.

- Resource over-provisioning.
  As previous mentioned, VMs are either selected by Cloud users or automatically selected by software platforms. In either case, it requires an estimation of required resources. However, The accurate estimation is almost impossible because of unpredictable workloads; A simple way is to reserve more resources for ensuring the QoS at peak hours [5], rather than completely rely on auto-scaling, simply because auto-scaling is more expensive than reservation. However, the peak hours only account for a short period, therefore, in most of time, resources are wasted. In IaaS, the types of VM are a part of the contract, Cloud providers cannot simply change the type of VMs after provisioning. In PaaS, Cloud providers can change the type of VMs to more suitable sizes.

- Unbalanced usage of resources.
  Specific applications consume unbalanced resources which leads to vast amount of resource wastage [25]. For example, computation intensive tasks consume much more CPU than RAM; a fixed type of VM provides much more RAM than it needs. Because the tasks use too much CPU, they prevent other tasks from co-allocating. This also causes wastage.

- Heavy overhead of VM hypervisors and redundant operating systems (OSs) cause resource wastage.

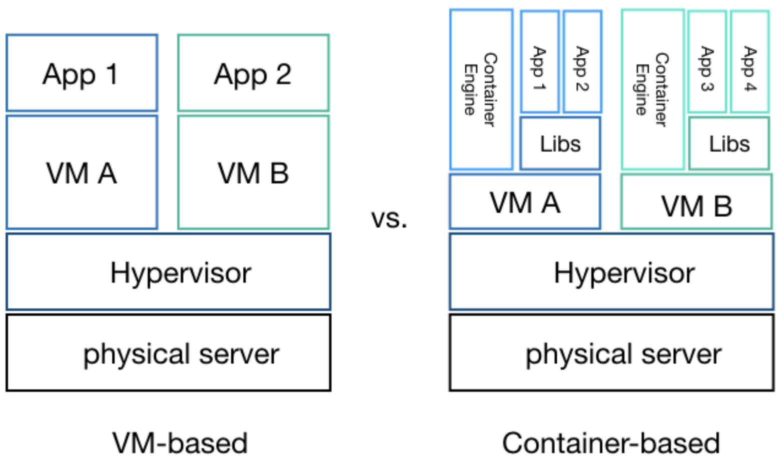

Figure 1.2: A comparison between VM-based and Container-based virtualization

> In VM-based resource allocation, heavy overhead is caused by the hypervisor of VMs and the separated operating systems running in the PM. A hypervisor manages and monitors the VMs running on a PM. The overhead of a hypervisor is heavier with the increasing numbers of VM. Redundant operating system is another reason for overhead, as normal applications do not need specific operating systems; Commonly used OSs - such as Linux-based: RedHat, or Windows server versions - are well enough for their needs. Therefore, running applications in separated operating system simultaneously is unnecessary.

The recent development of container technology [23] is an operating system level of virtualization. Each container wraps an application and multiple containers can be installed in a same operating system within a VM (see Figure 1.2 right-hand side). Container technology has attracted the attention of both industrial and academia because it has the potential to solve these problems. Firstly, it solves the over-provisioning problem by allocating multiple containers in the same VM. Secondly, the unbalance resource problem can be solved by combining different types of applications (e.g CPU-intensive and Memory-intensive). Thirdly, because containers share a same VM, the overhead of hypervisor and OSs are naturally solved with less deployment of VMs.

Apart from virtualization, different server consolidation strategies should be applied in different scenarios. This is because Cloud data center has a highly dynamic nature where it constantly receives new requests for resource allocation and releases of old resources, consequently, the energy efficiency is constantly dropping.

Server consolidation strategies can be applied in multiple resource re-allocation scenarios as shown in Figure 1.3 including new application allocation, periodic optimization, overloading and under-loading adjustment. Periodic optimization can be seen as static server consolidation, because these operation normally take a number of applications and re-allocate them into a number of PMs. This optimization is quite time-consuming, therefore, it is conducted in an off-line fashion. Overloading is a scenario that the CPU utilization of a PM reaches a predefined threshold: e.g. 80%, then it is likely that the applications running inside will reach the bottleneck of CPU. At this occasion, it assumes that migrates one of the applications to another PM can solve the problem. Under-loading is the scenario when the CPU utilization of a PM is under a predefined threshold: e.g 50%, then all applications running inside should be migrated to other PMs so that this PM can be turned off.

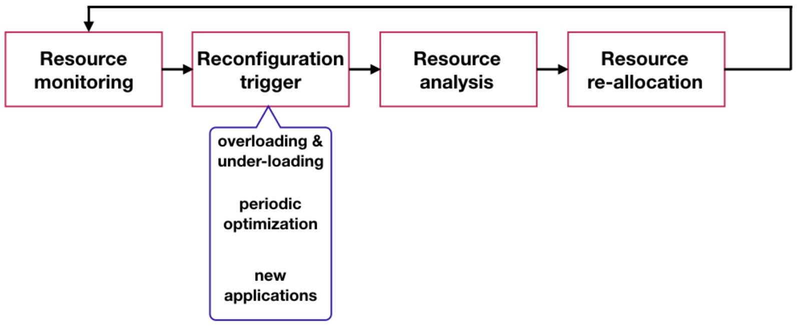

Figure 1.3: A workflow of resource management [16]

These two cases re-allocate one application each time, it requires a fast on-line operation. Therefore, they can be considered as a dynamic consolidation. New application allocation can be seen as either static: allocate a batch of new applications, or dynamic: allocate a new application each time. These three consolidation scenarios can be seen distinct optimization tasks which have a common goal: minimizing the energy consumption.

In this thesis, our goal is to help Cloud providers to increase their profits by saving money from the energy consumed by servers. Specifically, we achieve this goal by providing better resource allocation techniques using the container-based server consolidation to optimize the resource utilization in different scenarios.

The problem can be described as, a data center has a number of PMs, a number of types of VM and a number of containers to be allocated. They can be represented as resources such as CPU cores and RAM. The task is to allocate these containers to a number of VMs, each of VM has a type. Furthermore, allocating these VMs to a minimum number of PMs. The decision variable is the location of containers and VMs as well as the types of VMs. The constraints are the aggregated resources of containers inside a VM cannot exceed the VM's capacity. Similarly, the aggregated resources of VMs inside a PM cannot exceed the PM's capacity. The main difficulty of the problem is a large number of combination container allocation and the choices of VMs which is too time-consuming for a brutal-force approach.

Specifically, the new challenge posed by Container-based server consolidation is even harder than VM-based consolidation. Traditional VM-based server consolidation are modeled as bin-packing problems [13]. 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. In container-based server consolidation, there are two levels of allocation: Containers are allocated to VM as the first level and VMs are allocated to PMs as the second level. These two levels of allocation interact with each other, for the first level of allocation affects the decision in the second level. Therefore, it is necessary to propose new approaches for the problem.

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 [20]. 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 [19] 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 [13] 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 [14]. Therefore, these two allocations should be considered simultaneously.

1.2 Motivation

In this thesis, we aims 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. The initialization,
   At this stage, a set of containers is allocated to a set of empty VMs and these VMs are allocated to a set of PMs. Initalization problem is also refered as placement problem [10] which is fundamental for server consolidation problem. This problem is inherently more difficult than previous VM-based consolidation problem, since container-based consolidation naturely has a two-level structure. Only a few research focus on this problem, Piraghaj [19] designs a dynamic allocation system. She proposes a two-step procedure. Since, these two-level structure interact each other, separate solution certainly leads to a local optima. Therefore, in this thesis, we will solve the problem simutaneously.

   In this objective, we will establish the fundamental concepts in studying this joint allocation of containers and VMs including new problem models: price and power model, new problem constraints, and optimization objectives. The major challenges for this objective is to design representations and several EC approaches to solve this problem. More specifically, in designing the EC approach, new search mechanisms, operators will be designed and new representations will be proposed to fit the problem.

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 operations degrade 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. 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. It is also referred as a dynamic placement problem.

   To solve a dynamic placement, heuristics and dispatching rules are often used [3,7,21, 22]. In this scenario, a dispatching rule is considered as a function that determines the priorities of PMs that a container can be placed. However, dynamic placement is much complex than bin-packing problem [13]. Because of its dynamic nature, human designed heuristics are ill-equipped in approximating solutions when the environment has changed [24]. Multi-objective genetic algorithm (GA) [27] has been applied. However, GA is too slow for dynamic problem.

   We intend to develop a hyper-heuristic method - Genetic Programming (GP) technique [1] or artificial immune system [9]- to learn from the best previous allocation and automatic evolves dispatching rules to solve this problem. GP has been applied in generating dispatching rules for bin-packing problem [4, 24] and other scheduling problems [18]. 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 they are usually conducted in an off-line fashion. 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 [11,17] and agent-based management systems [28]. 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

In this thesis, we aims at providing a series of approaches to continuously optimize the a joint allocation of VMs and containers that considers three types of consolidation: Initialization, global consolidation, Dynamic consolidation. In addition, the static allocation normally involves with large amount of variables which is particular difficult to optimize. We are also going to propose a method to solve this problem. These approaches combine element of AI planning, to ensure the objectives and constraint fulfillment, and of Evolutionary Computation, to evolve a population of near-optimal solutions. The research aims at determining a flexible way in creation of solutions to solve server consolidation problems. As discussed in the previous section, the research goal can be achieved in the following objectives and sub-objectives.

1. The initialization Problem,
   Currently, most research focus on VM-based server consolidation technique. They often modeled this problem as a vector bin-packing problem [29]. 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 [20] solve this problem in separated steps without optimization. Therefore, this is the first research that trying to solve the problem.

   (a) *Modeling*
   Our first sub objective is to propose a description of model for the initialization problem. In order to achieve this goal, we will first review the related models including VM-based placement models and bi-level optimization models. Furthermore, we are going to consider the differences and design the constraints and other characteristics.

   (b) *Representation*
   Based on this new model, we are going to develop a representation that is suitable for this problem. We will first review a number representation of bi-level optimization problem and design a few representations. Furthermore, experiments will conduct to test the effectiveness of these repreesentations.

   (c) *New operators and searching mechanisms*
   In order to utilize Evolutionary Computation (EC) to solve this problem, we are going to develop searching mechanisms according to the nature of problem as well as the selected representation. In order to achieve this goal, we will design several new operators. In order to evaluate the quality of these components, we will perform analytical analysis on the result.

2. Global consolidation problem,
   As previous section mentioned, global consolidation is a time-dependent problem. 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.

   The objective of global consolidation has three sub-objectives. In order to measure the degree of robustness, we need to design a robustness measure. The second sub-objective is to design static consolidation algorithm with considering its previous immediate result. The third objective extends the second objective to a more general case, considering both previous immediate and next allocation. The evaluation of algorithm is based on analytical analysis of fitness functions and robustness measure.

   (a) *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 in a static placement process. It may cause more migration in the next consolidation. Therefore, it needs a time-aware measure of the robustness of system. A data center should be both consolidated as well as robustness after consolidate. Therefore, in this objective, the first sub-problem we are going to solve is to propose a robustness measure.
   (b) *Design an consolidation method consider previous consolidation result*
       Based on the robustness measure, we will first design an allocation method which takes the previous allocation into account. It has two optimization objectives, maximize the robustness and also minimize the energy consolidation.
   (c) *Design a time-aware consolidation method*
       We will generalize the previous sub-objective to a more general one: design a time-aware allocation method which takes previous and next allocation into consider.

3. 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 [13] 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. Therefore, in this objective, we will use GP to automatically generate heuristics or dispatching rules.

   (a) Construct Functional Set and Primitive Set for the problem
       As the basic component of a dispatching rule, primitive set contains the states of environment including: status of VMs, features of workloads. The functional set contains the operators which combines low level features. The effectiveness of functional set and primitive set is tested by applying the constructed dispatching rules on dynamic consolidation problem.
   (b) 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.

4. Large-scale Static Consolidation Problem

   (a) 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 propose a method that categorizes servers so that only a small number of servers 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.

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

2.1.1 An Overview of Cloud Computing

Cloud computing is a paradigm that allows Cloud users and End users to acess Cloud services and applications based on their requirements regardless of where the services are. The advent of Cloud computing meets the trend of fast growing applications where these applications are deploying in a third-party data centers and application users can use them without installing on their local computers.

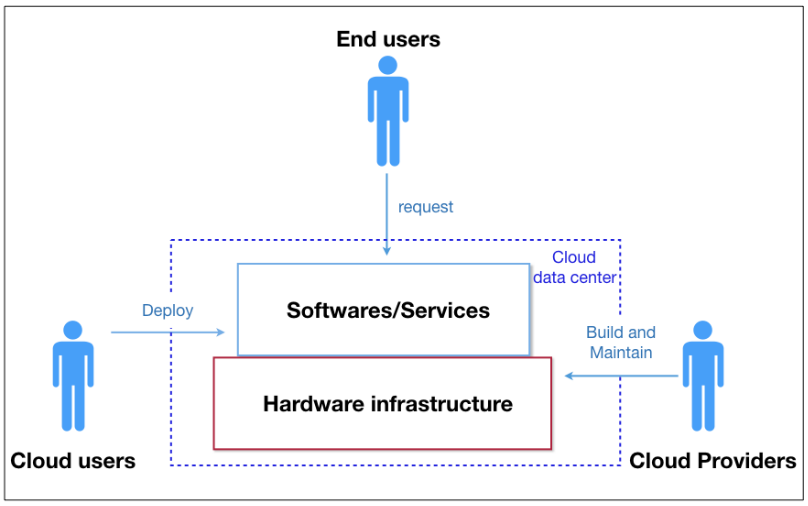

Figure 2.1: Stakeholders of Cloud computing

Cloud computing involves three stakeholders [10] (see Figure 2.1): Cloud provider, Cloud user and End User. Each of them has their responsibility, goal, and objectives.

- *Cloud providers* build data centers, provide maintenance and resource management on the hardware infrastructure. Their 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.

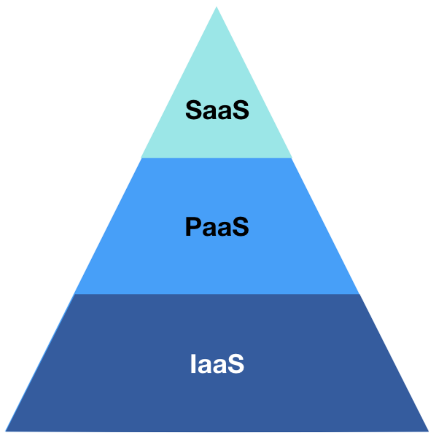

Figure 2.2: Pyramid of service models in Cloud computing. IaaS provides the fundamental resources such as CPU cores, RAM. The resources are usually wrapped with various types of virtual machine. PaaS provides a software platform sitting on top of IaaS for Cloud users to develop applications. SaaS is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

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* develop and deploy applications on Cloud. Each application generates time-vary CPU utilization. Their goal is also increase the profit mainly through two objectives, attracting more End users and reduce the expense of resources. The first objective can be 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* are the final customers in this chain. They consume services directly from Cloud users and indirectly from Cloud provider. Their 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.

In this thesis, we focus on a core issue of helping Cloud providers to increase their profits. This goal can be done by two ways, attracting more Cloud users to use Cloud services and reducing the expense.

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. There are three traditional service models [15]: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The relationship among three service models can be pictured as a pyramid (see 2.2). Their detailed responsibilities among stakeholders are explained as below:

- IaaS, a Cloud provider hosts hardwares such as PMs and cooling infrastructure on behalf of Cloud users. A Cloud provider also provides maintenance, management of hardware resources.

  In IaaS, Cloud provider offers the fundamental resources such as CPU cores, RAMs, and network bandwidth. These 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.

From the resource management perspective, the constraint in allocation of VMs is that the aggregated VMs' resources in a PM cannot exceed the capacity of the PM. After these VMs have been allocated, their types cannot be changed. During the life cycle of an application, Cloud providers can dynamically adjust the locations of VMs, provision new VMs (same type) for the replicas of an application, as well as turning on/off PMs.

After receiving the requests, Cloud providers choose a set of PMs and allocate the required VMs in them. Then, Cloud users can access the VMs through remote consoles. Cloud providers manage the VMs by monitoring the resource utilization of VMs and PMs. If a VM has run out of resources, an auto-scaling strategy is automatically applied to duplicate the users' VM. Finally, Cloud providers optimize the locations of VMs by consolidating them into a minimum number of PMs.

- PaaS, a Cloud provider builds a software platform on top of hardwares provided by IaaS. This platform allows a complete life-cycle of software development including development, testing and deployment.

  From resource management perspective, as shown in Figure 2.2, 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.

  A typical procedure of a Cloud user deploying their applications in an PaaS cloud includes several steps. In the first step, Cloud users need to provide the initial estimation of the quantity of resources instead of types of VM. Then, Cloud providers determine the types of VM for applications according to the estimated resources. After this step, resource management system conducts the provisioning and allocating as the same steps in IaaS. During the life cycle of applications, similar to IaaS, Cloud providers can also adjust the location of VMs, add new VMs, and control the status of PMs. Different from IaaS, Cloud providers can change the type of VM for an application according to it needs.

- SaaS describes the relationship between Cloud users and End users. End users create workloads for applications. Although this service model does not directly related to the resource management, it provides the fundamental reasons for resource management and optimization. Because of the dynamic nature of workloads, the underlying resources must also be dynamic adjusted to meet the requirement.

Our proposed optimization approaches are based on a new service model: Container as a Service (CaaS) proposed by Piraghaj and et al. [19] which is a variant of PaaS. The reasons for us to establish our approaches on CaaS are listed as followed:

- CaaS uses a new virtualization technology called containers as its fundamental resource allocation unit. CaaS, like PaaS, can be built based on IaaS without changing the underlying structure, but it allows a finer resource allocation by adjusting containers. Therefore, CaaS is a promising trend.
- CaaS has a fine granularity level of resource management which has shown opportunities to improve the resource utilization, however, it often proposes new optimization challenges which have not been studied yet.

2.2 VM-based Static Consolidation Methods

So far in the industry, most Cloud data center is based on virtual machine technology. Therefore, VM-based resource management is the mainstream in both industry and academia.

As previous mentioned, server consolidation is one of the technique to reduce the power consumption. Various techniques have been proposed in this field, these techniques can be roughly grouped into static and dynamic approaches.

Static approaches adjust the allocation of VMs in a periodical fashion (e.g. weekly or monthly).

Dynamic appproaches are conducted by a runtime placement manager to migrate VMs automatically in response to workload variations.

2.3 Dyanmic Consolidation Methods

2.4 Evolutionary Computation Approaches on Server Consolidation Problem

Bibliography

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

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

[3] 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.

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

[5] CHAISIRI, S., LEE, B.-S., AND NIYATO, D. Optimization of Resource Provisioning Cost in Cloud Computing. *IEEE Trans. Services Computing 5*, 2 (2012), 164–177.

[6] CHO, J., AND KIM, Y. Improving energy efficiency of dedicated cooling system and its contribution towards meeting an energy-optimized data center. *Applied Energy 165* (Mar. 2016), 967–982.

[7] 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.

[8] 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.

[9] HOFMEYR, S. A., AND FORREST, S. Architecture for an Artificial Immune System. *Evolutionary Computation 8*, 4 (2000), 443–473.

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

[11] 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.

[12] KAPLAN, J. M., FORREST, W., AND KINDLER, N. Revolutionizing data center energy efficiency, 2008.

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

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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).

[21] 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.

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

[23] 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.

[24] 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.

[25] TOMÁS, L., AND TORDSSON, J. Improving cloud infrastructure utilization through overbooking. *CAC* (2013), 1.

[26] 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.

[27] XU, J., AND FORTES, J. A. B. Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. *GreenCom/CPSCom* (2010).

[28] 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.

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

[30] 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.