# 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 [11], *"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)* or *Containers*. 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 providers offer the fundamental resources: data centers and computation powers that 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.

In this thesis, our goal is to help Cloud providers to reduce expense from the energy consumption by servers in data centers because energy is the major source of expense [8]. Specifically, we achieve this goal by providing better resource allocation techniques based on container technology and server consolidation to optimize the resource utilization in three resource allocation scenarios.

The problem can be described as, given a number of Physical Machines (PMs) which can be represented as the resources such as a number of CPU cores and memory they contained; a number of different fixed configurations of VMs, each configuration can also be represented as aforementioned resources; and a number of containers which represent the resource requirements of applications. The task is to allocate these containers to a number of different VMs. Furthermore, allocating these VMs to a number of PMs. The objective is to minimize the number of PMs used to accommodate these containers. The decision variable is the location of containers and VMs as well as the selection of the configurations 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 task is generally considered as a server consolidation problem.

Server consolidation problems have different assumptions, objectives and requirements in multiple resource re-allocation scenarios as shown in Figure 1.1 including new application allocation, periodic optimization, overloading and under-loading adjustment. Periodic

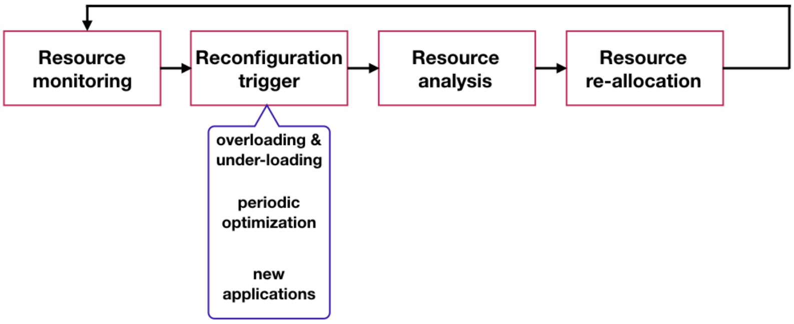


Figure 1.1: A workflow of resource management [12]

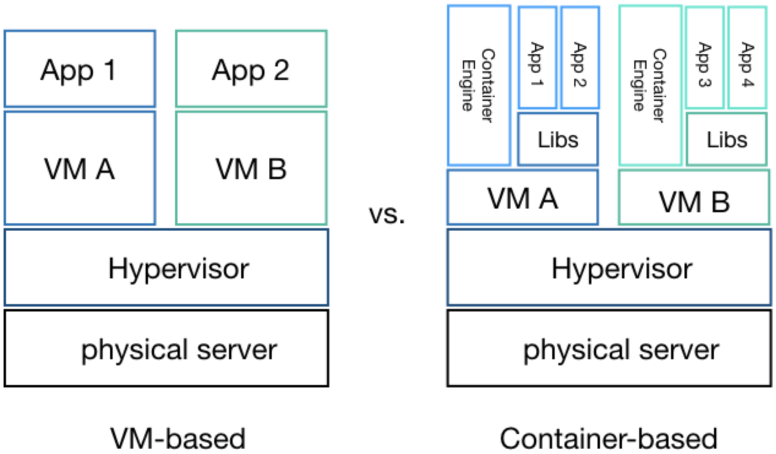


Figure 1.2: A comparison between VM-based and Container-based virtualization

optimization can be seen as static server consolidation, because these optimization normally involves large amount of variables, therefore, it is quite time-consuming. It takes a number of existing applications and re-allocate them into a number of PMs. This problem is conducted in an off-line fashion. Overloading and under-loading can be categories to dynamic consolidation problem, since they consider allocating one application each time and require a fast on-line operation. 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 thesis, we consider it as a static problem. These three consolidation problems can be seen distinct optimization tasks which have a common goal: minimizing the energy consumption of a data center.

Apart from these different resource allocation scenarios, we would also like to consider the server consolidation problem based on a new technology. Container technology [19] 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). The reason for choosing container as the fundamental resource management unit is because Container-based resource allocation can overcome 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 re-

sources. 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 [4], 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 [21]. 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)
  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.

Container-based allocation solves the over-provisioning problem by allocating multiple containers in the same VM. In addition, the unbalance resource problem can be solved by combining different types of applications (e.g CPU-intensive and Memory-intensive). Furthermore, because containers share a same VM, the overhead of hypervisor and OSs are naturally solved with less deployment of VMs.

However, 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 [9]. 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 [16]. 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 [15] 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 [9] 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 [10]. Therefore, these two allocations should be considered simultaneously.

## 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 container and VM (new application 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 empty VMs and these VMs are allocated to a set of PMs. 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 [15] 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. Robust 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 [2,5,17, 18]. 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 [9]. Because of its dynamic nature, human designed heuristics are ill-equipped in approximating solutions when the environment has changed [20]. Multi-objective genetic algorithm (GA) [22] 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 [6]- 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 [3, 20] and other scheduling problems [14]. 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 [7, 13] and agent-based management systems [23]. 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 [24]. 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 [16] 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 [9] 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.

# Bibliography

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

[2] 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.

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

[4] 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.

[5] 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.

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

[7] 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.

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

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] 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).

[17] 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.

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

[19] 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.

[20] 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.

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

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

[23] 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.

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