Chapter 1

Introduction

1.1 Problem Statement

Cloud computing is a computing model offers a network of servers to their clients in a on-demand fashion. From NIST's definition [22], *"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."*

Cloud computing has completely reformed the software industry [6] by providing three major benefits to web-based software or web service providers. First, service providers do not need upfront investment in hardwares (e.g servers and networking devices) and pay for hardwares' maintenance. Second, service providers will not worried about the limited resources will obstruct the performance of their services when unexpected high demand occurs. The elastic nature of cloud can dynamic allocate and release resources for a service. In addition, software providers can pay as much as the resource usage under a *pay-as-you-go* policy. Third, service providers can publish and update their applications at any location as long as there is an Internet connection. These advantages allow anyone or organization to deploy their softwares on Cloud in a reasonable price.

From Cloud providers' perspective, they are trying to make the most profit on data centers. On one hand, cloud providers are trying to improve the quality of Cloud service to attract more service providers. On the other hand, they want to cut enormous energy consumption - as much as 25,000 households [18] - to lower the expense.

Energy consumption in data centers are derived from several parts as illustrated in Figure 1.2. Regardless the energy consumption of refrigeration system (or cooling system), the majority are from servers. According to Hameed et al [12], servers are far from energy-efficient. The main reason for the wastage is that the energy consumption of servers remains high even when the utilization are low. Therefore, a concept of *energy proportional computing* [3] raised to address the disproportionate between utilization and energy consumption. This leads to using vitualization technology to achieve server consolidation.

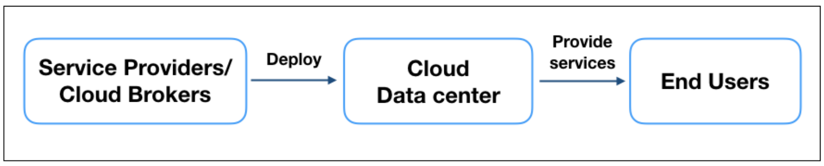

Figure 1.1: Stakeholders of Cloud computing

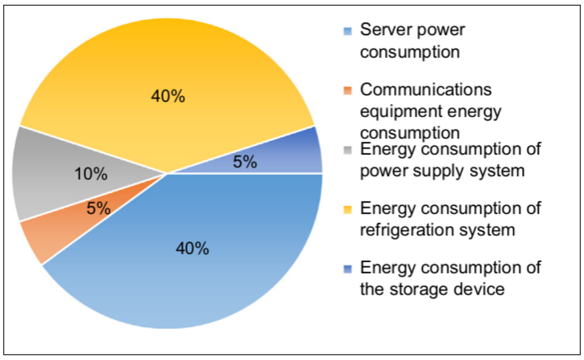

Figure 1.2: Energy consumption distribution of data centers [27]

Virtualization [33] partitions a physical machine's resources (e.g. CPU, memory and disk) into several isolated units called virtual machines (VMs) where each VM allows an operating system running on them. This technology rooted back in the 1960s' and was originally invented to enable isolated software testing. Soon, people realized it can be a way to improve the utilization of hardware resources. Thereafter, a resource management strategy of server consolidation was invented.

Server consolidation [39] resolves the low utilization problem by gathering virtual machines (VMs) into a fewer number of physical machines (PMs), so that the resource utilization of PMs are maintained at a high level. In the meanwhile, idle servers are turned off to save energy.

Despite the usefulness of server consolidation, it is a difficult task. Server consolidation is often considered as a global optimization problem where its goal is to minimize the overall energy consumption. It is often modeled as a bin-packing problem [20] which is a well-known NP-hard problem meaning it is unlikely to find an optimal solution of a large problem. Previous research have studied the problem extensively. Deterministic methods such as Integer Linear Programming [32] and Mixed Integer Programming [34] are unsuitable for a large scale problem because of the long computation time. More research proposed heuristic methods to approximate the optimal solution such as First Fit Decreasing (FFD) [24], Best Fit Decreasing (BFD) [36]. Manually designed heuristics are designed to tackle the special requirements such as a bin-item incomplete scenario [11] and Multi-tier Applications [17, 19]. Although these greedy-based heuristics can quickly solve the consolidation problem, as Mann's research [20] shown, server consolidation is a lot more harder than bin-packing problem because of multi-dimension, many constraints. Therefore, a simple greedy-based heuristic (e.g FFD) leads to a bad performance and human designed heuristics cannot handle it well when Cloud environment has changed.

In addition, virtualization technology has evolved to allow finer granularity resource management. A recent development of Container technique [30] has drove the attention of both industrial and academia. Container is an operating system level of virtualization which means multiple containers can be installed in a same operating system (see Figure 1.3). Each container provides an isolated environment for an application. In short, a VM is partitioned into smaller manageable units. This new concept starts a new service model called Container as a Service (CaaS) [26]. From Cloud customers' perspective, CaaS has advantages of both IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) but without their disadvantages. On one hand - similar to PaaS - it frees the customers' from low level resource management so that they can focus on application development. On the other hand - similar to IaaS - it allows customers to customize their software environment

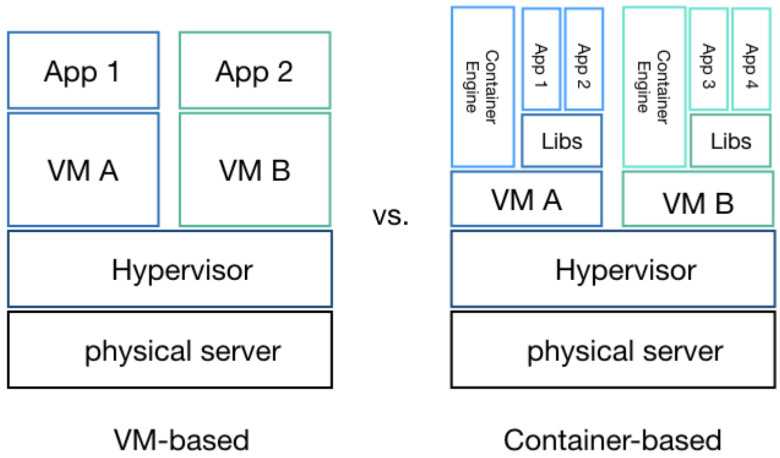

Figure 1.3: A comparison between VM-based and Container-based virtualization

without being constrained by platforms. From Cloud providers' perspective, CaaS allows Cloud providers to manage both the deployment of applications and resource allocation. Hence, Cloud providers have a complete control of resources which may lead to a better utilization of resources. However the management difficulty also increases. Currently, vast amount of research focus on VM-based server consolidation, which can not be directly used on Contained-based Cloud. This thesis, therefore, aims at providing an end-to-end solution for Container-based server consolidation which includes three stages: initialization, static container-based server consolidation and dynamic container placement.

1.2 Motivation

The motivation for this thesis mainly includes two parts, in the first part, we illustrate the roots of container-based server consolidation problem. In the second parts, we explain the motivations for the objectives.

1.2.1 Motivation For Container-based Server Consolidation Problem

This container technology certainly brings many advantages to current Cloud industry[9]. However, it also brings difficulties for server consolidation. Server consolidation problems are typically modeled as vector bin-packing problem which are NP-hard. Container-based server consolidation adds another level of abstraction which makes it a two-level vector bin-packing problem. Current server consolidation methods are mostly VM-based which can not be directly applied on this problem, because two-level of bin-packing problems interact with each other. Piraghaj [25] proposes a two-step procedure; it first maps tasks to VMs and then allocate containers to VMs. As Mann illustrated in [21], these two steps should be conducted simultaneously, otherwise it leads to local optimal. Other research [8, 13, 1] propose greedy-based heuristics on container allocation problem. They can be easily stuck at local optimal. Therefore, it motivates us to provide *global optimized* resource allocation solution for container-based data centers.

1.2.2 Motivation For Research Objectives

The container-based consolidation problem, similar to VM-based consolidation, can be seen as a continuous optimization procedure with three stages: initialization, off-line static joint allocation of container and VM, and on-line dynamic consolidation. Different stages have distinctive goals, therefore, they are separated research questions. In this thesis, we aims at providing an end-to-end solution to all three problems. In addition to these three research questions, a scalability problem of static optimization is considered as an optional objective.

1. The initialization stage is first step in continuous server consolidation. At this stage, a set of applications or containers is allocated to empty VMs and these VMs are allocated to physical servers. This seemingly two-step procedure is interconnected, therefore, should be conducted simultaneously. Previous research [15] focus on VM-based optimization and they cannot solve the two-level optimization problem.

   This project will establish the fundamental concepts in studying the joint allocation of containers and VMs including new problem models: price and power model, new problem constraints, and optimization objectives. This study will also explore different representations that suitable for this problem.

2. Server consolidation can be considered as static or dynamic problem [35]. Cloud data center has a highly dynamic nature with arrival and release of VMs. Therefore, after the initial allocation, the energy efficiency keeps dropping. At a certain time point, e.g. a fixed time interval, a *static server consolidation* is conducted to improve the global energy efficiency. Similar with VM-based consolidation, the problem is considered as a multi-objective problem with minimization of migration cost as well as keeping a good energy efficiency. Distinct from previous studies, the problem model has become a two-level of bin-packing problem which both container and VM migration cost should be considered.

3. Dynamic consolidation is another method to maintain a high energy efficiency. Unlike static consolidation is conducted periodically in a global scale, dynamic consolidation is applied on single VM or container at any time point. At large, data center system monitors all the states of servers for overloaded and underloaded servers. Once an overloaded server is detected, one of the VM or container running inside the server will be migrated to other machine so that the applications do not suffer from a performance degradation; for an underloaded server, all its applications will be moved to other servers so that it can be turned off. In conclusion, the main goal for dynamic consolidation is to optimize the global energy consumption as well as prevent overloading. In a container-based environment, it involves three steps .

   - *When to migrate?* refers to determine the time point that a physical server is overloaded.
   - *Which container to migrate?* refers to determine which container need to be migrated so that it optimize the global energy consumption.
   - *Where to migrate?* refers to determine which VM and host that a container is migrated to.

   Specifically, we focus on the third question: dynamic placement problem. Previous research employ simple heuristics [29, 10, 4], they are fast but could not perform well. Multi-objective genetic algorithm (GA) [36] has been applied. However, GA is too slow for dynamic problem.

To solve a dynamic placement with large number of variables, heuristics and dispatching rules are often used[28]. In this scenario, a dispatching rule is considered as a function that determines the priorities of servers that a container can be placed. However, dynamic placement is much complex than bin-packing problem [20]. Therefore, we intend to develop a hyper-heuristic method - Genetic Programming (GP) technique [2] or artificial immune system [14]- to automatic evolve dispatching rules to solve this problem. GP has been applied in generating dispatching rules for bin-packing problem [5, 31] and it produces promising solutions.

4. Cloud data center typically has hundreds of thousands servers and more. Large scale of server consolidation has always been a challenge. Many approaches have been proposed in the literature to resolve the problem. There are mainly two ways, both rely on distributed methods, hierarchical-based [16, 23] and agent-based management systems [37]. 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 static methods where all the states of machines are collected and analyzed. In fact, it is infeasible and unnecessary to check all the states of machines since the search space is too large and most machines 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

The overall goal of this thesis is to propose an end-to-end server consolidation approach that considers all three stages: Initialization, Off-line Static Joint Allocation of Container and VM, On-line Dynamic Container Placement Problem. In addition, the static allocation normally involves with large amount of variables which is particular difficult to optimize. We 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 to determine a flexible way in which planning and EC can be combined to allow the 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 [38]. 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.

   (a) *Modeling*
   Previous VM-based models do not consider this structure, therefore, our first sub problem 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 model and bilevel optimization model. 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 discover a representation that suitable for this problem.

(c) *New operators and searching mechanisms*
In order to utilize Evolutionary Computation (EC) to solve this problem, we are going to design searching mechanisms according to the nature of problem. In order to achieve this goal, we will design several new operators.

2. Off-line Static Joint Allocation of Container and VM Problem,
A static allocation can be seen as a resource scheduling problem. A schedule is robust if it is able to absorb some degree of uncertainty in tasks duration while maintaining a stable solution [7]. Cloud resource management is a continuous process, after each static allocation, the system should be able to maintain a stable status with the least adjustment. The development of static allocation approach has three sub-objectives.

   (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-series 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 allocation method consider previous allocation*
   Based on the robustness measure, we will first design an allocation method which takes previous allocation into account. It has two objectives, maximize the robustness and also minimize the energy consolidation.

   (c) *Design a time-series-aware allocation method*
   Last but not the least, we will generalize the previous sub-objective to a more general one: design a time-series-aware allocation method which takes several allocation into consider.

3. On-line Dynamic Container Placement Problem with a GP approach,

   (a) Representation
   In order to utilize a hyper-heuristic method such as GP to solve the problem, the first step is to design a representation.

   (b) 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.

   (c) Develop GP-based methods for evolving Dispatching rules

   (d) One heuristic can solve all the problem or different heuristics to solve specific problems.

4. Large-scale Static Consolidation Problem

   (a) Propose a preprocessing method to eliminate variables

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] ANSELMI, J., AMALDI, E., AND CREMONESI, P. Service Consolidation with End-to-End Response Time Constraints. *EUROMICRO-SEAA* (2008), 345–352.

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

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

[4] 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.

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

[6] BUYYA, R., YEO, C. S., VENUGOPAL, S., BROBERG, J., AND BRANDIC, I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems 25*, 6 (June 2009), 599–616.

[7] CANON, L.-C., AND JEANNOT, E. Evaluation and Optimization of the Robustness of DAG Schedules in Heterogeneous Environments. *IEEE Transactions on Parallel and Distributed Systems 21*, 4 (2010), 532–546.

[8] DONG, Z., ZHUANG, W., AND ROJAS-CESSA, R. Energy-aware scheduling schemes for cloud data centers on Google trace data. *OnlineGreenComm* (2014), 1–6.

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

[10] 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.

[11] GUPTA, R., BOSE, S. K., SUNDARRAJAN, S., CHEBIYAM, M., AND CHAKRABARTI, A. A Two Stage Heuristic Algorithm for Solving the Server Consolidation Problem with Item-Item and Bin-Item Incompatibility Constraints. *IEEE SCC* (2008).

[12] 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.

[13] 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).

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

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

[16] 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.

[17] JUNG, G., JOSHI, K. R., HILTUNEN, M. A., SCHLICHTING, R. D., AND PU, C. Generating Adaptation Policies for Multi-tier Applications in Consolidated Server Environments. *ICAC* (2008).

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

[19] LI, B., AND JIANXIN, L. *An Energy-saving Application Live Placement Approach for Cloud Computing Environment*. ACM International Conference on Cloud Computing, 2009.

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

[21] 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.

[22] 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.

[23] 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.

[24] PANIGRAHY, R., TALWAR, K., UYEDA, L., AND WIEDER, U. Heuristics for vector bin packing. *research microsoft com* (2011).

[25] 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.

[26] 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).

[27] RONG, H., ZHANG, H., XIAO, S., LI, C., AND HU, C. Optimizing energy consumption for data centers. *Renewable and Sustainable Energy Reviews 58* (May 2016), 674–691.

[28] 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.

[29] Shi, W., and Hong, B. Towards Profitable Virtual Machine Placement in the Data Center. *UCC* (2011), 138–145.

[30] 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.

[31] 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.

[32] Speitkamp, B., and Bichler, M. A mathematical programming approach for server consolidation problems in virtualized data centers. *IEEE Transactions on services ...* (2010).

[33] 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.

[34] Wang, Y., and Xia, Y. Energy Optimal VM Placement in the Cloud. In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD* (2016), IEEE, pp. 84–91.

[35] Xiao, Z., Jiang, J., Zhu, Y., Ming, Z., Zhong, S.-h., and Cai, S.-B. A solution of dynamic VMs placement problem for energy consumption optimization based on evolutionary game theory. *Journal of Systems and Software 101* (2015), 260–272.

[36] Xu, J., and Fortes, J. A. B. Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. *GreenCom/CPSCom* (2010).

[37] 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.

[38] Zhang, J., Huang, H., and Wang, X. Resource provision algorithms in cloud computing - A survey. *J. Network and Computer Applications 64* (2016), 23–42.

[39] 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.