AN ANALYSIS AND COMPARISON OF CLOUD DATA CENTER
ENERGY-efficient RESOURCE management TECHNOLOGY

[1] School of Computer Science, Fudan University, Shanghai, China,
lzh@fudan.edu.cn, {zhihui.lu,soichi.takashige,ks,yumiko.sugita,yf,tomohiro.morimura,ke,yutaka.kudo.sn}
@hitachi.com

Extension 6011 at SCC Industry 2013

Abstract
Nowadays, cloud data centers began to support more and more popular online services such as web search, e-commerce, social networking, video on demand and software as a service (SaaS), so the massive scale of data centers brings the challenges of energy efficiency. Therefore, the concept of an energy-efficient Green data center has been proposed. To build an energy-efficient cloud data center, cloud data center resource dynamic provisioning and consolidation technology is involved. In this paper, we first provide a survey of current industry and academic efforts on cloud data center energy-efficient management technology, focusing on the cloud data center resource dynamic provisioning technology and resource consolidation technology. We first focus on an analysis and comparison of cloud resource predictive dynamic provisioning technology. We analyze and discuss the main resource prediction methods and models, including basic models, feedback based models and multiple time series models. We describe the relationship between these categories as well as the characteristics of the models. After that, we analyze and compare Cloud resource reactive dynamic provisioning technology. And then we analyze cloud resource consolidation technology. Furthermore, we also give a prospect on cloud resource management technology standardization trends. Lastly, we analyze the prospective research direction.

Keywords: [Cloud Computing, IDC, Cloud Data Center, Virtual Machine-VM, Resource Consolidation, Energy-efficient management technology, Resource Prediction, Resource Management, VM Migration]

1. INTRODUCTION
With the arrival of the twenty-first century, Web technologies have matured increasingly with the introduction of multiple applications such as e-commerce, real-time communication, video on demand, and social networking and so on. These entities require high-speed networks to connect and guarantee the performance of the running applications. To meet the resource demands of these applications, plenty of Internet Data Centers (IDCs) are built around the world. Benefiting from Internet economies and recent developments in Web technologies, data centers have emerged as a backbone of today’s IT infrastructure to provision resources to multiple applications for satisfying their availability and performance requirements. The rise and rapid development of cloud computing virtualization technologies are bringing the traditional IDC a revolutionary change to Cloud data center. Server, storage and network virtualization are at the base of the flexibility, affordability and scalability of any cloud-based data center offering.

Meanwhile, with the increasing scale of Internet applications every year, the size of data center scales up in proportion to the service expansion, and data centers consume a great deal of power energy every year. The more hosts within a data center, the more power energy is consumed. It was reported that 2012 saw an increased focus on the data center industry from the media, public and government bodies, all concerned about its use of energy (DCD Intelligence 2014). There is a 19% increase in the amount of electricity consumed globally by data centers between 2011 and 2012 (DCD Intelligence 2014). The results from the 2013 annual DCD Intelligence (DCDi) Industry Census show that the rise in power for 2013 was over 7% (DCD Intelligence 2014). The facts tell us that such a dramatic energy consumption increase in cloud data center requires a scalable and dependable IT infrastructure comprising of servers, storage, network bandwidth, electrical Grid and cooling system.

In modern data centers, there are hundreds of thousands of hosts used to meet application demands. However, the data center resources are usually in low utilization. Based on the report of Data Center Efficiency Trends for 2014 from “Energy Manager Today” (Aaron R. 2013), in current data center, server utilization rates are typically very low, currently averaging in the 6–12 percent range. A completely idle server still draws 60 percent of its maximum power. While large data centers enjoy economies of scale by amortizing long-term capital investments over large number of machines, they also incur tremendous energy costs in terms of power distribution and cooling (Qi Zh., et al., 2013). In particular, it has been reported that energy-related costs account for approximately 12% of overall data center expenditures. For large companies like Google, a 3% reduction in energy cost can translate to over a million dollars in cost savings (Qi Zh., et al., 2013). On the other

http://www.hipore.com/ijsc
hand, governmental agencies continue to implement standards and regulations to promote energy-efficient computing (Green Computing). As a result, reducing energy consumption has become a primary concern for today’s data center operators (Qi Zh., et al., 2013). The goal of the energy-efficient resource management is to dynamically adjust and minimize the number of active machines in a data center in order to reduce energy consumption while meeting the Service Level Agreements (SLA) of applications.

In the context of energy-efficient resource management, energy non-proportionality of IT resource, resource over-provisioning for near-peak performance should be considered by the resource management solutions. One of the simplest yet most effective approaches for reducing energy costs is to dynamically adjust the data center capacity by turning on needed machines and turning off unused machines, or to set them to a sleep state. This is supported by the evidence that an idle machine can consume as much as 50-60% of the power when the machine is fully utilized (Qi Zh. 2013). Unsurprisingly, a number of efforts are trying to leverage this fact to save energy by minimizing the number of active machines using a combination of server consolidation and dynamic capacity provisioning (Qi Zh. 2013):

Server consolidation aims at finding an assignment of workload to machines in order to minimize the number of used machines (Qi Zh. 2013). Server consolidation and storage consolidation with virtualization can help reduce the number of data center running servers, storage, desktop, and network devices to reduce complexity and make IT management simpler. Typically, server consolidation is achieved through (1) resource-aware workload scheduling and (2) dynamically adjusting workload placement using migration (Qi Zh. 2013). The former approach relies on the scheduler to find a good initial placement of workloads, whereas the second approach realizes that the initial workload placement can become sub-optimal over time, and improves workload placement using techniques such as VM migration.

On the other hand, dynamic capacity provisioning aims at dynamically controlling the number of active machines by switching machines on and off (or in and out of "sleep" state), based on various factors such as the workload arrival rate, workload performance requirement and the electricity price (Qi Zh. 2013). While over-provisioning the data center capacity can lead to sub-optimal energy savings, under-provisioning the data center capacity can cause significant performance penalty in terms of scheduling delay, which is the time a resource request has to wait before it is scheduled in the data center. In data centers, some application workload exhibits periodical nature. Sometimes workload is high and the resource is at high utilization, while other time the load is low and part of the resource is idle. It is typical that the load is orderly during some period. This introduces the technology of resource prediction. Many resource usage prediction algorithms have been exploited to predict some future parameters, such as system workload, the number of physical machines needed and so on. At the same time, we need to design reactive techniques, as the name suggests, referring to the methods that can quickly react to the abrupt fluctuation of the current system and applications status.

Energy saving technologies can be implemented through data center hardware and physical infrastructure, or upper resource management software, as shown in Figure 1. Some energy-aware hardware technology generally uses dynamic voltage and frequency scaling (DVFS), and many cooling infrastructure, such as temperature-aware air conditioner, has already been deployed. As for management software, the main concern is with the energy-efficient resource management. Cloud resource consolidation and dynamic resource provisioning focuses on energy-efficient cloud resource management field, where the main idea is to improve the utilization efficiency of the cloud system resource while meeting the Service Level Agreements (SLA) of applications. Baliga J., et al. (Baliga J., et al., 2011) present Green Cloud computing: an analysis of energy consumption in cloud computing. The analysis considers both public and private clouds, and includes energy consumption in switching and transmission as well as data processing and data storage. Buyya et al. (Buyya et al., 2010) proposed the GreenCloud project aimed at the development of energy-efficient provisioning of Cloud resources, while meeting QoS requirements defined by the SLAs established through a negotiation between Cloud providers and consumers. Beloglazov A., et al. (Beloglazov A., et al., 2011) give a good survey on the causes and problems of high power / energy consumption, and presents a taxonomy of energy-efficient design of computing systems covering the hardware, operating system, virtualization, and data center levels.

Figure 1. Energy Saving Technology Architecture in Cloud Data Center

Consolidating many VM servers in a small number can contribute to reducing the cost of energy in a substantial way (Medina, V., et al., 2014). Consolidating cloud data center resources through virtualization strategy not only increases utilization without incurring additional risk but it will ultimately help avoid costly, unnecessary new resource construction of data centers. The consolidation mechanism is an example of power saving management through policies that handle the idle, standby, and sleep states. Consolidation
can be used to minimize the number of currently active resources so that standby and sleep states can be used. Underused servers can be consolidated into fewer hosts and offloaded hosts can be placed into a sleep mode (Medina, V., et al., 2014). Therefore, it realizes energy saving during resource consolidation process, and an energy saving object can often be achieved through resource consolidation in turn.

Dynamic resource provisioning has a lot of potential for reducing power consumption in data centers. Most dynamic resource provisioning approaches can be categorized into two types: predictive (or proactive) and reactive (or control-theoretic). However, there are many challenges that hinder the successful deployment of dynamic resource provisioning. A common assumption in resource provisioning is that workload demand can be predicted (Gandhi A., et al., 2013). Therefore, proactive or prediction-based resource provisioning is required so as to deal with the periodic resource usage pattern. Unfortunately, this is not always the case. Even for long-term (daily) workloads with periodic trends, short-term fluctuations cannot be predicted based on periodicity (Gandhi A., et al., 2013). This observation suggests that purely predictive approaches might be insufficient for handling data center workloads. Ignoring this unpredictable portion can lead to a steep increase in response times. Further, another challenge in dealing with unpredictable workload demand is the possibility of load spikes (Gandhi A., et al., 2013). While load spikes, or abrupt changes in load, are not necessarily a daily occurrence in data centers, several instances of load spikes have been documented for web workloads. Important events, such as the September 11 attacks, earthquakes or other natural disasters, slashdot effects, Black Friday shopping, or sporting events, such as the Super Bowl or the Soccer World Cup, are common causes of load spikes for website traffic. Service outages or server failures can also result in abrupt changes in load caused by a sharp drop in capacity. Most of the above events cannot be predicted in advance. Therefore, some researchers propose reactive resource provisioning schemes.

The main contribution of this paper focuses on two aspects: 1. Cloud resource predictive and reactive provisioning method analysis and comparison. In this paper, we analyze and discuss the main cloud resource predictive and reactive methods. 2. Virtualized resource consolidation technology analysis. We analyze and compare the cloud virtualized resource consolidation schemes. Furthermore we analyze the characteristics and factors which impact the cloud resource consolidation.

The remainder of this paper is organized as follows: In Sec. 2, we make a detailed survey on current research on key technologies for building cloud virtualized data center and realizing cloud resource energy-efficient management. In Sec. 3, we introduce and describe the main cloud resource prediction models, and have an analysis and comparison on these models in Sec. 4. We analyze and compare cloud resource reactive provisioning schemes in Sec 5. Next, we analyze and compare cloud resource consolidation schemes in Sec 6. After that, we give a prospect for cloud resource management technology standardization trends in Sec 7. Finally, we conclude the paper and propose our future work in Sec 8.

2. SURVEY ON CLOUD VIRTUALIZED DATA CENTER RESOURCE ENERGY-EFFICIENT MANAGEMENT TECHNOLOGY

2.1 SURVEY ON CLOUD DATA CENTER RESOURCE DYNAMIC PROVISIONING TECHNOLOGY

Cloud computing has emerged as one of the most promising business models for the future. For a growing number of businesses, the journey to cloud computing starts with a private cloud implementation, and this mainly focuses on IaaS (Infrastructure as a Service). In recent years, a great number of IT enterprises have been committed to offering products related to this area.

Currently, there are four mainstream production cloud virtualization platforms from different companies: VMware®’s vSphere®, Microsoft®’s Hyper-V®, open source Xen® sponsored by Citrix®, and kernel-based KVM® supported by Red Hat®. The comparison of these four platforms is described in Table 1. Xen supports CPU DVFS with P-states and C-states (CPU sleep state) (Beloglazov A., et al., 2011). Xen also supports offline and live migration of VMs. KVM supports the S4 (hibernate) and S3 (sleep/stand by) power states. Similar to Xen, VMware supports host-level power management via DVFS. And VMware VMotion® enables live migration of VMs between physical nodes (Beloglazov A., et al., 2011). Currently, VMware’s vCenter® and Citrix’s XenServer® have begun to provide the power management function, and VMware Distributed Resource Scheduler (DRS) contains a subsystem called VMware Distributed Power Management (DPM) to reduce power consumption by a pool of servers by dynamically switching off spare servers, but they don’t support high level energy-efficient policy. The free Microsoft Assessment and Planning (MAP) ToolKit can consolidate servers and implement some power management features based on Hyper-V. The four most popular open source cloud management platforms are OpenStack® (OpenStack, 2014), CloudStack® (Apache CloudStack, 2014), OpenNebula® (OpenNebula, 2014) and Eucalyptus (Eucalyptus, 2014). These open source cloud management platforms have provided basic resource management function, but need to extend to support realizing energy-efficient management policy and function module. As a representative, OpenStack Neat (OpenStack Neat, 2014) is an extension to OpenStack implementing dynamic consolidation of Virtual Machines (VMs) using live migration. The major objective of dynamic VM consolidation is to improve the utilization of physical resources and reduce energy consumption by re-allocating VMs using live migration according to their real-time
resource demand and switching idle hosts to the sleep mode. The aim of the OpenStack Neat project is to provide an extensible framework for dynamic consolidation of VMs based on the OpenStack platform.

**Table 1. Comparison of Main IaaS Management Products**

<table>
<thead>
<tr>
<th>Company</th>
<th>Virtualization Product</th>
<th>Management Product</th>
<th>Main Management Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware (VMware, 2014)</td>
<td>vSphere (ESX/ESXi)</td>
<td>vCenter family</td>
<td>Integrated management Workload balancing Resource optimization Priority settings/affinity rules Power management etc.</td>
</tr>
<tr>
<td>Microsoft (Microsoft, 2014)</td>
<td>Hyper-V</td>
<td>System Center family</td>
<td>Comprehensive configuration Multi-hypervisor support Resource optimization etc.</td>
</tr>
<tr>
<td>Citrix (Citrix, 2014)</td>
<td>Xen</td>
<td>XenServer</td>
<td>Workload balancing Memory optimization Power management etc.</td>
</tr>
<tr>
<td>Red Hat (Red Hat, 2014)</td>
<td>KVM</td>
<td>CloudForms</td>
<td>Application lifecycle mgmt. Compute resource mgmt. etc.</td>
</tr>
</tbody>
</table>

**2.1.1 Survey on Cloud Data Center Resource Predictive Dynamic Provisioning Technology**

In recent years, resource prediction techniques have been used in the cloud data center environment as a necessary step to dynamic resource provisioning. As a supplement, it helps to manage resources more efficiently. Prediction techniques have been in existence long before and there are many advanced models. In cloud resource dynamic provisioning schemes, some prediction methods are totally based on these traditional models, but there are some novel prediction models transforming from traditional models and more suitable for cloud resources.

In terms of traditional prediction techniques, there are some mature models. Mikhail Dashevskiy et al. (Dashevskiy, M., et al., 2009) consider Fractional Autoregressive Integrated Moving Average (FARIMA) processes which provide a unified approach to characterizing both short-range and long-range dependence. Taylor J.W. et al. (Taylor, J.W., et al., 2007) study some short-term load prediction methods, including ARIMA modeling, periodic AR modeling, an extension for double seasonality of Holt-Winters exponential smoothing and a recently proposed alternative exponential smoothing formulation. Antti Sorjamaa et al. (Sorjamaa, A., et al., 2007) propose a global methodology for the long-term prediction of time series. This methodology combines direct prediction strategy and sophisticated input selection criteria to forecast future values. In (Taylor J.W., et al., 2006; Tan, J., et al., 2011), the authors present a prediction method based on PCA (principal component analysis).

With the development of cloud computing, prediction techniques are introduced in order to provide the virtualized resources more dynamically and efficiently. In recent years, a great number of prediction methods are used in the cloud. Besides these, we present a few well performed models used in a data center environment. At present, most models are built based on time series analysis. Depending on the time series, we divide these prediction models into three categories: the first category is basic models and the second is feedback based prediction models. Both of these two categories are built based on single time series. The third category is for multiple time series prediction models. The classification chart is shown in Fig. 2.

![Figure 2. Classification of Resource Prediction Model](http://www.hipore.com/ijsc)
predictor based on the Autoregressive Integrated Moving Average (ARIMA) model, to achieve adaptive resource allocation for cloud applications on each VM. Then they propose the prediction-based dynamic resource scheduling algorithms to dynamically consolidate the VMs with adaptive resource allocation to reduce the number of physical machines. Extensive experimental results show that our scheduling is able to realize automatic elastic resource allocation with acceptable effect on SLAs.

In some resource management schemes, feedback control is added to the prediction models to improve resource allocation. Wesam Dawoud et al (Dawoud, W., et al., 2011) present elastic VM architecture for cloud resources provisioning based on feedback control. In this architecture, each controller is designed to predict the next resource allocation based on the last allocation and consumption. Ziming Shen et al (Shen, Z., et al.,2011) present CloudScale, a system that automates fine-grained elastic resource scaling for multi-tenant cloud computing infrastructures. It employs online resource demand prediction and prediction error handling to achieve adaptive resource allocation. Jie Zhu et al. (Zhu, J., et al., 2011) present a dynamic allocation framework for Database-as-a-Service. In their prediction model, in order to alleviate underestimate errors, a small offset is added to the predicted value. Pradeep Padala et al. (Padala, P., et al., 2009) present AutoControl, a resource control system of multiple virtualized resources. In this system, there is a model estimator, which inputs past allocation, past performance and leverage autoregressive moving average (ARMA) model to achieve the future performance value. With the predicted performance value, optimizer module determines the resource allocation.

There is another kind of prediction method, which adopts multiple time series to build prediction models. Jian Tan et al. (Tan, J., et al.,2011), Liang, Jin et al. (Liang, J., et al.,2004) present multi-resource prediction models for resource sharing environments, like cloud, data center and so on. In their models, they not only consider the autocorrelation of a single resource, but also the cross correlation between different resources. Arjit Khan et al. (Khan, A., et al., 2012) present a multiple time series approach for workload characterization and prediction in the cloud. In their approach, it treats server workload data samples as multiple time series, and divides servers into different clusters. Then it characterizes the correlations in these clusters to predict for each cluster. Yexi Jiang et al. (Yexi, J., et al., 2011) present an online system to model and predict the cloud VM demand. They utilize two-level ensemble method to capture the characteristics of the high transient demand time series. The first level is a regression based ensemble that combines the results of different prediction models of the same VM type. The second level ensemble considers the relationship between different VM types, and utilizes their correlation to help improve the robustness of prediction.

In the on-demand cloud environment, web application providers have the potential to scale virtual resources up or down to achieve cost-effective outcomes. To address this challenge, Jiang jing et al. (Jiang, J., et al., 2013) propose a novel cloud resource auto-scaling scheme at the virtual machine (VM) level for web application providers. The scheme automatically predicts the number of web requests and discovers an optimal cloud resource demand with cost-latency trade-off. Based on this demand, the scheme makes a resource scaling decision that is up or down or NOP (no operation) in each time-unit re-allocation. Their experiment results demonstrate that the proposed scheme achieves resource autoscaling with an optimal cost-latency trade-off, as well as low SLA violations. On-demand resource provisioning is with great challenge in cloud systems. The key problem is how to know the future workload in advance to help determine resource allocation. There is various prediction models are developed to predict the future workloads and provide resources. The major problem of previous research is that they assume that application workload has static pattern. In the paper (Feifei Zh., et al., 2013), we present a Pattern Sensitive Resource Provisioning Scheme, named PSRPS. It can recognize application workload dynamic patterns and choose suitable prediction models for prediction online. Besides, when there is maladjustment in prediction models, PSRPS can switch prediction models or adjust the parameters of the model by itself to adaptively to guarantee prediction accuracy. This paper extends paper (Feifei Zh., et al., 2013)’s work.

Gandhi A., et al.(Gandhi A.,et al., 2014) propose a new cloud service, Dependable Compute Cloud (DC2), that automatically scales the infrastructure to meet the user-specified performance requirements. DC2 employs predictive Kalman filtering to automatically learn the (possibly changing) system parameters for each application, allowing it to proactively scale the infrastructure to meet performance guarantees. DC2 is designed for the cloud - it is application-agnostic and does not require any offline application profiling or benchmarking. Their implementation results on OpenStack using a multi-tier application under a range of workload traces demonstrate the robustness and superiority of DC2 over existing rule-based approaches.

**2.1.2 Survey on Cloud Data Center Resource Reactive Dynamic Provisioning Technology**

In general, predictive approaches are very successful when dealing with periodic or seasonal workloads. However, these approaches fail when the workload is bursty and unpredictable, or when the workload demand suddenly increases: it is clearly hard to predict what will happen in the future when demand is bursty and future arrivals are unknown(Gandhi A., et al., 2013). Therefore, some studies also propose cloud resource reactive dynamic provisioning technology. In this situation, we need to consider how to reduce the setup time of VM in a short time with a reactive way, especially for public cloud, and some researchers use
VM image pre-copying method to reduce VM launching time. At the same time, we need to avoid resource over-provisioning for workload peak time, resulting in resource waste and excessive energy consumption.

Wang et al. (Peijian W., et al., 2010) employ a reactive feedback mechanism to manage the power-performance tradeoff in multi-tier systems. The authors use DVFS along with capacity provisioning to react to degradation in observed response times. Wang et al. leverage queueing theoretic results to help guide their system. However, the queueing models employed by Wang et al. are much simpler, and do not take setup time into account.

Gandhi A., et al. (Gandhi A., et al., 2012) introduce a dynamic capacity management policy, AutoScale, that greatly reduces the number of servers needed in data centers driven by unpredictable, time-varying load, while meeting response time SLAs. AutoScale scales the data center capacity, adding or removing servers as needed. AutoScale has two key features: (i) it autonomically maintains just the right amount of spare capacity to handle bursts in the request rate; and (ii) it is robust not just to changes in the request rate of real-world traces, but also request size and server efficiency.

In the paper (Guo, T, et al., 2014), Tian Guo et al. describe Seagull, a system designed to facilitate cloud bursting by determining which applications should be transitioned into the cloud and automating the movement process at the proper time. Seagull optimizes the bursting of applications using an optimization algorithm as well as a more efficient but approximate greedy heuristic. Seagull also optimizes the overhead of deploying applications into the cloud using an intelligent pre-copying mechanism that proactively replicates virtualized applications, lowering the bursting time from hours to minutes.

As the popularity of cloud computing continues to rise, more and more applications are being deployed in public clouds. To conserve provisioning costs while achieving performance objectives, clients should automatically scale up and down applications deployed in the cloud to match changing workload demands. The cloud provider, on the other hand, should attempt to consolidate load onto highly utilized physical machines, in order to reduce wasted power consumption. Tighe, M et al. (Tighe, M. et al., 2014) propose a new algorithm combining both the automatic scaling of applications with dynamic consolidation of virtual machines, in order to meet the goals of both the cloud client and provider. This allows the cloud provider to make better use of their infrastructure, and reduce power consumption.

Reducing the energy footprint of warehouse-scale computer (WSC) systems is key to their affordability, yet difficult to achieve in practice. The lack of energy proportionality of typical WSC hardware and the fact that important workloads (such as search) require all servers to remain up regardless of traffic intensity renders existing power management techniques ineffective at reducing WSC energy use. Lo, D. et al. (Lo, D., et al., 2014) present PEGASUS, a feedback-based controller that significantly improves the energy proportionality of WSC systems, as demonstrated by a real implementation in a Google search cluster. PEGASUS uses request latency statistics to dynamically adjust server power management limits in a fine-grain manner, running each server just fast enough to meet global service-level latency objectives. In large cluster experiments, PEGASUS reduces power consumption by up to 20%.

Swama et al. (Mylavarapu, S., et al., 2010) came up with a better capacity planning algorithm that “could ensure that it plans for peak usage but do not provision for it”. They modeled the problem as a stochastic optimization problem with the objective of minimizing the number of servers while considering two important constraints a) stochastic nature of workloads and b) minimizing the application SLA violations. Padala et al. (Padala, P., et al. 2009) presented a resource control system that automatically adapted to dynamic workload changes to achieve application SLOs. It was a combination of an online model estimator and a novel multi-input, multi-output (MIMO) resource controller, in which the former one captured the complex relationship between application performance and resource allocations, while the latter one allocated the right amount of multiple virtualized resources to achieve application SLOs.

2.2 SURVEY ON CLOUD DATA CENTER RESOURCE CONSOLIDATION TECHNOLOGY

Whether predictive provisioning approaches or reactive provisioning approaches, we need deploy VM resources on physical server to contain applications. Consolidating multiple applications on a single physical server can solve issues related to low utilization, however, how to efficiently and accurately perform server consolidation at enterprise datacenter level is still an unsolved research problem that faces significant technical challenges (Lei L., 2014) including how to accurately measure and characterize an application’s resource requirements, how optimally to distribute the virtual machines hosting the applications over the physical resources, and how to avoid performance interference among the virtual machines collocating in the same physical machines (Lei L., 2014).

As a first step toward enabling energy efficient consolidation, Srikantaiah S. et al. (Srikantaiah, S., et al., 2008) study the inter-relationships between energy consumption, resource utilization, and performance of consolidated workloads. The study reveals the energy performance trade-offs for consolidation and shows that optimal operating points exist. They model the consolidation problem as a modified bin packing problem and illustrate it with an example.

Verma et al. (Verma, A. et al., 2008) investigated the problem of dynamic placement of applications in virtualized systems, while minimizing power consumption and meeting the SLAs. To address the problem, the authors proposed the pMapper application placement framework. It consists of

http://www.hipore.com/ijsc
three managers and an arbitrator, which coordinates their actions and makes allocation decisions. Performance Manager monitors the behavior of applications and resizes the VMs according to the current resource requirements and SLAs. Power Manager is in charge of adjusting hardware power states and applying DVFS. Migration Manager issues instructions for VM live migration to consolidate the workload. Arbitrator has a global view of the system and makes decisions about new placements of VMs and determines the VM reallocations necessary to achieve a new placement.

Effective consolidation of different applications on common resources is often akin to black art as application performance interference may result in unpredictable system and workload delays. In the paper (Lei L., et al., 2013), Lu lei et al. consider the problem of fair load balancing on multiple servers within a virtualized data center setting. They especially focus on multi-tiered applications with different resource demands per tier and address the problem on how to best match each application tier on each resource, such that performance interference is minimized. To address this problem, they propose a two-step approach.

First, a fair load balancing scheme assigns different virtual machines (VMs) across different servers; this process is formulated as a multi-dimensional vector scheduling problem that uses a new polynomial-time approximation scheme (PTAS) to minimize the maximum utilization across all server resources and results in multiple load balancing solutions. Second, a queueing network analytic model is applied on the proposed min-max solutions in order to select the optimal one. Experimental results show that the proposed mechanism is robust as it always predicts the optimal consolidation strategy.

In particular, the consolidation of multiple customer applications onto multi-core servers introduces performance interference between colocated workloads, significantly impacting application QoS. To address this challenge, Ripal Nathuji et al. (Nathuji, R., et al., 2010) advocate that the cloud should transparently provision additional resources as necessary to achieve the performance that customers would have realized if they were running in isolation. Accordingly, they have developed Q-Clouds, a QoS-aware control framework that tunes resource allocations to mitigate performance interference effects. Q-Clouds uses online feedback to build a multi-input multi-output (MIMO) model that captures performance interference interactions, and uses it to perform closed loop resource management. In addition, they utilize this functionality to allow applications to specify multiple levels of QoS as application Q-states. For such applications, Q-Clouds dynamically provisions underutilized resources to enable elevated QoS levels, thereby improving system efficiency. Experimental evaluations of their solution using benchmark applications illustrate the benefits: performance interference is mitigated completely when feasible, and system utilization is improved by up to 35% using Qstates.

Dejan et al. describe the design and implementation of Deep-Dive, a system for transparently identifying and managing interference (Novakovic, D., et al., 2013). DeepDive successfully addresses several important challenges, including lack of performance information from applications, and large overhead of detailed interference analysis. They first show that it is possible to use easily-obtainable, low-level metrics to clearly discern when interference is occurring and what resource is causing it. Next, using realistic workloads, they demonstrate that DeepDive quickly learns about interference across colocated VMs. Finally, they show DeepDive’s ability to deal efficiently with interference when it is detected, by using a low-overhead approach to identifying a VM placement that alleviates interference.

Dejan et al. (Vasić, N., et al., 2012) also propose DejaVu – a framework that deals with interference by estimating an “interference index”. Their approach to dealing with performance interference on the virtualized hosting platform recognizes the difficulty of pinpointing the cause of interference, and the inability of cloud users to change the hosting platform itself to eliminate interference. DejaVu uses a pragmatic approach in which it probes for interference and adjusts to it by provisioning the service with more resources.

Large-scale datacenters (DCs) host tens of thousands of diverse applications each day. However, interference between colocated workloads and the difficulty to match applications to one of the many hardware platforms available can degrade performance, violating the quality of service (QoS) guarantees that many cloud workloads require. While previous work has identified the impact of heterogeneity and interference, existing solutions are computationally intensive, cannot be applied online and do not scale beyond few applications. Christina D. et al. (Christina D., et al., 2013) present Paragon, an online and scalable DC scheduler that is heterogeneity and interference-aware. Paragon is derived from robust analytical methods and instead of profiling each application in detail; it leverages information the system already has about applications it has previously seen. It uses collaborative filtering techniques to quickly and accurately classify an unknown, incoming workload with respect to heterogeneity and interference in multiple shared resources, by identifying similarities to previously scheduled applications. The classification allows Paragon to greedily schedule applications in a manner that minimizes interference and maximizes server utilization. Paragon scales to tens of thousands of servers with marginal scheduling overheads in terms of time or state.

The cost and benefit of virtual machine migration for resource consolidation are also taken into consideration for some research such as (Qiang H., et al., 2011; Verma, A., et al., 2010; Haikun L., et al., 2011). In (Qiang H., et al., 2011), the authors studied the power consumption of virtual machine live migration. They conducted experiments and
the results showed that the power influence of migration to the original server decreases when the CPU usage of the migrated VM increases, but to the destination server, the influence is stable. In (Verma A., et al., 2010), researchers found that live migration required a significant amount of space CPU on the source server, and the amount of CPU required would increase with an increase in the number of active pages of the VM being migrated. Besides, a co-located VM would impact a VM being migrated by taking away resources from the physical server. Haikun L., et al.(Haikun L., et al., 2011) thoroughly analyze the key parameters that affect the migration cost from theory to practice. They construct two application-oblivious models for the cost prediction by using learned knowledge about the workloads at the hypervisor (also called VMM) level. This should be the first kind of work to estimate VM live migration cost in terms of both performance and energy in a quantitative approach. There exist unconsidered conflicting factors impacting the VM migration process such as load volume, power consumption and resource wastage. Sallam, Ahmed et al. (Sallam, A., et al., 2014) consider the migration process as a multi-objective problem where the objectives are typically non-commensurable. Therefore, they propose a novel migration policy consolidated by a new elastic multi-objective optimization strategy to evaluate different objectives (including migration cost) simultaneously, and to provide the flexibility for manipulating different cases. They have tested the proposed policy through an extensive set of simulation experiments using CloudSim, and the results ensure the efficiency of our policy to control the system performance by adjusting the migration objectives to suit various workload consolidation situations.

Beloglazov A., et al. (Beloglazov A., et al., 2012) propose a novel adaptive heuristics for dynamic consolidation of VMs based on an analysis of historical data from the VM resource usage by VMs. The proposed algorithms significantly reduce energy consumption, while ensuring a high level of adherence to SLA. They validate the high efficiency of the proposed algorithms by extensive simulations using real-world workload traces from more than a thousand PlanetLab VMs.

3. Introduction and Description of Cloud Resource Predictive Dynamic Provisioning Technology Models

In recent years, resource predictive dynamic provisioning technologies have been used in cloud data center environments, as a necessary pre-step to realize cloud resource dynamic provisioning on predicted demand. We have listed the main cloud resource prediction models in Figure 2. We introduce and describe the main mechanism of these models in this section.

3.1 Basic Prediction Models

Most classical methods for prediction are based on time series analysis and there also are some advanced models, such as ARMA models, ARIMA models, and state-space models etc. Because of the computational complexity of these methods, most prediction models adopted in cloud are simplified. Here we will introduce some basic prediction methods.

Niblabja Roy et al. (Roy, N., et al., 2011) develop a model-predictive algorithm for workload prediction that is used for resource autoscaling in cloud. In their strategy, they use a second order autoregressive moving average method (ARMA) filter. The equation for the filter used is given as

\[ \lambda(t + 1) = \beta \times \lambda(t) + \gamma \times \lambda(t - 1) + (1 - \beta + \gamma) (\lambda(t - 2)) \]

(1)

The values for the variables \( \beta \) and \( \gamma \) are given by the values 0.8 and 0.15 respectively.

This ARMA model is a simple case and there are some deficiencies. Firstly, they use history data directly to model the workload pattern, without a pre-processing, which may result in jitter in prediction. Secondly, assigning the variables in the model to specified values will affect the prediction accuracy.

Danilo Ardagna et al. (Ardagna, D., et al., 2011) propose capacity allocation techniques able to coordinate multiple distributed resource controllers working on distributed cloud sites. They leverage exponential smoothing to predict the local arrival rate. The detailed description of the ES-based prediction model is as follows. Suppose the time scale \( T_i \), at sample \( t \), the ES model predicts the local arrival rate at \( T_i \) steps ahead, \( \hat{A}^i_k(t) \), as a weighted average of the last sample \( \hat{A}^i_k(t) \) and of corresponding predicted sample \( \hat{A}^i_k(t) \), that is equal to:

\[ \hat{A}^i_k(T_i) = \frac{1}{T_i} \sum_{t=1}^{T_i} \hat{A}^i_k(t) \]

(2)

\[ \hat{A}^i_k(t + T_i) = \gamma^i_k(t) \hat{A}^i_k(t) + (1 - \gamma^i_k(t)) \hat{A}^i_k(t), t > T_i \]

(3)

Where \( \hat{A}^i_k(T_i) \) is the initial predicted value and \( 0 < \gamma^i_k(t) < 1 \) is the smoothing factor at current sample \( t \) related to the site \( i \) and the class \( k \) that determines how much weight is given to each sample. They obtain a dynamic ES model by re-evaluating the smoothing factor \( \gamma^i_k(t) \) at each prediction sample \( t \). the smoothing parameter is defined as the absolute value of the ratio of the smoothed error, \( \hat{A}^i_k(t) \), to the absolute error, \( E^i_k(t) \), \( \gamma^i_k(t) = \frac{\hat{A}^i_k(t)}{E^i_k(t)} \) (Trigg, D., et al., 1967).

In this model, the evaluation of the smoothing factor is critical in that the selection of this factor has a direct impact on the algorithm accuracy. For a better fitting to the system, they leverage dynamic ES model, so they have to re-evaluate its smoothing factor frequently.
Brian Guenter et al. (Guenter, B., et al., 2011) present a method to predicate the number of physical machines needed in some future time. In the paper, they present ACES, an automated server provisioning system that aims to meet workload demand while minimizing unmet load demand. The ACES has key components: a load predictor for proactive provisioning and an optimizer framework. The predictor is based on regression analysis to predict load in the near future. The method is as follows:

\[ m_0 = [a_1, \ldots, a_p] \begin{bmatrix} m_{-p} \\ \vdots \\ m_{-1} \end{bmatrix} \]

(4)

The future demand \( m_0 \) is predicted by taking the dot product of coefficient vector \( a \) with a vector of past demand values, \( m_t(i<0) \). Where \( (a_1, \ldots, a_p) \) denote the \( p \) coefficients of a weighted liner predictor and are computed by solving the least-squares problem.

This method directly develops the prediction model with historical data, without considering the workload patterns. It is simple, but may have some improvement on accuracy.

Gong, Chen et al. develop a prediction algorithm in (Chen, G., et al., 2008). In order to achieve energy-aware server provisioning, they exploit short-term load forecasting method. In this method, let \( y(t) \) be the stochastic periodic time series under consideration, with a specified time unit. It can represent the parameters measured at regular time intervals. Suppose the periodic component has a period of \( T \) time units. The value of \( y(t) \) in terms of all previous measurements as:

\[ y(t) = \sum_{k=1}^{n} a_k y(t - kT) + \sum_{j=1}^{m} b_j \Delta y(t - j) \]

(5)

\[ \Delta y(t - j) = y(t - j) - \frac{1}{n} \sum_{k=1}^{n} y(t - j - kT) \]

(6)

There are two parts in the model. The part with parameters \( a_k \) does periodic prediction—it is an auto-regression model for the value of \( y \) over a period of \( T \). The part with parameters \( b_j \) gives local adjustment, meaning the correlations between \( y(t) \) and the values immediately before it. The integer \( n \) and \( m \) are their orders, respectively. It is called a SPAR (Sparse Periodic Auto-Regression) model and has a good expansibility to multiple seasonal components.

In this model, there is an assumption that the workload is periodic. In some systems, however, workload is not cyclic. So this assumption limits its use.

Zhenhuan Gong et al. (Gong, Z., et al., 2010) present a novel predictive elastic resource scaling (PRESS) scheme for cloud systems. This approach leverages light-weight signal processing and statistical learning algorithms to achieve online predictions of dynamic application resource requirements. For workloads with repeating patterns, PRESS derives a signature for the pattern of historical resource usage, and uses that signature in its prediction. PRESS uses signal processing techniques to discover the signature, or decide that one does not exist. It employs a Fast Fourier Transform (FFT) to calculate the dominant frequencies of resource-usage variation in the observed load pattern. If there are multiple dominating frequencies, PRESS picks the lowest dominating frequency \( f_0 \). With \( f_0 \), PRESS derives a pattern window size of \( Z \) samples: \( Z = (1/ f_0) \times r \) where \( r \) denotes the sampling rate. It then splits the original time series into \( Q = [W/Z] \) pattern windows: \( P_1 = \{l_1, \ldots, l_Z\} \). \( P_2 = \{l_{Z+1}, \ldots, l_{2Z}\} \ldots \). \( P_i = \{l_{(i-1)Z+1}, \ldots, l_{iZ}\} \). PRESS evaluates the similarity between all pairs of different pattern windows \( P_i \) and \( P_j \) to discover whether it includes repeating patterns. If all pattern windows are similar, PRESS treats the resource time series as having repeating behavior, and uses the average value of the samples in each position of pattern windows to make its prediction. However, for applications don’t have repeating patterns, PRESS uses a discrete-time Markov chain with a finite number of states to build a short-term prediction.

This prediction model can conduct a prediction without assuming any prior knowledge, like the repeating pattern, period. But sometimes the lowest frequency is not the best choice. Besides, if there is no repeating pattern, it leverages the Markov chain to give a prediction, and the accuracy will be impacted.

3.2 Feedback Based Prediction Models

Despite the potential for basic prediction models to perform very well in some systems, in order to guarantee application performance, there have been some prediction models combined with control theory. With performance feedback information, the predicted values can be remedied adaptively to obtain higher accuracy.

Jie Zhu et al. (Zhu, J., et al., 2011) propose a dynamic resource allocation framework (DRAF) for Database-as-a-Service, in which there is a prediction model. In their model, time series is assumed to be represented as trend plus noise, i.e. \( T_{l,t}(t) = T_{l, t} + e_{l,t}, t \in T_j \), where \( T_{l, t}(t) \) is the trend and \( e_{l,t} \) is the noise. \( T_{l, t}(t) \) is obtained through regression analysis, and \( e_{l,t} \) is assumed to have a normal distribution unless additional information about noise becomes available. A trend pattern is represented as \( T_{l, t}(t) = b_0 + b_1 t + b_2 t^2 \). A least-squares method is used to abstract patterns. Noise is described in two parameters: the mean value \( \mu \) and the standard deviation \( \sigma \). According to the normal distribution table, \( \mu + 1.96\sigma \) can cover over 95% of noise. Therefore, \( e_{l,t}=\mu+1.96\sigma \). In every interval, predictor (a component) adjusts the model based on the analyzers output. Lastly, a small offset is added to the predicted curve.

This prediction model leverages a simple method to predict the workload. It neglects the cyclical component in the time series, which in some applications is a major characteristic. Meanwhile, the remedy of this method is coarse.
Zhiming Shen et al. (Shen, Z., et al., 2011) present CloudScale, a system that employs online resource demand prediction and prediction error handling to achieve adaptive resource allocation in the cloud. Online resource demand prediction frequently makes over- and under-estimation errors. Over-estimation may result in resource waste, but it will not impact the application performance. On the contrary, under-estimation will result in SLO violations for insufficient resource allocated to application. In CloudScale, the basic prediction method is based on the model presented in (Barham, P., et al., 2003). The main work to alleviate under-estimation errors is prediction error correction. Here, we introduce how the prediction error correction module works. The module incorporates both proactive and reactive approaches to handle under-estimation errors. CloudScale uses an online adaptive padding scheme to avoid under-estimation errors by adding a small extra value to the predicted resource demand. When the application suffers SLO violations during resource under-estimation, it uses fast under-estimation correction to solve the problem. A simple solution is to immediately raise the resource cap to the maximum possible value. Another method is to raise the resource cap gradually by multiplying the current resource cap by a ratio a>1 until the error is corrected.

Pradeep Padala et al. (Padala, P., et al., 2009) present a resource control system AutoControl, a control system of multiple virtualized resources. In this system, a model estimator inputs past performance and past allocation to get the performance target in the future, and then the optimizer computes the resource needed. For every control interval, the model estimation re-computes a linear model that approximates the quantitative relationship between the resource allocations to application a (u_t) and its normalized performance (y_t) around the current operating point. The relationship is represented by the following auto-regression moving average (ARMA) model:

\[
y_a(k) = a_1(k)y_a(k-1) + a_2(k)y_a(k-2) + b_0^T(k)u_a(k) + b_1^T(k)u_a(k-1)
\]

(7)

Where \(a_1(k)\) and \(a_2(k)\) capture the correlation between the application’s past and present performance, and \(b_0(k)\) and \(b_1(k)\) are vectors of coefficients capturing the correlation between the current performance and recent resource allocations. With the performance target, optimizer determines the resource allocations required.

This model reflects the concept of control theory. The resource need in the future is determined both by application’s performance and past allocation. In this model, the prediction is simple, and the key is an optimizer which transforms the performance target to the resource needed.

### 3.3 Multiple Time Series Prediction Models

In the multiple time series prediction, prediction models are developed based on both autocorrelation of a single resource and cross correlation between different resources. Considering the cross correlation can help to obtain a more accurate prediction model on the premise that there indeed exist some relationships among these resources.

Liang, Jin et al. (Liang, J., et al., 2004) propose a multi-resource prediction model (MMModel) in a distributed resource sharing environment. This model uses both the autocorrelation and the cross correlation to achieve higher prediction accuracy. The autocorrelation characterizes the statistical relationship of same resource at different times, and the cross correlation characterizes the statistical relationship between two resources. The multi-resource prediction model is as follows:

\[
\hat{x}_k = \sum_{i=1}^{p} a_i x_{k-i} + \sum_{i=1}^{q} b_i y_{k-i}
\]

(8)

\[
\hat{y}_k = \sum_{i=1}^{p} c_i y_{k-i} + \sum_{i=1}^{q} d_i x_{k-i}
\]

(9)

This means the resource value \(x_t\) is predicted from not only the history value \(x\) itself but the resource value \(y_t\). The same is true for the prediction of \(y_t\). Here \(p\) is the auto-regression order and \(q\) is the cross regression order.

In the case of MMModel, model fitting is similar to that of an AR model. In general, this model can achieve a higher accuracy compared with AR models considering only autocorrelation. There are some assumptions that the mean and autocorrelation of the resource are time invariant. Because of the cross correlation, there is another assumption that the cross correlation is also time invariant. All these assumptions limit the use of this model.

Jian Tan et al. (Tan, J., et al., 2011) propose two ways, from microscopic and macroscopic perspectives, to predict the resource consumption for data centers by statistically characterizing resource usage patterns. The first approach focuses on the usage prediction for a specific node, and the second approach is based on Principal Component Analysis (PCA) to identify resource usage patterns across different nodes. Auto-regression models for a VM: let \(X^j_T(t) = (X^j_{11}(t), X^j_{12}(t))\) denote the stationary multivariate time series obtained by preprocessing the raw data. \(X^j_{11}(t)\) denotes the CPU usage and \(X^j_{12}(t)\) denotes the memory usage. They apply standard AR models to characterize data. It relies on both CPU and memory series for prediction, using

\[
X^j_{11}(t) = b_{j,11}X^j_{11}(t-1) + b_{j,12}X^j_{12}(t-1) + \beta^j_1 e^j_1(t)
\]

(10)

\[
X^j_{12}(t) = b_{j,11}X^j_{11}(t-1) + b_{j,12}X^j_{12}(t-1) + \beta^j_1 e^j_1(t)
\]

(11)

Where \(e^j_1(t), e^j_2(t)\) are white noise(\(WN(0, \sigma^2)\)). The coefficients are estimated with the Yule-Walker method (Peter, J.B., et al., 2002). PCA based prediction across nodes: this method uses principal component analysis to identify the resource usage patterns across the whole data center, and then conducts AR model prediction on a subset of principal components that account for most of the variability in the measurements. After forecasting resource usages for the
small number of principal components, it will then map the predictions along these identified principal components to the resource usage observation space as the prediction.

The whole prediction strategy has two levels. The first one characterizes the usage pattern for each VM using an AR model. The higher level uses PCA to identify the principal components, and with this subset to forecast the whole system resource consumption.

Arijit Khan et al. (Khan, A., et al., 2012) present a multiple time series approach for workload characterization and prediction in the cloud. This approach firstly groups servers into some co-clusters by a co-clustering method. In the workload prediction, the first step is to analyze the temporal correlation of server workload at different time granularity, and determine the optimal interval for developing a prediction model. Step 2 is to identify the predictable co-clusters from the original co-clusters. Step 3 is to construct the prediction model. In order to predict workload changes for those predictable co-clusters, they divide them into prediction groups and a prediction model can be developed for each of these groups. Paroli et al. model the workload variations across all co-clusters in a group as a continuous-time Markov process (Paroli, R., et al., 2002). Using a Hidden Markov Model (HMM), they define some parameters. Once the state transition and observation probabilities have been estimated, given the current observation at any time t, the probability of observing \( o_{t+1} \) at time \( t+1 \) can be calculated as:

\[
\sum_{H_t} P(H_t | o_t) \sum_{H_{t+1}} P(H_{t+1} | H_t) P(o_{t+1} | H_{t+1})
\]

(12)

With this probability estimation, it can predict the most likely observation at time \( t+1 \) as

\[
o_{t+1} = \arg \max_{o_t} P(o_t | o_t), o_t \in O
\]

(13)

Note that the next observation represents a possible combination of which co-clusters in a prediction group would appear. Therefore, if a co-cluster is predicted to appear in the next time interval, we can predict accordingly the discretized workload levels of all associated servers.

To improve prediction accuracy, this approach develops a model that captures workload patterns along both temporal and spatial dimensions. Developing a prediction model specialized for a group can characterize the series patterns more precisely. While all the prediction work is used for the predictable co-cluster, there is little concern for others without predictability.

Yexi Jiang et al. (Yexi, J., et al., 2011) present an online temporal mining system called ASAP, to model and predict cloud VM demands. It uses a two-level ensemble method to capture the characteristics of the high transient demands time series. The first level is a regression based ensemble that combines the results of different prediction models of the same VM type. The weights of these methods will be updated by calculating the relative error. The models used in this level include moving average, auto-regression, artificial neural network, support vector machine, and gene expression programming. The second level ensemble considers the relationship between different VM types, and utilizes their correlation to help improve the robustness of prediction. With the correlation information between time series to develop prediction model helps to alleviate the disturbance of the “noisy” data in real cloud service scenario. Suppose \( \text{cov}_{ij}^{(t)} \) denotes the covariance between resource types \( i \) and its \( j_{th} \) correlated resource, the post-processed predicted demand for type \( i \) at time \( t \) should be

\[
\hat{q}_i^{(t)} = \frac{\sum_{j=1}^{k} \text{cov}_{ij}^{(t-1)} s_{ij} \hat{v}_j^{(t)}}{\sum_{j=1}^{k} \text{cov}_{ij}}
\]

(14)

Where \( s_{ij} = \hat{v}_i / \hat{v}_j \) denotes the difference of scale between two time series and \( k \) is the number of strong correlated time series.

This model considers both the autocorrelation of each VM type and the cross correlation between different types to improve the prediction accuracy. Meanwhile, in the first level, it integrates many prediction models to characterize the pattern of series, which leads to greater computational capability at some degree.


4.1 Classification

In section 3, we divide the cloud resource prediction models into three categories: basic models, feedback based models and multiple time series models. From the description above we find whether feedback models or multiple time series models, their most fundamental work of prediction are based on basic models. So we can depict the relationships of these categories as follows, shown in Fig. 3.
4.2 Comparison and Analysis

We analyze and compare all of the models in each category in Table 2, 3, and 4 respectively. With respect to each category, we list different metrics to depict the models in it.

4.2.1 Basic Prediction Models

In Table 2, except for the FFT based model, their core methods for prediction are auto-regression. In the prediction area, auto-regression is a fundamental but widely used method. The AR model is a special case of the ARMA model. In (Peter, J.B., et al., 2002), it shows the whole theories about different prediction models, including the ARMA model. The AR model is simple and the accuracy is acceptable for some systems. The key issue of this method is estimation parameters, and (Peter, J.B., et al., 2002) shows some approaches for it. In the models here, they all leverage the least square method to estimate parameters. For the ES model, the key is to evaluate $\gamma_i(t)$ precisely, which influences the prediction accuracy directly. The problems with these models are that they are suitable for short-term prediction, but for long term prediction, they cannot guarantee accuracy. This is the same for the FFT based model. If there is a repeating pattern in the time series, it can predict the whole future values in a pattern window. However, if it fails to find a repeating pattern, it will adopt Markov chain to predict, which just give a short term prediction.

4.2.2 Feedback Based Prediction Models

The main feature of feedback based prediction models in Table 3 is to automatically provide resources for application and avoid SLOs violations. In order to guarantee SLOs, the first thing is adding an offset to the predicted value. Secondly, with feedback information, schemes rebuild the prediction models in every control interval. The new prediction models consider the past performance of applications relying on the past allocation, and make some adjustments. Thus, it can reflect the latest status of application better. In general, feedback based models can achieve better performance than basic models.

<table>
<thead>
<tr>
<th>Table 2. Comparison and analysis on basic models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction models</td>
</tr>
<tr>
<td>ARMA model (Roy, N., et al., 2011)</td>
</tr>
<tr>
<td>Exponential Smoothing based model (Ardagna, D., et al., 2011)</td>
</tr>
<tr>
<td>FFT based model (Gong, Z., et al., 2010)</td>
</tr>
<tr>
<td>Line prediction (Zauner, B., et al., 2011)</td>
</tr>
<tr>
<td>Sparse periodic auto-regression (Chen, G., et al., 2008)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3. Comparison and analysis on feedback based models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheme</td>
</tr>
<tr>
<td>CloudScale (Shen, Z., et al., 2011)</td>
</tr>
<tr>
<td>DRAF (Zhu, J., et al., 2011)</td>
</tr>
<tr>
<td>AutoControl (Padala, P., et al., 2009)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4. Comparison and analysis on multiple time series models</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheme</td>
</tr>
<tr>
<td>Multi-resource prediction (Tan, J., et al., 2011)</td>
</tr>
<tr>
<td>MModel (Liang, J., et al., 2004)</td>
</tr>
<tr>
<td>Group-level approach (Khan, A., et al., 2012)</td>
</tr>
<tr>
<td>ASAP (Yexi, J., et al., 2011)</td>
</tr>
</tbody>
</table>

4.2.3 Multiple Time Series Prediction Models

In Table 4, the first two models consider the correlations between different resources because in some systems a resource usage will influence some other resource’s usage. So they leverage this characteristic. But the premise is that the correlation between resources indeed exists, otherwise, it will not help to improve the prediction accuracy. In the last two models, the purpose of introducing correlation is to alleviate the influence of “noisy” data. In the process of building prediction model, if there are much “noisy” data, the model may not characterize the pattern of the series.

http://www.hipore.com/ijsc
accurately. As for the introduction of cross correlation, we need to evaluate the necessity, because it will complicate the models.

In order to maximize resource utilization and meanwhile guarantee application SLOs, almost all resource dynamic provisioning schemes first adopt prediction techniques, especially for long-term demand forecast. Some new prediction models, such as feedback based models or multiple time series models perform better than the basic prediction model. Sometimes, we should balance the accuracy and complexity of the models and pick the most suitable one. High accuracy is not always required because different systems have different service requirements.

4.3 A BASIC PROCESS OF PREDICTION MODELING AND DYNAMIC RESOURCE PROVISIONING SCHEME

Here, we first describe a general procedure of prediction modeling which is a reasonable method widely used. Then we present an outline of cloud resource dynamic provisioning schemes based on the prediction model.

4.3.1 Time Series Modeling

In [Peter, J.B., et al., 2002], a general method is listed to build a prediction model. Although not all prediction methods do so, it gives us an overview on prediction.

- Plot the series and examine whether there is
  (a) a trend
  (b) a seasonal component
  (c) any apparent sharp changes in behavior
  (d) any outlying observations
- Remove the trend and seasonal components to get stationary residuals.
- Choose a model to fit the residuals.
- Forecasting will be achieved by predicting the residuals and then inverting the transformations to arrive at forecasts of the original series.

4.3.2 The Outline of Resource Predictive Dynamic Provisioning Scheme

Based on the discussion above, a reasonable resource predictive provisioning scheme should include the following steps: modeling, forecasting, error correction, performance analysis, re-modeling, as shown in Fig. 4. When we build a prediction model based resource dynamic provisioning scheme, we need not to follow these steps strictly, and we can adjust some intermediate modules flexibly.

Cloud resource reactive dynamic provisioning technology handles excess demand by computing additional capacity at short time scales (e.g., the number of core, amount of memory, or the number of servers), in response to excess workload that is beyond the predictive workload, i.e., the difference between the actual workload and forecast based workload. Summarized from the various research directions on reactive dynamic provisioning technology, we can conclude some of the key objectives with wide public concerns, shown in Figure 5.

Figure 5. Main Objectives of Cloud Resource Reactive Dynamic Provisioning Technology

To handle a broader spectrum of possible changes in workload, including unpredictable changes in the request size and server efficiency, Gandhi A., et al. introduce the AutoScale policy (Gandhi A., et al., 2012). While AutoScale- addresses a problem that many others have looked at, it does so in a very different way. Whereas prior approaches aim at predicting the future request rate and scaling up the number of servers to meet this predicted rate, which is clearly difficult to do when request rate is, by definition, unpredictable. AutoScale-- does not attempt to predict future request rate. Instead, AutoScale-- demonstrates that it is possible to achieve SLAs for real-world workloads by simply being conservative in scaling down the number of servers: not turning servers off recklessly.

This article makes the following contributions.

- They overturn the common wisdom that says that capacity provisioning requires “knowing the future load and planning for it,” which is at the heart of existing predictive capacity management policies. Such predictions are simply not possible when workloads are unpredictable, and, they furthermore show they are unnecessary, at least for the range of variability in our workloads. They demonstrate that simply provisioning carefully and not turning servers off recklessly achieves better performance than existing policies that are based on predicting current load or over-provisioning to account for possible future load.

- They introduce our capacity inference algorithm, which allows us to determine the appropriate capacity at any point of time in response to changes in request rate, request size and/or server efficiency, without any knowledge of these quantities. They demonstrate that AutoScale, via the capacity inference algorithm, is robust to all forms of changes in load, including unpredictable changes in request size and unpredictable degradations in server speeds, within the range of our traces.

http://www.hipore.com/ijsc
Problem Analysis: AutoScale really takes a fundamentally different approach to dynamic capacity management than has been taken in the past. First, AutoScale does not try to predict the future request rate. Instead, AutoScale introduces a smart policy to automatically provision spare capacity, which can absorb unpredictable changes in request rate. Existing reactive approaches can be easily modified to be more conservative in giving away spare capacity so as to inherit AutoScale’s ability to absorb unpredictable changes in request rate. But the problem is when the heavy workload comes suddenly, we really need to start up more new VMs, and we need consider how to decrease the set up time for new VMs.

Therefore, we can see that the VM setup time is a big concern in cloud resource reactive dynamic provisioning methods to handle workload spike or cloud burst. Some researchers have used pre-copy VM image methods to solve this problem to some extent.

Gandhi A., et al. also present SoftScale, a practical approach to handling load spikes in multi-tier data centers without having to over-provision resources (Gandhi A., et al., 2013). SoftScale works by opportunistically stealing resources from other tiers to alleviate the bottleneck tier, even when the tiers are carefully provisioned at capacity. SoftScale is especially useful during the transient overload periods when additional capacity is being brought online. Importantly, SoftScale can be used in conjunction with existing dynamic server provisioning policies, such as AutoScale.

The key insight of Tian Guo et al. (Guo, T, et al., 2014) is that occasional pre-copying of virtual disk snapshots of a few overload-prone applications can significantly reduce the cloud bursting latency, since only the incremental delta of the disk state needs to be transferred to reconstruct the image in the cloud. Their work examines the impact of judiciously choosing the candidate applications for such pre-copying. They have developed a system called Seagull to address the above challenges; Seagull automatically detects when resources in a private cloud are overloaded, decides which applications can be moved to a public cloud at lowest cost, and then performs the migrations needed to dynamically expand capacity as efficiently as possible.

Seagull supports both horizontally and vertically scalable applications for cloud bursting. It also allows flexible policies to be specified in terms of which applications to periodically pre-copy into the cloud. By automating these processes, Seagull is able to respond quickly and efficiently to workload spikes, allowing the data center to be safely operated at a higher utilization level.

The decision of when to trigger a cloud burst involves monitoring one or more metrics and using a threshold on them. Depending on the scenario, Seagull can use system-level metrics (such as CPU utilization, disk/network utilization or memory page fault rate) or application-level metrics such as response time. They assume that the system administrator makes a one-time decision on which metrics are relevant and the corresponding thresholds. In case of system-level metrics, the desired metrics can be monitored at the hypervisor-level, and for application-level metrics, we assume the presence of a monitoring system such as Ganglia that supports extensions to monitor any application metric of interest. Once an overload warning is triggered, Seagull can use either an optimal ILP based algorithm or a greedy heuristic to decide which applications to move to the cloud.

Problem Analysis: Seagull work is that occasional pre-copying of virtual disk snapshots of a few overload-prone applications can significantly reduce the cloud bursting latency, since only the incremental delta of the disk state needs to be transferred to reconstruct the image in the cloud. The problem is how to signature and recognize which applications have overload-prone feature and then take efficient pre-copy action.

In AGILE, Hiep Nguyen, et al. (Nguyen, H., et al., 2013) also use dynamic VM migration to reduce application startup times. By combining the medium-term resource demand prediction with the black-box performance model, AGILE can predict whether an application will enter the overload state and how many new servers should be added to avoid this. The dynamic copy-rate scheme completes the cloning before the application enters the overload state with minimum disturbance to the running system. AGILE is light-weight: its slave modules impose less than 1% CPU overhead.

Problem Analysis: AGILE can predict whether an application will enter the overload state and then take the measure of dynamic cloning, but in the situation of short-term workload fluctuating, the medium-term resource demand prediction method will not be efficient.

In CloudScale solution (Shen, Z., et al., 2011), when applying the predictor to the resource scaling system, they found that the resource scaling system needs to address a set of new problems in order to reduce SLO violations. First, online resource demand prediction frequently makes over- and under-estimation errors. Over-estimations are wasteful, but can be corrected by the online resource demand prediction model after it is updated with true application resource demand data. Under-estimations are much worse since they prevent the system from knowingle the true application resource demand and may cause significant SLO violations. Second, co-located applications will conflict when the available resources are insufficient to accommodate all scale-up requirements.

CloudScale provides two complementary under-estimation error handling schemes: 1) online adaptive padding and 2) reactive error correction. Their approach is based on the observation that reactive error correction alone is often insufficient. When an underestimation error is detected, an SLO violation has probably already happened. Moreover, there is some delay before the scaling system can figure out the right resource cap. Thus, it is worthwhile to perform proactive padding to avoid under-estimation errors.
Problem Analysis: the prediction error correction module: When the prediction error correction module performs online adaptive padding that adds a dynamically determined cushion value to the predicted resource demand in order to avoid under-estimation errors. The problem is how to determine the suitable amount of this value to avoid resource over-provision.

6. ANALYSIS AND COMPARISON ON CLOUD DATA CENTER RESOURCE CONSOLIDATION TECHNOLOGY

The development of virtualization technology has made great contribution to today’s IDC: it can effectively help the IDC to improve its overall utilization of IT resources, as well as provide greater flexibility for IDCs. However, obviously, this requires a higher level of resource consolidation techniques than ever. Resource consolidation aims at finding an assignment of workload to machines in order to minimize the number of used machines. Server consolidation and storage consolidation with virtualization can help reduce the number of data center servers, storage, desktop, and network devices to reduce complexity and make IT management simpler. As mentioned above, typically, resource consolidation is achieved through (1) resource-aware workload scheduling and (2) dynamically adjusting workload placement using migration (Qi Zh. 2013).

Figure 6. Main Objectives of Cloud Resource Consolidation Technology

The workload scheduling of virtual machines is one of the most important consolidation issues in the virtualized cloud data center management field. It aims at obtaining the mapping between virtual machines and physical machines. More specifically, the further resource consolidation is the dynamically adjusting workload placement using migration. A virtual machine can be migrated to another host if the server begins to experience overload, failures or for energy saving purpose. Migration is a tool for load balancing, dealing with hardware failures, consolidating resource, or reallocating resources. In addition, the whole migration process should be adaptive to the dynamic changes of the overall operating environment, and make timely adjustment.

Summarized from the various research directions on cloud virtualized resource consolidation issues, we can conclude some of the key objectives with wide public concerns, shown in Figure 6.

Resource Utilization. Server virtualization and consolidation is an attractive and effective technology to improve the average utilization rates of IT resources which are currently in the 15-20% range (Van, H. N., et al. 2009). It enables smaller resource allocations than physical machine, which hence potentially benefits data centers by allowing several applications to make use of a physical machine. Many research efforts such as (Nakada, H. N., et al. 2009; Hirofuchi, T., et al. 2010; Gupta, R., et al. 2008) viewed this objective as the most basic and fundamental one. Commonly, they took the virtual machine requirements and physical machine size as input, considering resources like CPU, and memory, to compute for a solution with least physical machines.

Energy-saving. Energy costs represent a significant fraction of a datacenter’s budget and this fraction is expected to grow continuously in coming years, which has become an increasingly important concern for many businesses. We focus on energy-efficient resource management technology analysis in this paper. It is recognized that consolidation of VM services workloads and dynamic VM migration based on virtualization techniques show a great opportunity for increased server utilization rate and power decrease (Petrucci, V., et al. 2009). VM migration provides mechanism to realize energy-saving through moving applications running on several underutilized servers to a reduced number of highly used servers. Several recent papers focus on this aspect, e.g. (Rao, L., et al. 2010; Jung, G., et al. 2010; Petrucci, V. et al. 2009; Minghong L., et al. 2013). Most of them proposed their own power model as well as workload model to evaluate the current state, and work on this energy optimization problem.

Migration Cost. Undoubtedly, virtual machine live migration enhances the flexibility in data centers, and gives the possibility for the re-location of VMs and re-distribution of resources. But some researchers also emphasis that the action of “live migration” is definitely not cost-free, on the contrary, the end-to-end performance, power consumption and thermal impacts are significant during the process (Qiang H., et al. 2011; Frnak Y., et al. 2011). Based on this fact, many recent researches such as (Quaenter, B., et al. 2011; Jung, G., et al. 2010; Shrivastava, V., et al. 2011) begin to take migration cost into account and view it as a secondary goal besides the others. The evaluation indicator of this might be migration times or size of VM to be migrated. Some logical policies or special user requirements may also be taken into consideration when executing VM migration. For example, a user might require that two of their virtual machines never run on the same physical host so one will remain available even if a physical host fails (Hysen, C., et al. 2007), or try to deploy a set of VMs in separate physical nodes so as to not to compete over the same resources (Tsakalozos, K. et al. 2010). These may be translated into some placement constraints, such as sets of virtual machines that must always/never be on the same host, as well as lists of permitted/forbidden hosts for certain virtual machines.

User SLA/Application SLO. The applications running inside the virtual machine are much more important than the
VM itself. Therefore, energy-efficient resource consolidation technology must not be at the expense of the users’ SLA and applications’ SLO. What’s more, it is quite difficult to satisfy service-level objectives (SLOs) of applications on shared infrastructure, as application workloads and resource consumption patterns change over time (Padala, P., et al. 2009). Therefore, cloud providers have to deal with the power-performance tradeoff—minimizing energy consumption while meeting consumer SLA. To solve this problem, research such as (Mylavarapu, S., et al., 2010; Padala, P., et al. 2009; Van, H. N., et al., 2009; Petrucci, V. et al., 2009) focused on the complex relationship between application performance and resource reallocations (including VM migration), and further determined the new mapping.

Performance Interference. Despite the benefits of virtualization, including its ability to slice a PM well in terms of CPU and memory space allocation, performance isolation is far from perfect in these environments. As cloud providers continue to utilize virtualization technologies in their systems, this can become problematic. In particular, the consolidation of multiple customer applications onto multicore servers introduces performance interference between collocated workloads, significantly impacting application QoS. Specifically, a challenging problem for providers is identifying (and managing) performance interference between the VMs that are co-located at each PM. Effectively dealing with performance interference is challenging for many reasons.

First, the IaaS provider is oblivious to its customers’ applications and workloads, and it cannot easily determine that interference is occurring.

Second, interference is complex in nature and may be due to any server component (e.g., shared hardware cache, memory, I/O).

Further, interference might only manifest when the co-located VMs are concurrently competing for hardware resources. Some approaches have struggled to improve the performance interference issues (Lei L., et al., 2013; Nathuji, R., et al., 2010; Novakovic, D., et al., 2013; Vasić, N., et al., 2012; Christina D. et al., 2013). Some systems may attempt to ensure the workload on each VM is within a small tolerance of the other workloads on the same physical hosts, or may attempt to avoid congestion through live migration (Yi Zh., et al., 2009; Gerofi, B., et al., 2010).

Above lists some major objectives in the cloud resource consolidation scheme. These objectives are not mutually exclusive, though with possible conflicts, and can be somewhat combined together so as to achieve more comprehensive and better results. Some recent research has made effort towards addressing this multi-objective virtual machine placement problem and has made some progress, e.g. (Günter, B., et al., 2011; Jung, G., et al., 2010; Xu, J., et al., 2010; Salam, A., et al., 2014).

7. PROSPECT ON CLOUD RESOURCE MANAGEMENT TECHNOLOGY STANDARDIZATION TREND

As a growing number of cloud computing products and solutions emerged, some potential problems have been exposed. For example, users may worry about the vendor lock-in, in other words, it might be impossible to move service from one cloud to another. Or, the different management method and process offered by each cloud provider may cause interaction problems. Therefore, it becomes a must to establish some powerful standards, in order to unify the multitude of cloud vendors. Some international standardization organizations begin to work towards international standards of cloud computing resource management, but most of them don’t consider energy-efficient resource management.

DMTF Open Cloud Standards Incubator (DMTF, 2009) established 3 whitepapers in 2009 on use cases and reference architecture as they relate to the interfaces between a cloud service provider and a cloud service consumer. The goal of the Incubator is to define a set of architectural semantics that unify the interoperable management of enterprise and cloud computing. These papers summarize the core use cases, reference architecture, and service lifecycle. These building blocks will be used to specify the cloud provider interfaces, data artifacts, and profiles to achieve interoperable management. After Open Cloud Standards Incubator, DMTF has announced the normal cloud resource management standard (DMTF, 2011) in 2011: the Cloud Infrastructure Management Interface (CIMI). CIMI is a self-service interface for infrastructure clouds, allowing users to dynamically provision configure and administer their cloud usage with a high-level interface that greatly simplifies cloud systems management. The specification standardizes interactions between cloud environments to achieve interoperable cloud infrastructure management between service providers and their consumers and developers, enabling users to manage their cloud infrastructure use easily and without complexity. However, they do not consider energy-efficient management standardized interface, but energy-saving policies can be inserted as a flexible strategy.

Two of the recent representative drafts from IETF are on the subject of Virtual Resource Operations & Management in the Data Center (IETF, 2011.7), and Policies and dynamic data migration in DC (IETF, 2011.6). The former describes the problem of operational and management challenges that virtualization brings in the (carrier) data center as an enabler of new technologies such as self-provisioning and elastic capacity and related benefits of consolidation, reduced total cost of ownership, and energy management. While the latter one describes some examples of the policies and dynamic information that need to migrate with VM, the influence if they are not migrated with VM, the problems that need to be considered with migration

http://www.hipore.com/ijsc
policies and dynamic information. It also describes some existing network management protocols standardized by IETF and the advantages and disadvantages of them for operating policies and dynamic information migration respectively. When the operators carry out VM migration to realize resource consolidation, the goal of this standard draft is to justify that it is necessary to make effort on policy and dynamic information migration together with VM migration for large virtualized Data Center.

8. Conclusion and Future Work

Cloud computing as a new service model has attracted more and more attention. In the industry, there have been some successful examples, such as Google® AppEngine®, Microsoft Azure®, IBM® Blue Cloud®, Amazon® EC2® and S3® etc. Managing data center resources at large scale in both public and private clouds is quite challenging. Power is an expensive resource in data centers. Unfortunately, a lot of power in data centers is actually wasted. One of the big reasons for this waste is low server utilization (Gandhi A., et al., 2013). Servers in data centers are often left “always on”, leading to only 10-30% server utilization (Gandhi A., et al., 2013). In fact, some related reports tell that the average data center server utilization is only 18% despite years of deploying virtualization aimed at improving server utilization. From a power perspective, low utilization is problematic because servers that are on, while idle, still utilize 60% or more of peak power (Gandhi A., et al., 2013).

Further, servers that are on necessitate support from other IT infrastructure including the cooling, networking, and storage systems. Thus, idle servers also lead to indirect power consumption. Given the importance of reducing power consumption in data centers and the fact that data centers have low server utilization, we conclude the resource dynamic provisioning and consolidation solution is a key technology trend for building energy-efficient cloud data center. In this paper, we first make a survey of virtualized cloud data center technology, including current industry efforts and academic efforts, focusing on cloud resource dynamic provisioning technology and cloud resource consolidation technology. Next, we analyze and discuss cloud resource prediction models, including basic models, feedback based models and multiple time series models. Furthermore, we describe the relationship between these categories and the cloud resource prediction models’ characteristics. We list and compare cloud reactive dynamic resource provisioning scheme. After that, we analyze and compare the cloud resource consolidation technology for energy-efficient cloud data center. Furthermore, we analyze the factors which impact the efficiency of VM consolidation.

Dynamic resource provisioning and efficient resource consolidation may lead to significant reduction of the cloud data center energy consumption, while meeting the SLA requirements. Efficient resource prediction is extremely important for cloud data centers comprising multiple computer and storage nodes. There have been several resource prediction methods used in cloud resource dynamic provisioning areas. These methods estimate application workload or resource demand at some time in the future, and provide appropriate resources to meet service demand in advance. The more accuracy in the prediction method, the better service the cloud data center can provide. At the same time, the reactive resource provisioning methods also have played an important role to complement the predictive provisioning method. VM set up time is still a challenge issue when the reactive resource provisioning method is used to handle workload fluctuating or spike. We have built a cloud data center resource management prototype system based on OpenStack and we implement a load prediction algorithm based on auto-regression and we also realize migration-based resource peak load shifting strategies to support resource consolidation. These studies have gained some achievement (Wei F. et al., 2012; Feifei Zh. et al., 2013; Jie B., et al., 2014). Cloud resource consolidation takes dynamic server provisioning one step further and allows different application instances to be co-located on the same physical server (Gandhi A., et al., 2013). The basic idea is to consolidate the workload from several, possibly under-utilized, servers onto fewer servers through migration. Once a decision to migrate one VM is made, we need to determine its best destination to realize resource consolidation while avoiding performance interference. VM consolidation allows the unneeded servers to be turned off or become sleep, resulting in lower power consumption, and higher system utilization. Therefore, after VM consolidation, it is necessary to efficiently determine when and which physical machines should be switch off to save power, or switch back to the active stage for handling the new workload (Beloglazov A., et al., 2011). Data center operators should consider that frequently switching off and back servers increases the wear-and-tear cost of servers, which might also increase the risk of hardware failure. On the other hand, cloud provider is unaware of the application types when consolidating different workloads: different applications have different resource usage pattern. This makes it challenging to co-locate different application instances together on a physical server. For example, it might be more beneficial to co-locate a CPU-intensive application with an I/O-intensive application rather than co-locating two CPU-intensive applications. Identifying the resource usage patterns of different applications is often very difficult because of the heterogeneous nature of application workloads. There is also the concern of security risk, for instance, VM colocation might expose critical applications to security risks because of other suspicious and dangerous applications. VM colocation attack is an important issue to consider while carrying out VM consolidation.
9. ACKNOWLEDGMENT

This paper work is based on the Fudan-Hitachi Innovative Software Technology Joint Laboratory project cloud data center virtualized resource management system and Hitachi Yokohama Research Laboratory project cloud data center resource scheduling management system. This work is also supported by 2014-2016 PuJiang Program of Shanghai under Grant No. 14PJ1431100.

10. REFERENCES


http://www.hipore.com/ijsc

49
Authors

Dr. ZhiHui Lu is an Associate Professor in School of Computer Science, Pudan University, China. Now he is also a senior researcher of IT Management Research Department, Yokohama Research Laboratory, Hitachi Ltd, Japan. His research interests are cloud computing, system resource management, and network multimedia technology. He is an expert member of China SOA and cloud computing standard work group led by CEST. He is also an academic alliance member of DMTF WS-Management Work Group.

Mr. Soichi Takashige is a researcher of IT Management Research Department, Yokohama Research Laboratory, Hitachi Ltd, Japan. His research interests are cloud data center resource management based on OpenStack. He is a team leader of OpenStack research team in YRL.

Ms. Yumiko Sugita is a chief staff researcher of IT Management Research Department, Yokohama Research Laboratory, Hitachi Ltd, Japan. Her research interests are cloud data center resource management and scheduling technology, and she is a Linux Architect.

Dr. Tomohiro Morimura is unit manager of IT Management Research Department, Yokohama Research Laboratory, Hitachi Ltd, Japan. His research interests are IT system management, virtual network and storage management, and IaaS cloud resource management technology.

Dr. Yutaka Kudo is general manager of IT Management Research Department, Yokohama Research Laboratory, Hitachi Ltd, Japan. His research interests are IT system management, service management, and cloud data center resource management technology, and PaaS Cloud management.