International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Efficient Resource Allocation for Heterogeneous Workloads in IaaS Clouds

Rama Manikanta Pola[1], Krishna Chaitanya Paluru[2], M Venkata Ruthvik Raja[3], Ramesh T [4]

[1,2,3]Department of Computer Science and Engineering, R.M.K. Engineering College, Tamil Nadu, India
[4]Associate Professor, Department of Computer Science and Engineering, R.M.K. Engineering College, Tamil Nadu, India

---

Abstract - *Nowadays user demands the large computing resources which can store large data in it, this creates a attentions towards Infrastructure-as-a-service (IaaS) cloud technology. Current IaaS clouds provision resources in terms of virtual machines (VMs) with homogeneous resource configurations where different types of resources in VMs have a similar share of the capacity in a physical machine (PM). Among the users they have different demand for the resources, the capacity of the resources may differ for each. In a high performing computing job need a high capacity resource such as they need more CPU cores. This needs a large amount of memory for big data processing application. In the existing system, the dominant resources get starved due to the high demand for resources and non-dominant resources get wasted due to less utility of the resource. In homogenous resource allocation have these two problems. In order to solve the problem, we implement the heterogeneous resource allocation approach, called skewness-avoidance multi-resource allocation (SAMR). This will enhance the resources to the diversified requirements on different types of resources. Through our solution, we can overcome the unwanted allocation of resources and the resources get split and utilized in a well-versed manner. We use VM allocation algorithm to ensure heterogeneous workloads are allocated appropriately to avoid skewed resource utilization in PMs, and a model-based approach to estimate the appropriate number of active PMs to operate SAMR. This system provides a low complexity in designing the model for practical operation and accurate estimation. The simulation result of the proposed system is far better than the existing system due to the SAMR.*

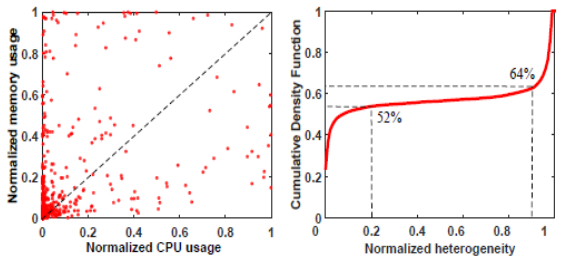
Keywords— Cloud computing, heterogeneous workloads, resource allocation.

1. INTRODUCTION

A public cloud is a common cloud through which service provider provide resources to the users via the internet. Resources get differ but they have storage capabilities and virtual machines. It attracted much attention from academic and industry side. Users are able to benefit from the clouds by highly elastic, scalable and economical resource utilization. But the public cloud is in high demand due to the multiple users and difficult for the user to purchase any items through the public cloud. At the same time on the server side, it is difficult to maintain the hardware at peak load. In recent years, many efforts have been devoted to the problem of resource management in IaaS public clouds such as Amazon EC2 [8] and Rackspace cloud [9]. The above-mentioned studies show their strength in each aspect of resource scheduling. In existing work, the cloud provider allocates a virtual machine (VM) with homogeneous resource configurations. Specifically, homogeneous resource allocation offers resources. In VM all the resources share the physical machine capacity efficiently. The resources are classified into two types they are dominant and non-dominant.

Both dominant resource and non-dominant resource are allocated with the same share in such manner even if the demands for different resources from a user are different. Obviously, using a homogeneous resource allocation approach to serve users with different demands on various resources is not efficient in terms of green and economical computing [10]. For instance, if users need Linux servers with 16 CPU cores but only 1GB memory, they still require to purchase m4.4xlarge (with 16 vCPU and 64 GB RAM) or c4.4xlarge (with 16 vCPU and 30 GB RAM) in Amazon EC2 [8] (July 2, 2015), or Compute1- 30 (with 16 vCPU and 30 GB RAM) or I/O1-60 (with 16 vCPU and 60 GB RAM) in Rackspace [9] (July 2, 2015) to satisfy users' demands. In this case, large memory will be wasted. As the energy consumption by PMs in data centers and the corresponding cooling system is the largest portion of cloud costs [10], [11], [12], homogeneous resource

allocation that provisions large amounts of idle resources wastes tremendous energy. Even in the most energy-efficient data centers, the idle physical resources may still contribute more than one half of the energy consumption in their peak loads. Besides, for cloud users, purchasing the appropriate amounts of resources for their practical demands is able to reduce their monetary costs, especially when the resource demands are mostly heterogeneous traces.



(a) Resource usage of CPU and RAM (normalized to $(0, 1)$)

(b) CDF of heterogeneity

Fig -1: Resource usage analysis of Google Cluster

The resource type is classified in a different manner based on the need of the users. We observe that most resource demands of the applications in cloud workloads are diversified on multiple-resource types (e.g., number of CPU cores, RAM size, disk size, bandwidth, etc.). As shown in Fig. 1, we analyzed the normalized resource (CPU and RAM) usages of a cloud computing trace from Google [13], [14] which consists of a large amount of cloud computing jobs. It is clear that different jobs in Google trace have different demands in various resource types. Fig. 1(a) shows the comparisons of normalized CPU and RAM usages for the first 1000 jobs in Google trace. We can see that most jobs do not utilize the same share of different resource types. Allocating resource according to the dominant source naturally wastes many non-dominant resources. Fig. 1(b) analyzes the distribution of the heterogeneity(defined as the difference between CPU and RAM usage, or CPU usage RAMusagej) for all jobs in Google trace. It reveals that more than 40% of the jobs are highly unbalanced between CPU and memory usage, and there are approximately 36% of jobs with heterogeneity higher than 90%. Homogeneous resource allocation will not be cost-efficient for such heterogeneous workloads in the1 clouds because the non-dominant resources will be wasted significantly. Therefore, a flexible and economical resource allocation method for heterogeneous workloads needed. Nevertheless, consideration of heterogeneous workloads in resource allocation results in a number of challenges the complexity of provisioning algorithms for homogeneous resource allocation [15], [16] is already high and the computational time is long given a large number of PMs in data centers nowadays. To cope with the heterogeneous workloads, this paper proposes a skewness-avoidance multi-resource (SAMR) allocation algorithm to efficiently allocate heterogeneous workloads into PMs. SAMR designs a heterogeneous VM offering strategy that provides flexible VM types for heterogeneous workloads. To measure the skewness of multi-resource utilization in the data center and reduce its impact, SAMR defines the multi-resource skewness factor as the metric that measures both the inter-node and inter-node resource balancing. In resource allocation the process, SAMR first predicts the required number of PMs under the predefined VM allocation delay constraint. Then SAMR schedules the VM requests based on skewness factors to reduce both the inner-node resource balance among multiple resources and the inter-node resource balance among PMs in the data center. By such manner, the total number of PMs are reduced significantly while the resource skewness is also controlled to an acceptable level. Based on our earlier work in [15] which provisions heterogeneous workloads for preset delay constraint, in this paper, we propose a skewness factor based scheme to further optimize the resource allocation for heterogeneous workloads in clouds. Our approach can reduce the resource provider of workload by 45%. It took 11% to reduce the cloud workload when compare with the single-dimensional method and the multi-resource allocation method without skewness consideration, respectively. Organization. The rest of the paper is organized as follows. We first review the related work in Section 2. Section 3 describes the system model of our proposed algorithmSAMR and Section 4 provides a detailed description of our proposed heterogeneous resource allocation algorithm SAMR. Section 5 introduces our developed resource prediction model based on Markov Chain. We present experimental results and discussions in Section 6 and draw an important conclusion in Section 7.

2. RELATED WORK

The review of resource allocation in the cloud center is classified into two categories they are homogenous and heterogeneous resource allocation.

2.1 Homogeneous Resource Allocation

The main goal of the homogeneous resource allocation is to mapping the VMs into PMs under some specific goals. Bin packing is a typical VM scheduling and placement method that has been explored by many heuristic policies [17], [18], [19], [1] such as first fit, best fit and worst fit and others. Some recent studies [18], [20] show that the impact on resource usage among various heuristic policies is similar. However, these policies cannot apply directly to heterogeneous resource provisioning because they may cause resource usage imbalance among different resource types. Some recent works investigated the scheduling of jobs with specific deadlines [3], [4], [5], [6]. As cloud workload is highly dynamic, elastic VM provisioning is difficult due to load burstiness. Ali-Eldin et al. [21] proposed using an adaptive elasticity control to react to sudden workload changes. Niu et al. [22] designed an elastic approach to dynamically resize the virtual clusters for HPC applications. Thus, Deng et al. [7] recently proposed a portfolio scheduling framework that attempts to select the optimal scheduling approach for different workload patterns with limited time. So far, all the research works assume that cloud providers offer VMs homogeneously and all resources are allocated according to their dominant resources. As discussed in Section 1, such a single-dimensional resource allocation method is inefficient on resource usage as well as the cost of both users and cloud providers. Another significant problem in homogeneous resource allocation is resource provisioning which targets on determining the required resources for cloud workloads. To achieve green and power-proportional computing [10], cloud providers always seek elastic management on their physical resources [23], [12], [15], [11], [24]. Li et al. [23] and Xiao et al. [11] both designed similar elastic PM provisioning strategy based on predicted workloads. They adjust the number of PMs by consolidating VMs in over-provisioned cases and powering on extra PMs in under-provisioned cases. Such heuristic adjusting is simple to implement, but the prediction accuracy is low. Model-based PM provisioning approaches [16], [12], [25], [15], on the other hand, are able to achieve more precise prediction. Lin et al. [12] and Chen et al. [25] both proposed algorithms that minimize the cost of data center to seek power-proportional PM provisioning. Hacker et al. [16] proposed hybrid provisioning for both HPC and cloud workloads to cover their features in resource allocation (HPC jobs are all queued by the scheduling system, but jobs in public clouds use all-or-nothing policy). However the above mentioned studies consider CPU as a dominant resource in the single-dimensional resource allocation. In order to handle the provision problem for heterogeneous workload, this paper proposed a model where it provide a minimum number of resources to satisfy the users .

2.2 Heterogeneous Resource Allocation

There have been a number of attempts made on heterogeneous resource allocation [26], [27], [28], [29], [30], [31] for cloud data centers. Dominant resource fairness (DRF) [28] is a typical method based on max-min fairness scheme. It focuses on sharing cloud resources fairly among several users with a heterogeneous resource requirements on different resources. Each user takes the same share on its dominant resource so that the performance of each user is nearly fair because of the performance relies on the dominant resource significantly. Motivated by this work, a number of extensions based on DRF have been proposed [27], [31], [30]. Bhattacharya et al. [31] proposed a hierarchical version of DRF that allocates resources fairly among users with hierarchical organizations such as different departments in a school or company. Wang et al. [27] extended DRF from one single PM to multiple heterogeneous PMs and guarantee that no user can acquire more resource without decreasing that of others. Joe et al. [30] claimed that DRF is inefficient and proposed a multi-resource allocating framework which consists of two fairness functions: DRF and GFJ (Generalized Fairness on Jobs). Conditions of efficiency for these two functions are derived in their work. Ghodsi et al. [29] studied a constrained maximum fairness scheme that has two important properties compared with current multi-resource schedulers including DRF: incentivizing the pooling of shared resources and robustness on users'

constraints. These DRF-based approaches mainly focus on performance fairness among users in private clouds. They do not address the skewed resource utilization. Zhang et al. [32], [33] recently proposed a heterogeneity-aware capacity provisioning approach which considers both workload heterogeneity and hardware heterogeneity in IaaS public clouds. They divided user requests into different classes (such as VMs) and fit these classes into different PMs using dynamic programming. Garg et al. [34] proposed an admission control and scheduling mechanism to reduce costs in clouds and guarantee the performance of the user's jobs with heterogeneous resource demands. These works made contributions on serving heterogeneous workloads in clouds. But they did not consider the resource starvation problem which is the key issue in heterogeneous resource provisioning in clouds. In this paper we allocate a resources based on skenwss avoidance mechanism where reducing the PMs in order satisfy the user with minimum number of resource allocation.

3. SYSTEM OVERVIEW

In system overview we can analysis the system architecture of SAMR and the proposed solution for heterogeneous resource allocation.
Table 1 lists the key notations used throughout this paper.

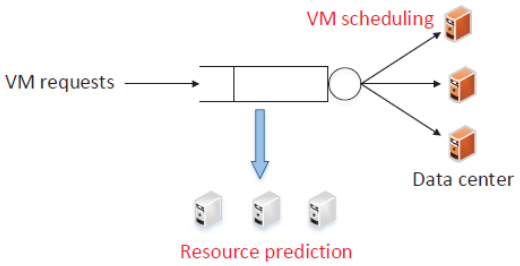


Fig -2: System architecture of SAMR.

Similar to other works that optimize the resource usages in the clouds [10], [11], [12], we use the number of active PMs as the main metric to measure the degree of energy consumption in clouds. To check the performance of the clouds we reduce the active PMs in datacenter and provide a same workload to the cloud and analysis the result. It provide a same result as the PMs number in higher than this ratio. This create an attraction among the cloud operator.

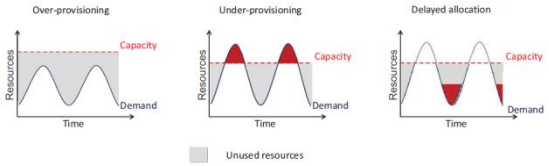


Fig -3: The cases of over-provisioning, under-provisioning and delay caused by under-provisioning

K	Number of resource types
N_{total}	Total number of PMs in the considered data center
$\vec{R}$	r_i is the capacity of type-i resource in a PM, $i = [1, 2, ..., K]$
$\vec{M}$	m_i, $m_i < r_i$ is the maximum resource for type-i resource in a VM, $i = [1, 2, ..., K]$
X	Total number of VM types
$\vec{V}^x$	The resource configuration of type-x VM, v_i^x ($v_i^x \leq m_i$) represents the amount of type-i resource, $x = [1, 2, ..., X]$ and $i = [1, 2, ..., K]$
$\vec{C}$	c_i is the total consumed type-i resource in a PM, $c_i \leq r_i$, $i = [1, 2, ..., K]$
$\vec{U}$	u_i is the utilization of type-i resource in a PM, $u_i \in [0, 1]$, $i = [1, 2, ..., K]$
λ_x	Arrival rate of type-x requests, $x = [1, 2, ..., X]$
μ_x	Service rate of type-x requests, $x = [1, 2, ..., X]$
D	Predefined VM allocation delay threshold
d	Actual average VM allocation delay in a time slot
N	Provisioned number of active PMs (predicted by the model)
$\vec{S}$	s_n is the skewness factor for n^{th} active PM, $n = [1, 2, ..., N]$

Table -1: Notations used in algorithms and models

Normally cloud provider charge the users based on the usage of the VM and its running time. Fig. 2 shows the system model of our proposed heterogeneous resource allocation approach SAMR. Generally, we assume that a cloud data center with Ntotal PMs offers K different resource types (e.g., CPU, RAM, Disk, ...). The cloud system offers X different VM types, each of which is with a resource combination V~x = fvx i ji = 1; 2; :::;Kg(x = 1; 2; :::;X) where vx i denotes the resource capacity of ith resource type in xth VM type. Cloud users submit their VM requests (also denoted as workloads in this paper) to the cloud data center according to their heterogeneous resource demands and choose the VM types that are most appropriate in terms of satisfying the user demands while minimizing the resource wastage. We refer a request for xth type of VM as a type-x request in workloads. All VM requests are maintained by a scheduling queue. For each request from users, resource (or VM) scheduler allocates the resources for requested VM in N current active PMs if

the resource slot of the VM is available. Otherwise, the request will be delayed waiting for more PMs to power up and join the service. Based on the arrival and service rate of the request, SAMR allocate the resources for the users based on a Markov Chain model periodically in every time slot with a duration of t to satisfy the user experience in terms of VM allocation delay. By such manner, we focus on solving the problem in a small time period to increase the prediction accuracy. After the online prediction of required resources, the cloud system provisions corresponding number of active PMs N in the coming time slot. In VM scheduling phase during each time slot with the length t, cloud providers allocate resources and host each VM into PMs using SAMR allocation algorithm. In cloud service, one of the most significant impacts on user experience is the service delay caused by schedulers. Here we consider the resource (or VM) allocation delay as the main metric for service-level-agreements (SLA) between users and cloud providers. Specifically, SAMR uses a VM allocation delay threshold D to be the maximum SLA value that cloud providers should comply with. Thus, there is a trade off between cost and SLA (as shown in Fig. 3) for cloud providers. To cope with the large amount of random request arrivals from users, it is important to provision enough active PMs. However, maintaining too many active PMs may cope well even under peak load but wastes energy unnecessary. It is challenging to find the adequate number of active PMs. In our proposed model we satisfy the SLA value for resource prediction through Markov chain model. Precisely, the model determines the number of active PMs, N, such that the average VM allocation delay d is smaller than the agreed threshold D. We use the Markov Chain model to determine the adequate number of active PMs for operation. The model assumes heterogeneous workloads and balanced utilization of all types of resources within a PM. To realize the balanced utilization, we define a multi-resource skewness as the metric to measure the degree of unbalancing among multiple resource types as well as multiple PMs. Mostly SAMR aim to reduce the skewness in datacenter to avoid the data starvation.

4. SKEWNESS-AVOIDANCE MULTI-RESOURCE ALLOCATION

Here we implement a model to avoid the skewness in the Multi resource allocation algorithm. Firstly, we introduce new notions of VM offering for heterogeneous workloads in clouds. Then we define skewness factor as the metric to characterize the skewness of multiple resources in a data center. Finally, based on definition of skewness factor, we propose a SAMR allocation algorithm to reduce resource usage while maintaining the VM allocation delay experienced by users to a level not exceeding the predefined threshold.

4.1 New Notions of VM Offering

Generally, we consider a cloud data center with Ntotal PMs, each of which have K types of computing resources. We denote ~R =< r1; r2; :::; rK >to be the vector describing the capacity of K types of resources and ~C =< c1; c2; :::; cK >to be the vector that describing the amount of resource used in a PM. To support better utilization of resources for cloud applications with heterogeneous resource demands, it is necessary to consider a new VM offering package to cover the flexible resource allocation according to different resource types. We propose SAMR to offer a series of amounts for each resource type and allow arbitrary resource combinations that a user can pick. For instance, a cloud provider offers and charges VMs according to K resource types (e.g., CPU, RAM, disk storage, bandwidth,...) and the maximum amount of type-i resource (i = 1; 2; :::;K, we refer ith resource type as type-i resource in this paper) is mi. For each type of resource, there is a list of possible amounts for users to choose, and we consider a list of power of 2 for the amounts (e.g., 1; 2; 4; 8; :::) for convenience (SAMR can actually support arbitrary sizes of VMs). Thus, the total number of VM types are X = QK i=1 (log2(mi) + 1). We use ~V x =< v1; v2; :::; vK >x, for x = [1; 2; :::;X], to present a resource combination for type-x VM. SAMR allows users to select the suitable number of resource for each type. Thus, users are able to purchase the appropriate VMs that optimally satisfy their demands to avoid over-investments. We use an example to illustrate above VM offering package. A cloud system may identify two resource types: CPU and memory. The amounts of CPU (number of cores), memory (GB) are expressed by V~x =< v1; v2 >x. If each PM have 16 CPU cores and 32 GB memory and it allows the maximum VM to use all the

resources. Users can select 1 core, 2 cores, 4 cores, ..., or 16 cores of CPU combining with 1 GB, 2 GB, 4 GB, ..., or 32 GB of memory for their VMs.

4.2 Multi-Resource Skewness

As we already mentioned that the hetergenous workload can lead to data starvation if the workloads are not properly managed. Although live migration can be used to consolidate the resource utilization in data centers to unlock the wasted resources, live migration operations result in service interruption and additional energy consumption. Thus the SAMR avoid these type of starvation by various resource types during the allocation. Migration could be used to further reduce the skewness in the runtime of cloud data center if necessary. Skewness [11], [35] is widely used as a metric for quantifying the resource balancing of multiple resources. To better serve the heterogeneous workloads, we develop a new definition of skewness in SAMR, namely skewness factor. Let G = f1; 2; :::;Kg be the set that carries all different resource types. We define the mean difference of the
utilizations of K resource types as

$$Diff = \frac{\sum_{(i\in\mathbf{G},j\in\mathbf{G},i\neq j)} |u_i - u_j|}{K \cdot (K-1)}, \qquad (1)$$

where ui is the utilization of ith resource type in a PM. Then the average utilization of all resource types in a
PM is U, which can be calculated by

$$\overline{U} = \frac{\sum_{i=1}^{K} u_i}{K}. \qquad (2)$$

The skewness factor of nth PM in a cloud data center is defined by

$$s_n = \frac{Diff}{\overline{U}} = \frac{\sum_{(i\in\mathbf{G},j\in\mathbf{G},i\neq j)} |u_i - u_j|}{(K-1)\cdot\sum_{i=1}^{K} u_i}. \qquad (3)$$

The concept of skewness factor is denoted as a factor that quantifies the degree of skewness in resource utilization in a data center with multiple resources. The degree of skewness factor has the following implication
and usages.

- The value of skewness factor is non-negative (sn _ 0), where 0 indicates that all different types of resources are utilized at the same level. The skewness factor closer to 0 reveals lower degree of unbalanced resource usages in a PM. Thus, our scheduling goal is to minimize the average skewness factor. In contrast, a larger skewness factor implies higher skewness, which means that the resource usages are skewed to some specific resource types or some PMs. It also indicates that the PMs have a high probability of resource starvation.
- The skewness factor is the main metric in skewnessavoidance resource allocation for heterogenous workloads. Thus, in the definition of skewness factor, we consider two aspects of the characteristics of the resource usages in PMs to keep the inner-node and inter-node resource balancing. The first aspect is the mean differences between the utilizations of multi-resources within a PM, or inner-node aspect. A higher degree of difference leads to a higher
- skewness factor, which is translated to higher degree of unbalanced resource usage. The second aspect in skewness factor is the mean of utilization of multiresources in a PM. When the first aspect, the mean difference, is identical in each PM in data center, SAMR always choose the PM with the lowest mean utilization to host new VM requests such that the inter-node balance between PMs is covered in the definition of skewness factor.
- The resource scheduler makes scheduling decisions according to the skewness factors of all active PMs in data center. For each VM request arrival, the scheduler calculates the skewness factor for each PM as if the VM request was hosted in the PM. Thus, the scheduler is able to find the PM with the most
- skewness reduction after hosting the VM request. This strategy not only keeps the mean skewness factor of the PM low, but also maintain a low mean skewness factor across PMs. The detailed operation of the skewness-avoidance resource allocation algorithm is provided in the next subsection.

4.3 Skewness-Avoidance Resource allocation

Due to the need of multi resources allocation we implement the SAMR as the resource allocation algorithm for allocating the resources for heterogeneous workload. From Algorthim1 we initially set the N number of physical machine (PM) with prediction model. Let N`be the starting PM at the beginning slot of the process. The PM need to serve the workloads. Based on the prediction the addition and deletion of the PM will be takes place. Each time the user request for the VM then the system conducts the following steps: 1) The system automatically allocate the Vacant VM for the user based on the scheduler request 2) During each search the scheduler will check that there is enough source of PM is available.

Algorithm 1 Allocation algorithm of SAMR

```
Provision N PMs with predition model in Section 5
Let N' be the current number of PMs at the beginning of the time slot
if N > N' then
    Powering on N − N' PMs
else if N < N' then
    Shut down N' − N PMs
if a type-x VM request arrives at cloud system with demand V^x then
    opt = 0
    s_opt = 0
    for n = 1 to N do
        if C + V^x ≤ R then
            Compute s_n with Eq. 3
            Compute new s'_n if host the type-x request
            if s_n − s'_n > s_opt then
                opt = n
                s_opt = s_n − s'_n
    if opt == 0 then
        Power on a PM to allocate the request
        Delay the VM allocation for time t_power
        N = N + 1
    else
        Allocate this VM request to opt^th PM: C = C + V^x
if a type-x VM finishes in the n^th PM then
    Recycle the resource: C = C − V^x
```

If a PM has enough resources to host the requested VM, the scheduler calculates the new multi-resource skewness factor and records the PM with maximum decease in skewness factor. For the PM without enough resources, the scheduler simply skips the calculation.

3) Then each and every active PMs is checked constantly and the scheduler select the PM which has less skewness factor this can help the user to utilize the various resources. If there is no active PMs to host the requested PM, this power up the new VM. This request will experience additional delay (tpower) due to the waiting time for powering up a PM.

4) After each VM finishes its execution, the system recycles the resources allocated to the VM. These resources will become available immediately for new requests.

5. RESOURCE PREDICTION MODEL

In this section, we introduce the resource prediction model of SAMR. The objective of the model is to provision the active number of PMs, N, at the beginning of each time slot. To form an analytical relationship between operational configurations and performance outcomes, we develop a Markov Chain model describing the evolution of resource usage for SAMR in the cloud

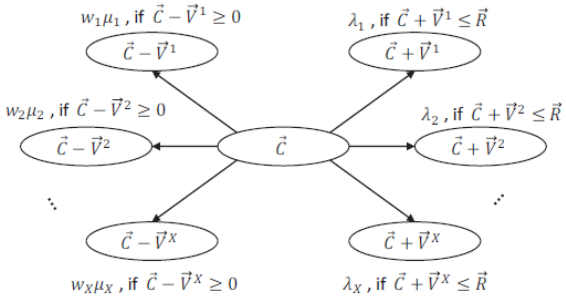


Fig -4: State transitions in the model.

data center. With the model, we can determine the optimal number of PMs for cost-effective provisioning while meeting VM allocation delay requirement. One of the benefits of utilizing the cloud computing is cost effectiveness for users and service providers. This reduce the cost for storing the large data instead external memories. However, due to the complexity in multiple dimensional resource type management, large scale deployment of PMs, and the highly dynamic nature of workloads, it is a non-trivial task to predict the suitable number of active PMs that can meet the user requirement. Modeling all Ntotal PMs and all K types of resource in a

data center leads to a model complexity level of O((QK i=1 ri)3Ntotal) and O((QK i=1 ri)2Ntotal) for computation and space complexity, respectively. For example, with 1000 PMs, 2 types of resources, each with 10 options, the system evolves over 104000 different states. It is computationally intensive to solve a model involving such a huge number of states. Since the resources allocated to a VM must come from a single PM, we see an opportunity to utilize this feature for model simplification. Instead of considering
all PMs simultaneously, we can develop a model to analyze each PM separately which significantly reduces the complexity. We observe that the utilizations of different types of resources among different PMs in data center are similar in a long run under SAMR allocation algorithm because the essence of SAMR is keeping the utilizations balanced among different PMs. Since all active PMs share similar statistical behavior of the resource utilization, we focus on modeling a particular PM in the system. Such approximation method can largely reduce the complexity while providing an acceptable prediction precision. The model permits the determination of allocation delay given a particular number of active PMs, N. With the model, we propose a binary search to find the suitable number of active PMs such that the delay condition of d _ D can be met. In our model, we first predict the workloads at the beginning of each time slot. There are many load prediction methods available in the literature [11], [36], we simply use the Exponential Weighted Moving Average (EWMA) in our paper. EWMA is a common method used to predict an outcome based on past values. At a given time _, the predicted value of a variable can be calculated by

$$P_d = \frac{\sum_{x=1}^{X} P_{d_x} \lambda_x}{\sum_{x=1}^{X} \lambda_x}.$$

P_D-Time slot

For the VM allocation algorithm execution, as it performs linear check on each active PM, the complexity is O(Ntotal). The overall complexity of our solution is thus linear to the number of PMs.

6. EVALUATION

In Evaluation section we analysis the heterogeneous resource allocation method through simulation experiment. Initially we setup the simulator with heterogeneous workload data. Then by using SAMR we compare our solution with the existing method.

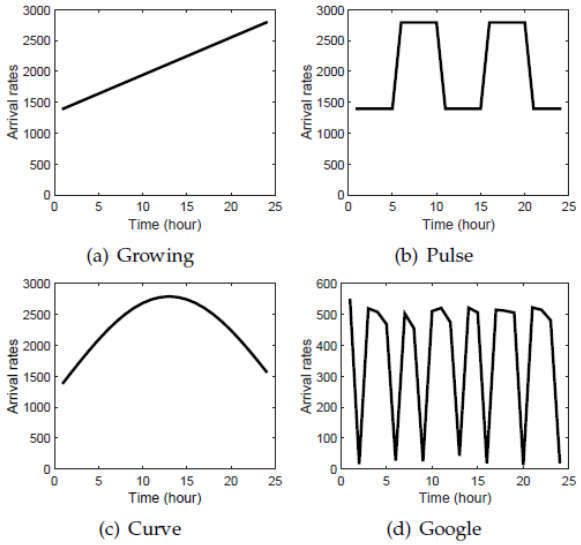


Fig -5: Three synthetic workload patterns and one real world cloud trace from Google.

6.1 Experimental setup

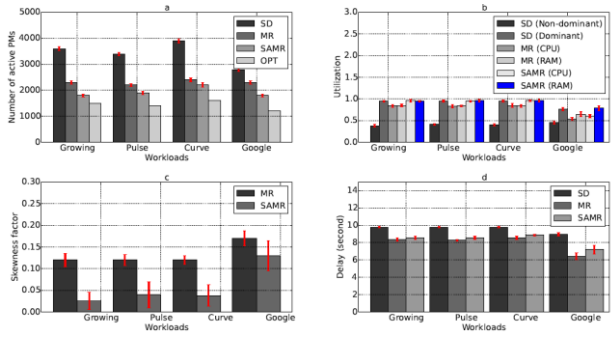


Fig -6: Overall results of four metrics under four workloads.

Simulator: The above figure shows the overall result of four metrics under four different work load. We simulate the Iaas cloud and the bars in the graph shows a average value and the red mark indicate the confidential interval. It is approximately 95%. Stimulator maintain the resources usage of PM in cloud and support to lease and release the

sources for the users based on the request. The resource will be allocated from VM. The VM started with the delay of 10 second. We study the following performance metrics in each time slot: number of PMs per time slot, mean utilization of all active PMs, multi-resource skewness factor and average VM allocation delay. The number of PMs is the main metric which can impact the other three metrics.

Comparisons: The effectiveness of the SAMR is evaluated for maintain the heterogeneous cloud work load. Normally the SNMR is compared with the following method

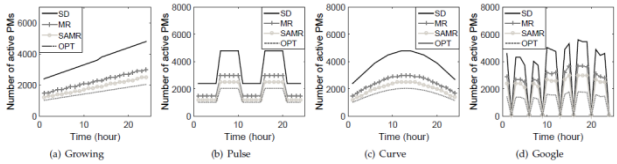
1) single-dimensional (SD) which is used as a homogenous resource allocation approach. It is used as a current Iaas cloud.

Resource allocation in SD is according to the dominant resource, other resources have the same share of dominant resource regardless of users' demands. For scheduling policy,

we simply choose first fit because different scheduling policies in SD have similar performance impact on resource usage. In first fit, the provisioned PMs are collected to form a list of active PMs and the order of PMs in the list is not critical. Based on the request from the user the scheduler check the available resources. If the resources is available then the PM is created or else the user in delay condition to get the resources.

2) multi-resource (MR) it is a combination of multiple resources without skewness factor. MR also uses first fit policy to host VMs in cloud data center. 3) optimal (OPT). An optimal resource allocation (OPT) is compared as the ideal provisioning method with oracle information of workloads. The optimization result of the OPT is calculated by dividing the total resource demands in each time slot by the capacity of the PMs.

Workloads: There are two kind of work load one is synthetic and other is real world cloud trace. Synthetic workload is classified as growing, curve and pulse. The arrival rate for these synthetic pattern may range from 1400 to 2800.



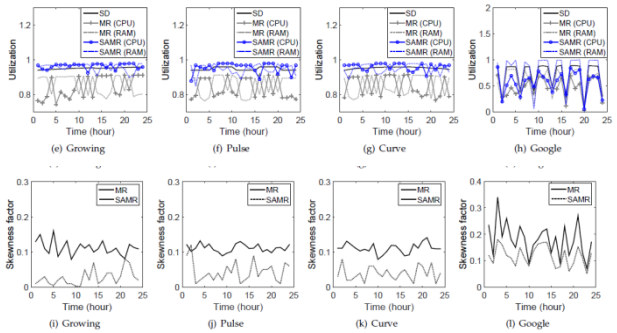


Fig -7: Detailed results of three metrics under four workload

6.2 Experimental results

From the experimental result of the four workload which illustrated in fig 6 based on the resource allocation management. The red marks in the bar indicate the confidence intervals. From the analysis it is understood that the heterogeneous result management method (SAMR and MR) reducing the resources in terms of number of active PMs for the same workloads.

SAMR further reduces the required number of PMs by another 11%, or around 45% compared with SD. It shows that SAMR is able to effectively reduce the resource usage by avoiding resource starvation in cloud data center. Besides, the number of active PMs for SAMR is quite close to the optimal solution with only 13% difference. Note that the presented number of active PMs for SAMR is the actual required number for the given workloads. Based on our experiment records,the predicted numbers of PMs from our model have no

more than 5% (4:3% on average) error rates compared

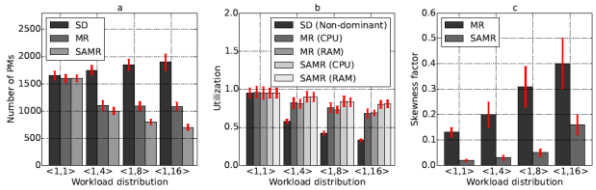


Fig -8: Sensitivity studies for different degrees of heterogeneity (workload distributions). The bars in the figure show average values and the red lines indicate 95% confidence intervals.

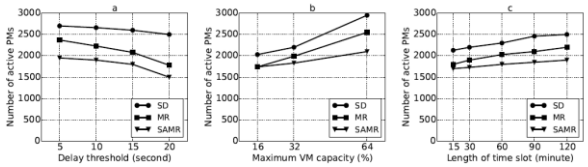


Fig -9: Sensitivity studies for delay threshold, maximum VM

capacity and length of time slot using Google trace. with the actual required numbers presented in the figure. Secondly, although the utilization of dominant resource using SD method is high as shown in Fig. 6(b), the non-dominant resources are under-utilized. However, the resource utilizations in MR and SAMR policies are balanced. This is the reason that SD must provision more PMs. Thirdly, the effectiveness of resource allocation in SAMR is validated by the skewness factor shown in Fig. 6(c), where the average resource skewness factors in SAMR method are less than that in MR. Finally, all three policies achieve the predefined VM allocation delay threshold as shown in Fig. 6(d). SD holds slight higher average delays than SAMR and MR, which is due to the fact that SD always reacts slowly to the workload
dynamicity and cause more under-provisioned cases to make the delay longer.

Impacts by the amount of workloads. Fig. 7 shows the detailed results of all methods for different metrics under four workloads. We highlight and analyze the following phenomenons in the results. Firstly, heterogeneous resource allocation methods significantly reduce the required number of PMs in each time slot for 4 workloads as in Fig. 7(a) to Fig. 7(d). Secondly, from Fig. 7(e) to Fig. 7(h) we can see that SAMR is able to maintain high PM utilization in data center but the PM utilization of MR method fluctuates, falling down
under 80% frequently. This is due to the starvation or unbalanced usage among multiple resource types in MR as shown in Fig. 7(i) to Fig. 7(l). Thirdly, we observe that the utilization of CPU and RAM resources using SAMR are close in the three synthetic workloads but the difference in Google trace is large as shown in Fig. 7(e) to Fig. 7(h). This is caused by the fact that the total demands of RAM is more than that of CPU in traces from Google Cluster. It can also be verified by the higher resource skewness factors in Fig. 7(i) to Fig. 7(l), where the skewness factors in Google trace are much higher
than the other three workloads.

Impacts by workload heterogeneity. We first investigate the performance under different workload distributions with different degrees of heterogeneity. We run four experiments using Growing pattern in this study. In each experiment, the workload consists of only two types of VMs (the amounts of two types of VM are the same) with the same heterogeneity degree. Specifically,we use <1; 1 >+ <1; 1 >, <1; 4 >+ <4; 1 >, <1; 8 > + <8; 1 >, and <1; 16 >+ <16; 1 >in the first, second, third and fourth experiments, respectively. For all the experiments, we keep the total amounts of dominant resource identical in order to compare the impacts of heterogeneity on resource usage. Fig. 8 shows the results using SD, MR and SAMR with different heterogeneity. It can be seen that the required number of PMs increases as the heterogeneity increases in SD method but the number of PMs required in MR and SAMR falls with the increase of heterogeneity of the workloads. The reason is that large amounts of resources are wasted in SD, while MR and SAMR are capable to provide balanced utilization of resources. This phenomenon again shows the advantage of heterogeneous resource management
for serving diversified workloads in IaaS clouds

Impacts by delay threshold. Fig. 9(a) shows the results for varying the delay threshold D for Google trace. We use a set of delay threshold (minutes): 15; 30; 60; 90; 120. We can see from the figure that the number of active PMs in each time slot reduces as we allow higher delay threshold. This is because a larger D value permits more requests in the waiting queue for powering up additional PMs, and thus the cloud system is able to serve more VMs with current active PMs. In practice, cloud providers is able to set an appropriate D to achieve a good balance between quality of service and power
consumption.

Impacts by maximum VM capacitiy. In Fig. 9(b), we design an experiment on Google trace where the cloud providers offer different maximum VM capacity. For example, a cloud system with the normalized maximum resource mi offers (log2mi _ 100 + 1) options on resource type-i. We test three maximum resource values 16%; 32%; 64%, respectively. From the figure we can see that with

bigger VMs offered by providers, more PMs are needed to serve the same amount of workloads. The reason is that bigger VMs have higher chance to be delayed when the utilization of resources in the data center is high.

Impacts by time slot length. Fig. 9(c) shows the results for varying slot length from 15 minutes to 120 minutes using Google trace. Our heterogeneous resource management allows cloud providers to specify time slot according to their requirements. As shown in the figure, the number of active PMs can be further optimized with smaller time slots. These results suggest that we can obtain better optimization effect if our proposed prediction model and PM provisioning can be executed more frequently.

7. CONCLUSION

In the Real world, the demand for resources get vary based on the jobs. The existing system may cause the starvation or wastage of the resources. In this paper, this paper first emphasized the need to have a flexible VM offering for VM requests with different resource demands on different resource types. We then implement the heterogeneous resource allocation called skewness-avoidance multi-resource (SAMR) allocation. VM allocation algorithm is used to ensure heterogenous workloads are allocated appropriately to avoid skewed resource utilization in PMs, and a model-based approach to estimate the appropriate number of active PMs to operate SAMR. We evaluate the result of our proposed system through simulation. The development of Markov chain results in low complexity for practical operation and accurate estimation. We compare our result with the single dimensional method and multi-resource method without considering skewness. We found that avoiding the heterogeneity workload there may be a huge loss in the resource allocation. method without skewness consideration. From the Specifically, by conducting simulation studies with three synthetic workloads and one cloud trace from Google, it revealed that our proposed allocation approach that is aware of heterogenous VMs is able to significantly reduce the active PMs in the data center, by 45% and 11% on average compared with single-dimensional and multi-resource schemes, respectively. Through our solution, we can maintain the resource allocation.

REFERENCES

[1] S. Genaud and J. Gossa, "Cost-wait trade-offs in client-side resource provisioning with elastic clouds," in Proc. of 2011 IEEE International Conference on Cloud Computing (CLOUD'10). IEEE, 2011, pp. 1–8.

[2] E. Michon, J. Gossa, S. Genaud et al., "Free elasticity and free cpu power for scientific workloads on iaas clouds." in ICPADS. Citeseer, 2012, pp. 85–92.

[3] P. Marshall, H. Tufo, and K. Keahey, "Provisioning policies for elastic computing environments," in Proc. of 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW). IEEE, 2012, pp. 1085–1094.

[4] L. Wang, J. Zhan, W. Shi, and Y. Liang, "In cloud, can scientific communities benefit from the economies of scale?" IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 2, pp. 296–303, 2012.

[5] R. V. den Bossche, K. Vanmechelen, and J. Broeckhove, "Costoptimal scheduling in hybrid iaas clouds for deadline constrained workloads," in IEEE CLOUD'10, 2010.

[6] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Costand deadline-constrained provisioning for scientific workflow ensembles in iaas clouds," in Proc. of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC'12). IEEE Computer Society Press, 2012, p. 22.

[7] K. Deng, J. Song, K. Ren, and A. Iosup, "Exploring portfolio scheduling for long-term execution of scientific workloads in iaas clouds," in Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis (SC'13). ACM, 2013, p. 55.

[8] Amazon Pricing, https://aws.amazon.com/ec2/pricing/.

[9] Rackspace Cloud Pricing, http://www.rackspace.com/cloud/ servers.

[10]L. A. Barroso and U. H¨ olzle, "The case for energy-proportional computing," IEEE computer, vol. 40, no. 12, pp. 33–37, 2007.

[11] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing

environment," IEEE Transactions on Parallel and Distributed Systems, 2013.
[12] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in INFOCOM' 11, 2011.
[13] Google Inc, http://code.google.com/p/googleclusterdata/.
[14] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in Proceedings of the Third ACM Symposium on Cloud Computing. ACM, 2012.
[15] L. Wei, B. He, and C. H. Foh, "Towards Multi-Resource physical machine provisioning for IaaS clouds," in IEEE ICC 2014 – Selected Areas in Communications Symposium (ICC'14 SAC), 2014.
[16] T. J. Hacker and K. Mahadik, "Flexible resource allocation for reliable virtual cluster computing systems," in Proc. of SC'11, 2011.
[17] D. Villegas, A. Antoniou, S. M. Sadjadi, and A. Iosup, "An analysis of provisioning and allocation policies for infrastructureas- a-service clouds," in Proc. of 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'12). IEEE, 2012, pp. 612–619.
[18] K. Mills, J. Filliben, and C. Dabrowski, "Comparing vm-placement algorithms for on-demand clouds," in Proc. of CLOUDCOM'11, 2011.
[19] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: A survey," in Approximation algorithms for NP-hard problems. PWS Publishing Co., 1996, pp. 46–93.
[20] D. Xie, N. Ding, Y. C. Hu, and R. Kompella, "The only constant is change: incorporating time-varying network reservations in data centers," ACM SIGCOMM Computer Communication Review.
[21] A. Ali-Eldin, M. Kihl, J. Tordsson, and E. Elmroth, "Efficient provisioning of bursty scientific workloads on the cloud using adaptive elasticity control," in Proc. of the 3rd workshop on Scientific Cloud Computing Date. ACM, 2012, pp. 31–40.
[22] S. Niu, J. Zhai, X. Ma, X. Tang, and W. Chen, "Cost-effective cloud hpc resource provisioning by building semi-elastic virtual clusters," in Proc. of International Conference for High Performance Computing, Networking, Storage and Analysis (SC'13). ACM, 2013, p. 56.
[23] J. Li, K. Shuang, S. Su, Q. Huang, P. Xu, X. Cheng, and J. Wang, "Reducing operational costs through consolidation with resource prediction in the cloud," in Proc. of CCGRID'12, 2012.
[24] M. Mao and M. Humphrey, "Auto-scaling to minimize cost and meet application deadlines in cloud workflows," in Proc. of SC'11, 2011.
[25] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services." in Proc. of NSDI'08, 2008.
[26] C. Delimitrou and C. Kozyrakis, "Qos-aware scheduling in heterogeneous datacenters with paragon," ACM Transactions on Computer Systems (TOCS), vol. 31, no. 4, p. 12, 2013.
[27] W. Wang, B. Li, and B. Liang, "Dominant resource fairness in cloud computing systems with heterogeneous servers," in INFOCOM' 14, 2014.