Low SLA violation and Low Energy consumption using VM Consolidation in Green Cloud Data Centers

Hong-fu Chou (Hung-pu Chou)

Research Question(s)

Clearly state the research question(s). The research question should be clear and focused and summarises the issue you will investigate.

Virtual Machines (VM) consolidation is an efficient way towards energy conservation in cloud data centers. The VM consolidation technique is applied to migrate VMs into lesser number of active Physical Machines (PMs), so that the PMs which have no VMs can be turned into sleep state. VM consolidation technique can reduce energy consumption of cloud data centers because of the energy consumption by the PM which is in sleep state. Because of VMs sharing the underlying physical resources, aggressive consolidation of VMs can lead to performance degradation. Furthermore, an application may encounter an unexpected resources requirement which may lead to increased response times or even failures. Before providing cloud services, cloud providers should sign Service Level Agreements (SLA) with customers. To provide reliable Quality of Service (QoS) for cloud providers is quite important of considering this research topic. To strike a tradeoff between energy and performance, minimizing energy consumption on the premise of meeting SLA is considered. One of the optimization challenges is to decide which VMs to migrate, when to migrate, where to migrate, and when and which servers to turn on/off. To achieve this goal optimally, it is important to predict the future host state accurately and make plan for migration of VMs based on the prediction. For example, if a host will be overloaded at next time unit, some VMs should be migrated from the host to keep the host from overloading, and if a host will be underloaded at next time unit, all VMs should be migrated from the host, so that the host can be turned off to save power. The design goal of the controller is to achieve the balance between server energy consumption and application performance. Because of the heterogeneity of cloud resources and various applications in the cloud environment, the workload on hosts is dynamically changing over time. It is essential to develop accurate workload prediction models for effective resource management and allocation. The disadvantage of VM consolidation process in cloud data centers is that they only concentrate on primitive system characteristics such as CPU utilization, memory and the number of active hosts. When originating their models and approaches as the decisive factors, these characteristics ignore the discrepancy in performance-to-power efficiency between heterogeneous infrastructures. Therefore, this is the reason that leads to unreasonable consolidation which may cause redundant number of VM migrations and energy waste. Advance artificial intelligence such as reinforcement learning can learn a management strategy without prior knowledge, which enables us to design a model-free resource allocation control system. For example, VM consolidation could be predicted by using artificial intelligence rather than based on the current resources utilization usage.

Considering these facts, this Ph.D. study aims to answer the following questions:

1. Does the linear or weighted linear regression be able to perform a better VM consolidation to reach a balance between reliable QoS and minimizing energy consumption?
2. How does the reinforcement learning benefit to VM consolidation and how to build up a learning-based framework for performing a prediction of resource utilization?

To achieve the goal optimally, it is important to predict the future host state accurately and make plan for migration of VMs based on the prediction. Therefore, the following sub-questions are proposed in response to the first research question:

a) How to decide efficiently based on energy saving strategy of which VMs to migrate, when to migrate, where to migrate, and when and which servers to turn on/off?
b) How to consider an intelligence VM consolidate policy from observing the relationship between the power consumption by servers and their CPU utilization?

Since the applications of tenants often encounter highly variable workloads, aggressive consolidation of VMs can lead to SLA violations when an application experiences an increasing demand. The decision-making prediction of machine learning improves the prediction accuracy by considering the degree of dispersion of resource utilization and reduces the frequency of VMs' migrations caused by abrupt workload peaks. Therefore, the following sub-questions will be applied:

a) How to achieve efficient resource management in cloud environments by improving the prediction accuracy?
b) How to depict the optimal matching relationship between resource requests and hosts using a better prediction accuracy?

The objective of this Ph.D. research is to develop the energy efficiency of VM consolidation for cloud centers in comparison with other competitive approaches.

Rationale and Significance of the Study

The key reasons for undertaking the research, with a focus on why the study is worth undertaking.

In recent years, the increasing complexity and quick expansion of network application service is urgent to integrate and consolidate the IT infrastructure for the easy centralized monitoring and management. The total ownership cost can be reduced constantly. Cloud computing has recently become one of the most popular topics for research. The customer accesses to provision resources on-demand by using pay-as-you-go model. Cloud service providers (such as Amazon, Google, Microsoft, Alibaba, etc.) are speeding up to establish large-scale data centers all over the world. With the rapid growth of the number of data centers, however, the operation costs increase dramatically due to the increasing energy consumption and environmental pollutants. It has been estimated that by the year 2014 infrastructure and energy expenses contributed about 75%, whereas IT contributed just 25% to the total cost of operating a data center. A recent study [1] shows that data centers expend about 3 percent of the world's total electricity, emitting around 200 million metric tons of carbon dioxide. In the US, data centers are predicted to consume 140 billion kWh per year by 2020, resulting in $78 billion per year in electricity bills. As a result, energy conservation has become one of the hotspots in the field of cloud computing. The dominant reason for the enormous quantities of electrical energy is not just because of the great number of resources consumption in large-scale data centers and low efficiency of power utilization ratio, but more critically, it is due to the inefficient use of these resources [2]. The authors in [3] found that the hosts' utilization is only maintained at 10-50% of their total capacity by data acquisition from more than 5000 hosts over a 6 months period. Moreover, idle hosts consume approximately 70% of their peak power [4]. Hence, maintaining hosts under loaded is unadvisable in terms of energy consumption. As a result, the varying workloads in applications keep the hosts at overloaded status may lead to the QoS requirements of user's application considered as SLA violations. To satisfy the above motivation, VM consolidation is an efficient way towards energy conservation in cloud data centers. This technique can reduce energy consumption of cloud data centers because of the energy by the PM which is in sleep state is significantly lower than the energy consumption by the PM which is in active state [5]. A live migration [6] approach is presented in [7] to guarantee high-level of energy efficient and SLA. The authors propose that one of the optimization challenges is to decide which VMs to migrate, when to migrate, where to migrate, and when and which servers to turn on/off. To achieve this goal optimally, it is important to predict the future host state accurately, and make plan for migration of VMs based on the prediction. For example, if a host will be overloaded at next time unit, some VMs should be migrated from the host to keep the host from overloading, and if a host will be underloaded at next time unit, all VMs should be migrated from the host, so that the host can be turn off to save power. There is a linear relationship between the power consumption by servers and their CPU utilization, even when Dynamic Voltage and Frequency Scaling (DVFS) is applied [8], [9].

Proposed Research Approaches

To this study, I initially focus on predicting the host CPU utilization to determine when a host is overloaded or underloaded by dynamic threshold based heuristics and decision-making approach. A prediction model is proposed to forecast the future CPU utilization and named Robust Simple Linear Regression (RobustSLR) prediction model [22]. The main objective of RobustSLR model is to minimize the power consumption and SLA violation level. The initial result is shown in this study to reveal the improvement of applying linear regression model. The linear regression is a strong statistical method used in machine learning schemes to estimate a prediction function. Different from the native simple linear regression, by adding the error directly or indirectly to the prediction, this research proposes to amend the prediction and squint towards over-prediction. Furthermore, the workload model for prediction of resource usage should be considered using the decision-making approach in this study. It improves the prediction accuracy by considering the degree of dispersion of resource utilization and reduces the frequency of VMs' migrations caused by abrupt workload peaks. The final goal of this study shall explore the optimal matching relationship between resource requests and hosts using decision-making machine-learning technique. This could lead us to provide an optimal solution of the tradeoff between reliable QoS and minimizing energy consumption.

Literature/Past Research Review

The purpose of this section is to identify the gap in literature to support the research direction. This should include a brief account of how the proposed project relates to existing knowledge and literature within the appropriate field.

In recent years, Cloud computing is a pay as you use model which delivers infrastructure (IaaS), platform (PaaS) and software (SaaS) as services to users as per their requirements. It also exposes data centers capabilities as network virtual services, which may include the set of required hardware and application with support of the database. This allows the users to deploy and access applications across the internet which is based on demand and QoS requirements. The rapid growth in demand for computational power driven by modern service applications combined with the shift to the Cloud computing model. This led to the establishment of large-scale virtualized data centers. However, data centers consume electrical energy of high operating costs and carbon dioxide emissions. Dynamic consolidation of VMs utilizing live relocation and exchanging sit out of gear hubs to the rest mode permits Cloud suppliers to optimize asset utilization. In any case, the commitment of giving tall quality of benefit to clients requires the energy-performance trade-off request. Due to the inconstancy of workloads experienced by up-to-date applications, the VM situation ought to be optimized persistently in a web way. To address the energy wastefulness, it may well be degree through the capabilities of the virtualization innovation. The virtualization innovation permits Cloud suppliers to make numerous VMs occurrences on a single physical server, in this way moving forward the utilization of assets and expanding the Return On Investment (ROI). The decrease in vitality utilization can be accomplished by exchanging sit still hubs to low-power modes (i.e. rest, hibernation), in this way dispensing with the sit still control utilization

(Figure1). In addition, the VMs can be powerfully solidified to the negligible number of physical hubs agreeing to their current asset prerequisites. It isn't trifling to realize the objective of productive asset administration in Clouds as present day benefit applications regularly involvement exceedingly variable workloads causing energetic asset utilization designs. In this manner, forceful union of VMs can lead to execution debasement when an application experiences an expanding request coming about in an unforeseen rise of the asset utilization. On the off chance that the asset necessities of an application are not met, the application debase with expanded reaction times, time-outs or failures. Reliable QoS guaranteed by SLAs bolster to set up between Cloud suppliers and their clients. It must be considered that the minimization of vitality utilization whereas assembly the SLAs and cloud suppliers bargain with the energy-performance trade-off.

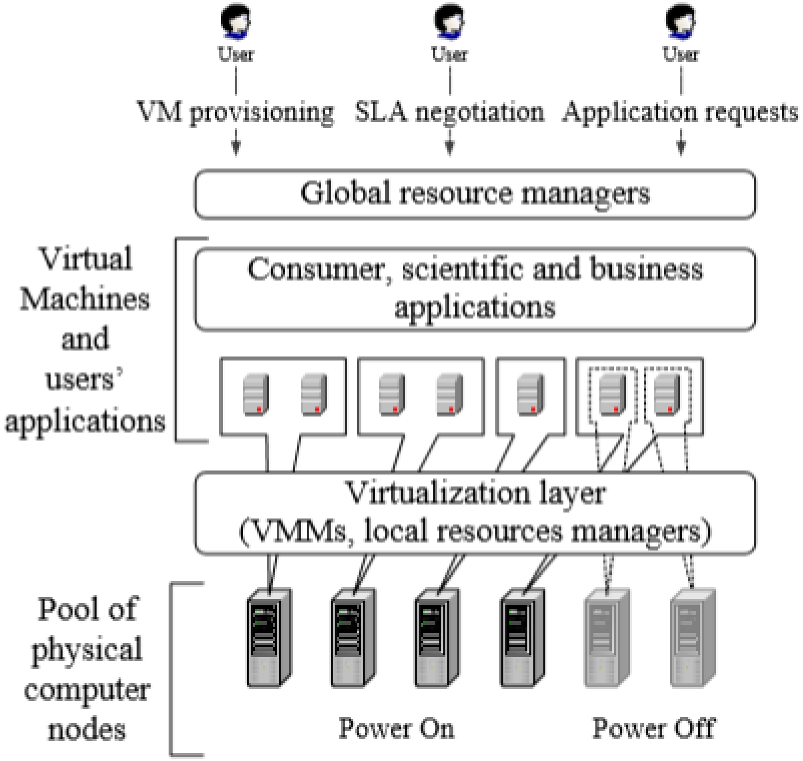

User
User
User
VM provisioning
SLA negotiation
Application requests
Global resource managers
Virtual Machines and users' applications
Consumer, scientific and business applications
Virtualization layer (VMMs, local resources managers)
Pool of physical computer nodes
Power On
Power Off

Fig.1 The System View [22]

There are two primary strategies in information centers for vitality utilization administration: Energetic Voltage and Recurrence Scaling (DVFS) method and VM union method. The DVFS procedure [10]-[16] is proposed to powerfully alter the chip's working recurrence and voltage concurring to the distinctive needs of the application program running on the chip for computing control in arrange to realize the reason of vitality sparing. Interval-based strategies alter the processor recurrence by foreseeing long run CPU utilization, inter-task strategies allot distinctive assignments to diverse speed of processors, and intra-task strategies alter the processor recurrence concurring to the structure of programs. In spite of the fact that DVFS innovation moves forward energy utilization, there is still much room for optimization. The VM combination issue may be a NP-hard issue [17][18]. So VM union issue is regularly defined as an optimization issue with the objective to discover a close ideal arrangement. The existing VM consolidation approaches can be primarily partitioned into two categories: threshold-based heuristics and decision-making based on factual investigation of verifiable information. Threshold-based heuristics set fitting edge to foresee the state of a host by comparing it with the limit, and after that choose the relocation of VMs. The threshold-based heuristics strategy can too be partitioned into two categories: inactive threshold-based heuristics and energetic threshold-based heuristics. The creators in [19] propose a inactive threshold-based heuristics strategy. They set two limits for have state expectation: an over-burden limit and a underloaded limit. A physical machine (PM) is overloaded if its CPU utilization is rise to or more noteworthy than 90%, and after that a few VMs ought to be migrated to dodge overload at next time unit. Something else, a PM is underloaded in case its CPU utilization is break even with or less than 10%, and after that all VMs ought to be moved to spare control. However, the inactive limit strategy isn't reasonable at most of the time, since the workloads in cloud information centers are energetic and complicated. The creators [20] propose two energetic threshold-based heuristics approaches: Median Absolute Deviation (MAD) and Inter Quartile Range (IQR). Frantic could be a degree of measurable scattering; it could be a vigorous degree of the changeability of a univariate test of quantitative information. IQR could be a degree of factual scattering, being break even with to the distinction between 75th and 25th percentiles, or between upper and lower quartiles. The creators alter the limits utilizing over measurable strategies based on the authentic information of CPU utilization. Subsequently, the edges can change with the workloads of has. In any case, the inherent drawbacks of measurable strategies have negative effect on the viability. For case, the past CPU utilizations of a have are utilized for the IQR strategy are same so the upper edge will be set to 100%. But, it clearly is an unacceptable edge. Diverse from the threshold-based heuristics, choice making arrangements don't set edge, they foresee the state of a have utilizing the verifiable information to prepare a few capacities. Neural organize forecast strategy is much more complicated, and is frequently utilized for medium term to long term expectation. Direct relapse is utilized to produce an evaluated future asset utilization of a PM from analyzing its past asset utilization measurements. It is frequently utilized for brief term forecast. In any case, the calculation is basically a ravenous calculation, VMs are regularly solidified forcefully which leads to outlandish union such as over the top VM movement. This issue will be the most thought of this study within the close future.

1. SLA-AWARE AND ENERGY-EFFICIENT VM CONSOLIDATION BASED ON ROBUST SIMPLE LINEAR REGRESSION MODEL

Simple Linear Regression(SLR) [21] could be a factual strategy that permits us to summarize and think about relationship between two ceaseless (quantitative) factors by decision-making approach: One variable, indicated x, is respected as the indicator variable, the other variable, signified y, is respected as the reaction variable. The reason of SLR is finding ‘‘the best fitting line’’ which is called relapse work appeared in (1).

$$\widehat{y_t} = b_0 + b_1 x_t \quad (1)$$

When we use (1) to predict the actual response y_t, the observed response for experimental unit *t*, the residual error can be calculated as following:

$$e_t = y_t - \widehat{y_t} \quad (2)$$

Using the squares criterion method, we should minimize the Q value:

$$Q = \sum_{t=1}^{n} (-\widehat{y_t}\,) \quad (3)$$

The values of b_0 and b_1 can be calculated if the equalities (4) and (5) are met.

$$\frac{\partial Q}{\partial b_0} = -2\sum_{t=1}^{n}(y_t - b_1x_t - b_0) = 0 \text{ (4)}$$

$$\frac{\partial Q}{\partial b_1} = -2\sum_{t=1}^{n}(y_t - b_1x_t - b_0) = 0 \text{ (5)}$$

Then get the least squares estimates for b_0 and b_1:

$$b_0 = \hat{y} - b_1\hat{x} \text{ (6)}$$

$$b_1 = \frac{\sum_{t=1}^{n}(x_t-\hat{x})(y_t-\hat{y})}{\sum_{t=1}^{n}(x_t-\hat{x})^2} \text{ (7)}$$

where $\hat{x}$ and $\hat{y}$ are the means of the x_t and y_t observations, respectively.
There are two assessment criteria for the SLR show forecast mistakes, the wrong negative forecast rate and the wrong positive forecast rate. The untrue negative forecast, which may lead to have over-burden, includes a definitive affect on the execution of has; but the wrong positive expectation, which may lead to VM migration, moreover has a vital impact on the execution of has. So we have to be make tradeoff between the untrue negative expectation rate and the untrue positive expectation rate. An adaptive dynamic strategy is displayed that corrects the expectation and squint towards over-prediction by including the mistake to the expectation.

a) Mean Square Error [22]
Ordinarily, it is indicated as MSE, is appeared in condition (8). We include MSE to the expectation straight forwardly, at that point we get the amendatory forecast which is appeared in equation (9).

$$\mathrm{MSE} = \frac{\sum_{t=1}^{n}(y_t-\hat{y})^2}{n-2} \text{ (8)}$$

$$\widehat{y_{n+1}} = b_0 + b_1x_{n+1} + \mathrm{MSE} \text{ (9)}$$

b) Root Mean Square Error [22]
Regularly, it is indicated as RMSE, is appeared in (10). We include RMSE to the forecast straight forwardly, at that point we get the amendatory forecast which is appeared in (11).

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \text{ (10)}$$

$$\widehat{y_{n+1}} = b_0 + b_1x_{n+1} + \mathrm{RMSE} \text{ (11)}$$

c) Standard error o f the slope estimate [22]
Typically, it is shown as se(b_1),is shown in (12). It presents adding se(b_1) to b_1, then get the amendatory prediction which is presented in (13).

$$\mathrm{se}(b_1) = \frac{\mathrm{RMSE}}{\sqrt{\sum(x_t-\hat{x})^2}} \text{ (12)}$$

$$\widehat{y_{n+1}} = b_0 + (b_1 + \mathrm{se}(b_1))x_{n+1} \text{ (13)}$$

Robust Simple Linear Regression algorithm [22] is shown as followings. It presents how to forcast the host CPU utilization according to the history of past CPU utilization.

Algorithm 1 RobustSimpleLinearRegression(RobustSLR)

```
Input: The set of utilization history, utilizationHistory;
Output: The prediction of utilization, prediction;
1: n=utilizationHistory.lenth;
2: for i=0 to n−1 do
3:     x[i]=i+1;
4:     y[i]=utilizationHistory[i];
5: end for
6: $\hat{x} = \sum x_i / n$;
7: $\hat{y} = \sum y_i / n$;
8: calculate $b_0$ and $b_1$ according to (6) and (7);
9: if use MSE to revise then
10:     calculate MSE according to (8);
11:     calculate prediction according to (9);
12: else if use RMSE to revise then
13:     calculate RMSE according to (10);
14:     calculate prediction according to (11);
15: else if use se($b_1$) to revise then
16:     calculate se($b_1$) according to (12);
17:     calculate prediction according to (13);
18: else
19:     we should not be here, report some error;
20: end if
21: return prediction;
```

Initially, the host utilization history to compute b_0 and b_1 is shown according to (6) and (7)(line 1-8). Then, the prediction error and the amendatory prediction are computed by the adaptive dynamic methods (line 9-17). Finally, the prediction of host CPU utilization for the next time unit is returned for detecting whether the host is overloaded (line 21).

2. HOST DETECTION

The VM live migration problem can be divided into four parts: (1) Host detection: detecting when a host is overloading or detecting when a host is underloading; (2) VM selection: selecting some VMs from the overloaded hosts and all VMs from the underloaded hosts for live migration; and (3) VM placement: making a placement plan for all VMs which have been chosen from the overloaded and underloaded hosts.

Host Overloading Detection:

To describe a Host Overloading Detection (HOD) algorithm[22], at first, a have is considered over-burden in case the current CPU utilization is more prominent than 90% of the have assets in the event that there are not sufficient data(at slightest 10 CPU utilization time unit) to be computed for the RobustSLR demonstrate. At that point, the expectation is calculated concurring to the RobustSLR calculation. At last, the HOD calculation is returned agreeing to the forecast whether the have is over-burden at the another time unit with respect to as the prescient CPU utilization is larger than 1 (100%).

Host underloading detection:

To illustrate the host underloading detection (HUD)[22], a host is regarded underloaded if the present day CPU utilization is much less than 10% of the have property in the tournament that there are now

not ample facts (less than 10 CPU utilization time unit) to be computed for the RobustSLR show. At that point, a decrease side for CPU utilization is calculated agreeing to (14), and the expectation is calculated concurring to the RobustSLR show. At lengthy last, the HUD calculation is lower back agreeing to the expectation whether or not the have is underloaded at the following time unit with recognize to as the prescient CPU utilization is decrease than Tl. An versatile decrease utilization aspect is displayed based totally on a full of life size strategy, the interquartile run (IQR) method [22]. It is moreover referred to as the middle fifty with appreciate to as a diploma of factual scattering and being damage even with to the distinction between the 0.33 and first quartiles: IQR=Q3−Q1. Not at all like the (overall) extend, the interquartile run may additionally be a robust measurement, having a breakdown factor of 25%, and is in this way regularly favoured to the standard run. This IQR approach can be utilized to calculate a decrease threshold for CPU utilization.
A lower threshold is set using the IQR method:

$$\mathrm{T_l} = 0.4(1 - \mathrm{s} \times \mathrm{IQR}) \text{ (14)}$$

where s is a safety parameter for VM consolidate. The higher s is corresponding to the lower level of SLA violations, but the more the energy consumption, and vice versa.

3. VM SELECTION

Once it has been chosen that a have is overloaded, the following step is to choose VMs to migrate from this have. The depicted approaches are connected iteratively. After a determination of a VM to migrate, the have is checked once more for being over-burden. On the off chance that it is still considered as being overloaded, the VM selection arrangement is connected again to choose another VM emigrate from the host. This can be rehashed until the host is considered as being not overloaded. One or more VMs ought to be chosen from the overloaded host, and all the VMs ought to be chosen to migrate from the underloaded host. The Minimum Migration Time (MMT) arrangement [22] is recommended that's broadly utilized for VM choice in this study. The MMT approach chooses a VM which has the least relocation time from source have to target host. From the formulation of MMT, the VM which takes the least memory will be chosen. The overloaded have applies the MMT policy iteratively until it isn't overloaded. The MMT policy migrates a VM *v* that costs the minimum time migrate to the other VMs. The migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host ***j.*** Let V_j be a set of VMs currently allocated to the host *j*. The MMT policy finds a VM *v* that satisfies conditions formalized in equation (15)

$$v \in V_j | \forall a \in V_j, \frac{RAM_u(v)}{NET_j} \leq \frac{RAM_u(a)}{NET_j} \text{ (15)}$$

where $RAM_u(a)$ is the amount of RAM currently utilized by the VM *a*; and NET_j is the spare network bandwidth available for the host *j*.

4. VM PLACEMENT

The VM placement can be seen as a bin packing issue with variable container sizes and costs, where bins represent the physical nodes; items are the VMs that need to be allocated; bin sizes are the accessible CPU capacities of the nodes; and costs compare to the control utilization by the hubs. As the bin packing issue is NP-hard to illuminate it, a modification is applied of the Best Fit Decreasing(BFD) calculation [22] that's appeared to utilize no more than 11/9•OPT + 1bins (where Pick is the number of bins given by the ideal arrangement). The Robust Power Aware Best Fit Decreasing (PABFD) algorithm for VM placement is presented in Algorithm 2 that the algorithm searches all the VMs in decreasing order according to their CPU utilizations first(line 1 to line 4). The determination of the host detection is chosen as HOD or HUD is presented in line 6 and line 9. After that each VM is relocated to such a host which has the least power increasing after the VM is migrated to the host(line 12 to line 15). The algorithm is essentially a greedy algorithm, so VMs are often consolidated aggressively.

Algorithm 2 RobustSLR Power Aware Best Fit Decreasing (RPABFD) [28]

```
Input: hostList,vmList;
Output: allocation of VMs;
1: vmList.sortDecreasingUtilization();
2: for vmList.contain(vm) is true do
3:     min=MAX;
4:     alloctedHost =NULL;
5:     for hostList.contain(host) is true do
6:         if HOD(host) is true then
7:             continue;
8:         end if
9:         if HUD(host) is true then
10:            continue;
11:        end if
12:        if host can allocate resources for vm then
13:            power =getPower(host,vm);
14:            if power < min then
15:                alloctedHost =host;
16:                min=power;
17:            end if
18:        end if
19:    if alloctedHost is not NULL then
20:        allocation.add(vm,allocatedHost);
21:    end if
22:    end for
23: end for
24: return allocation;
```

5. POWER MODEL

In cloud data centers, the power consumption of computing nodes is primarily decided by the CPU, memory, disk capacity and organize interfacing. One of the foremost common vitality utilization models is the linear correlation between control utilization and CPU utilization. However, an idle server consumes 70% of the full-load power [23].

In the target system, the energy consumption of the *i*th server at time t is defined as:

$$\mathrm{EC_i(t)} = \begin{cases} \rho \times \mathrm{EC_i^{full}} \text{ the } \mathrm{H_i} \text{ is idle} \\ \mathrm{EC_i^{idle}} + (1-\rho) \times \mathrm{EC_i^{full}} \times \mathrm{U_i(t)} \text{ the } \mathrm{H_i} \text{ is busy} \end{cases} \quad (16)$$

where $\mathrm{H_i}$ is the *i*th server in the data center and $\mathrm{EC_i^{full}}$ and $\mathrm{EC_i^{idle}}$ are the maximum energy consumption when the server is fully utilized and idle respectively. The ρ is the fraction of energy consumed by a idle server.

When the server is idle, ρ is a static coefficients representing the energy ratio of the idle processor (i.e., 70%). The $\mathrm{EC_i^{full}}$ is presented the energy consumption of the physical node *i* under full-loading. Since the CPU utilization dynamically changes according to the workload, the CPU utilization is a function of the time t as an independent variable, which is denoted as U(t). The energy consumption of server $\mathrm{H_i}$ regarding the process can be presented as

$$\mathrm{EC_i} = \int_{\mathrm{t_0}}^{\mathrm{t_1}} \mathrm{EC_i(t)dt} \ (17)$$

Then, the total energy consumption of a cloud data center with n nodes is

$$\mathrm{EC} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{x_i EC_i} \ (18)$$

where

$$\mathrm{x_i} = \begin{cases} 0, \text{the } \mathrm{H_i} \text{ is shutdown} \\ \qquad 1, \text{other} \end{cases}$$

6. INTRODUCTION OF CLOUDSIM SIMULATION TOOLKIT

As Cloud computing could be a new concept and is still in an awfully early organize of its advancement, analysts and framework designers are working on moving forward the innovation to provide way better on preparing, quality & taken a toll parameters. But most of the investigate is centered on moving forward the execution of provisioning arrangements and to test such inquire about on genuine cloud environment like Amazon EC2, Microsoft Purplish blue, Google App Motor for distinctive applications models beneath variable conditions is greatly challenging as: 1).Clouds display changing requests, supply designs, framework sizes, and assets (equipment, program, and network). 2).Users have heterogeneous, energetic, and competing QoS requirements. 3).Applications have changing execution, workload, and energetic application scaling requirements. Benchmarking the application execution on the genuine open cloud foundation like google cloud, Microsoft Purplish blue, etc., are not appropriate due to their multi-tenant nature as well as variable workload fulfillment. Besides, there are exceptionally few admin level arrangements that a user/researcher can be able to alter. Thus, this leads the propagation of comes about that can be depended upon as an greatly troublesome undertaking. It is troublesome on real-world open Cloud frameworks to attempt benchmarking experimentations as repeatable, tried and true, and adaptable situations. The need of the recreation tool(s) emerges, which may gotten to be a practical elective to evaluate/benchmark the test workloads in a controlled and completely configurable environment that can repeatable over numerous emphases and duplicate the comes about for analysis.

This simulation-based approach can give different benefits over the researcher's community because it permits them to: 1) Test administrations in a repeatable and controllable environment. 2) Tuning the framework bottlenecks (execution issues) some time recently deploying on genuine clouds. 3) Simulating the desired foundation (little or huge scale) to assess diverse sets of workloads as well as asset execution. Requiring the capacity to creating, testing and arrangement of versatile application provisioning techniques. To simulate the working of unused administrations on a few test systems, CloudSim Recreation Toolkit is the more generalized and compelling test system for testing Cloud computing-related speculation. CloudSim is an extensible recreation system created by a group of researchers(under the direction of Dr. Raj Kumar Buyya) at Cloud Computing and Disseminated Frameworks (CLOUDS) Research facility, College of Melbourne. This toolkit permits consistent modeling, recreation, and experimentation related to cloud-based foundations and application administrations which are the discharged adaptations distributed on Cloudsim's GitHub venture page. This reenactment toolkit permits the analysts as well as cloud designers to test the execution of the potential cloud application for execution testing in a controlled. And it too permits fine tuning the generally benefit execution indeed some time recently it is conveyed within the generation environment. The most highlights of Cloudsim are organized as follows. 1) A self-contained stage for modeling Clouds, benefit brokers, provisioning, and allotment policies. 2) Facilitates the recreation of arrange associations over the recreated framework elements. 3) Facility for reenactment of unified Cloud environment that inter-networks assets from both private and open spaces, a highlight basic for investigate considers related to Cloudbursts and programmed application scaling. 4) Availability of a virtualization motor that encourages the creation and administration of different, autonomous, and co-hosted virtualized administrations on a information center node. 5) Flexibility to switch between space-shared and time-shared assignment of preparing centers to virtualized services.

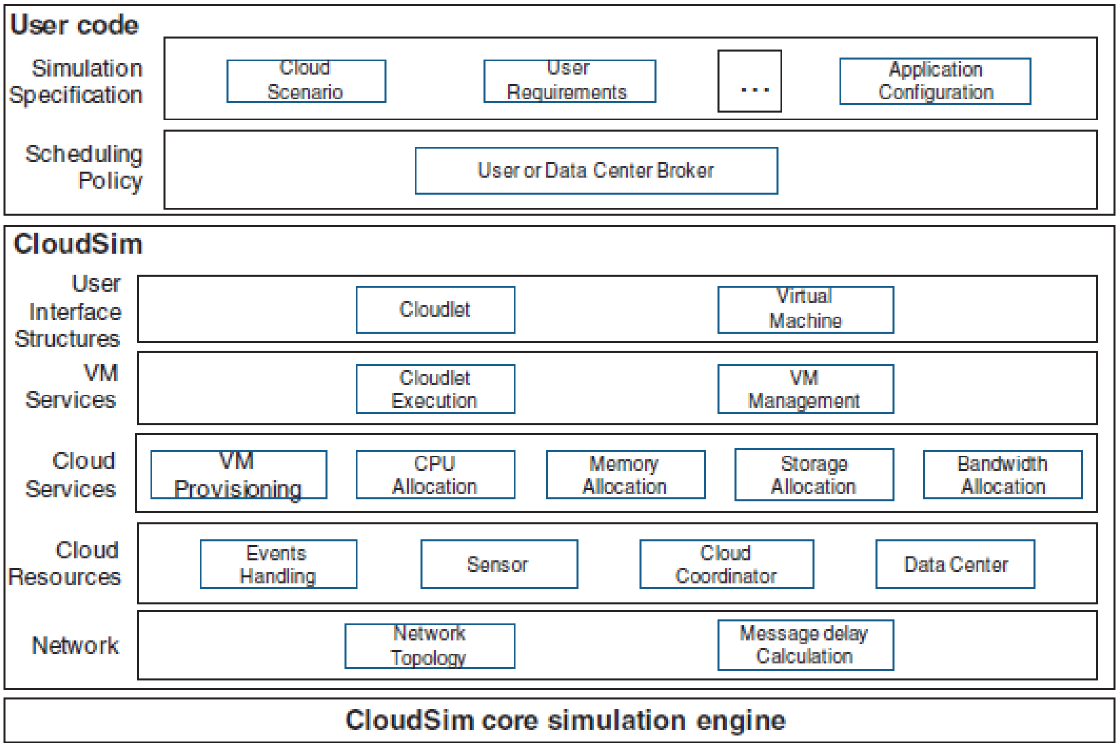


Fig2. Cloudsim Architecture [26]

Fig.2 illustrates the layered design of CloudSim Reenactment Toolkit. The CloudSim Center recreation motor gives back for modeling and reenactment of virtualized Cloud-based information center situations counting lining and preparing of occasions, creation of cloud framework substances (like information center, have, virtual machines, brokers, administrations, etc.) communication between components and administration of the recreation clock. The CloudSim layer gives committed administration interfacing for Virtual Machines, memory, capacity, and transfer speed. Too, it oversees the other principal issues, such as provisioning of has to Virtual Machines, overseeing application execution, and checking energetic framework state(e.g. Arrange topology, sensors, capacity characteristics, etc), etc. The Client Code layer may be a custom layer where the client composes their claim code to rethink the characteristics of the invigorating environment as per their modern investigate findings.

7. LEARNING-BASED METHODOLOGY

The plan intention of the controller is to obtain the stability between server power consumption and software performance. As the decision-making method goes through RobustSLR method, this approach pursues the ultimate purpose via the linear regression approach. Several adaptive heuristics has introduced to determine VMs' migration destinations, however the problem is the solely use of the modern CPU utilization as the most important criterion to determine VMs' migration destinations. Hence, it can lead to growing the quantity of VMs migration and quantity of statistics transmission in the consolidation process. Regarding to learning-based methodology, reinforcement studying (RL) [25] can research a administration method besides prior knowledge, which can layout a model-free useful resource allocation manage system. This methodology should assist to enhance the aim of this study. The short introduction must be introduced as follows. RL is an place of laptop mastering involved with how application experts have to be take things to do in an surroundings in organize to maximize the concept of combination compensate. RL is one of three vital computing device gaining knowledge of standards, close by directed getting to know and unsupervised learning. Reinforcement learning, due to its simplification, is examined in severa different disciplines, such as diversion hypothesis, manipulate hypothesis, operations investigate, statistics hypothesis, simulation-based optimization, multi-agent frameworks, swarm insights, measurements and hereditary calculations. Within the previous investigate, reinforcement gaining knowledge of is known as inexact full of life programming, or neuro-dynamic programming. The problems of intrigued in reinforcement studying have too been viewed inside the speculation of perfect control. This is regularly worried in the main with the presence and characterization of perfect arrangements. Based on their right computation, and much less with getting to know or guess, RL is specifically well-fitted inside the nonattendance of a numerical exhibit of the environment. RL may also be utilized to make clear how stability may additionally emerge underneath bounded soundness in monetary things and diversion hypothesis.

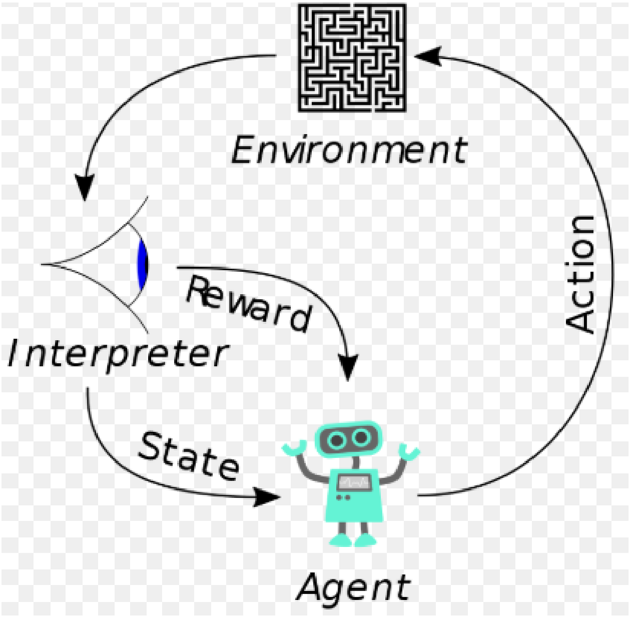


Fig.3 The typical framing of a RL scenario [27]

Fig.3 appears an operator takes activities in an environment. Usually translated into a compensate and a representation of the state which are encouraged back into the specialist. RL requires that the reason and objective of the framework ought to be formalized in terms of the compensate flag to be accomplished. Reward is an real scalar esteem which shows an operator getting from the environment when it takes off current state at time t. In spite of the fact that the objective of an specialist is to maximize the rewards it receives, it endeavors to maximize long-term rewards instead of quick rewards. Hence, it’s completely vital to set the compensate flag that clearly demonstrates the objective.

Design of the Study

An outline of the research design. This includes methodology, methods and analysis.

Following the thread of RobustSLR and linear regression, the dynamic VM consolidation could be extended to consider the **Loess method** (from the German l¨oss – short for local regression) proposed by [24]. The main idea of the method such as local regression is that fitting simple models and localization of data subsets approximate the original data together. The observations $(\hat{x}, \hat{y})$ are assigned neighborhood weights using the tricube weight function shown in (18).

$$T(u) = \begin{cases} (1 - |u|^3)^3 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases} \text{ (18)}$$

The neighbourhood weight for the observation $(\hat{x}, \hat{y})$ is defined by the function in (19)

$$w_i(x) = T\left(\frac{\Delta_i(x)}{\Delta_q(x)}\right), \Delta_i(x) = |x_i - x| \text{ (19)}$$

q is the number of observations in the subset of data localized around x. The size of the subset is defined by a parameter of the method called the bandwidth. As in (1), the line is fitted to the data using the weighted least squares method with weight $w_i(x)$ at $(\hat{x}_i, \hat{y_i})$ by minimizing the function shown in (20).

$$\sum_{t=1}^{n} w_t(x)(y_t - b_0 + b_1 x_t)^2 \text{ (20)}$$

According to the Algorithm 1, the weighted Loess method could be considered as minimizing the function in (20). This could be denoted as WLRMSE. The simulation of this approach compares the performance with the other prediction models.

Next, the decision-making prediction of artificial intelligence improves the prediction accuracy by considering the degree of dispersion of resource utilization and reduces the frequency of VMs' migrations caused by abrupt workload peaks. RL is a good choice to study further. The design goal of the controller is to achieve the balance between server energy consumption and application performance. Reinforcement learning can learn a management strategy without prior knowledge, which enables us to design a model-free resource allocation control system. In reinforcement learning, I will consider the need to define the state that reflect the current state of the agent; manage the number of actions, and the reward function that represents the effect of executing actions in the state. The output of the reward function is used to take more accurate actions in the next state observation. By means of the advance machine learning, the optimal matching relationship could be explored between the workload to be allocated and the host to the available resource. To further achieve the goal, this study will pursue the path that the host efficiency ratio after allocation is the highest and the probability of SLA violation is the lowest.

References

A list of sources referred to in the proposal.

[1] Z. Asad and M. A. Rehman Chaudhry, ``A two-way street: Green big data processing for a greener smart grid,'' *IEEE Syst. J.*, vol. 11, no. 2, pp. 784_795, Jun. 2017.

[2] L. Salimian, F. Sa_ Esfahani, and M.-H. Nadimi-Shahraki, ``An adaptive fuzzy threshold-based approach for energy and performance ef_cient consolidation of virtual machines,'' *Computing*, vol. 98, no. 6, pp. 641_660,
Jun. 2016.
[3] L. A. Barroso and U. Hölzle, ``The case for energy-proportional computing,'' *Computer*, vol. 40, no. 12, pp. 33_37, Dec. 2007.
[4] X. Fan, W. D. Weber, and L. A. Barroso, ``Power provisioning for a warehouse-sized computer,'' *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, pp. 13_23, 2007.
[5] M. D. de Assunção, J.-P. Gelas, L. Lefèvre, and A.-C. Orgerie, ``The green Grid'5000: Instrumenting and using a grid with energy sensors,'' in *Proc. 5th Int. Workshop Distrib. Cooper. Lab., Instrum. Grid (INGRID)*, Poznan, Poland, 2010, pp. 25_42.
[6] C. Clark *et al.*, ``Live migration of virtual machines,'' in *Proc. 2nd Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, 2005,pp. 273_286.
[7] L. Salimian, F. Safi Esfahani, and M.-H. Nadimi-Shahraki, ''An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines,'' Computing, vol. 98, no. 6, pp. 641–660, Jun. 2016.
[8] X. Fan, W.-D. Weber, and L.-A. Barroso, ``Power provisioning for a warehouse-sized computer,'' in *Proc. 34th Annu. Int. Symp. Comput. Archit. (ISCA)*, New York, NY, USA, 2007, pp. 13_23.
[9] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, ``Power and performance management of virtualized computing environments via lookahead control,'' *Cluster Comput.*, vol. 12, no. 1, pp. 1_15,
Mar. 2009.
[10] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, ''A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters,'' Future Generat. Comput. Syst., vol. 37, pp. 141–147, Jul. 2014.
[11] A. Wierman, L. L. H. Andrew, and A. Tang, ''Power-aware speed scaling in processor sharing systems,'' in Proc. 28th Conf. Comput. Commun. (INFOCOM), Apr. 2009, pp. 2007–2015.
[12] L. L. H. Andrew, M. Lin, and A. Wierman, ''Optimality, fairness, and robustness in speed scaling designs,'' in Proc. ACM Int. Conf. Meas. Modeling Comput. Syst. (SIGMETRICS), 2010, pp. 37–48.
[13] K. Flautner, S. Reinhardt, and T. Mudge, ''Automatic performance setting for dynamic voltage scaling,'' Wireless Netw., vol. 8, no. 5, pp. 507–520, 2002.
[14] A. Weissel and F. Bellosa, ''Process cruise control-event-driven clock scaling for dynamic power management,'' in Proc. Int. Conf. Compil., Archit., Synth. Embedded Syst., 2002, pp. 238–246.
[15] S. Lee and T. Sakurai, ''Run-time voltage hopping for low-power realtime systems,'' Proc. 37th Annu. ACM/IEEE Design Autom. Conf. (DAC), Jun. 2000, pp. 806–809.
[16] J. R. Lorch and A. J. Smith, ''Improving dynamic voltage scaling algorithms with PACE,''
ACMSIGMETRICS Perform. Eval. Rev.,vol.29,no.1, pp. 50–61, Jun. 2001.
[17] E. Feller, C. Morin, and A. Esnault, ''A case for fully decentralized dynamic VM consolidation in clouds,''in Proc.IEEE4th Int.Conf.Cloud Comput. Technol. Sci., Dec. 2012, pp. 26–33.
[18] T.Wood, P.Shenoy, A.Venkataramani, and M.Yousif,''Sandpiper:Blackbox and gray-box resource management for virtual machines,'' Comput. Netw., vol. 53, no. 17, pp. 2923–2938, Dec. 2009.
[19] F. Ahamed, S. Shahrestani, and B. Javadi, ''Security aware and energyefficient virtual machine consolidation in cloud computing systems,'' in Proc. IEEE Trustcom/BigDataSE/ISPA, Aug. 2016, pp. 1516–1523.
[20] A. Beloglazov, J. Abawajy, and R. Buyya, ''Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,''FutureGenerat.Comput.Syst.,vol.28,no.5,pp.755–768,2012.
[21] STAT501: Simple Linear Regression. Accessed:2018.[Online]. Available: https://onlinecourses.science.psu.edu/stat501/node/250

[22] A. Beloglazov and R. Buyya, ‘‘Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers,’’ Concurrency Comput., Pract. Exper., vol. 24, no. 13, pp. 1397–1420, 2012.

[23]R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, ‘‘No power struggles: Coordinated multi-level power management for the data center,’’ Solutions, vol. 36, no. 1, pp. 48–59, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1346289

[24] Cleveland WS. Robust locally weighted regression and smoothing scatterplots. Journal of the American statistical association 1979;74(368):829–836.

[25] Kaelbling, Leslie P.*;* Littman, Michael L.*;* Moore, Andrew W. *(1996).* "Reinforcement Learning: A Survey"*. Journal of Artificial Intelligence Research. 4: 237–285.* arXiv*:*cs/9605103*.* doi*:*10.1613/jair.301*. Archived from* the original *on 2001-11-20.*

[26] Online source https://www.cloudsimtutorials.online/cloudsim-simulation-toolkit-an-introduction/

[27] *Online source* https://en.wikipedia.org/wiki/Reinforcement_learning

[28] Lianpeng Li, Jian Dong, Decheng Zuo” SLA-Aware and Energy-Efficient VM Consolidation in Cloud Data Centers Using Robust Linear Regression Prediction Model”, IEEE Access, Vol. 7, pp9490 – 9500, 2019

Ethical Approval

If ethics approval is required, please attach a copy of the letter(s) of approval. If not yet granted, this must be indicated, along with details of which committee(s) will be approached. A copy of approval letter(s) must be submitted to the Postgraduate Office within 6 months of the confirmation of candidature.

N/A

Resources and Budget

Identify any additional resources you will need over and above those provided by AUT. Provide a planned budget for your research and how this will be funded.

Software named Spirent Temeva requires budget NZD$ 5000 for SaaS to a collection of applications and measure the performance and scale of networks and clouds. (Spirent Temeva (Test. Measure. Validate.) is a Software-as-a-Service (SaaS) portal providing convenient access to the most up-to-date network and cloud testing solutions).

Location

Indicate where the research will be conducted, and if not at AUT, provide a brief justification and indicate how communication and resource issues will be addressed.

AUT city campus.

Progress and Activity to Date

Specifically address the past 6 months and any issues raised in the previous Progress Report.

a) EXPERIMENT SETUP

I have simulated a data center that comprises 400 HP ProLiant ML110 G4 servers and 400 HP ProLiant ML110 G5 servers using the CloudSim toolkit. The configuration of hosts are given in Table 1. The power consumption characteristics of the servers are shown in Table 2, and the characteristics of the VMs are listed in Table 3.

TABLE 1. Configuration of hosts.

Hosts	CPU type	Freq(GHz)	Core	RAM(GB)
HP ProLiant ML110 G4	Intel Xeon 3040	1.86	2	4
HP ProLiant ML110 G5	Intel Xeon 3075	2.86	2	4

TABLE 2. Power consumption by the selected servers at different load levels in Watts.

Server	Sleep	0%	20%	40%	60%	80%	100%
HP ProLiant G4	10	86	92.6	99.5	106	112	117
HP ProLiant G5	10	93.7	101	110	121	129	135

TABLE 3. Four kinds of VM types

VM type	CPU(MIPS)	RAM(GB)
High CPU medium instance	2500	0.85
Large instance	2000	1.7
Small instance	1000	1.7
Micro instance	500	0.61

TABLE 4. Workload data characteristics(CPU utilization).

Date	Hosts	VMs	Mean	St.dev
03/03/2011	800	1052	12.31%	17.09%
random	50	50	N/A	N/A

The evaluation is presented as the prediction Model using random workload and a real world workload:

- RandomWorkload: The users submit requests for provisioning of 50 heterogeneous VMs to the 50 hosts. Each application has 300 bytes input and 300 bytes output. Each VM runs the application and the CPU utilizations are generated according to a random variable. It has been run the simulation experiment for 24 hours.
- Real Workload (PlanetLab data): PlanetLab is the monitoring part of the CoMon project. It collected the CPU utilization data every 5 minutes from thousands of servers located at more than 500 places around the world.

b) PERFORMANCE METRICS

Several metrics has been used to evaluate the performance of the algorithms. The main metrics are Energy Consumption by physical nodes and SLA Violation.

- Energy Consumption: The mode of energy consumption of the servers in this study is shown in Table2. The energy consumption of the server in sleep state is much less than the server in active state.
- SLA Violation: While a cloud provider is unable to provide service to customers corresponding to service level agreement, a SLA violation (SLAV) will happen. SLAV [23] is an independent metric that can be measured by SLA violation time per active host (SLATAH) and performance degradation due to

migration(PDM). The two metrics are unrelated and have the identical effect on SLAV. The SLAV metric can be computed as following[22]:

$$SLAV = SLATAH \times PDM \quad (21)$$

- SLA violation time per active host (SLATAH): While a host is experiencing the 100% utilization, it cannot provide service, so SLATAH can be computed as follows:

$$\mathrm{SLATAH} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{oi}}{T_{ai}} \quad (22)$$

where N is the number of hosts. T_{oi} is the total time during which the host i is experiencing the 100% utilization; the total time of the host i which in active state is T_{ai}

- Performance degradation due to migration (PDM): Live migration of VMs has negative effect on application performance. The PDM can be computed as follows:

$$\mathrm{PDM} = \frac{1}{N}\sum_{i=1}^{N}\frac{C_{di}}{C_{ri}} \quad (23)$$

where N is the number of VMs; C_{di} is the performance degradation of the VM i due to migrations, we set it as 10% of the CPU utilization. C_{ri} is the total CPU capacity requested by the VM i.

- Average SLA violation: The metric can be calculated as follows:

$$\text{Average SLAV}=\frac{\sum_{k=1}^{N}(\text{requested MIPS})-\sum_{k=1}^{N}(\text{allocated MIPS})}{N}$$

where N is the number of VMs.

- Overall SLA violation: The metric can be calculated as follows:

$$\text{Overall SLAV} =\frac{\sum_{k=1}^{N}(\text{requested MIPS})-\sum_{k=1}^{N}(\text{allocated MIPS})}{\sum_{k=1}^{N}(\text{allocated MIPS})}$$

where N is the number of VMs.

- Number of VM migrations: The metric can be calculated as follows:

$$\mathrm{Migrations}(F, t_1, t_2) = \sum_{i=1}^{N}\int_{t_1}^{t_2}\mathrm{Mig}_i(F) \quad (24)$$

where N is the number of hosts, F is the placement of VMs at host *i*, Migi(F) is the number of migrations of host *i* from time t_1 to time t_2.

- Number of host shutdowns: The metric can be calculated as follows:

$$H = \frac{1}{n}\sum_{i=1}^{n}h_i \quad (25)$$

where h_i is the number of active hosts at the time i. H is the number of host shutdowns from time 1 to time n.

c) COMPARISON WITH OTHER BENCHMARKS

‘’20110303’’ data set is selected using the PlanetLab as the workload. The notation of RobustSLR algorithms with the appropriate safety parameter s are defined as follows: type of consolidation approach-VM selection-safety parameter s; MSE-MMT-1.2, RMSE-MMT-1.1 and, se(b1)-MMT-1.25 are chosen for comparison. The comparison benchmarks are NPA (NonePowerAware)[8],DVFS [9], dynamic threshold based heuristics algorithms such as THR-MMT-1.0 [22], THR-MMT-0.8 [22], IQR-MMT-1.5 [22], MAD-MMT-2.5 [22], and decision-making algorithms such as LR-MMT-1.2 [22], and LRR-MMT1.2 [22]. The hosts which use the NPA policy consume their maximum power all the time. The THR-MMT-1.0 algorithm uses the fixed threshold of 100%. Table 5 show the details of experimental results, and Figs. 4-6 illustrate the Energy consumption, SLA violations, number of VM migrations for the main algorithms respectively. From the simulation results, it could be summarized to the following conclusions: (1) VM consolidation technique significantly has better performance than NPA and DVFS; (2) Owing to reducing the level of SLA violations, dynamic threshold-based heuristics algorithms perform better than the static threshold-based heuristics algorithm(THR-MMT-1.0). (3) VM consolidation policies including dynamic threshold based heuristics algorithms and decision-making algorithms can reduce the metric of Energy consumption by at most 23.43%, average 15.10% and metric of SLAV by at most 99.16%, average 97.27% ,and the metric of number of VM migrations by at most 92.31%, average 85.91%. The best trade-off consolidation is rmse_mmt_1.1 which has the result of power consumption 154.99 KW and SLA 98*10^{-7}. The conclusion is that decision-making algorithms outperform dynamic threshold based algorithm and obtain a better trade-off result. The reason is that the number of VM migration is reduced by decision-making approach which alleviates inappropriate consolidation. Also noted that the proposed approach wlrmse_mmt has the same trend of decision-making approach, but having worse performance comparing to the best one of mse_mmt. The reason of the worth performance is that the prediction model of wlrmse_mmt is too aggressive and overfitting to migrate VMs as the number of VM migration is greater than the others. The second reason is the suggest weighting is not appropriate to this end. The ongoing work should be investigated for the optimal weighting of the WLRMSE. For example, the better choice of the weighting should be considered as decision making by certain policy.

Table 5 Simulation results of the best algorithm combinations and benchmark algorithms.

	Energy (KW)	SLA 10^{-7}	SLATAH	Overall SLAV	Average SLAV	# of VM migration
npa	2410.8	0	0	0	0	0
dvfs	803.91	0	0	0	0	0
thr_mmt_1	163.23	2807	15.81%	0.54%	9.10%	45517
thr_mmt_0.8	191.73	324	4.95%	0.07%	10.14%	26634
iqr_mmt_1.5	188.81	303	4.87%	0.07%	9.98%	26476
mad_mmt_2.5	184.88	331	5.03%	0.08%	10.18%	26292
lr_mmt_1.2	163.15	463	5.84%	0.14%	9.60%	27632
lrr_mmt_1.2	163.15	463	5.84%	0.14%	9.60%	27632
mse_mmt_1.2	154.97	116	3.53%	0.12%	9.06%	11921
rmse_mmt_1.1	154.99	98	3.31%	0.09%	9.12%	11127
se_mmt_1.25	164.2	80	2.62%	0.04%	9.73%	11657
wlrmse_mmt_1.1	171.93	173	3.67%	0.10%	9.59%	17437

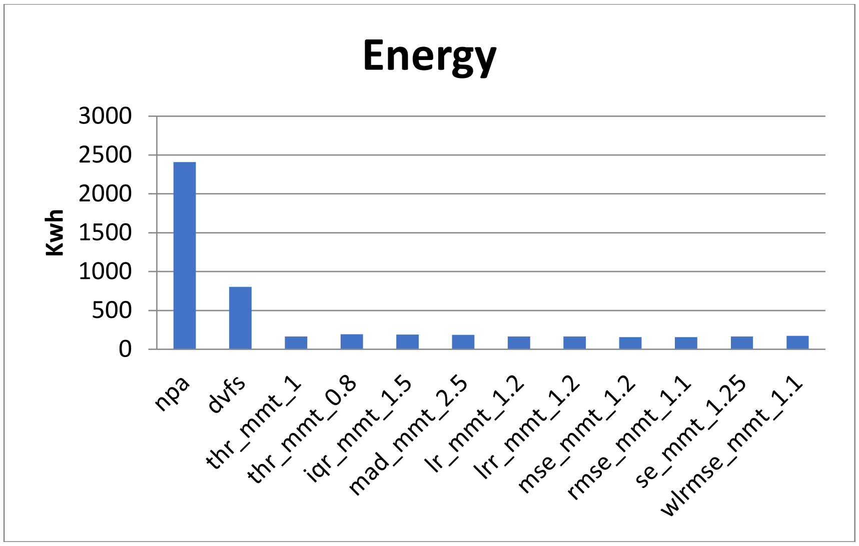


Fig.4 Energy consumption for the main algorithms.

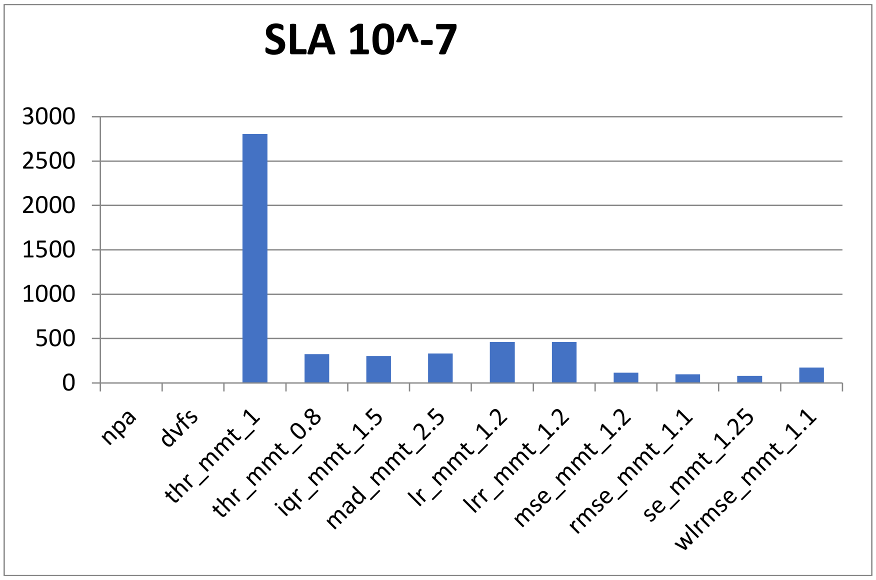


Fig.5 SLAV for the main algorithms

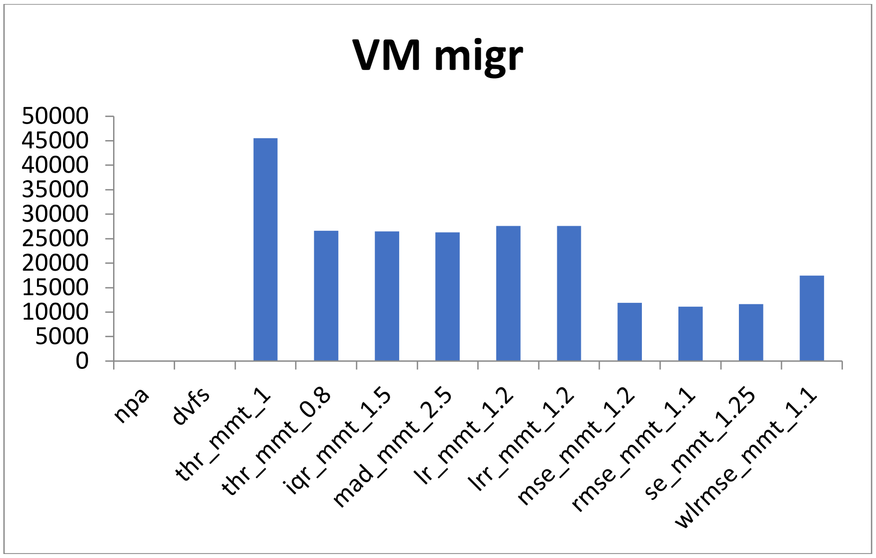
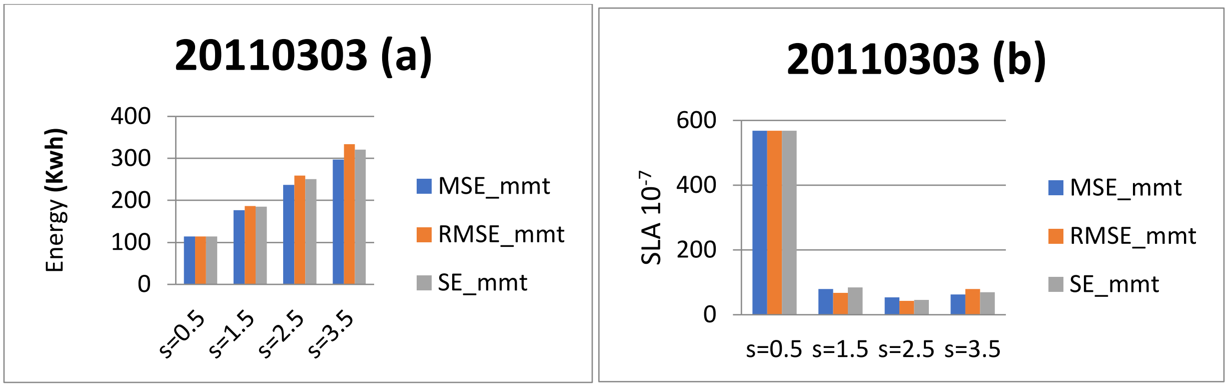


Fig.6 Number of VM migrations for the main algorithms

d) ANALYSIS OF SAFETY PARAMETER S

Because the safety parameter s in (14) is a key parameter for our model, these experiments engage to choose through the best s for each combination. As a point of view, the safety parameter s determines the threshold of HUD algorithm. As s is getting larger and the threshold is getting smaller, the HUD can be more easily to determine host as underloaded. This leads to more aggressively favour for VM consolidation and larger number of VM migration. Hence, it can be expected for lower level of SLA violations and higher energy consumption. This is also shown the necessity of using more accurate decision-making prediction. For each detection algorithm, the parameters are varied from 0.5 to 3.5 increased by 1. Figure 7 shows the energy consumption, SLA violations under different value of s using the ‘‘20110303’’ and random worload. There is a trend that energy consumption grows as the increasing s parameter. The higher s is corresponding to the lower level of SLA violations, but the more the energy consumption as shown in Fig.7 (a), (b), (e) and (d). However, the random workload tends to increase SLA violations while s is larger than 3.5. The level of SLA violations could be deteriorated by unreasonable consolidation while considering random workload. As shown in Fig.7 (c) and (f), the number of VM migration for the ‘‘20110303’’ increase while s is larger than 1.5 and the higher s for random is corresponding to the number of VM migration.



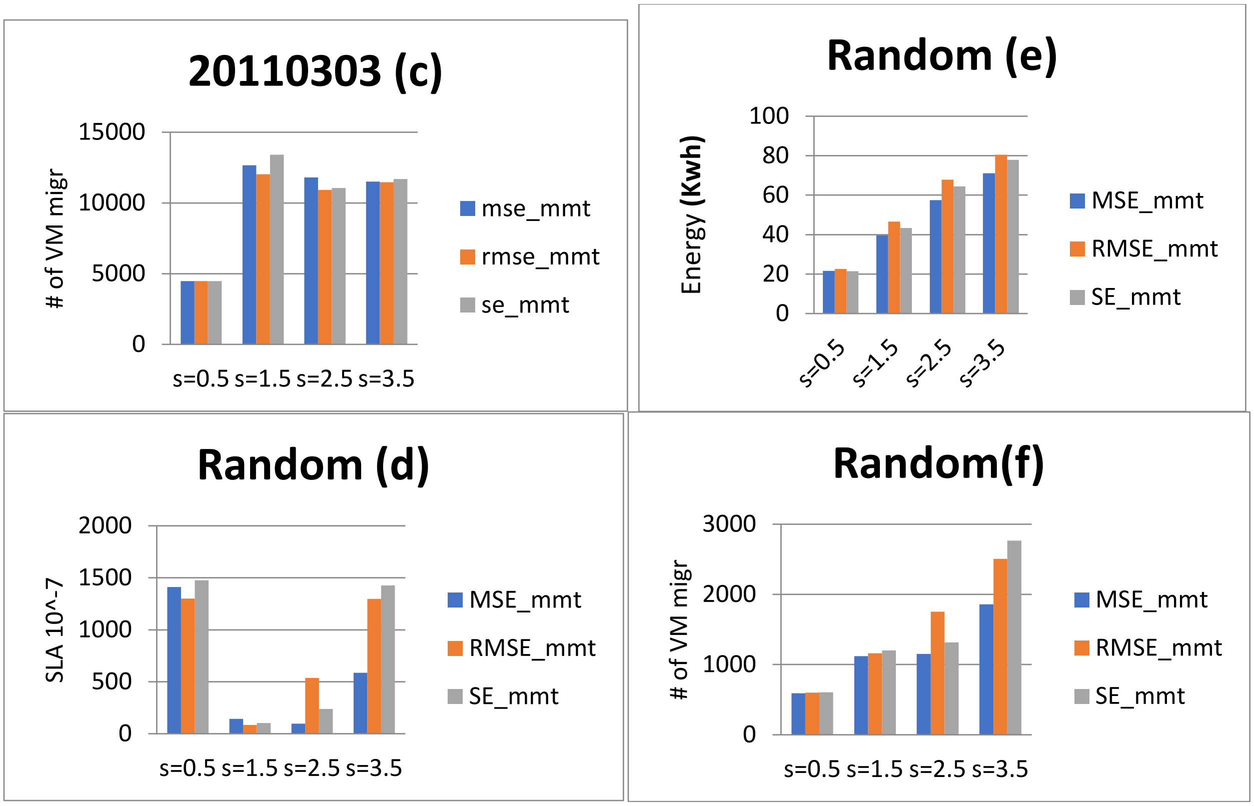


Fig 7 Energy consumption, SLAV and number of VM migration under different value of s for real workload. (a) 20110303-Energy. (b) 20110303-SLAV. (c) 20110303-VM migr. (d) Random- Energy. (e) Random-SLAV. (f) Random- VM migr.