Robust Server Consolidation: Coping with Peak Demand Underestimation

Diarmuid Grimes, Deepak Mehta, Barry O'Sullivan[1],
Robert Birke, Lydia Chen, Thomas Scherer[2] and Ignacio Castineiras[3]
[1]*University College Cork* [2]*IBM Research – Zurich* [3]*Cork Institute of Technology*

Submission Type: Research

***Abstract*—Energy consumption in data centres accounts for a significant proportion of national energy usage in many countries. One approach for reducing energy consumption is to improve the server usage efficiency via workload consolidation. However, there are two primary reasons why this is not done to a large extent. The first reason is that greater consolidation could result in violations of Service Level Agreements if resources are over-utilised. The second reason is that users specify the requirements of a VM based on the maximum estimated usage for each resource over the whole life span of the VM, and usually over-estimate these maximum values to avoid possible contract violations. Typically, the VM will have significantly lower resource usage in most time periods.**

Recently, a number of methods have been proposed to predict resource usage of VMs. We show that although these prediction techniques are efficient when their performances are measured using well known metrics, a low prediction error can still result in significant violations of Service Level Agreements if not handled properly during workload allocation. Our results emphasise the importance of analysing workload prediction in conjunction with workload allocation techniques. In this work, we examine the practical impact of using predicted resource usage for optimal server consolidation. In particular, we investigate the occurrences of over-utilised resources on servers due to under-predicted resource usage. We propose methods to reduce the likelihood of such occurrences, both through the enforcement of safety capacities on the server side, and through biasing towards over-prediction on the VM side. The results indicate that an appropriate balance can be found between energy savings and non-violation of Service Level Agreements.

I. Introduction

Data centres are significant energy consumers due to the pervasive use of cloud computing, big data, internet-of-things, e-commerce, etc. The continuing upward trends in these areas suggest a further increase of the energy consumption in the next years. Reducing the energy consumption while at the same time ensuring quality of performance for users has become an important goal for many data centres.

The two primary energy consumers within a data centre are IT and HVAC (Heating Ventilation and Air Conditioning), i.e. powering of servers to run virtual machines (VMs) and powering of the ventilation and cooling system to reduce hotspots and prevent server overheating, which can negatively impact server performance and damage equipment. In a typical data centre, approximately 60% of the energy is consumed by IT and 35% is consumed by HVAC. Consequently, research into reducing energy consumption has focused on these two areas.

From the IT workload side, server consolidation is one approach which can be implemented to reduce energy consumption. Servers are often under-utilised, and with a significant proportion of energy consumption incurred by just the idle power of a server to be maintained in an ON state. The reasons for the lack of consolidation are two-fold. Firstly, there is the concern that the greater the utilisation the greater the risk of a negative impact on VM performance, which might lead to Service Level Agreement (SLA) violations. The second reason is that the user must provide a value for each associated resource (CPU, memory, network, etc.) such that the VM will consume less than this value in every time period. Typically, the user over-estimates the maximum usage over all time periods to avoid possible contract violations. Furthermore, these over-estimated values will then be used for all time periods, when in fact actual resource usage may be far beneath this value for most of the operating time.

In order to improve consolidation, accurate predictions of the time-variable resource usage are required. Recently, a new approach has been proposed for predicting resource usage of VMs [1]: PRACTISE is a neural network based framework that can efficiently and accurately predict future loads, peak loads, and their timing. It has been shown that PRACTISE can outperform both ARIMA and baseline neural network models in terms of average prediction errors and that the method is particularly robust from short-term (e.g. hours) to long-term (e.g. several days) prediction windows.

However, in the first step the predictions were not evaluated in conjunction with server consolidation. In this work, we address this and specifically investigate the impact of prediction errors on resource over-utilisation. We present an analysis of both the savings that can be

achieved in terms of energy reduction through consolidation and, more importantly, the impact on VM/server performance with respect to resource over-utilisation. Based on the initial results, we propose a number of techniques for reducing the likelihood of resource over-utilisation on servers.

In Section II, we first provide some background for the current work, and describe in more detail the prediction method. In the following three sections, we describe the optimisation problem considered, present a mathematical model of the problem, and then introduce an efficient, scalable approach to workload allocation which was used for testing the impact of prediction. A number of metrics are introduced in Section VI for quantifying the impact of server over-utilisation. Section VII presents the experimental evaluation of the prediction method, and proposes techniques for reducing resource over-utilisation while maintaining energy reductions through consolidation.

II. Background

Data centre energy requirements have grown massively in the last few years. One of the optimisation challenges for reducing energy requirements is to keep servers well utilised by deciding which virtual machines to migrate, where to migrate, when to migrate, and when and which servers to switch on/off. Achieving this goal optimally requires the capability of predicting the future time-variable resource demands of VMs accurately and computing the plan for migrating VMs for efficient server consolidation quickly.

A. Server consolidation

Many data centres have infrastructure for VM migration in place. There are several reasons for migrating VMs from their current servers to different ones. Generally, if the load on a server is very high, one might want to move some or all the virtual machines from that server to others. Also, if there is an energy efficient server with sufficient resources, one might want to reassign some VMs to that server so that the overall energy consumption is reduced. Finally, if a server is under-utilized, one might want to shut down that server after having migrated the corresponding VMs to other servers. In general, the challenge is to consolidate servers so that energy consumption can be reduced without increasing the number of SLA violations.

Server consolidation can be static or dynamic [2]. In static consolidation, peak resource demands are typically used for each VM, and the assignment of VMs to servers may not be recomputed for long periods of time. This technique is done in an off-line fashion. In contrast, dynamic consolidation is implemented on shorter timescales, preferably shorter than periods of significant variability of the resource demand. It is a reactive approach in which servers are continuously monitored and the reconfiguration of VMs to servers is triggered when the servers are either over-loaded or under-utilised. Although dynamic consolidation may reduce the energy consumption for a given time-period, it might violate SLAs due to its myopic nature. We are therefore interested in minimising both energy consumption and SLA violations.

A mixed integer programming approach to dynamically configuring the consolidation of multiple services or applications in a virtualised server cluster has been proposed [3]. That work focuses on power efficiency, and considers the costs of turning on/off the servers. However, it assumes homogeneous resource requirements of VMs over time. In [4], a data centre is viewed as a dynamic bin packing system where servers host applications with varying resource requirements and varying relative placement constraints. The objective is to minimise the transition time for migrations of virtual machines, whereas we are concerned with minimising the energy consumption. A constraint optimisation model for energy-cost aware data centre assignment systems which allocates virtual machines with time-variable demands is studied in [5]. In [6] an energy-aware framework is proposed for the reallocation of virtual machines in a data centre to reduce the power consumption, the goal is to find the best possible placement of virtual machines for a given time-period subject to service level agreements.

Most of the work on server consolidation either assumes an oracle that can forecast resource requirements accurately or does not analyse the impact of prediction on the consolidation approach. We do not make such an assumption and specifically analyse the impact of uncertainty in the prediction on the server consolidation. A notable difference is the work of [7] where VM multiplexing is considered for the purpose of consolidation. The main idea is to partition VMs into VM groups based on the performance requirement of each VM (and in particular VMs with complementary resource requirements across time). For each VM group, the joint-VM sizing algorithm is employed to determine the capacity being allocated. Although by exploiting VM multiplexing, it is possible to achieve more compact static consolidation, it might still not achieve the efficiency of dynamic consolidation in terms of energy savings.

B. Online workload prediction

It is of paramount importance for data centres to provide sufficient resources, catering to multiple demands of virtual machines, particularly the peak demands [7], [8], [9], while maintaining energy efficiency [10]. Most of the related work focuses on developing consolida-

tion strategies, using either heuristics [7] or machine learning [9], and often assumes that accurate workload information is provided, except [7]. In contrast to this, we argue that obtaining accurate workload predictions for different resources, e.g., CPU, memory, disk and network, is indeed the first key step to efficient resource management [11], [12], [13] for large scale data centres. More importantly, the capability to capture peak demands is crucial to optimise the resource usage and energy efficiency of data centres.

Prior art [14], [15], [16] pointed out that resource demands typically show a strong time variability, e.g., seasonality within a day, week or month. Time series prediction methodologies such as autoregression (AR), moving average (MA), and ARIMA [17] are commonly adopted to capture the temporal patterns. For the workload time series considered in this study, ARIMA models fall short in capturing the peak demands [18], whereas the enhanced neural network model PRACTISE, which is based on information of autocorrelation, shows a strong promise in capturing highly bursty time series, and particularly their peaks.

To motivate the effectiveness of PRACTISE in predicting peak demands of time series, we further extend the original PRACTISE framework and develop an on-line version. In particular, we address the practical issues of developing an on-line process for time series prediction: i.e. how to obtain sufficient historical information and how much time to predict ahead. In the following, we first describe such a general on-line prediction process, which can be integrated with different kinds of time series prediction methodologies, and then provide a high-level summary of the PRACTISE training and prediction methodology.

C. Prediction process

To simplify the prediction process, we define fixed length timeslots (e.g. 10 minutes), a constant prediction horizon N_p (e.g. 6 timeslots, corresponding to 1 hour into the future), and a fixed number of training data samples N_t (e.g. 1008 samples, covering one week). Since in general monitoring data is not sampled in regular intervals, we resample the incoming data and store it into a circular buffer of length $N_c = N_t+1+N_p$ as illustrated in Fig. 1. Each element of this buffer corresponds to one timeslot as defined above. To minimize storage overhead, we calculate the cumulative moving average for timeslot i recursively with

$$m_i^{(k_i+1)} = \begin{cases} x_i^{(1)} & \text{for } k_i = 0 \\ \frac{x_i^{(k_i+1)} + k_i \cdot m_i^{(k_i)}}{k_i+1} & \text{for } k_i \geq 1\,, \end{cases} \quad (1)$$

where $x_i^{(k+1)}$ is the (k_i+1)-th incoming value belonging to timeslot i and $m_i^{(k_i)}$ is the previous averaged value of timeslot i. If the timestamp of an incoming value exceeds the window of the current timeslot, the buffer is shifted to the new timeslot (and the data of the oldest timeslots is discarded if necessary).

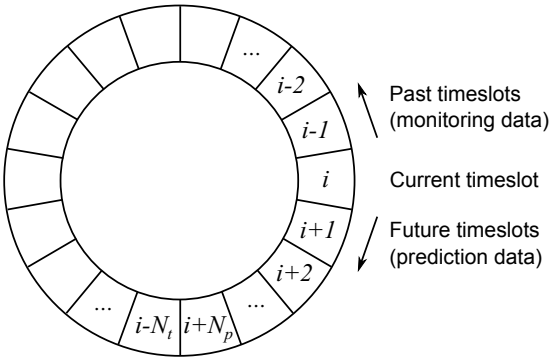

Fig. 1: Circular buffer for workload monitoring and prediction data.

Training of the workload prediction models automatically starts when the N_t-th timeslot is populated. The specific training process for PRACTISE is explained in more detail in the next subsection.

After the training is complete, the samples stored in the circular buffer are used as inputs to the prediction model. The number and indices of the samples required to obtain a prediction for timeslot $i+1$ depend on the specific prediction model. In the general case, all samples up to timeslot i need to be available. To obtain a prediction for timeslot $i+2$, we therefore first obtain the prediction for timeslot $i+1$ and can then use this value as an input sample for the prediction of timeslot $i+2$.

D. PRACTISE

PRACTISE [1] is a neural network based framework that efficiently and accurately predicts data centre workload, and in particular peak loads and their timing. Fig. 2 provides an overview of the training and prediction methodology. Monitoring data is fed to the feature selection module, the selected features are then used as inputs for the training of the neural network and the ensemble averaging module processes the aggregated results. Finally, there is a module that continually evaluates the prediction accuracy and triggers retraining of the prediction model if large errors are detected.

Artificial neural networks are composed of interconnected neurons whose associated weights are used to approximate non-linear functions of the inputs. During the training process, these weights are tuned based on representative input data. Selecting features that reliably capture repeating patterns and changing trends is key to building an accurate and efficient neural network

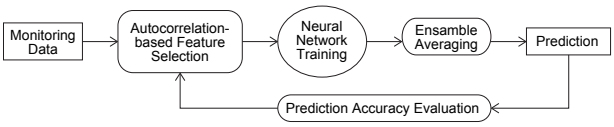

Fig. 2: Overview of the training and prediction methodology of PRACTISE.

model. PRACTISE identifies the relative peak points of the autocorrelation function by using a local maximum function and selects the respective lag values as features for the training of the neural network. Autocorrelation is a mathematical representation of the degree of similarity in a time series and a lagged version of itself over successive time intervals and as such is well suited to capture the periodic patterns commonly observed in data centre workload.

PRACTISE splits the input data into a training set to tune the weights, a validation set to determine the convergence point, and a test set to evaluate the training accuracy. A number of neural networks are trained in parallel, using randomly split data sets. The resulting models each produce their own prediction results. To obtain the final prediction, outliers are filtered out and the average of the remaining data is computed.

As shown in [1], compared to ARIMA and baseline neural network algorithms, PRACTISE achieves up to 3 times better prediction accuracy in terms of average prediction errors, and dramatic improvements (2- to 9-fold) with respect to the prediction timings. In particular PRACTICE, in contrast to classic time series models, is able to efficiently capture the peak loads in terms of their intensities and timing.

III. Problem Description

We first describe the optimisation problem relevant for analysing the impact of resource prediction on workload allocation. The problem is to place a set of virtual machines on a set of servers in order to minimise the energy cost of the data centre. The resource usage (including CPU) of a virtual machine changes over time. In each time-period, we must ensure that the servers have enough resources (e.g. CPU, memory, disk, network) to handle the virtual machines allocated to them.

Let $V = \{v_1, \ldots, v_n\}$ be the set of virtual machines, and $S = \{s_1, \ldots, s_m\}$ be the set of servers.

a) **Virtual Machines:** A virtual machine v_i is characterised by a set $A_i \subseteq S$ of allowed servers where it can be hosted, and a potential current server denoted by $init_i^{vm}$ (which might be null if the VM is a new request). A virtual machine v_i has a predicted resource consumption R_{ik} for the next time-period for each resource k in the set of resources K. In the following we will consider CPU to be the resource with index $k = 0$. The power consumption of a running server is computed based on the proportion of the CPU used by the VMs on the server [19].

b) **Servers:** A server s_j can be in one of two different states in any time-period: ON=1 or STBY=0 (stand-by). It is characterised by: resource capacities R_{jk}^{max}; a fixed power consumption P_j^{idle} (in Watts) when the server is ON to run the operating system and other permanent tasks, and P_j^{stby} when the server is in STBY mode; a maximum number N_j^{max} of virtual machines that can be allocated to it at any time-period; the current state $init_j^s \in \{0, 1\}$; and a unit cost τ_j per unit of CPU usage (also in Watts), computed based on R_{j0}^{max} and maximum power rating P_j^{max} of the server:

$$\forall_{s_j \in S} : \tau_j = (P_j^{max} - P_j^{idle})/R_{j0}^{max} \quad (2)$$

We assume that R_{j0}^{max} is the CPU capacity after excluding the basic consumption of CPU that is required when a server is in idle mode.

c) **Migrations:** The maximum number of VM migrations among all virtual machines from one time-period to the next is denoted by Mig^{max} and the cost of a migration by Mig^{cost}.

IV. Problem Formulation: Constraint Optimisation Model

The constraint optimisation model of the problem is presented below.

A. Variables

- Let $x_i \in A_i$ be the main integer decision variables that denote the server on which virtual machine v_i is to be run in the next time-period. The forbidden servers for each virtual machine are trivially enforced through the assignment of x to a value from its domain.
- Let $res_{jk} \in [0, R_{jk}^{max}]$ be the non-negative continuous variable that measures the resource consumption of resource k on server s_j.
- Let $nvm_j \in [0, N_j^{max}]$ be an integer variable that denotes the number of virtual machines running on server s_j.
- Let $cs \in [0, Mig^{max}]$ be an integer variable that denotes the number of virtual machines that are to change servers from the current time-period to the next time-period.
- Let $o_j \in \{0, 1\}$ be a Boolean variable that is set to 1 if s_j is required to be ON in the next time-period, 0 otherwise.

For each server $s_j \in S$ and virtual machine $v_i \in V$, variables o_j^c and x_{io}^c are created for denoting the current state of the server j and the current server of the VM.

B. Constraints

d) Capacity Constraint.: The following constraints link the CPU and other resource loads of a server to the virtual machines assigned to it.

$$\forall_{s_j \in S, k \in K} : res_{jk} = \sum_{v_i \in V \wedge x_i = j} R_{ik} \tag{3}$$

The constraint on the server resources not exceeding their capacities is trivially enforced through the upper bounds of the domains of res_{jk}.

e) Cardinality Constraint.: The maximum number of virtual machines that can run on a server is restricted:

$$\forall_{s_j \in S} : nvm_j = |\{v_i | v_i \in V \wedge x_i = j\}| \tag{4}$$

f) Migration Constraint.: The number of server changes over all virtual machines is constrained:

$$cs = |\{v_i | v_i \in V \wedge x_i \neq x_i^c\}| \tag{5}$$

g) ON Constraint.: A server is ON if it is hosting at least one virtual machine:

$$\forall_{v_i \in V} : \quad x_i = j \implies o_j = 1 \tag{6}$$

h) Initial State.: If the initial configuration is given then the constraints $o_j^c = init_j^s$ and $x_i^c = init_i^{vm}$ are enforced for each $s_j \in S$ and $v_i \in V$ respectively. Otherwise, the constraints $o_j^c = o_j$ and $x_i^c = x_i$ are enforced.

C. Objective Function

The objective is to minimise the sum of the following costs:

i) Migration Cost: The migration cost is the total number of server changes over all virtual machines multiplied by the cost of migration:

$$Mig^{cost} \times cs$$

j) Server Usage Cost: The total server usage cost is the sum of all CPU costs incurred over all servers:

$$\sum_{s_j \in S} \Big(\tau_j \times res_{j0} \; + \; P_j^{idle} \times o_j \; + \; P_j^{stby} \times (1 - o_j) \Big)$$

V. Solution Approach

A VM can be associated with one of the following states: `running`, `suspended`, `newRequest`.

- The state `running` means that a VM is currently running on a server.
- The state `suspended` means that a VM is assigned to a server but it is not running on it. However, it may be consuming memory.
- The state `newRequest` means that a VM is not created and it is not assigned to any server.

Algorithm 1 uses a greedy approach for assigning VMs to servers to find a feasible solution. It receives as input the set of VMs V, the set of servers S, and a vector o that denotes the current servers where the VMs are running.

The algorithm first tries to assign the VMs that are already running to their respective current servers for the next time period (Lines 3–11). If for any reason a VM cannot be assigned to its current server (e.g. due to an increase in resource consumption of the already assigned VMs of this server) then such VMs are collected in the set $unassigned$.

Algorithm 1 findGreedySolution(V, S, o)

1: $uvars \leftarrow \{x_i | v_i \in V\}$; $sol \leftarrow \emptyset$; $fvars \leftarrow \emptyset$;
2: $unassigned \leftarrow \emptyset$
3: **for** $v_i \in \{v_j \in V | state(v_j) = \texttt{running}\}$ **do**
4: **if** x_i can be assigned to o_i by satisfying all the constraints **then**
5: $x_i \leftarrow o_i$
6: $nvm_{x_i} \leftarrow nvm_{x_i} + 1$
7: update resource consumption of server x_i
8: **else**
9: $unassigned \leftarrow unassigned \cup \{v_i\}$
10: **end if**
11: **end for**
12: **for** $v_i \in unassigned$ **do**
13: $x_i \leftarrow \texttt{findSatisfiableMachine}(v_i)$
14: $nvm_{x_i} \leftarrow nvm_{x_i} + 1$
15: update resource consumption of server x_i
16: **end for**
17: **for** $v_i \in \{v_j \in V | state(v_j) = \texttt{newRequest}\}$ **do**
18: $x_i \leftarrow \texttt{findSatisfiableMachine}(v_i)$
19: $nvm_{x_i} \leftarrow nvm_{x_i} + 1$
20: update resource consumption of server x_i
21: **end for**
22: **for** $m_j \in M$ (which is an ordered set of machines) **do**
23: **for** $v_i \in \{v_j \in V | x_i = m_j\}$ **do**
24: $m_k \leftarrow findBetterMachine(v_i, m_j)$
25: **if** $m_k \neq m_j$ **then**
26: $x_i \leftarrow m_k$
27: $nvm_j \leftarrow nvm_j - 1$
28: $nvm_k] \leftarrow nvm_k + 1$
29: update usages of servers m_j and m_k accordingly
30: **end if**
31: **end for**
32: **if** $nvm_j = 0$ **then**
33: put server m_j in the standby mode
34: **else**
35: put server m_j in the ON mode
36: **end if**
37: **end for**

A new server is sought for each unassigned VM, resulting in the migration of such VMs (Lines 12–16). The algorithm then tries to satisfy the newly arrived requests for allocating VMs to servers (Lines 17–21). The function `findSatisfiableMachine` returns a server where a given VM can be created and run.

Once all the VMs are allocated to servers or migrated to resolve overloaded servers, the algorithm further tries to migrate VMs from least power-efficient servers to most power-efficient servers (Lines 23–31). Beloglazov et al. [19] propose a workload allocation approach called *Modified Best Fit Decreasing* where VMs are sorted in terms of decreasing CPU requirements and allocated to a server that provides the least increase in power consumption. We generate a static ordering on servers acoording to the power efficiency relative to CPU usage. Based on this ordering, VMs running on the least efficient server are migrated to the most efficient server with available capacity where possible and subject to the limit on the number of migrations allowed. This consolidation will result in the least inefficient servers not running any VM, and consequently such servers have their state switched to standby (Lines 32–36), further reducing the energy cost.

VI. Metrics for evaluating overconsumption of resources

We now introduce a set of metrics to quantify the over-utilisation of resources due to inaccuracies in predicted consumption. Let T be a given set of time-periods. Let res^a_{jkt} be the actual (average) utilisation of a resource $k \in K$ on server $s_j \in S$ during time-period t. Let vm_{jt} be the set of virtual machines running on a server $s_j \in S$ during time-period t. Let $rm_{jt} \subseteq vm_{jt}$ be a minimal set of virtual machines that if removed from server $s_j \in S$ would result in no resource being over-utilised on server s_j during time-period t.

1) The average over-utilisation of a resource k over all servers over all time-periods can be computed as follows:

$$ARO_k = \frac{\sum_{t \in T} \sum_{s_j \in S} \max\left(res^a_{jkt} - R^{max}_{jk}, 0\right)}{|T|}$$

2) The following computes the average number of servers that are over-utilised over a given set of time-periods:

$$\text{NSO} = \frac{\sum_{t \in T} |\{s_j | res^a_{jkt} > R^{max}_{jk}, s_j \in S, k \in K\}|}{|T|}$$

3) The average number of virtual machines whose performance can be affected due to over-utilisation of one or more resources is given by:

$$\text{NVO} = \frac{\sum_{t \in T} \sum_{s_j \in S, \exists_{k \in K} res^a_{jkt} > R^{max}_{jk}} |vm_{jt}|}{|T|}$$

TABLE I: Specification ranges for 132 servers.

	Power (*W*)		CPU (*GHz*)		Memory
	Max	Idle	Cores	Speed	(*GB*)
Min	90	30	8	2000	16
Avg	184.6	100.5	14.5	2657.6	117.7
Max	460	300	32	3200	256

4) The average minimal number of virtual machines that would need to be removed from the over-utilised servers is computed as:

$$\text{NVR} = \frac{\sum_{t \in T} \sum_{s_j \in S} |rm_{jt}|}{|T|}$$

VII. Evaluation of server performance with workload prediction.

The impact of inaccuracies in the prediction on workload optimisation was evaluated in simulation mode as follows. The simulated data centre involved 132 servers, of 9 different types, with specifications ranging as shown in Table I. It should be noted that in this work we only consider IT power consumption. We do not consider the thermal impact of consolidation on the HVAC power consumption.

Actual and predicted usage data was collected for each of the 4 resources CPU, memory, disk, and network for 1008 time points for 583 VMs emulating the typical workload patterns experienced by a webfarm based on the traces from the Wikipedia Grid Report [20]. This amounted to a sample every 10 minutes for 7 days. Figure 3 presents CPU usage data for two randomly chosen VMs across the 7 days. One can observe a clear periodicity in both cases, albeit there are variations in the magnitude and there is an inverse correlation between the usage of the two VMs.

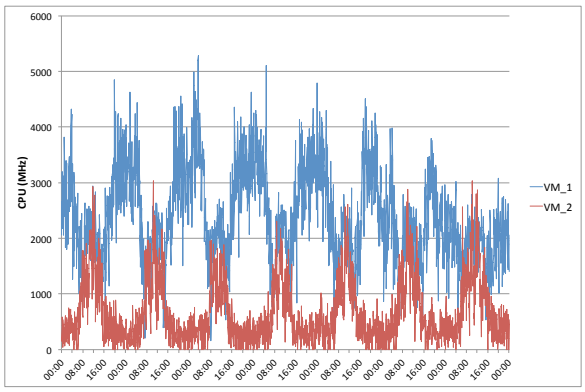

Fig. 3: CPU usage for two sample VMs.

Server consolidation was performed based on *predicted* resource usage, i.e. VMs were migrated to servers (in order of server preference) provided that the server had sufficient available capacity on each of its four

resources according to the VM's predicted resource usage. The purpose of these experiments was solely to assess the impact of inaccuracies when performing server consolidation. Therefore, to remove vagaries imposed by the migration limit, the experimental setup allowed unlimited migrations in each time-period for this simulation study.

The allocation in each time-period was then evaluated based on *actual* resource usage. In particular, actual CPU usage was used to compute the IT power consumption of the servers in each time-period. Actual values were also used for identifying the number of servers that had over-consumed resources in a time-period (i.e. where predicted usage of a resource was less than the capacity but actual usage was greater than the capacity), and for computing the metrics described in Section VI.

There were three baselines for the workload allocation experiment, providing lower and upper bounds on the performance with predicted VM resource usage.

1) The first baseline did not allow migrations so all VMs were run on the same server from the initial round robin allocation for all time periods. VMs were initially spread across all servers which, coupled with the limit on migrations, means that this gives an upper bound on IT power consumption.
2) The second baseline allowed unlimited migrations, with server consolidation performed based on *actual* resource consumption. This represents the oracle, i.e. how good would performance have been if the prediction was always 100% accurate.
3) The final baseline (*StaticMaxVal*) provided a stricter upper bound than the 0 migration case. Server consolidation was performed with unlimited migrations, but the values used for resource usage corresponded to the maximum over all 1008 time periods for that resource on that VM. (In practice the user would specify a value (much) greater than this for each resource in order to guarantee that there is no SLA violation on the user's side.) Therefore the value for each resource requirement of each VM was fixed across time for this baseline, and thus migrations were only needed in the first time period.

A. Impact of prediction accuracy

Baseline results are first provided for perspective in terms of power consumption (Fig. 4a) and number of servers in ON state (Fig. 4b) in each time period. Average power consumption for the *0 migration* case was 18.2kW, with all 132 servers running in every time period. Consolidation using the *StaticMaxVal* approach was able to reduce average power consumption to 14.9kW, requiring 13 fewer servers.

In comparison, the dynamic consolidation approaches yielded much better results. The oracle had an average consumption of 10.6kW (over 40% reduction in energy consumption) with just 87.1 servers operating on average over all time-periods. Allocation based on predicted usage resulted in similar behaviour to the oracle, with average power consumption of 10.6kWh (and a near-identical value when predicted CPU usage was used in calculation instead of actual CPU usage), and average number of operating servers of 87.4.

Finally, one can observe in Fig. 4a periodic peaks and troughs in total CPU consumption, with the result that for the oracle often only half as many servers were required for the off-peak times. This shows the benefits that can be achieved through *dynamic* consolidation and migration, unlike the VM-multiplexing approach of [7] which uses a static consolidation.

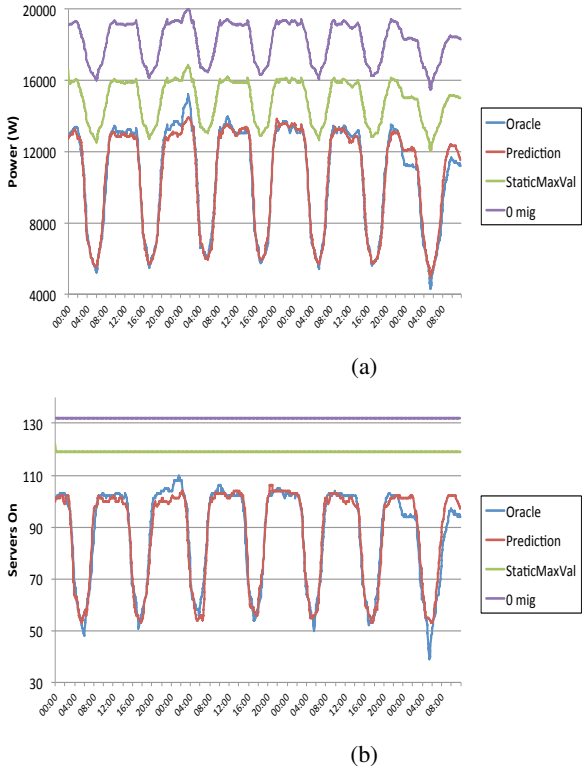

Fig. 4: (a) Power consumption and (b) Number of servers in state ON per time period.

However, analysis of the results where prediction values were used for deciding allocation revealed that there was at least one server over-utilised in 995 of the 1008 time periods. In every case, CPU was over-utilised while there were a small number of cases where disk was also over-utilised (albeit always by less than 0.1% of disk capacity).

Table II provides more detailed information regarding

the CPU over-utilisation of servers. The results reveal that a significant number of servers were over-utilised in the 995 runs where server over-utilisation occurred, with 22 servers on average (the maximum number of over-utilised servers in a time-period was 87).

The number of VMs affected by this over-utilisation was typically in the hundreds (note this is not per server, but summed across all over-utilised servers for each time period), although in most cases the removal of only one or two VMs would have been sufficient for the server to return to safe performance.

The ARO_{cpu} metric shows that the CPU resource of servers was typically over-utilised by approximately 7.3%. However the maximum of the over-utilisation in a run was as much as 77% over the CPU capacity of the servers averaged across all such servers.

TABLE II: Over-utilised server resource (CPU) statistics.

Servers On	NSO	NVO	NVR	ARO_{cpu}	$ARO_{cpu}\%$
87.4	22.0	189.8	1.3	1877.2	7.3

B. Safety capacity for servers

We have shown that prediction inaccuracies (particularly under-predicting resource consumption) can have a detrimental impact on the performance of VMs. One method to mitigate this is to introduce a safety capacity for each resource on each server. This allows a percentage of resource capacity to be kept as a buffer for inaccuracies when allocating based on predicted values.

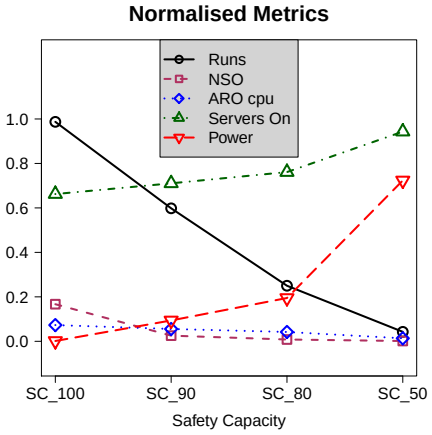

Fig. 5: Safety Capacity Impact on Normalised Metrics

Figure 5 presents results for sample safety capacities of 100% (i.e. no buffer), 90%, 80% and 50%. The metrics shown are normalised values for: number of runs (i.e. time periods) out of 1008 where at least one server was over-utilised, NSO, ARO_{cpu}, average number of servers running, and average power consumption (normalised with respect to the first two baselines).

The first three metrics show the positive impact of safety capacity in reducing server over-utilisation. For example, the number of runs where server over-utilisation occurs drops by nearly half with a safety capacity of 90%, and is down to just one quarter of runs with a safety capacity of 80%. However, even with a safety capacity of 50% (i.e. holding back half the capacity of each resource in reserve), over-utilisation still occurred in 4% of the runs.

Furthermore, there is an obvious tradeoff between buffer size and possible savings. Increasing the size of buffer used will naturally decrease the likelihood of over-utilisation, but will also reduce the consolidation that can be performed and thereby decrease the possible IT energy savings. This can be seen in the results for average number of servers running and average power consumption in Figure 5. Indeed, all 132 servers were running in the majority of runs when the safety capacity was 50%. This resulted in average power consumption of 16kW, which is 2kW less than the *0-migration* baseline and 6kW greater than the oracle.

C. Biasing towards over-prediction of VM demand

The results in the previous section illustrated that safety capacity alone is not a sufficient measure to minimise the likelihood of over-utilised servers. An additional method is to incorporate the prediction *error* in the VM demand and bias towards over-prediction, adding the error directly to the prediction. There are a number of ways the error could be computed, here we used an exponentially weighted moving average of the absolute prediction error (*EWMAE*).

A final method we investigated (referred to as *MaxLastEWMAE* in the figures below) was to also factor in the most recent resource usage when biasing towards over-prediction. More specifically the biased demand value (R^B_{ikt}) of a VM v_i for each resource k in each time-period t is computed as the maximum of the actual resource usage of that VM in the previous time-period ($R^a_{ik(t-1)}$), and the error-adjusted prediction.

$$R^B_{ikt} = \max(R^a_{ik(t-1)}, R_{ikt} + Err_{ik}) \qquad (7)$$

We first present results in terms of impact on server over-utilisation, in particular the number of time periods where at least one server was over-utilised, the average number of VMs/servers affected by the over-utilisation across all time periods, and the percentage of resource (CPU) that was over-utilised in the relevant cases. Fig. 6 compares the default prediction with the two adjusted predictions (*EWMAE* for the exponentially weighted moving average of the absolute prediction error alone, and *MaxLastEWMAE* for the maximum of most recent

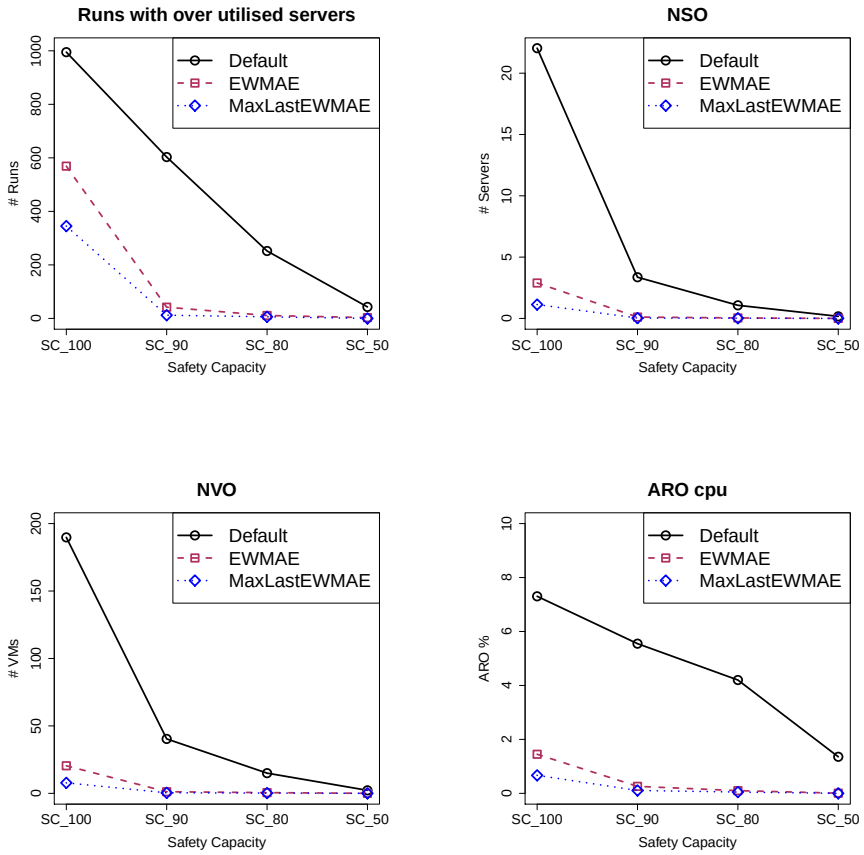

Fig. 6: Impact of safety capacity and VM demand error margins on server over-utilisation.

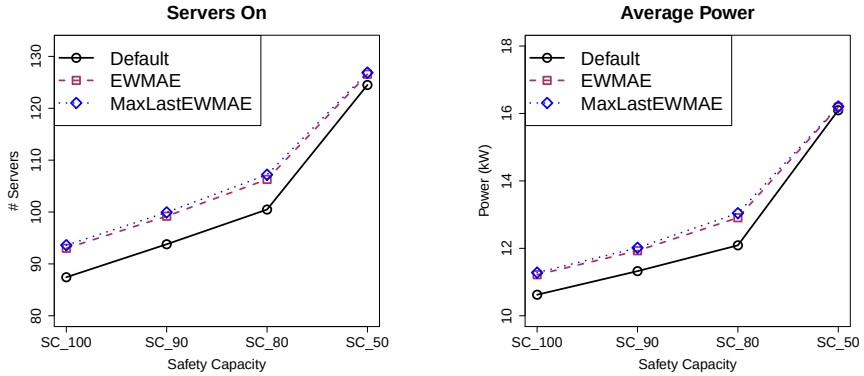

Fig. 7: Impact of safety capacity and VM demand error margins on consolidation.

usage value and error-adjusted prediction) for different safety capacities.

There is a significant improvement in performance in both cases with the latter, as expected, slightly better than the former. The combination of adjusted prediction with safety capacity of even just 90% removed nearly all server over-utilisation. For example, there were only 12 runs out of the 1008 where a server was over-utilised with *MaxLastEWMAE* with 90% safety capacity, falling to only 6 runs with a safety capacity of 80%. Consequently the average number of servers/vms affected and $\%\ ARO_{cpu}$ were all less than 1 with 90% safety capacity.

The negative impact of over-prediction bias is shown in Fig. 7, with 6 more servers running with *EWMAE* and *MaxLastEWMAE* on average than the default for safety capacities 100%, 90%, and 80%. This was reflected in the average power consumption which increased by just under 1kW. Given the reduction in server over-utilisation (and by extension possible SLA violations), we believe this moderate increase in energy consumption is an acceptable tradeoff.

VIII. Conclusion

Reducing energy consumption in data centres is of growing importance, with server consolidation offering one of the largest potential savings. In order to enable dynamic consolidation, resource usage of VMs must be predicted efficiently and accurately. Recent advances in this area have been encouraging, but the impact of inaccuracies in the prediction has not been considered to any great level to date. In this work we have presented an analysis of the behaviour of prediction-based server consolidation utilising values from a state-of-the-art resource prediction method.

Although the prediction method has been shown to be very accurate relative to a number of common metrics, it was found in this work that consolidation using these values still resulted in considerable server over-utilisation. Based on these findings, we proposed a number of methods to mitigate the impact of prediction inaccuracy while still allowing for considerable energy savings. In particular, combining overbiasing on the predicted demand side with resource buffers on the supply side was found to provide a good tradeoff between minimising SLA violations and minimising energy consumption.

Acknowledgment

This work was supported by FP7 Grant 608826 (GENiC - Globally Optimised Energy Efficient Data Centres). The Insight Centre for Data Analytics is supported by SFI Grant SFI/12/RC/2289.

References

[1] J. Xue, F. Yan, R. Birke, L. Y. Chen, T. Scherer, and E. Smirni, "Practise: Robust prediction of data center time series," in *Network and Service Management (CNSM), 2015 11th International Conference on*. IEEE, 2015, pp. 126–134.

[2] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing sla violations," in *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*. IEEE, 2007, pp. 119–128.

[3] V. Petrucci, O. Loques, and D. Mosse, "A dynamic configuration model for power-efficient virtualized server clusters," in *Proceedings of the 11th Brazilian Workshop on Real-Time and Embedded Systems*, 2009.

[4] E. Bin, O. Biran, O. Boni, E. Hadad, E. K. Kolodner, Y. Moatti, and D. H. Lorenz, "Guaranteeing high availability goals for virtual machine placement," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011, pp. 700–709.

[5] H. Cambazard, D. Mehta, B. O'Sullivan, and H. Simonis, "Bin packing with linear usage costs - an application to energy management in data centres," in *Principles and Practice of Constraint Programming - 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings*, ser. Lecture Notes in Computer Science, C. Schulte, Ed., vol. 8124. Springer, 2013, pp. 47–62. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40627-0_7

[6] C. Dupont, G. Giuliani, F. Hermenier, T. Schulze, and A. Somov, "An energy aware framework for virtual machine placement in cloud federated data centres," in *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on*. IEEE, 2012, pp. 1–10.

[7] X. Meng, C. Isci, J. O. Kephart, L. Zhang, E. Bouillet, and D. E. Pendarakis, "Efficient resource provisioning in compute clouds via VM multiplexing," in *ICAC*, 2010, pp. 11–20.

[8] T. Wood, L. Cherkasova, K. M. Ozonat, and P. J. Shenoy, "Profiling and modeling resource usage of virtualized applications," in *ACM/IFIP/USENIX Middleware*, 2008, pp. 366–387.

[9] Z. Gong and X. Gu, "PAC: pattern-driven application consolidation for efficient cloud computing," in *MASCOTS*, 2010, pp. 24–33.

[10] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini, "Statistical profiling-based techniques for effective power provisioning in data centers," in *Eurosys*, 2009, pp. 317–330.

[11] N. Tran and D. A. Reed, "Automatic ARIMA time series modeling for adaptive I/O prefetching," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 4, pp. 362–377, 2004.

[12] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," in *Sigmetrics*, 2005, pp. 303–314.

[13] J. Xue, R. Birke, E. Smirni, and L. Y. Chen, "Managing data center tickets:prediction and active sizing," in *IEEE DSN*, 2016.

[14] F. Brosig, F. Gorsler, N. Huber, and S. Kounev, "Evaluating approaches for performance prediction in virtualized environments," in *IEEE MASCOTS*, 2013, pp. 404–408.

[15] R. Birke, L. Y. Chen, and E. Smirni, "Data centers in the cloud: A large scale performance study," in *IEEE Cloud*, 2012, pp. 336–343.

[16] ——, "Multi-resource characterization and their (in)dependencies in production datacenters," in *IEEE NOMs*, 2014, pp. 1–6.

[17] C. Chatfield, *The analysis of time series: an introduction*. CRC press, 2013.

[18] J. Xue, F. Yan, R. Birke, L. Y. Chen, T. Scherer, and E. Smirni, "PRACTISE: robust prediction of data center time series," in *IEEE CNSM*, 2015, pp. 126–134.

[19] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755–768, 2012.

[20] "Wikipedia grid report," https://ganglia.wikimedia.org/.