Ismaeel *et al. Journal of Cloud Computing: Advances, Systems and Applications* (2018) 7:10
https://doi.org/10.1186/s13677-018-0111-x

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH

Open Access

Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres

Salam Ismaeel, Raed Karim and Ali Miri*

Abstract

Data center power consumption is among the largest commodity expenditures for many organizations. Reduction of power used in cloud data centres with heterogeneous physical resources can be achieved through Virtual-Machine (VM) consolidation which reduces the number of Physical Machines (PMs) used, subject to Quality of Service (QoS) constraints. This paper provides an in-depth survey of the most recent techniques and algorithms used in proactive dynamic VM consolidation focused on energy consumption. We present a general framework that can be used on multiple phases of a complete consolidation process.

Keywords: Cloud computing, Data centre management, Workload prediction, Energy efficiency, VM placement, Review

Introduction

Recent years has seen an exponential increase in the use of the cloud computing industry in satisfying Information Technology (IT) requirements. Data center power usage has been among one of the large commodity IT service expenditure for many organizations. The global data center electricity usage in 2012 was around 300 – 400 TWh, about 2% of global electricity usage and it is expected to triple by 2020 [16, 181], see Fig. 1 [136]. With up to 88% of this power going to powering and cooling IT equipment's, any energy use reduction can result in major power and cost savings. For example, an estimate by Amazon shows the cost of energy for its data centers has reached 42% of total cost of its operation [139]. In addition, according to Environmental Protection Agency, each 1000kWh of power consumption emits 0.72 tons of CO2 [171]. Hence, the reduction of energy usage has become one of the key objectives in the design of any modern data centers.

Today Data centres often consist of a large number of Physical Machines (PMs), which are grouped into multiple management clusters. Each of these clusters manages and controls a large number of PMs. A cluster can be homogeneous in that all of its managed PMs are identical, or it could be heterogeneous in that it manages PMs with different resource make and capacities [44].

Virtual Machines (VMs) are virtualized environments with predetermined virtual resources such as CPU, memory storage and bandwidth configured with an operating system and/or middle-ware and one or more application programs. VMs can execute workloads like any PM. Cloud service providers offer their computing resources to their clients based on Service Level Agreement (SLA). Services provided are typically in a form of VMs, which place on different PMs to carry out various tasks. The virtualization ability not only enables service providers to charge their clients based on their usage in a pay as-you-go scheme, but also it provides clients the ability to scale up or scale down resource utilization, as their needs vary. These advantageous partially stem from the fact that virtualization technology enables multiple virtual servers to run on the same PM, resulting in better resource utilization and reduction of aggregate power consumption [68, 89, 90, 102].

Data centre energy efficiency measures reduction of energy used by hardware or software equipment in data centres for a given service or level of activity. Hardware equipment includes both IT equipment (e.g. network and servers) and supporting equipment (e.g. power supply, cooling and data center building itself), whereas

* Correspondence: ali.miri@ryerson.ca
Department of Computer Science, Ryerson University, 350 Victoria St, Toronto, ON M5B 2K3, Canada

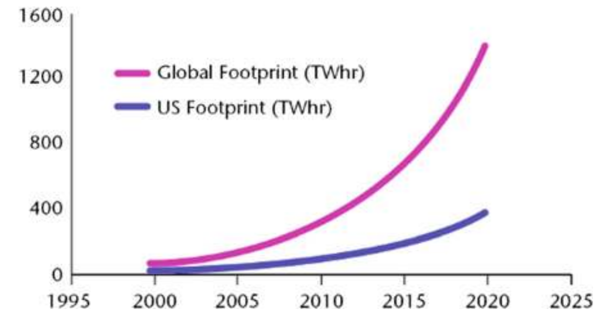

Fig. 1 Projection of data centres electricity use

software equipment may include Cloud Management Systems (CMSs) used to manage the entire data centre or end-users' applications [66, 125]. Given that a large part of power consumption of data centers is in their hardware equipment, this paper focuses on the problem of reducing energy consumption through efficient management of PMs and VMs in Cloud Data Centre (CDC) [18, 53]. We consider four different strategies:

- VM Resizing is the process of changing the number of resources reserved for VMs through either adding or removing resource elements, or increasing or decreasing the capacity of each resource element in a VM. All these processes will be done without executing a reboot, an application restart, reconfiguration or recreation of a VM [28]. This will attempt to adjust PMs to their actual load and typically results in a reduction of power use [27, 73, 105, 163].
- Optimal initial placement seeks to optimally assign VM or group of VMs to servers - as part of an initial state such that the mapping minimizes the total inter-rack PMs used or traffic load in the network to reduce energy [58]. Deterministic Algorithms will discuss these algorithms, such as those in ([112, 144, 168]).
- Overbooking of physical resources refers to the strategy of overlaying requested virtual resources onto physical resources at a higher ratio than 1:1 [135]. This strategy can result in better utilization of PM idle resources, which might have been otherwise reserved. However, special care must be taken to reduce risks associated with unmet Quality of Service (QoS) demand over peak PM resource utilization [13, 116].
- VM Consolidation is the process of using minimum active PMs as possible through migrating VMs over time in an optimal fashion to reduce resource consumption [109, 118, 142].

There are two general types of VM consolidation: static and dynamic. In static consolidation, sizing and placement of VMs on PMs are pre-determined when a job arrives and the placement does not change over a period of time. This type of VM consolidation therefore is often suitable for short running jobs for a couple of hours, where PMs resources for different types of VMs are predefined [157]. Energy reduction will be mostly based on simple heuristics or historical VMs demand patterns. Although this may result in an increase of the cost of application provider during low demand resource period, whereas during high utilization periods, the available resources may be insufficient [183]. Dynamic VM consolidation can result in utilization of fewer PMs by re-allocation or live migration of VMs among PMs without significant interruption of services. It takes into consideration the performance as it based on QoS which is predefined via SLA between the tenant and the service provider. This will increase the power efficiency in data centres by turning off unused servers to save power [1, 66]. Dynamic provisioning-based energy consumption can represent the most efficient methods to improve the utilization of the resources and reduce energy [1, 14, 19].

Approaches took to dynamic provisioning fall under reactive or proactive categories. Reactive provisioning is to change initial placement after the system reaches a certain undesired state. The change may be made because of the performance, maintenance, power or load issues, or SLA violations. In proactive, monitoring, historical data and prediction algorithms are used to change the VM's initial placement before the system reaches a certain condition [90, 121]. Proactive provisioning uses prediction based approaches that help prepare ahead of changes in the workload and system usage [158].

This article provides a comprehensive literature survey of most recent proactive dynamic provisioning framework in a data centre with a focus on energy conversation. Dynamic consolidation frameworks typically consist of a large number of overlapped domains, which we have divided into the following five main subsystems, Fig. 2:

i. **Workload Prediction Subsystem** focuses on clustering process, VM and user behavior estimation, prediction window size, and forecasting process as a part of workload prediction subsystem.
ii. **Resource State Subsystem** is used to identify the state of physical and virtual resources. This subsystem will not include the monitoring and tracking tools only but it will be a focus on algorithms and techniques used in defining the PMs states.
iii. **VM Selection Subsystem** focuses on VM selection criteria.
iv. **VM Placement and Migration Subsystem** deal with the question of how to migrate selected VMs.

Networking strategies play a pivotal role in this framework, as network infrastructure topology and routing

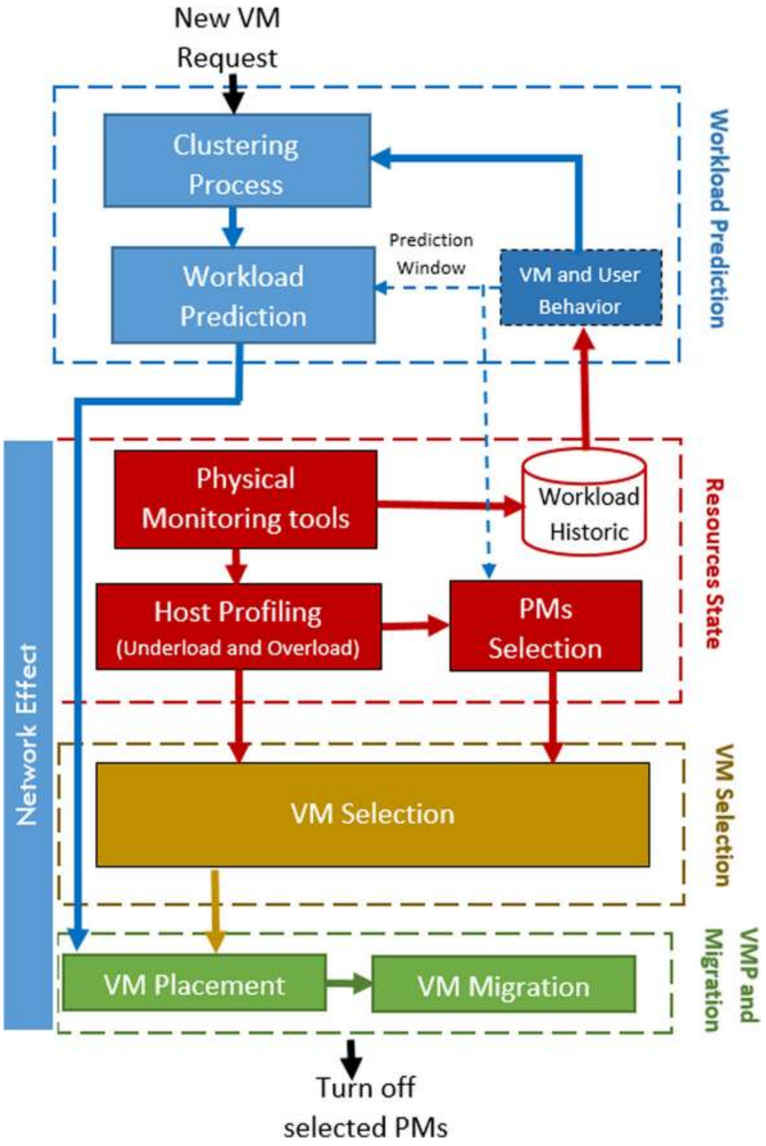

Fig. 2 Proactive dynamic VM consolidation literature review framework

protocols can have a direct impact on migration or consolidation with minimum network load [104].

Our survey will provide answers to the following questions:

- How to predict workloads? How can we predict future VM requests?
- What is the current state of the resources? How can we monitor and track the behavior of physical and virtual resources?
- Which VMs to migrate? And Where?
- How to migrate selected VMs?

The rest of the paper is organized according to subsystems described above. Workload Prediction Subsystem reviews application and techniques in workload prediction subsystems. Resources State Subsystem reviews algorithm related to data centre resource states, which are then used in VM selection subsystems in VM Selection Subsystem. VM and host selection presented in VM Placement (VMP) Subsystem. VM placement and migration subsystem covered in VM Migration, while network effect on all these subsystems are covered in Network Effect. Comparison of work on related subsystem can be found in Analysis of the State-of-the-Art Surveys in the Literature, followed by conclusions in Conclusions.

Workload Prediction Subsystem

Resource estimation underlies various workload management strategies including dynamic provisioning, workload scheduling, and admission control. All these approaches possess a prediction module in common which provides estimations to determine respectively whether or not to add more resources, rearrange the order of query execution, and admit or reject a new incoming query [89]. Prediction of the future resource behavior is a crucial process for efficient resource utilization in dynamic cloud computing environment because workload forecasting for short or long periods will be necessary to real-time control, resource allocation, capacity planning and data centre energy saving in cloud computing [90].

In recent years, cloud workload prediction is becoming more and more important. Many performance prediction algorithms and tools have been developed, which can be applied to predict the future CPU, memory load, VMs, etc. [97].

For propose of energy conservation, proactive approach to forecast required resources based on demand history, must overcome some or all of the following challenges [173]:

- Finding a way to make predictions that take into account both user, virtual and physical resources variations,
- Overcoming the problem of time varying demands,
- Estimating the required observation window size, and
- Detecting when the prediction is likely to be incorrect and how we can overcome the problem.

In a cloud environment, it is too difficult to predict the demand for each type of resource separately [44, 89]:

- Typically, VM requests consist of different amounts and types of cloud resources (e.g., CPU, memory, bandwidth, etc.). The multi-resource nature of these VMs poses a unique challenge when it comes to developing prediction techniques.
- Different cloud clients may request different amounts of VM resources which may be assigned on the same PM and not on separate machines. Therefore, it is both impractical and too difficult to predict the demand for each type of resource separately.

So, it is logical to create different categories of VM clusters, and then develop prediction techniques for each of these clusters. Thus, this review subsystem will not cover the most practical and recent published prediction algorithms but will include clustering literature, most useful prediction window size recommendation, and even literature discussed VM and user behaviors. We classify workload prediction subsystem into four functional areas, each will review publications in a separate subsection, namely [86, 90]:

- **Clustering Process**: Review the recent literature in clustering applied.
- **Prediction Process**: Algorithms and techniques used to forecast the future resource demand values.
- **Prediction and Observation Windows Size**: Prediction window used to identify the length of the time period in the future for which the workload needs to be predicted. While observation window used to identify the length of the time period required to monitor past workload variations.
- **User and VM Behavior**: Current approaches in analyzing and supporting users and resources behaviors, which has a strong influence on the overall cloud workload. This component analyzes VM and user behaviors during the time of requesting VMs. Uncovering the dependency relationships between users and VMs helps improve the prediction accuracy and excluding unwanted (noise) data [99].

Clustering Process

The objective of the prediction subsystem is to use previous usage patterns to estimate future VM request workloads in a data center. In a cloud environment, it is too difficult to predict the demand for each type of resource separately for the following reasons [89]:

- Typically, VMs consists of different amounts and types of cloud resources (e.g., CPU, memory, bandwidth, etc.). The multi-resource nature of these VMs poses a unique challenge when it comes to developing prediction techniques.
- Different cloud clients may request different amounts of the same resources. Therefore, it is both impractical and too difficult to predict the demand for each type of resource separately.

So, it is logical in any proactive VM consolidation to use clustering. Clustering, precisely partitioned clustering, use to map each request received into one of a set of clusters with different types of VMs or tasks during the predefined period of time. Notice that for fuzzy partitioning, a point can belong to more than one group [96]. The prediction algorithm used to predict the number of VM in each cluster rather than predicting each type of VM [86].

In this section, a brief summary of the latest useful clustering techniques in the literature. A recommended general clustering system will be described by end of this section.

The K-means method is one of the most famous and widely used clustering algorithms. Given a data set of N points, a partitioning method constructs K ($N \geq K$) partitions of the data, with each partition representing a cluster. Where a number of clusters in the data should be pre-specified and each data point belongs to exactly one group. The basic K-means-described in Algorithm 1-works as follows [167]:

Algorithm 1 Basic K-means algorithm.

```
Inputs: Historical VM requests, number of clusters
Outputs: Centre of these clusters
Select $K$ points as initial centroids.
repeat
    From $K$ clusters by assigning each point to its closest centroid.
    Re-calculate centroids of each cluster.
Until Centroids do not change.
```

K-means has been used by Dabbagh *et al* [44] and Chowdhury *et al* [40] to create a set of clusters to group all types of VM requests. Each request represents a VM with CPU and memory for Google traces data [146]. K-means algorithm inputs included Google traces and the number of clusters, while the output was centres of these clusters. The selection of **K** should be balanced between two conflicting objectives: reducing errors and maintaining low overhead [89].

Khan *et al* introduced a co-clustering algorithm to identify VM groups and the time periods in which certain workload patterns appear in a group. Then, they used Hidden Markov Model (HMM) to explore the temporal correlations in workload pattern changes. This help to predict individual VM's workload based on the groups found in clustering step [101].

A kernel Fuzzy C-means FCM clustering algorithm was used to forecast the future CPU loads by Xu *et al* [179]. They divided historical long CPU load time series data into short equal sequences and used kernel FCM to put the subsequences into different clusters.

Canali and Lancellotti [33] used Principal Component Analysis (PCA) as an automated methodology to cluster VMs by leveraging the similarity between VMs' behavior. They considered VMs as a member of classes running in the same software component. This methodology has been applied to two case studies, a virtualization tested and a real enterprise data center. This methodology can reduce the amount of collected data, thus effectively contribute to addressing the scalability issues of the monitoring system. This technique is very useful for monitoring and reporting but it is difficult to use it as an input to a prediction algorithm to forecast each type of VMs in nearest future. This is because: (1) PCA relies on linear assumptions (2) PCA based on mean vector and covariance matrix, some distributions may be characterized by this but not all.

Claudia and Lancellotti combined the Bhattacharyya distance and ensemble techniques to evaluate the similarity between the probability distributions of multiple VM resource usage [32]. They considered both system and network related data. Their proposal achieves high and stable performance in automatic VM clustering through their experiments on real-data collected from an enterprise data center. VM Clustering was used to reduce the amount of data required in cloud monitoring.

The workload is always driven by the users, therefore realistic workload models must include user behavioral patterns linked to tasks. The approaches previously described completely focus on tasks, neglecting the impact of user behavior on the overall environment workload [133]. Raed *et al* [99] labeled clusters with ranges of workload percentages into Very Big, Big, Medium and Small with 25% workload difference between each cluster. They incorporated users' behaviors and VM clustering with multi-way prediction technique to estimate incoming workload at a data centre. They got more accurate prediction results by comparing theirs with other well-known prediction models.

General clustering system used in VM clustering with user behaviors proposed by our previous work [86, 90]. General clustering system consists of the following components (see Fig. 3):

User and VM Behavior: used to analyze VM and user behaviors on real time VMs request. Uncovering the

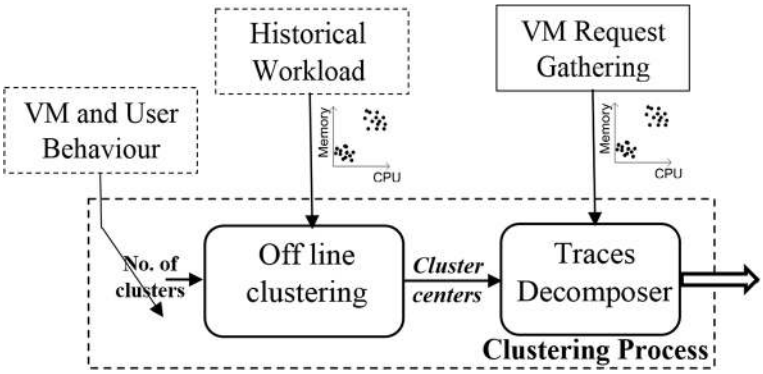

Fig. 3 Recommended Clustering System [90]

dependency relationships between users and VMs helps improve the prediction accuracy and excluding unwanted (noise) data [99].

Off-line clustering: is used to create a set of clusters for different types of VMs and users using long term historical data. The centres of these clusters used to classify incoming request during a specific time frame. *Trace decomposer*: maps each request received during a given observation time into one cluster according to long term cluster centres calculated off-line.

User and VM behaviors have a strong influence on the overall cloud workload. Comprehensive workload models must consider both VMs and users behaviors to reflect realistic conditions by excluding unwanted VMs or users from future workload estimation process.

Historical Workload represents the historical data, which should be updated periodically and used to predict the next period VM request for each observation. Also, it used to calculate centres of clusters from time to time using long term observations.

VM Request Gathering includes cloud monitoring tools which can help in detecting and tracing the variations or failure of resources and applications during an observation.

The number of clusters should balance two conflicting objectives: (1) reducing errors and (2) maintaining low overhead. For example, in Google workload trace, Xia *et al* [178] and Rasheduzzaman *et al* [145] chose **K** = 6, **K** = 5, respectively, for K-means clustering algorithm. Xia and Rasheduzzaman depend on the minimum value of **K** to reduce error. They didn't take into consideration the effect of increasing **K** on the performance of the predictor. This problem was discussed in Dabbagh et al [44, 48]; They suggested to choose **K** = 4 for his work. While Moreno et al [133] selection was **K** = 3 as the best selection because he included users behavioral patterns.

Ismaeel and Miri compared between K-means and FCM for different numbers of VM clusters and User clusters [90]. They concluded that, although the FCM algorithm needs long off-line training time, it produces better results than the K-means for a fewer number of clusters. FCM provided a fewer number of clusters with a small error by balancing reducing errors and maintaining low overhead requirements through the use of minimizing the number of inputs in the prediction process [86].

After studying these techniques it is observed that various clustering techniques currently used for analyzing workload characteristics do not provide a structured model which can be used for conducting simulations. All we can do is to compare on the basis of execution time and cluster quality. Workload analyses need to explore more than coarse-grain statistics and cluster centroids. To capture the patterns of clustered individuals it is also necessary to conduct an analysis of the parameters and study the trends of each cluster characteristic. This will lead us to conclude that the need for new methodologies especially for real time and online streaming data [15, 133].

Prediction Process

As discussed in the previous section, a proactive dynamic VM consolidation is to triggering resource requests, this can be taken by forecasting future resource demand values based on demand history. Since workloads tend to trace of resources patterns based on time, it is expected that time series forecasting methods are reliably predicted resource demand [173].

In recent years, many performance prediction algorithms and tools have been developed, which can be applied to predict the future CPU, memory load, VMs … etc. Their focuses were on how to save energy, improve performance and increase profit and so on [97]. In next subsections, the most recent prediction techniques, especially ML techniques, applied in the field of VM consolidation based energy consumption will be reviewed. Before we do that, a simple description of the basic principle of prediction problem and these techniques will give.

Basically, prediction problem is to estimate the value of an output Y from the set(s) of readily available input(s) X, and can be formulated simply by:

$$\hat{Y}(k) = \hat{f}(X(k)) \quad \dots \tag{1}$$

Where $\hat{Y}(k)$ is the predicted value(s) and $\hat{f}(X(k))$ is the estimated relation between inputs and outputs of the system. This relation is either linear or nonlinear. In Linear Regression (LR) models, the relation between one or more input variables and dependent output variable(s) described by using a linear equation to observed data, like Auto-regressive (AR), Moving average (MA) and Gray Forecasting Model (GFM), and Wiener filter [173].

Sometimes linear models are not sufficient to capture the real-world phenomena, and thus nonlinear models are necessary. But in many situations, we do not know much about the underlying nature of the process being modeled, or else modeling it precisely is too difficult. In these cases, we typically turn to a few models in Machine Learning (ML) that are widely-used and quite effective for many problems. These methods include basis function regression including Radial Basis Functions (RBF), Artificial Neural Networks (ANN), and K-Nearest Neighbors (KNN) [80].

Auto-regressive Integrated Moving Average (ARIMA)

The basic assumption made to implement this mode is that the considered time series is linear and follow a particular statistical distribution, such as Normal distribution. It Combination of AR and MA models.

In an AR(T_p) model the future value of a variable is assumed to be a linear combination of T_p past observations and a random error together with a constant term. Mathematically the AR(T_p) model can be expressed as:

$$\hat{Y}(k) = const + \sum_{i=1}^{T_p} \phi(i) Y(k-i) + \in(k) \tag{2}$$

where T_p is an integer constant represents the order of the model, $\phi(k)$ and $\epsilon(k)$ are the actual value and random error at time period k, respectively, $\phi(i)(i = 1, 2, \ldots, T_p)$ are model parameters and *const* is a constant. Yule-Walker equations usually used to relate AR model parameter to the auto-covariance of the random process [29]. While the model order T_p selected by using different criteria like Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) and Cross Validation (CV) [4].

MA model uses past errors as the explanatory variables. The MA model is given by:

$$\hat{Y}(k) = \mu + \sum_{j=1}^{o} \theta(j) \in (k-j) + \in(k) \tag{3}$$

where O is the order of the model with white noise process, $\theta(j)(j = 1, 2, \ldots, O)$ is model parameters and μ is mean of the series. Fitting an MA model to a time series is more complicated than fitting an AR model because in the former one the random error terms are not predicted.

Auto-regressive and MA models can be effectively combined together to form the ARMA (T_p, O) models, as represented by:

$$\hat{Y}(k) = const + \sum_{i=1}^{Tp} \varphi(i) Y(k-i) + \sum_{j=1}^{o} \theta(j) \in (k-j) + \in(k) \tag{4}$$

Usually, ARMA models are manipulated using the lag operator, refer to [166] for more details.

If the original process *Y (k)* is not stationary, we can look at the first order difference process $\Delta Y (k + 1) = Y (k) - Y(k - 1)$ or the second order differences $\Delta 2Y (k + 1) = Y (k) - 2Y (k - 1) + Y (k - 2)$ and so on.

The process *Y (k + 1)* is said to be an Auto-regressive Integrated Moving Average process, ARIMA(T_p, d, O), if $\Delta^d Y$ (*k*) is an ARMA(T_p, O) process.

Researches on workload prediction have been done based on statistical approaches, such as [31, 57, 108] who proposed an ARIMA algorithm. The basic assumption that the considered time series is linear and follows a particular statistical distribution, such as Normal distribution. If the original process is not stationary, we can look at the first order difference process or the second order differences and so on. If we ever find that the difference process is a stationary process we can look for an ARMA model of that. AR, MA, ARMA and ARIMA techniques can be used to model many time series. A key tool in identifying a model is an estimate of the auto-covariance function [173].

Gray Forecasting Model (GFM)

Grey forecast can be used to predict the behavior of non-linear time series. This is a non-statistical forecasting method that is particularly effective when the number of observations is insufficient.

Grey forecasting model, precisely GM(1, 1) model, is one of the most widely used technique in the Grey system [42, 94]. In this technique, the predicted value of $\hat{Y}(k)$ can be obtained by accumulated generation sequence of the original data sequence.

$$\hat{Y}^{(1)}(k) = \sum_{i=1}^{k} Y^{(i)}, k = 1, 2, \ldots, T_n \tag{5}$$

Where the sequence $\hat{Y}^{(0)} = X^{(0)}(1), X^{(0)}(2), \ldots, X^{(0)}(T_p)$ is an original data sequence, T_n is the sample size of data, and $\hat{Y}^{(1)} = X^{(1)}(1), X^{(1)}(2), \ldots, X^{(1)}(T_p)$ is the accumulated generation sequence of $Y^{(0)}$.

The GM(1,1) model can be represented by a first order difference equation with time response equation given by:

$$\hat{Y}^{(1)}(k+1) = \left(Y^{(0)}(1) - \frac{b}{a}\right)e^{-ak} + \frac{b}{a},\ k = 1, 2, \ldots, T_n - 1 \tag{6}$$

where $\hat{x}(k + 1)$ denotes the prediction x x at time k + 1 and a and b represent the adjusting and effect factors, respectively. These coefficients, or parameter series, $[a, b]^T$ can be obtained by ordinary least squares method, as described in [175].

The main characteristics of GFM are it is simple and has the ability in time series prediction with least amount of historical data. This is done by extracting of actual laws in a system using existing data [117], where number of historical data must be more or equal to four. But the main drawback is, it assumes new data grows exponentially and they use time dependency rather than data dependency in a time series forecasting model.

Jheng *et al* [94] proposed a GFM to predict the workload of the PMs in a cloud data centre. The main characteristics of GFM are the simplicity and the ability to predict with least amount of historical data. This is done by extracting of actual laws in a system using existing data [117], where the number of historical data must be more or equal to four. But, the main drawbacks are (1) it assumes new data grows exponentially and (2) it uses time dependency rather than data dependency in a time series forecasting model.

Wiener Filter

Wiener filter is an optimal-linear discrete time filter which can be used to produce an estimate of a desired or target random process by linear time-invariant multi-filtering of an observed noisy process assuming known stationary signal and noise spectra and additive noise [22].

Dabbagh *et al* [44, 48], proposed a framework to predict the number of VM requests, to be arriving in the near future, along with the amount of CPU and memory resources associated with each of these requests. The K-means clustering was used to create a set of clusters contain all types of VM requests. Stochastic Wiener Filter (SWF) was used to estimate the workload of each cluster. Although, Wiener filter is unreliable for the dynamic behavior of demand cloud resources because it is suitable to estimate the target random process by Linear Time-Invariant (LTI) for known stationary signal and noise spectra [30], Dabbagh *et al* improved the original Wiener filter to support online learning, making it more adaptive to changes in workload characteristics.

An alternative approach to address the prediction problem is LR [7], which models the relationship between one or more input variables and dependent output variable by using a linear equation to observed data. Sometimes linear models are not sufficient to capture the real-world phenomena, and thus nonlinear models are necessary. In regression, all such models will have the same basic form, i.e. Eq. 1. But in many situations, we do not know much about the underlying nature of the process being modeled, or else modeling it precisely is too difficult. In these cases, we typically turn to a few models in ML that are widely-used and quite effective for many problems. These methods include basis function regression (including RBFs), ANNs, and KNNs [80].

Many researchers use combination pre-described techniques and other to increase prediction accuracy. Cao *et al* [35], suggested an ensemble model for online CPU load prediction. Their model has multiple predictor sets include Auto-regression model, Weighted Nearest Neighbors (WNN) model, Exponential Smoothing Model (ESM), most similar pattern model, and WNN model for differenced data (DWNN). Each predictor has a specific membership which can dynamically adjust. CPU workload has been estimated by these combined sets through the scoring algorithm. The main drawbacks in this predictors are: 1) it consists of two levels of prediction; all the predictors have specific weight and it is very difficult to find the optimal weight for each predictor; 2) relatively time-consuming in applying different algorithms at the same time; 3) most of the suggested set of predictors are based on statistical approaches.

Basis Function Regression

A one dimension basis function can simple represented by:

$$\hat{Y}(k) = \sum\nolimits_i^{M_b} w_i b_i(X) = \mathbf{b}(x)^T W \tag{7}$$

where $\boldsymbol{b}(x) = [b_1(x), \ldots, b_{M_b}(x)]^T$ and $M_b$are the number of basis functions and $\boldsymbol{w} = [w_1, \ldots, w_{M_b}]^T$.

Two common choices of basis functions are **polynomials** and **RBF**. Radial basis functions and the resulting regression model are given by [4]:

$$\mathbf{b}(x) = \sum e^{-\frac{(x(k)-c_k)^2}{2\sigma^2}} \tag{8}$$

$$\hat{Y}(x) = \sum w_k e^{-\frac{(x(k)-c_k)^2}{2\sigma^2}} \tag{9}$$

where c_k is the center of the basis function and σ^2 determines the width of the basis function. Both of these are parameters of the model that must be determined somehow.

In practice, there are many other possible choices for basis functions, including sinusoidal functions, and other types of polynomials. Also, basis functions from different families, such as monomials and RBFs, can be combined. We might, for example, form a basis using the first few polynomials and a collection of RBFs.

In general, we ideally want to choose a family of basis functions such that we get a good fit to the data with a small basis set so that the number of weights to be estimated is not too large.

To fit these models least-squares regression can be used to minimize the sum of squared residual error between model predictions and the training data outputs, refer to [4] for more details.

Artificial Neural Networks

Another choice of basis function is the sigmoid function, the most common choice of sigmoid is:

$$\mathbf{b}(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

Sigmoids can be combined to create a model called an ANN. For regression with multi-dimensional inputs$X \in \mathbb{R}_2^k$, and multi-dimensional outputs $Y \in \mathbb{R}_1^2$, and for 1D case model:

$$\hat{Y}(x) = \sum w_j^{(1)} \mathbf{b}\left(w_j^{(1)} x + bias_j^{(2)}\right) + bias^{(1)} \tag{11}$$

Hence, the neural network is a linear combination of shifted (smoothed) step functions, linear ramps, and the bias term. This objective function cannot be optimized in closed-form, and numerical optimization procedures must be used. Neural network and LR are widely applied in previous works to forecast VMs workload in cloud environments [103]. The main problem with this approach, and in most of the LR applications, as they considered the fact that future workload could be independent of their previous workload pattern [144]. On the other hand, the workload has an obvious nonlinear feature [39], and LR demands workloads that have simpler behavior than those that ANN-based method [31].

Several studies use ANN as prediction model [38, 140, 150]. Although ANN represents a universal approximation, but still have the drawbacks of in choosing a suitable algorithm, network structure, and initial condition. For butter performance, ANN may be combined with the typical prediction methods such as Sliding Window Method (SWM) [85], Auto-regression model [39], and Fuzzy System (FS) [23, 39, 144].

Dynamic behavior forecasting problem can be resolved with ANN [38, 140, 150], Adaptive Neuro-Fuzzy Inference System (ANFIS) [23, 39, 144], Support Vector Machine (SVM) [7], and latent feature learning based models [35, 36, 39].

Bey *et al* [23]; combined Adaptive Network-based Fuzzy Inference Systems (ANFIS) and clustering process to estimate the future value of CPU load. The model carried out on real CPU load time series to determine the optimal number of clustering for one machine. The results of their work showed that the CPU load prediction using ANFIS model for each category performs better than using one ANFIS for the whole of CPU time series without clustering.

Bey's work was improved by Chen *et al* [39], an ensemble model and subtractive-fuzzy clustering based fuzzy neural network was adopted. Fuzzy-Neural network performance was optimized using fuzzy-subtractive clustering algorithm. The Fuzzy-subtractive algorithm is composed of FCM clustering algorithm and subtractive clustering algorithm.

In [144], a neural network model was proposed to predict workload patterns in VMs, while Fuzzy Expert System (FES) was used to control near future changes in workload patterns for every VM. This scheme has been used to determine the time that VMs will be overloaded and need to be migrated.

Combining fuzzy and NN improves the modeling and prediction process, even ANFIS has better performance than NN [12, 87], but both of them require training before use.

Karim *et al* [99] proposed a model for predicting incoming number and types of VM based on user requirements using multi-way prediction technique. They incorporated user behavior to improve the prediction results. On the other hand, our previous works [89, 90] proposed framework combines clustering algorithm and Extreme Learning Machine (ELM) to forecast the VM requests in a CDC, Fig. 4. This work considers a single network to predict the number of VMs requested in each cluster and the optimal weights for the predictor in one step. This work was developed to use clustering not only on VM requests but also on user requests to filter for unexpected VM requests caused by unpredictable users' actions. We suggested to use this type of prediction to overcome the following challenges:

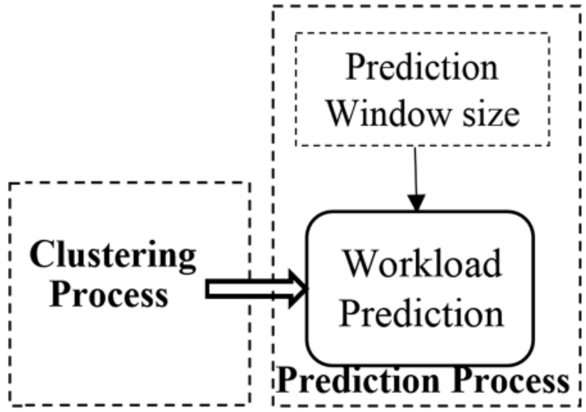

Fig. 4 Recommended prediction system

- Finding a way to make predictions that take into account both user and VM variations. Most related work in workload prediction only takes into account VM variability.
- Overcoming the problem of multivariate time varying VM requests
- Eliminating the restrictions on observation window size and number of VM clusters

Table 1 shows the prediction techniques can be divided into statistical, machine learning and hybrid approaches. In this table, Owin and Pwin are the observation and prediction windows, respectively.

Dabbagh et al [44, 50] estimated prediction window size based on the difference between the energy cost for keeping the PM idle and PM OFF/ON power cost, as described in the following equations:

Prediction Window Size

It is the time period for which the workload needs to be predicted to decide whether PMs need to be switched to sleep mode. It totally depends on the configuration of CDC, especially the server hardware, and its values affecting on workload prediction section. It represents the algorithms and optimization techniques used to determine the minimum number of times the prediction calculations must be performed. Based on prediction window, clustering and prediction algorithms used, the time required to monitor which is called observation window, should determine.

Table 1 Workload prediction techniques

	Techniques	References	Parameter		Clustering	User behavior	Window size
			VM	PM			
Statistical	ARIMA	[31, 127]	√				Fixed
		[57]		√			Fixed
	GFM	[94]		√			Fixed
	HMM	[101]	√		co-clustering		Fixed
	Bays Model	[54]		√		√	Owin=1/2 Pwin
	Multi-Way Data Analysis	[99]	√		FCM	√	Fixed
Hybrid	AR Model, ESM, WNN, DWNN	[35]		√			Owin=2 Pwin
	ECNN and LR	[85]		√			Fixed
	Ensemble Model based FNN	[39]		√	FCM/subtractive		Fixed
	Static and adaptive Winner Filter	[44]	√		k-means		Fixed/overlapped
ML	SVM, NN, and LR	[7]		√			Fixed
	GA to optimize Elman NN	[179]		√	Kernal FCM		Fixed/overlapped
	NN and Fuzzy expert	[144]	√				Fixed
	ELM	[89]	√		k-mean	√	Fixed/overlapped
	Multivariate ELM	[90]	√		FCM	√	Fixed

$$E_{sleep} = E_0 + P_{sleep} \cdot \left(T_p - T_0\right) \tag{12}$$

Where T_p is the length prediction window, P_{sleep} is the consumed power when in the sleep mode, E_0 is the energy needed to switch the PM to the sleep mode plus the energy needed to wake up it later, and T_0 is the transitional switching time. The estimated time required to keep the PM ON and idle (T_b) consumes an amount of energy that is equal to the energy consumed due to mode transition plus that consumed while the PM is in the sleep mode during that same period:

$$P_{idle} \cdot T_b = E0 + P_{sleep} \cdot (T_b - T_0) \tag{13}$$

Where T_b is the beak-even time. This means energy can be saved by switching PM to sleep mode if and only if the PM stays idle for a time period longer than T_b. That is, $T_p \geq T_b$ must hold in order for the power switching decisions to be energy efficient.

According to above, if we have PMs Profiles we can easily estimate the value of T_p. Dabbagh used the energy measurement study of PMs conducted in [153] to estimate the break-even time, T_b.

On the other hand, Prevost et al [139], presented a dynamic prediction quantization method to determine the optimal number of prediction calculation intervals to be performed within required future load SLAs [86].

Observation Window Size

As defined in Workload Prediction Subsystem, it is a process of observing and monitoring past workload variations during a time period. It is a specific time frame used to classify gathering data (new request and/or already exist VMs) that will be used in clusters process described in Clustering Process.

Di *et al* [54] found, based on their experiments, that maximum prediction accuracy for Google trace data [146] is to set the observation window to the half of the prediction window length. This setting is absolutely different from the well-known that a large observation window size leads to higher accuracy. This is may be a special case for Google host load used, which fluctuates much more drastically with higher noise.

Unlike Di *et al*, Ismaeel and Miri [89] and Dabbagh *et al* [44, 50] used to estimate the size of the observation window after classifying the workload into clusters. Where Ismaeel and Miri find a unique observation window size for all clusters, and Dabbagh *et al* find different observation window size for each cluster.

Dabbagh *et al* used experiments to estimate the length of the observation window in each cluster. They increased the size of observation window gradually until the reaches a point beyond which the prediction error can no longer be reduced even window size increase.

Ismaeel and Miri [79] selected the observation window to be 3 times longer than the prediction one, as in Eq. 14:

$$C_i(k+1) = (C_i(k), C_i(k-1), C_i(k-2)) \tag{14}$$

Where C_i represents the number of VM requests in the i^{th} cluster, and k represents the sampling interval. This means the observation window is 3 times equal prediction window size. This is due to the dynamic behavior of cloud provisioning, making the predicted output not only depends on the current state of the input but also totally affected by the previous state of the output. This work has been developed to eliminate restrictions on observation window size in an on-line multivariate time series ELM [90]. They used the current state and previous sates to cover all possible observation window states.

User and VM Behavior

Analyzing and supporting behaviors of users and tasks is a crucial process for both data centre providers and their perspective users. The behavior analysis within a specific course of time helps decision makers plan ahead for incoming workloads into data centers and make sure all requirements are fulfilled. As more and more data is stored and processed in data centres, it becomes a challenging task to anticipate the behavior of users and tasks. The workload can be modeled and analyzed in order to simulate requests and consumption patterns in data center environments [133]. The workload (CPU and memory) analysis captures both user and task behaviors. Then, users and tasks are clustered based on characteristics defined during the workload modeling.

Clustering is an effective unsupervised learning technique that group together items that are naturally similar to each other based on a certain metric [120]. The **k**-mean clustering technique is used by dividing observations into k clusters and data are grouped around cluster centroids. Important applications of CPU and memory data modeling and clustering are improving resource utilization, reducing energy waste and supporting accurate forecasting. In their model [133], users with profiles U submitting tasks with profiles *T*. The expectation $E(u_i)$ of a user profile is given by its probability $P(u_i)$, and the expectation $E(t_i)$ of a task profile is given by its probability $P(t_i)$ conditioned to the probability of $P(u_j)$.

$$E(u_i) = u_i P(u_i) \text{and } E(t_i) = t_i P(t_i)\left|P\left(u_j\right)\right. \tag{15}$$

Behavior prediction models predict application behaviors as well as VM behaviors in the cloud by tracing recently observed patterns which can be used to guide dynamic management decision. Adapting to frequent changes of workloads in order to calculate the required resources has been dealt with using heuristic techniques (predefined thresholds) at the Service Level Agreement (SLA) time to manage the scaling process as the application behaviors change dynamically [172]. Auto-scaling is another technique that performs scaling operations (adding or removing resources) without needs of human interactions [155]. Another technique that monitors changes in behaviors is History Table Predictor (HTP) [143]. In the history table, each row presents a pattern of the changes. When a new pattern found, the model attempts to find a match in the table to predict the next phase or to store that new pattern in the case of no matches is found. A more effective technique is the Statistical Metric Model (SMM) [152] that outperforms the HTP technique and other historical predictors for its long term global patterns modeling in application behavior, and its effective response to variable patterns.

The SMM model can be applied in cloud environments using common resource components which represent the behavior of the workload; in particular, these components are Memory utilization U_{mem}, CPU utilization U_{cpu}, and network utilization U_{net}. The three components can be combined using the load volume notation introduced in [177] and formulated as Datacenters:

$$LV = \frac{1}{(1-U_{mem})}\cdot\frac{1}{(1-U_{CPU})}\cdot\frac{1}{(1-U_{net})} \tag{16}$$

Data mining techniques can be used to discover Frequent Workload Patterns (FWPs) according to the previous history of resource usages [110]. The resource allocations can be determined by using the Association Rules Technique (ART) according to the prediction of resource availability in a given time period. The idea of using ART is on the discovered data patterns is to find out the possibility that the same patterns will repeat in future. In other words, ART can be used to represent the correlation between data patterns.

The technique mentioned above work well in discovering patterns in workload data and prepare them for subsequent operations such as resource allocations and predictions. Different other techniques can be used to perform these operations. The assumption is that since the data has already been trained, the resource allocations and predictions will be improved. However, this is not always the case. For example, Ismaeel et al [86] have tested his ML model by feeding it with the trained data (using clustering algorithms) to predict incoming requests to a CDC. The results were not promising (i.e. not a good prediction accuracy). More comprehensive techniques can do dual processes. They can be used not only to discover data patterns and hidden relationships (training) but also to perform these subsequent operations and produce more accurate results. Some examples of these techniques are Multi-Way Data Analysis [98] and Pearson Correlation Coefficient [97].

Resources State Subsystem

As described in in Introduction, VM planner needs to optimally assign VM or group of VMs to server-racks such that the mapping minimizes the total inter-rack PMs used or traffic load in the network to reduce energy, as an initial state [58]. Dynamic VM consolidation within a typical data centre can be done though migrating VMs over time in an optimal fashion. The typical data centre, a data centre with old and new PMs with different types. In other words, in a typical data centre, PM's power consumption is not constant but depends on the PM's load [119]. So, for any efficient VM consolidation, it is very important to identify the state of physical resources before and after initial assignment of VMs.

The objective of this section is not to review the practical useful monitoring tools in CDC only, but also to discuss the most recent algorithms and techniques used in literature to define PMs state. According to these algorithms, host(s) will be selected. This host represents the best placement of new or selected VM to migrate. Although, host selection may depend on several factors like workload dependencies, security, and network load, in next sections will cover selection process based on PM load. Host Underload Detection and Host Overload Detection discusses algorithms related to select the host that will be switched off (Host underload) and the host that will move some of the VMs from because of overloading (Host overload). But before that the practical CDC monitoring tools will be discussed.

CDC Monitoring tools

CDC state monitoring represents all physical components considerations and monitoring by tracking the behavior of these resources. In other words, it is the process of continuously measuring and accessing infrastructure and application behaviorism terms of performance, reliability, and power usage while maintaining a good QoS. Perfect CDC monitoring tools used in dynamic consolidation most able to:

- provide power information as well as the state of PMs and VMs,
- combine monitoring data arrived at different sampling rate from unrelated monitoring systems,
- analyze the measurement data, and select the most affected parameters to reduce the storage and computation load, and
- Select the suitable PM to switch on or off.

Monitoring the power consumption is required not only for understanding how power is consumed, but also for assessing the impact of energy management policies [148]. It will help in detecting and tracing the variations or failure of resources and applications [77]. There are many tools used in cloud monitoring such as: *Collectd*, *Nagios*, and *Ganglia*, which are providing the capability to monitor the computing, networking and storage resources utilization [93]; *Ceilometer* from OpenStack is used to reliably collect measurements of the utilization of the physical and virtual resources comprising deployed cloud [91]. Ceilometer collects data from different levels of the entire computing infrastructure (e.g., VM container, hypervisors, storage, and network) and the software resources (e.g., web server, application server, database server, and virtual applications) [77, 107]; Data Center Infrastructure Manager (DCIM) which provides detailed information about a server configuration, hardware, network connections, installed software, and so on. DCIM profiles the power consumed by each part of hardware in data centre [86]. Cloud monitoring tools and platforms properties, issues, analyzing, and comparisons surveys can be found in [2, 43, 64, 76, 78, 91, 92].

Host Underload Detection

Host underload refers to the state of a host in which all VMs should be migrated from. In the literature, the two common techniques used for determining host underload state are the least utilized host and static threshold [14]. It is the process of finding the host with the minimum utilization compared to the other hosts. i.e. the host that all VMs should be migrated from, so it should be switched off. If all VMs from the source host cannot be allocated, the host is kept active.

Several algorithms are used to determine the underloaded PMs, most of these algorithms depend on the CPU load in the PM. See Table 2 that summarizes as follows [14, 21]:

Least utilized: This technique uses CPU usage of the PM as a measure of determining underloaded PMs. PM is considered as being underloaded when it uses minimum resources. This algorithm is cost-effective because any monitoring system for the CPU utilization will be sufficient to decide which PM is the underload. But, it does not consider the number of VMs on that Host and the cost of moving such them to other PM.

Static Threshold: It depends on the mean of the latest CPU utilization measurements and compares it with a predefined threshold. If the mean CPU utilization is lower than the threshold, a host underload is detected. Put in Kashyap *et al* [100] use 0.2 for host CPU underload threshold. The problem is that using constant values of the threshold will be useless especially in a heterogeneous environment. Because it is difficult to find an optimal value of this threshold useful for all host.

Available capacity: This approach considers the available resource capacity instead of resource utilization as a measure of determining underloaded PMs. This is done by selecting a PM with an available capacity which is the least among all candidate PMs. The main drawback of this technique is that PMs with adequate resources not necessarily has less power than the others. Also, it does not consider the number of VM on a specific host.

Migration delay: PMs will be selected based on minimum time to complete all VMs migration process to other PMs. After pre-estimated the migration delay for each VM for different PMs. This technique needs a

Table 2 Host underload detection algorithms

Algorithm	Policies				Characteristic
	Available Capacity	Migration Delay	Number of VM	Host power	
Least utilized host [21]	√				Base on a host with minimum resources Not cover number of VMs on the host
Static Threshold [21]	√				Depend on the mean of the last CPU used Difficult to find the optimal value of the threshold
Available capacity [14]	√				Base on available host capacity compared to others; Not necessary PMs has less power than the other
Migration delay [14]		√			Base on minimum time to complete VMs migration process; Need a lot of predication and estimation
Hybrid [14]	√	√	√		based on MCDM More complicated and difficult for practical
Weighted CPU utilization [124]	√		√		Combines host utilization and number of VMs Need less computation then hybrid

lot of prediction and estimation which may cost more power, regardless the complexity of VM migration cost estimation in networking and or the energy consumption estimation of moving specific VM on a predetermined PM.

Hybrid: A multi-criteria decision-making method that takes into consideration available capacity of the PM, the number of VMs on the PM, and the migration delays of VMs. Although this algorithm may give more accurate result it will be more complicated and difficult for practical implementations.

Weighted CPU utilization and VMs on Host: It combines the Host CPU utilization CPU_{Hi} and number of VMs on the Host VM_{Hi} according to following equation [81]:

$$U_{H_i} = \alpha \cdot CPU_{H_i} + \beta \cdot VM_{H_i} \quad \ldots \qquad (17)$$

where U_{Hi} is the utilization of host H_i, α and β are weighted for CPU_{Hi} and VM_{Hi}, respectively. Such that, $\alpha+\beta = 1$, $0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$. And their values are optimized based on workload type with hill claiming method.

The same technique used in [124], by combining of the CPU utilization and the number of VMs according to reward function. Comparing this technique with Hybrid we can notice the following: 1) it reduces the number of required migrated VMs; 2) host with least number of VMs has a better chance to be switched to sleep mode in comparison with a host with more VMs; 3) it depends on both Host utilization and VM number of VMs on that Host. It still needs more computation to find the optimal values of α and β for each Host.

Host Overload Detection

Host overload detection is the process of deciding if a host is considered to be overloaded so that some VMs should be migrated from it to other active or reactivated hosts to avoid violating the QoS requirements. Static utilization thresholds, Adaptive utilization base, and Regression Based are some of the useful techniques [18]. Abdelsamea et al. [1] classified the host overload detection process into, see Table 3:

Static utilization threshold

It is exactly the same as the static threshold in host underload algorithm. The algorithm compares the

Table 3 Host overload detection algorithms

Algorithms		Characteristics
Static utilization threshold [21, 100]		Depend on the mean of the last CPU used. Unsuitable for dynamic and unpredictable workload
Adaptive utilization threshold [18]	Median Absolute Deviation	Depend on statistical dispersion Similar to the static threshold
	Inter-quartile range [130]	Same as Median but compare third and first quartiles Provide poor prediction of host overloading
Regression based algorithms [1, 17]	Local regression algorithms	By fitting simple models to CPU utilization
	regression robust	Use regression to predict the future CPU utilization
	Markov overload detection	Add constraint in estimation increase the computation

latest CPU utilization measurements with a predefined threshold [21]. As discussed in CDC Monitoring tools, this technique is unsuitable for dynamic and unpredictable workloads. e.g. Kashyap *et al* [100] use 0.8 for host CPU overload threshold.

Adaptive utilization threshold

This is done by using an adaptive threshold based on a statistical analysis of historical data of the VMs. Beloglazov [18, 20] proposed two adjustment criteria Median Absolute Deviation (MAD) and Interquartile Range. MAD depends on statistical dispersion, where a PM with large CPU utilization deviations is weighted more heavily than the others. Once the threshold is calculated, the algorithm acts similarly to the static threshold algorithm by comparing the current CPU utilization with the calculated threshold. Interquartile Range follows the same principle of MAD but the distance is calculated by computing the difference between the third and first quartiles in descriptive statistics [130].

Generally, adaptive utilization threshold algorithms are more robust than static CPU utilization threshold algorithms in case of dynamic environments. However, these algorithms provide a poor prediction, and most of them depend on single resource usage value, which can lead to hasty decisions, unnecessary live migration overhead and stability issues [128].

Masoumzadeh and Hlavacs [122] proposed an intelligent and adaptive threshold-based algorithm for detecting overloaded hosts by Dynamic Fuzzy Q-learning (DFQL). The main deference with previous technique that the algorithm benefits from experiment gained by learning procedure to decide better about the numerical value of CPU utilization threshold in the future.

Prediction-based algorithms

They are based on the estimation of the future CPU utilization. They provide better predictions of host overloading but are more complex. Prediction algorithms include:

Local algorithms: this is done by fitting simple models to localized observations of the CPU utilization, in order to build a curve that approximates the CPU utilization.

Robust algorithms: the algorithm estimates the local parameter and uses them to predict the future CPU utilization at the next time step, taking into account the VM migration time which should be estimated [128].

Markov overload detection: In this algorithm, a constraint on the overload time fraction value will be added as a parameter of the algorithm, while maximizing the time between VM migrations, thus improving the quality of VM consolidation but increase the computation [17].

K-nearest neighbor: Farahnakian *et al* [62, 63] proposed two regression methods to predict CPU utilization of a PM. These methods use the LR and the KNN regression algorithms, respectively, to approximate a function based on the data collected during the lifetimes of the VMs. Therefore, they used the function to predict an overloaded or an under-loaded PM for reducing the SLA violations and energy consumption.

Host overload detection will become more complex problem when a VM has multiple (e.g. CPU, memory, storage capacity, etc.). As an example, authors [149] propose to use Multi-Criteria Decision Making (MCDM) algorithms as a promising to tackle the problem of VM selection that involves multiple computing resources. In this approach, these resources can represent the multiple criteria in the problem domain of VM sections. Using common MCDM algorithms such as Analytic Hierarchy Process (AHP) and Analytic Network Process (ANP), pair-wise comparisons can be performed so a decision maker (e.g. a cloud engineer or a data scientist) can determine the importance of each computing resource by assigning a weight (e.g. 1 to 10) or he/she determines the influence of one criterion on the others. There are not too many efforts made to tackle the VM selection problem based on multiple resources. The current research focuses on VM as a whole component such as the work presented in [106, 169].

VM Selection Subsystem

In dynamic VM consolidation base energy consumption, energy saving will be done through migrating all VMs from low usage host (underloaded) to switch it to sleep mode or it can be shut down. In contrast, due to the variability of workloads and keeping SLA, if a host usage is high (overloaded) some of the VMs moved to hosts which have a moderate load [59, 81]. VM Selection is the process of selecting one or more VMs from the full set of VMs allocated to the server and the future predicted new VMs, which must be located or reallocated to other servers [19]. VM selection answers to the two simple questions: which VMs to migrate, and where. The main function of VM selection subsystem is to determine the best subset of VMs to migrate that will provide the most beneficial system reconfiguration in terms of energy consumption and many other parameters like security and bandwidth.

The VM selection is a process of picking the best one or more VMs from overloaded PMs to migrate them with minimum energy consumption constraints. Unlike Abdelsamea et al [1] who classified the selection techniques into techniques with fixed or multiple criteria, this survey will divide the techniques into conventional and ML approaches, Conventional VM Selection Techniques and Machine Learning VM Selection Techniques, respectively.

Conventional VM Selection Techniques

Conventional in this section means the techniques without ML application. The first three of them described in details by Beloglazov [18, 21].

Random Choice (RC): This is the simplest policy, in which the selection of VM is based on the uniform random process [18].

Dynamic Management Algorithm (DMA): To reduce the processing overhead, the VM selection process should be based on the CPU utilization of VMs, i.e. the VMs with the lowest CPU utilization is selected.

Minimum Migration Time (MMT): Based on minimum time to complete the migration process relative to other VMs allocated on the same host, VM will be selected. Beloglazov [18, 20] suggested that the migration time is the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host.

Maximum Correlation (MC): In this algorithm, the VMs will be selected by calculating the probability correlation between resources usage by an application that runs on the oversubscribed server. If there is a higher correlation between the resource usages by applications running on an oversubscribed server, will lead to higher probability of server being overloaded. It means that if the correlation of the CPU utilization of VMs of a particular host is high then the probability of this host being overloaded is also high [131].

Constant Fixed Selection (CFS): It is almost the same as Random Choice policy but the selection will be constant either first, center or last position in the VM list which should be moved from the overloaded host [156].

Although DMA, MMT and CFS are indentation techniques and sufficient in static cloud environments but are not suitable for decision-making in dynamic environments. While MC need more calculation but give best selection approach because it selects the VM which will have less predicted correlation with a CPU usage of current PM.

Multi-objective optimization: Song *et al* [159] proposed a multi-objective optimization model based on analysis of the impact of CPU temperature, resource usage and power consumption in VM selection. The developed algorithm was evaluated by comprehensive experiments that are based on VM monitor Xen. Their results showed that combining all these factors can achieve the best VM selection with respect to resource usage, CPU temperature, and power consumption.

Machine Learning VM Selection Techniques

Most of these techniques are based on fuzzy logic because the selection process is a decision-making problem.

Fuzzy Q-Learning (FQL) [123]: It is an online decision making strategy. Its principle is to integrate multiple VM selection techniques and dynamically choose suitable VM selection approach for a current state. In other words, it is the process to find the optimal strategy to be used in the VM selection process [1].

Fuzzy VM selection [132]: This method has been proposed to select VM from an overloaded host. It incorporates the migration control in Fuzzy VM selection method. Simulation based on the CloudSim platform was used to show that this method provides the best performance considering all parameters. The main difficulty in the FS is to formulate the problem. This is because the VM selection process should take into consideration as many as unrelated elements. Fuzzy logic is able to relate these elements in a systematic manner [88].

VM Placement (VMP) Subsystem

Virtual Machine Placement (VMP) is the process of mapping VMs to PMs in such a way that the hosts (PMs) can be utilized to their maximum efficiency. This will help to shut down unused PMs depending on load conditions. Each of the VMP algorithms works well under certain specific conditions. Thus, it is important to choose a technique that suits the needs of the cloud user and cloud provider. Also, the parameters to these algorithms should be properly specified. The performance metrics are measured at both system level and application level. The system level metrics are measured in terms of CPU load and the application level metrics are measured in terms of response time of applications. Physical and virtual machines are characterized by their CPU (MIPS), RAM (MB) and bandwidth (Mbps). The goal of VMP problem is to determine the minimum number of PMs required by the set of VMs.

Fixed mapping VMP during the lifetime of the VM is called static VMP. While allowing to change initial placement due to reach a certain undesired state in the system performance, maintenance, power or load (Reactive), or before it reaches these conditions (Proactive) is called dynamic VMP [121].

Pires and Bar´an [115] listed all publications, up-to 2014, in different applications such as energy-efficiency, SLA, cloud service markets, QoS and carbon dioxide emissions. Also, there are comprehensive surveys can be found in [1, 6, 53, 121, 160, 184]. The following subsection will review the most up-to-date algorithms used for dynamic VMP to the maximum power efficiency of PMs in a data centre. In another word, we are going to review VMP algorithms used to map VMs to PMs in such a way the servers (PMs) can be utilized to their maximum power efficiency in a single CDC.

To simplify the review process, several classifications of VMP schemes proposed in the literature. In this work, VMP techniques have classified based on solution techniques into *Deterministic*, *Heuristic*, *Approximation* and *Meta-heuristic* algorithms [137]. We are going to give the simple description of each solution technique in general with a simple example without describing all technique in details. By the end of this section, a list of most recent literature will summarize the techniques, considered resources, a new aspect of the technique, evaluation process and comparison with other, Table 4.

Table 4 VMP algorithms based solution techniques

	Solution Category	References	Considered Resources	Aspect	Evaluation	Performance Better Than
Deterministic	Linear Programming	[37]	CUP, storage and network	PMs' resources subject to linear function	Simulation	Non
	Integer Linear Programming	[170]	network	Tree and forest formulated on graph	Simulation	BF
	Constraint Programming	[72]	CPU	Objective functions for optimality	Simulation	BF heuristic
	Constraint Programming	[95]	CPU and bandwidth	maximum link utilization optimization	Simulation	BFD and Random algorithms
	Convex function	[49]	CPU	PMs meeting all tasks' demands	Google Trace data	BF with Min, Max and random
Heuristic	Bin-packing	[162]	CPU, memory and network	Volume to size ratio	Simulation	FF, BF, FFD
	Bin-packing	[40]	CPU and memory	Redesign	CloudSim	
	PABFD	[20]	CPU	on-line	CloudSim	FF, BF, FFD
	Enhanced FFD	[11]	CPU	VM reuse strategy	CloudSim	FFD and Round-Robin
	PABFD + minimum correlation coefficient	[68]	CPU	PABFD with minimum correlation VMs	CloudSim	PADFB
Approximation	Utilization Aware BFD	[61]	CPU and memory	Use VM and PM prediction	CloudSim	Modified BFD, Modified FFD
	Utilization Aware BFD	[74]	number of network and server resources	Network connection	Simulation	FFD
	Utilization Aware BFD	[51]	Number of VMs	Use VM and PM prediction	Simulation	CPLEX Optimization Studio [84]
Meta-heuristic	GA	[114]	CPU, memory, bandwidth and storage	multi-objective formulation of the VMP	Itaipu Technological Park DC	brute force exhaustive search algorithm
	Hybrid genetic algorithm (HGA)	[168]	CPU and network	scalable with problem size	Simulation	GA
	Non-dominated Sorting Genetic Algorithm (NSGA)	[3]	CPU, memory and bandwidth	Non-dominated Sorting GA	Simulation	GA
	Ant colony optimization (ACO)	[65]	CPU, memory and bandwidth	Modeling Multidimensional bin-packing	Simulation	multi-dimensional bin-packing
	Ant Colony System (ACS)	[59]	CPU	Find near optimal	CloudSim	SLA violation and migrations
	multi-objective ACS	[69]	CPU, network and storage	Vector-algebra based resource utilization	Simulation	single-objective ACO, FFD

Deterministic Algorithms

This kind of algorithm is based on optimization techniques where VM sizes and constraints are pre-defined. The problem can be modeled as follows:

Let P is set of PMs, V is set of VMs, v_j is the maximum number of VMs can be hosted on $P_j \in P$. The following equation allows to model the objective function as follows [127]:

$$\textbf{Max}\left\{\sum_{i=1}^{P}\sum_{j=1}^{V} a_{ij}\cdot v_j\right\} + \textbf{Min}\left\{\sigma_p\right\} \tag{18}$$

Subjected to the following constraint:

$$\sum_{j=1}^{|V|} a_{ij} \le P_i, \ \ \forall_i \in 1, 2, \ldots, |P| \tag{19}$$

where a_{ij} are Boolean variables to assign the VM_j to the PM_i. σ_P is the standard deviation of the distribution of the VMs among the active PMs (i.e. each active PM exists at least has one of the $a_{ij} = 1$). σ_P can be given by:

$$\sigma_P = \sqrt{\frac{\sum_{i=1}^{P}\sum_{j=1}^{V}\left(a_{ij}\cdot v_j - \overline{V}\right)^2}{|V|-1}} \tag{20}$$

and $|V|$ given by

$$|\overline{V}| = \frac{\sum_{i=1}^{P}\sum_{j=1}^{V} a_{ij}\cdot v_j}{|V|} \tag{21}$$

Optimize the objective function, Eq. 18 will be done by minimizing σ_P, this means to maximize the number of assigned VMs to a given PM. This will lead to reducing the number of active servers and then the power consumption.

Algorithms listed in this category are: Linear programming (LP) [37], Binary integer programming (BIP) [170], Constraint programming [72, 95], Convex optimization [49], Pseudo-Boolean optimization and many other algorithms.

The simplest algorithm used is the LP, where the performance goal is linearly related to the placement of VMs. For example, the optimal placing of new VMs on different PMs with the assumption that the minimal number of PMs required and the resources in each server subject to a linear function [113]. In BIP each variable can only take on the value of 0 or 1, i.e. it represents the selection or rejection of a placement (PM). Constraints programming is to design some extension constraints for the LP, like restrict the number of VMs in a single PM, or limiting the number of VM migrations, etc. While convex optimization is a special class of mathematical optimization problems, that includes both least-squares and LP problems [25, 26]. The general common issues in these algorithms are:

- Need a long time to generate the optimal solution, depending on a number of constraints.
- Very useful in static VM consolidation, because it required exact size and constraints of the VM.

Heuristic Algorithms

Heuristic algorithms are used to find a solution step by step by taking a local best decision. In other words, the Np hard bin-packing problem principle is based on local best decision to pack a series of VMs having specified sizes into a least possible number of PMs [40]. Most approaches used in the literature are based on classic packing algorithms like First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD), First-Come First-Served (FCFS), and Best Fit Decreasing (BFD) algorithms [181]. These algorithms can be classified into online and offline algorithms. In online algorithms, such as FF, assign VMs to PMs as they arrive. There is no need for prior knowledge of the VMs which will be submitted in the future. While offline algorithms, which is useful in DCVM, do have the knowledge about all the VMs to be assigned thus they are able to sort them beforehand. In offline, such as FFD, VMs are assumed to arrive sequentially and are placed on the first PM which can accommodate them, starting from the first PM sorted according to a predefined metric, power efficiency in our case [66].

Algorithm 2 represents a simple Power Aware Best Fit Decreasing (PABFD) algorithm proposed by Beloglazov and Buyya [21]. This algorithm sorts the VMs according to their CPU utilization in decreasing order and then for each VM it checks all the PMs and finds the suitable PM where the increase of power consumption is minimum.

The quality of a polynomial time approximation algorithm A, is measured by its approximation ratio $R(A)$, to optimal algorithm OPT, Eq. 22 [53, 162].

$$R(A) = \lim_{n\to\infty}\ \sup_{OPT(L)} = n\frac{A(L)}{OPT(L)} \tag{22}$$

Where $A(L)$ is the number of PMs used under the algorithm A, $OPT(L)$ is the number of PMs used under optimal algorithm OPT and L is the list of VM sequence.

Algorithm 2 PABFD

```
Inputs: PMList, VMList
Outputs: VMs allocated with min power
for each VM in VMList do
    minPower ← MAX
    allocatedPM ← null
    for each PM in PMList do
        if PM has enough resources for VM then
            power ← estimatePower(PM, VM)
            if power < minPower then
                allocatedPM ← PM
                minPower ← power
            end if
        end if
        if allocatedPM ≠ null then
            allocation.add(VM.allocatedPM)
        end if
    end for
end for
```

Significant research has been done to improve bin-backing algorithms, like those used by CloudSim [18], Chowdhury *et al* [40] and Farahnakian *et al* [60]. But all of them have the following characteristics [125, 181]:

- Very fast and need fewer computation resources because it is done by comparing the VM's demand with server's available capacity, without considering the balanced utilization of multidimensional resources.
- Not guaranteed to be optimal but can be considered for immediate goals or suboptimal solutions.
- Minimum number of PMs used will not necessarily the solution for less energy because this totally depends on the PMs Hardware.

An interesting work has been done to improve these algorithms by extending them, such as, but not limited to:

- Extend classical BF heuristic by taking into account VMs' release times in order to reduce the number of active PMs over time [45–47].
- Use a heuristic algorithm to optimize network performance and reduce the energy consumption of PMs and network elements [55]
- Use multiple resources best fit and worst fit policies taking into account VMs' CPU, RAM, disk and bandwidth [138]
- Heuristic algorithms based on PMs fault-aware scheduling [154]

Approximation Algorithms

These algorithms depend on prediction algorithms where prices of resources are not known but for example, their probability distributions can be estimated such that network bandwidth of the VM as in [181]. Unlike deterministic algorithms which can be implemented using mean or maximum of the demand as its estimated value. As an example, Farahnakian *et al* [61] formulate a VM consolidation as a bin-packing problem considering both the current and future utilization of resources. The future utilization of resources was predicted using a KNN regression based model. Their experimental results show that this approach provides a substantial improvement over other heuristic algorithms in reducing energy consumption, a number of VM migrations and number of SLA violations.

Authors in [74] suggested a heuristic algorithm to solve multi-dimensional energy-efficient resource allocation. In their approach, they create multiple copies of VMs and then uses dynamic programming and local search to place these copies on the PMs. Local Search attempts to reduce the cost of energy by shutting down the underutilized servers, while dynamic programming initially identifies the number of VM clones to be placed on PMs. They minimize the length of networks connecting of all PMs to minimize the total connection costs and reduce energy.

Dalvandi *et at* [51], reduce power consumptions by maximizing the benefit from the overall traffic send by VMs to the root through proposing a time-aware VMP routing algorithm. Where each task requires a given number of network resources and server resources for a time duration. They formulate this problem as a mixed integer LP optimization based on a power utilization model. A heuristic algorithm is developed to fix the optimization issue. The main advantages of these approaches are:

- Not need to predefine constrained, because they depend on the probability of the parameters.
- Need less computation than the deterministic algorithms but more than heuristic ones.
- Useful for dynamic VM consolidation.

Meta-heuristic

Meta-heuristic or biology-based optimization is a way to solve the bin-packing problem with certain constraints. These approaches are based on Biology optimization techniques like Genetic algorithm (GA) [3], Ant Colony Optimization (ACO) method [65], and Hybrid Genetic Algorithm (HGA) [168]. These algorithms require more computation time and higher computing resources as compared to classic packing problem [1].

Tang and Pan [168] used an HGA for the energy efficient VM placement problem on PMs with communication network consideration in data centers. They developed a Java program that can randomly generate VM placement problems of different configurations, fixed and variable number of PMs with 20 and 80 random VMs. The experimental results show that the HGA is better than the original GA.

Feller *et al* [65] developed a multidimensional bin packing to place VMs into the least number of PMs necessary for the current workload based on ACO.

Genetic algorithms, nondominated sorting GA I and II were compared with common solution representation [3]. The simulation shows that the nondominated sorting GA II gives good and wind range of solutions compared to the former algorithms.

Lopez and Baran [114] proposed three objective functions to apply multi-objective mimetic in solving VMP problems, where the critical application was considered for a specific SLA. They concluded that by increasing the percentage of VMs with critical SLA, the number of solutions and execution time to find these solutions decrements.

VM Migration

Performing VM live migration in data centers is not a straightforward task. Several challenges need to be addressed such as maintaining a reasonable level of QoS requirements and optimum resource utilization for energy conservation [115]. The live migration process has been modeled and quantified in several articles. Two criteria can be identified for efficient VM migration: down time during the migration and the migration time itself [24]. Down time refers to the time when services are down due to the migration process. Migration time refers to the time required to transfer a VM from a node to another within a cluster [129]. Both criteria have low tendency meaning that we seek to minimize their values so that the migration process does not interrupt the provisioning process.

Different techniques have been used to execute live migrations. Some well-known techniques are described below:

Pure stop and copy technique: In this This technique uses CPU usage its content to the destination and then the new VM is restarted. This process is simple but the service downtime could be large and it is proportional to the allocated memory to the migrated VM [24].

Post copy technique: In this technique, only essential data structures are transferred to the destination which can be restarted. The other parts are migrated on demand across the data centre. This technique minimizes the migration downtime but the migration time still takes much time [41].

Pre-copy technique: This technique involves iteratively copying memory from the source VM to the destination server while keeping the migrated VM running. The iterative process is performed to consider any updates that could occur in the migrated VM so that updates are available at the destination server [82].

Hybrid technique: This technique combines the pre-copy and post-copy algorithms. Besides transferring the VCPU registers and devices states in post-copy, a small subset of memory is also transferred which is frequently accessed by the VM. Advantages of both the pre-copy and post copy can be exploited in the hybrid algorithm which makes it more suitable for VM migrations [151].

Due to the fact that live migration costs energy and any reconfiguration aims to reduce energy consumption, one of the most important tasks is to select those VMs whose replacements save at least as much energy as their migrations cost. To make energy efficient, decisions in terms of VMs migration requires a migration cost model that enables to quantify the energy overhead of VM live migration in advance. The LR technique is derived to model the energy overhead of live migration [71, 75, 83, 164]. The following model is used for energy consumption during live migration in a heterogeneous cluster:

$$\begin{aligned} E_{migration} &= E_{source} + E_{dest} \\ &= (\alpha_{source} + \alpha_{dest}) V_{migration} + \beta_{source} + \beta_{dest} \end{aligned} \tag{23}$$

where α_{source}, α_{dest}, β_{source}, and β_{dest} are parameters to be trained. The minimization function of energy consumption is modeled as follows:

$$\min \sum (Copr + Cmgr) \tag{24}$$

where C_{opr} denotes operational energy consumption cost and C_{mgr} denotes migration energy consumption cost. The C_{mgr} is the sum of the cost due to the size of system resource and the cost due to the bandwidth usage.

Akiyama *et al* [8] proposed to integrate a performance model and an energy model of live migration to simulate dynamic VM placement. The proposed performance model estimates how long a live migration takes under a given environment. The input is the size of the target VM, network bandwidth

available for migration, and workload running on the VM. This model is used to simulate dynamic VM

placements. Energy model estimates how much energy is lost by performing live migrations to process dynamic VM placements.

The input is a number of memory pages transferred during a live migration. The advantage of their approach is the combination of energy consumption models of the placement and migration operations since both operations complement each other in CDC environments. Moreover, it can simulate pre-copy live migration, as it works perfectly as a pre-copy live migration by reusing non-updated memory in the initial memory transfer. However, their model needs to be tested based on the hybrid migration technique where both the pre-copy and post copy algorithms are fused.

Network Effect

With increasing numbers of servers and switches in data centres, the communication bandwidth has to scale exponentially to meet increasing requirements of data accessing, processing and storing. On the other hand, Yang *et al* [180] reported that thousands of MapReduce programs implemented and run in different applications such as Yahoo, Facebook, Google's data centers every day, and petabytes of daily data flow are transmitted among distributed jobs within CDC. This incurs a very high cost and energy wastes. The energy consumption at the switches tier can be calculated as follows [34]:

$$P_{switch} = P_{chassis} + n_{inecards} \cdot P_{inecards} + \sum_{i=1}^{R} n_{ports} \cdot P_r \tag{25}$$

where $P_{chassis}$ and P_{switch} denote the power consumed by the switch-based hardware and an active line card. P_r denotes the power consumed by the live port which is running at the rate *r*. P_r denotes the switch's scaling transmission rate.

One way to deal with this scenario is to find efficient and cost effective approaches. In data centres, there are two main approaches for *network setup*: switches-centric and servers-centric [111].

Switches-centric: It implements the hierarchical network topology which is constructed from off-shelf components. In this approach, servers are positioned so they are at the leaves of the hierarchy of the network. The advantage of such approach is better to load balancing and less prone to bottleneck [10, 141]. The disadvantage is the limitation in terms of the scalability because of the size of routing tables in switches [111].
Servers-centric: It implements the Cayley graph [56]. The CDC network resides within servers as opposed to the first approach (switches-based network). It provides programmable capabilities and intelligent routing. Thus, servers not only process applications and data but also act as routers to relay traffic. The main advantages of this approach are: 1) the low cost of interconnections in data centres besides the ability to remove bottleneck at the architectural level. 2) It is highly scalable so expanding the network does not require to physically modify/upgrade existing servers. Liao *et al* [111] presented DPillar, highly scalable network architecture for data centres. DPillar is built with dual-port servers and n-port switches where server columns and switch columns are alternately placed along a cycle. One disadvantage of DPillar was not designed to produce a short path routing rather it focuses on simplicity. Erickson *et al* [56] proposed to improve the DPillar algorithm efficiency by developing a single-path routing algorithm that always produces the shortest path. Wang *et al* [176] presented a survey on different network topologies used in data centres and divided their networks into:
- Tree-based topologies: classified into Basic-tree, Fattree, and VL2. It is mainly based on switch routing architecture.
- Recursive-based topologies: classified into DCell, BCube, FiConn, FlatNet, and SprintNet. It is mainly based on server routing architecture.

Although Recursive-based topologies provide a substantial remedy to the problems of the Tree-based topologies, they still have their own shortcomings. DCell is built based on low-level links thus it may cause a bottleneck. BCube is considered a topology with high wiring complexity. FiConn suffers from deficiencies in fault tolerance, network capacity, and long path traffic. FlatNet and SprintNet have low scalability compared to other topologies. The improved version of DPillar topology [111, 134] has the remedy for these problems by providing highly scalable network topology, good fault tolerance, improved bottleneck throughput and latency, and shortest path routing within data centre network.

Analysis of the State-of-the-Art Surveys in the Literature

In this section, we review some of the existing surveys in the literature on VM consolidation operations as they pertain to energy efficiency and consumption models. We will then highlight the differences between the work presented in this paper and those in the literature.

The survey presented by Abdul *et al* [77] focused on energy-efficient resource allocation techniques. They defined four approaches for designing energy efficient cloud data centers: a) reducing energy using energy efficient resource allocation and management of data centers, b) ensuring the performance of infrastructure to reduce the usage of other devices, c) geographically distributing computational loads to meet end users' needs, d) minimizing self-management and flexibility. Key concepts of energy efficient resource allocation were identified:

- Resource adoption policy: the ability of a resource allocation mechanism to adapt to dynamic conditions.
- Objective function: single and composite objective functions. The former deals with the energy consumption dimension and the latter deals with the SLA violation dimension.
- Allocation method: power-aware and thermal-aware allocation methods.
- Allocation operation: service migration and service shutdown categories.
- Interoperability: the capability of energy optimization technique to work across multiple resource types during the resource allocation process.

The paper evaluated and summarized the work that proposes energy-efficient approaches with respect to the defined key concepts of resource allocation mechanisms.

Toni *et al* [126] survey focused on energy efficiency of infrastructure utilities of data centres that power ICT machinery. Two domains are covered, servers and networks.

The authors organized their paper based on two types of energy inefficiency terms, energy loss and energy waste. The four goals are defined for reducing energy:

- Minimizing the input energy that is not consumed by a subsystem.
- Reducing the overhead of supporting systems (e.g. cooling systems).
- Reducing idle run of the system.
- Minimizing energy consumption where the system performs redundant operations.

Based on the above criteria, the paper provided a literature review to find current research directions that focus on energy consumption efficiency in cloud infrastructures. It categorized cloud computing infrastructure software and hardware with a consideration of energy consumption. The paper put each category in the two defined domain in the context of the energy consumption domain, then it defined actions for reducing energy consumption, and presented challenges and research directions.

Dayarathana *et al* [52] presented an in-depth study of the existing work that addresses the problem of power consumption. They defined a general approach to manage data centre energy consumption which consists of four main steps: feature extraction, model construction, model validation and application of the model. The paper presented the surveyed work based on proposed energy consumption models: a) Additive server power models, i.e. models are based on aggregating energy consumed by server components (CPU, memory and I/O devices). b) System utilization models, i.e. models that leverage CPU utilization as their metrics in modeling the whole system power consumption.

c) Systems' performance setting related, i.e. regression based power modeling and queuing theory based. The paper presented work that proposes energy consumption models of a group of servers, data center networks, cooling systems, power condition systems. On the software level, the authors categorized the surveyed models into compute-intensive, data-intensive, communication intensive applications, OS and general software. Although this paper provides a comprehensive survey of energy consumption models. However it does not consider the VM consolidation techniques and operations and how energy consumption is handled and modeled in the literature.

Work that deals with resource management in cloud environments has been reviewed in [93]. It identified several challenges relate to providing predictable performance for cloud-hosted applications, achieving global manageability for cloud systems, engineering scalable resource management systems, understanding economic behavior and cloud pricing, and developing solutions for the mobile cloud paradigm. The surveyed work in this paper covers virtualization environments operations including: a) resource demand profiling, b) resource utilization estimation, c) resource pricing and profit maximization, d) application scaling and management, and e) cloud management systems. However, the paper did not provide information about user behaviors and profiling and their influences on the resource management process.

Uddin *et al* [171] analyzed three energy efficient algorithms for task scheduling to get the most efficient algorithm. The evaluated algorithms are Resource Aware Scheduling Algorithm (RASA), Two Phases Power Convergence (TPPC) and Power Aware Load Balancing (PALB). Three parameters are defined to evaluate the algorithms: a) power efficiency, b) cost effectiveness, and c) the amount of CO2 emissions. The authors focused on dynamics power management techniques.

Zhi-Hui *et al* [182] tackled the problem of cloud resource scheduling. They describe the problem as an NPhard problem whose intractability increases a lot with the increasing of the number of variables if deterministic techniques are used. They presented the taxonomy of three categories of cloud resource management and scheduling: a) scheduling in the application layer, b) scheduling in the virtualization layer, and c) scheduling in the deployment layer. Each layer has been analyzed and challenges have been identified. Hence each layer was divided into subcategories as follows:

- Scheduling in the application layer: includes scheduling for user QoS, scheduling for provider efficiency and scheduling for negotiation.
- Scheduling in the virtualization layer: includes scheduling for load balance, scheduling for energy conservation and scheduling for cost effectiveness.
- Scheduling in the deployment layer: includes scheduling for service placement, scheduling for partner federation and scheduling for data routing.

A state-of-art literature review under each sub category was presented which also illustrates different resource scheduling algorithms. Among the surveyed techniques, the paper focused on Evolutionary Computing as a promising optimization paradigm for solving the cloud resource scheduling problem.

Fischer *et al* [67] presented a survey study on virtualized networks as a promising solution for the high demands of applications and different services in data centres. The virtualized data center provides better management, low cost, and better resource utilization and energy efficiency, and virtualized networks is a subset of the virtualized data centres. A comprehensive review of the existing work in Virtual Network (VN) has been provided under different classifications. Some future research directions of VN was also presented such as virtualized edge data centres, virtual data centre embedding, security, pricing and programmable network. Although this paper introduced a rich survey on VN characteristics challenges and future research directions in CDC networks, there was a little emphasis on energy consumption and efficiency while reviewing current work or suggesting future research.

In [6], a comprehensive review of VM migration approaches in data centres was presented, their strengths, weaknesses and future research. The paper also discussed developing optimal methods for VM migration which are inspected through qualitative and quantitative parameters. The paper provided a review and comparisons of different VM consolidation approaches in the literature based on migration models and migration triggering points. Different server consolidation frameworks that reactively or proactively trigger migration was listed. The authors classified the VM migration optimization into three schemes: a) bandwidth optimization, b) DVFS enabled power optimization, and c) storage optimization. For each scheme, a brief overview of existing work was presented.

The literature review of VMP [115] presented in three optimization approaches: a) the mono-objective which considers the optimization of one objective or multiple objectives one at a time. b) the multi-objective which considers multiple objective functions fused into one objective function. c) the pure multi-objective. Objective functions have been classified based on the studies articles into 5 objectives: energy consumption minimization, network traffic minimization, economic costs minimization, performance maximization and resource utilization maximization.

Many literature surveys focus on some domains in VM consolidation based energy consumption process, as shown in Table 5. So, these surveys do not provide us with a complete plan of how the VM consolidation is implemented. They only emphasize on one or two domains of their interest. In other words, they concentrated on a partial consolidation domain, note cover all VM consolidation process. In this work, we provide an in-depth survey of the most recent techniques and algorithms used in proactive dynamic VM consolidation focused on energy consumption. The survey was presented based on a proposed general framework that can be used in multiple phases of a complete consolidation process. This will help researchers by providing a complete guidance on how these components work together (in a correlative manner) in a CDC environment to meet end user requirements according to the SLAs. Also, it will make easy to focus on fields requires further attention for future research.

Conclusions

This paper presents a comprehensive survey and analysis work on VM consolidation focusing on energy consumption in CDCs. In particular, the paper focuses on proactive dynamic VM consolidations in CDCs with heterogeneous environments. We presented a general framework with multiple phases that achieves a complete consolidation process.

Our framework includes and covers a comprehensive analysis of various techniques and algorithms used in implementing proactive dynamic VM consolidations. Our analysis identified a number of key observations:

- Consideration of QoS parameters related to VM performance such as availability, response time and reliability is a must as part of the consolidation process. It is very important to make sure that the level of QoS is maintained according to SLA while attempts are made to fully utilize data center resources.
- Most algorithms only consider CPU as their primary input. For better performance, these algorithms should be extended to consider other important resources, such as memory, storage, bandwidth, etc.
- Algorithms cannot always be compared to one another, as they may consider different input, operation criteria or goals [53].
- For VM clustering process, Prediction Process, the framework proposed by [86] is based on efficient use of historical VM request, user cluster algorithms, the

Table 5 Analysis of the existing survey papers in the literature

References	Hardware and Application modelling		VM consolidation			Energy Efficiency
		Migration	Placement	VM Selection	Networks	
[115]			√			√
[52, 77, 126, 171]	√					√
[67, 70, 126, 174]					√	
[70]	√				√	√
[5, 6, 52]		√				√
[147]			√		√	
[165]		√				
[93]		√	√			

current state of the data centre and an effective prediction window size. This model should be extended to include long term historical workload time series data and use this data to find the optimal number of clusters and centres of VM size clusters.

- Prediction window size can play an important role in the workload prediction calculations and its accuracy. Dabbagh et al [44, 46] is an important contribution in this area that takes into account prediction windows. More work should focus on further analysis of this problem to evaluate the current prediction models in the literature and use its result to propose an enhanced model to more accurately predict future workloads in CDCs.
- As stated in Clustering Process, identifying the behaviors of cloud users' requests resources strongly influence the overall cloud workload. Uncovering the dependency relationships between users and VMs helps improve the prediction accuracy and excluding unwanted data. More research work should take into consideration this behavior as in [90, 99].
- More work should be done on multi-criteria VM selection models that consider multiple infrastructure resources (e.g. CPU, memory, storage, and bandwidth), especially for energy consumption.
- The capabilities of the PMs in the data center play a key role in the consolidation process. Existing research mostly focused on workload characteristics. More focus is needed on taking into account PM characteristics [119].
- Most VMP algorithms compare their proposed algorithms against trivial heuristics. A comparison against real data can provide more meaningful results, which in return can result in improved algorithms.
- Simplifying assumptions made by algorithm designers, such as homogeneity of PMs or ignorance of PMs' power consumption characteristics degrade algorithm performance in realistic settings. In particular, the heterogeneity of PMs in terms of capacity and power efficiency needs to be taken into account when designing a VMP algorithm.
- Most of VMP algorithms and techniques have neglected the security related objective in the VM placement operations [121]. Security is one of the crucial factors which should be considered in the future VMP researches and studies.
- There are different VM selection criteria, each of them has weakness and strength for different application and specifications, VM Selection Subsystem. It is useful to have a rule base system, using fuzzy logic for example, to improve the process of selection between these techniques according to environment states [40, 161].
- Cloud computing QoS aware resource allocation polices plays an important role in energy efficient allocation of resources. A comprehensive study of services offered by cloud and workload distribution is needed to identify a common pattern of behaviors [77].
- VM migrations usually occur when there is over/under utilization of the resources. Extra VM migration may affect energy efficiency, leading to further power consumption. So, VM migration is a very critical process that should be optimally done to avoid unnecessary VM migration [9], and attempt to balances energy saving energy resulting from turning off PMs not in use with energy use required for the migration process.
- Failures due to power outages or network component are called correlated failures. The impact of these failures can cause reliability to be overestimated by at least two orders of magnitude [154]. Correlated failure impact on energy consolidation needs more attention in future research.

Abbreviations

ACO: Ant colony optimization; AHP: Analytic hierarchy process; AIC: Akaike information criterion; ANFIS: Adaptive neuro-fuzzy inference system; ANN: Artificial neural network; ANP: Analytic network process; AR: Auto-regressive; ARIMA: Auto-regressive integrated moving average; ART: Association rules technique; BF: Best fit; BIC: Bayesian information criterion; BIP: Binary integer programming; CDC: Cloud data centre; CFS: Constant fixed selection; CMS: Cloud management system; CPU: Central processing unit; CV: Cross validation; DCIM: Data center infrastructure manager; DFQL: Dynamic Fuzzy Q-learning; DMA: Dynamic management algorithm; DWNN: Weighted nearest neighbors for differenced data; ELM: Extreme learning machine; ESM: Exponential smoothing model; FCFS: First-Come First-Served; FCM: Fuzzy C-means; FF: First fit; FFD: First fit decreasing; FQL: Fuzzy Q-Learning; FS: Fuzzy system; FWP: Frequent workload pattern; GA: Genetic algorithm; GFM: Gray forecasting model; HGA: Hybrid genetic algorithm; HMM: Hidden Markov Model; HTP: History table predictor; KNN: K-nearest neighbor; LP: Linear programming; LR: Linear regression; LTI: Linear time-invariant; MA: Moving average; MAD: Median absolute deviation; MC: Maximum correlation; MCDM: Multi-criteria decision making; MIPS: Million instructions per second; ML: Machine learning; MMT: Minimum migration time; PABFD: Power aware best fit decreasing; PALB: Power aware load balancing; PCA: Principal component analysis; PM: Physical machine; QoS: Quality of service; RASA: Resource aware scheduling algorithm; RBF: Radial basis function; SLA: Service level agreement; SMM: Statistical metric model; SVM: Support vector machine; SWF: Stochastic Wiener Filter; SWM: Sliding window method; TPPC: Two phases power convergence; VM: Virtual machine; VMP: Virtual machine placement; VN: Virtual network; WNN: Weighted nearest neighbors

Acknowledgements

Authors are grateful to the reviewers of this manuscript for their expert advice. Authors are thankful to Department of Computer Science at Ryerson University, for support.

Authors' Contributions

The authors made substantive intellectual contributions to the research and manuscript. SI carried out most of the survey of the available literature and drafted the manuscript. RK is responsible for the overall technical approach and architecture, editing and preparation of the paper. AM formulated the concepts and the structure for the paper. He is responsible for analysing the results, and comparison with related work. Editing and revision of the paper. All authors read and approved the final manuscript.

Authors' information

Salam Ismaeel is presently a researcher and PhD Candidacy in the School of Computer Science at Ryerson University, Toronto, Canada. Currently his is working as a Senior Data Scientist at Ontario Government, Canada. Ismaeel served in academic positions for more than 15 years at the University of Technology, University of Baghdad and Future University. He was the dean of the Faculty of Computer and Informatics Engineering in SPY. He received the B.S. and M.S. in Computer and System Engineering form University of Technology, Iraq in 1995 and 1997 respectively. He received his 1st Ph.D. degree in Computer Engineering from AlNahrain University in 2003. His research interests span various topics in the areas of Cloud Computing, Energy-Aware Computing, Big Data modelling and analysis. He has won the Graduate Development Award (2015), paper of the month in FOS at Ryerson University. He is an IEEE member; reviewer in Applied Soft Computing Journal; and member in IEEE International Humanitarian Technology committee.

Raed Karim received his B.Sc., M.Sc. and Ph.D. degrees in Computer Science from Ryerson University in Toronto, Ontario, Canada in 2009, 2011 and 2015, respectively. He is currently working as an IT consultant focusing on Big Data Science and Data Analytics and Project Management. His research interests include cloud computing, QoS based service selection, energy efficiency computing in data centres, big data analytics, SOA, machine learning and recommender systems. He has published in various international conferences such as IEEE SERVICES, IEEE SCC, IEEE Cloud and Autonomic Computing, IEEE CLOUD, IEEE SOCA. He is a member of IEEE and ACM.

Ali Miri has been a Full Professor at the School of Computer Science, Ryerson University, Toronto since 2009. He is the Research Director, Privacy and Big Data Institute, Ryerson University, an Affiliated Scientist at Li Ka Shing Knowledge Institute, St. Michaels Hospital, and a member of Standards Council of Canada, Big Data Working Group. He has also been with the School of Information Technology and Engineering and the Department of Mathematics and Statistics since 2001, and has held visiting positions at the Fields Institute for Research in Mathematical Sciences, Toronto in 2006, and Universite de Cergy-Pontoise, France in 2007, and Alicante and Albecete Universities in Spain in 2008. His research interests include cloud computing and big data, computer networks, digital communication, and security and privacy technologies and their applications. He has authored and co-authored more than 200 referred articles, 6 books, and 5 patents in these fields. Dr. Miri has chaired over a dozen international conference and workshops, and had served on more than 80 technical program committees. He is a senior member of the IEEE, and a member of the Professional Engineers Ontario.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 10 December 2017 Accepted: 17 May 2018
Published online: 18 June 2018

References

1. Abdelsamea A, Hemayed EE, Eldeeb H, Elazhary H (2014) Virtual machine consolidation challenges: A review. International Journal of Innovation and Applied Studies 8(4):1504–1516
2. Aceto G, Botta A, De Donato W, Pescapè A (2013) Cloud monitoring: A survey. Comput Networks 57(9):2093–2115
3. Adamuthe AC, Pandharpatte RM, Thampi GT (2013) Multi-objective virtual machine placement in cloud environment. In: Cloud & Ubiquitous Computing & Emerging Technologies (CUBE), 2013 International Conference on. IEEE, pp 8–13
4. Adhikari R, Agrawal R (2013) An introductory study on time series modeling and forecasting. arXiv preprint arXiv 1302:6613
5. Ahmad RW, Gani A, Hamid SHA, Shiraz M, Xia F, Madani SA (2015) Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. J Supercomput:1–43
6. Ahmad RW, Gani A, Hamid SHA, Shiraz M, Yousafzai A, Xia F (2015) A survey on virtual machine migration and server consolidation frameworks for cloud data centers. J Network Comput Appl 52:11–25
7. Ajila S, Bankole A (2013) Cloud client prediction models using machine learning techniques. In: Proceedings of the 2013 IEEE 37th Annual Computer Software and Applications Conference (COMPSAC), pp 134–142
8. Akiyama S, Hirofuchi T, Honiden S (2014) Evaluating impact of live migration on data center energy saving. In: Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on, pp 759–762
9. Al-Dulaimy A, Itani W, Zekri A, Zantout R (2016) Power management in virtualized data centers: state of the art. J Cloud Comput 5(1):6
10. Al-Fares M, Loukissas A, Vahdat A (2008) A scalable, commodity data center network architecture. ACM SIGCOMM Comput Communication Rev 38(4):63–74
11. Alahmadi A, Alnowiser A, Zhu MM, Che D, Ghodous P (2014) Enhanced first-fit decreasing algorithm for energy-aware job scheduling in cloud. In: Computational Science and Computational Intelligence (CSCI), 2014 International Conference on, vol 2. IEEE, pp 69–74
12. Aljebory K, Ismaeel S, Alqaissi A (2009) Implementation of an intelligent SINS navigator based on ANFIS. In: Systems, Signals and Devices, 2009. SSD '09. 6th International Multi-Conference on, pp 1–7
13. Amarilla A (2018) Scalarization methods for many-objective virtual machine placement of elastic infrastructures in overbooked cloud computing data centers under uncertainty. arXiv preprint arXiv 1802:04245
14. Arianyan E, Taheri H, Sharifian S (2015) Novel energy and SLA efficient resource management heuristics for consolidation of virtual machines in cloud data centers. Comput Elect Eng
15. Arora S, Chana I (2014) A survey of clustering techniques for Big Data analysis. In Confluence The Next Generation Information Technology Summit (Confluence), 5th International Conference-. IEEE, p 59–65
16. Belady C (2011) Projecting annual new datacenter construction market size. Technical Report. Microsoft Corp., US
17. Beloglazov A, Buyya R (2013) Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. IEEE Transactions on Parallel and Distributed Systems 24(7):1366–1379
18. Beloglazov A (2013) Energy-efficient management of virtual machines in data centers for cloud computing, Ph.D. thesis, Department of Computing and Information Systems. The University of Melbourne
19. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems 28(5):755–768
20. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency and Computation: Practice and Experience 24(13):1397–1420
21. Beloglazov A, Buyya R (2015) Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds. Concurrency and Computation: Practice and Experience 27(5):1310–1333
22. Benesty J, Chen J, Huang YA, Doclo S (2005) Study of the Wiener Filter for Noise Reduction. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 9–41
23. Bey KB, Benhammadi F, Mokhtari A, Guessoum Z (2009) CPU load prediction model for distributed computing. In: Proceedings of The 8th IEEE International Symposium on Parallel and Distributed Computing. (ISPDC'09), pp 39–45
24. Bose SK, Sundarrajan S (2009) Optimizing migration of virtual machines across data-centers. In: Parallel Processing Workshops, 2009. ICPPW '09. International Conference on, pp 306–313
25. Boyd S (2015) EE364a: Lecture Notes: Convex Optimization I. Electrical Engineering Department, Stanford University. http://web.stanford.edu/class/ee364a/
26. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge, Cambridge University Press
27. Breitgand D, Da Silva DM, Epstein A, Glikson A, Hines MR, Ryu KD, Silva MA (2012) Dynamic virtual machine resizing in a cloud computing infrastructure. US Patent App 13(621):526
28. Breitgand D, Da Silva DM, Epstein A, Glikson A, Hines MR, Ryu KD, Silva MA (2018) Dynamic virtual machine resizing in a cloud computing infrastructure. US Patent 9,858,095
29. Brockwell PJ, Davis RA (2013) Time series: theory and methods. Springer Science & Business Media
30. Brown RG, Hwang PY Introduction to random signals and applied Kalman filtering. Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997. 1 (1997) Calheiros, R., Masoumi, E., Ranjan, R., Buyya, R.: Workload prediction using ARIMA model and its impact on cloud applications' QoS. IEEE Transactions on Cloud Computing 99:1, 2014–11

31. Canali C, Lancellotti R (2014) Exploiting ensemble techniques for automatic virtual machine clustering in cloud systems. Automated Software Engineering 21(3):319–344
32. Canali C, Lancellotti R (2014) Improving scalability of cloud monitoring through pca-based clustering of virtual machines. Journal of Computer Science and Technology 29(1):38–52
33. Cao B, Gao X, Chen G, Jin Y (2014) Nice: Networkaware vm consolidation scheme for energy conservation in data centers. In: Proceeding of 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS). IEEE: 166–173
34. Cao J, Fu J, Li M, Chen J (2014) CPU load prediction for cloud environment based on a dynamic ensemble model. Software: Practice and Experience 44(7):793–804
35. Cetinski K, Juric MB (2015) Ame-wpc: Advanced model for efficient workload prediction in the cloud. Journal of Network and Computer Applications 55:191–201
36. Chaisiri S, Lee BS, Niyato D (2009) Optimal virtual machine placement across multiple cloud providers. In: Services Computing Conference, 2009. APSCC 2009. IEEE, IEEE Asia-Pacific, pp 103–110
37. Chang YC, Chang RS, Chuang FW (2014) A predictive method for workload forecasting in the cloud environment. In: Advanced Technologies, Embedded and Multimedia for Human-Centric Computing. Springer, pp 577–585
38. Chen Z, Zhu Y, Di Y, Feng S (2015) Self-adaptive prediction of cloud resource demands using ensemble model and subtractive-fuzzy clustering based fuzzy neural network. Computational Intelligence And Neuroscience
39. Chowdhury MR, Mahmud MR, Rahman RM (2015) Implementation and performance analysis of various vm placement strategies in cloudsim. Journal of Cloud Computing 4(1):1–21
40. Clark C, Fraser K, Hand S, Hansen JG, Jul E, Limpach C, Pratt I, Warfield A (2005) Live migration of virtual machines. In: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2. USENIX Association, pp 273–286
41. Cui, J., Liu, S.f., Zeng, B., Xie, N.M.: A novel grey forecasting model and its optimization. Applied Mathematical Modelling 37(6), 4399–4406 (2013)
42. Da Cunha Rodrigues G, Calheiros RN, Guimaraes VT, Santos GLD, de Carvalho MB, Granville LZ, Tarouco LMR, Buyya R (2016) Monitoring of cloud computing environments: Concepts, solutions, trends, and future directions. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. SAC '16, ACM, New York, pp 378–383.
43. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2015) Energy-efficient resource allocation and provisioning framework for cloud data centers. IEEE Transactions on Network and Service Management 12(3):377–391
44. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2015) Exploiting task elasticity and price heterogeneity for maximizing cloud computing profits. IEEE Transactions on Emerging Topics in Computing PP(99):1
45. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2016) An energy-efficient vm prediction and migration framework for overcommitted clouds. IEEE Transactions on Cloud Computing PP(99):1
46. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2014) Release-time aware vm placement. In: Globecom Workshops (GC Wkshps), 2014. IEEE, pp 122–126
47. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2015) Efficient datacenter resource utilization through cloud resource over commitment. Memory 40(50):1–6
48. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2015) Online assignment and placement of cloud task requests with heterogeneous requirements. In: Global Communications Conference (GLOBECOM), 2015 IEEE. IEEE, pp 1–6
49. Dabbagh M, Hamdaoui B, Guizani M, Rayes A (2015) Toward energy-efficient cloud computing: Prediction, consolidation, and overcommitment. IEEE Network 29(2):56–61
50. Dalvandi, A., Gurusamy, M., Chua, K.C.: Time-aware vm-placement and routing with bandwidth guarantees in green cloud data centers. In: Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on. vol. 1, pp. 212–217. IEEE (2013)
51. Dayarathna M, Wen Y, Fan R (2016) Data center energy consumption modeling: A survey. IEEE Communications Surveys Tutorials 18(1):732–794
52. De Maio V, Kecskemeti G, Prodan R (2015) A workloadaware energy model for virtual machine migration. In: 2015 IEEE International Conference on Cluster Computing. IEEE, pp 274–283
53. Di S, Kondo D, Cirne W (2012) Host load prediction in a google compute cloud with a bayesian model. In: in Proceeding of The IEEE International Conference for High Performance Computing, Networking, Storage and Analysis (SC), pp 1–11
54. Dong J, Wang H, Jin X, Li Y, Zhang P, Cheng S (2013) Virtual machine placement for improving energy efficiency and network performance in iaas cloud. In: Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on. IEEE, pp 238–243
55. Erickson A, Stewart IA, Kiasari A, Navaridas J (2015) An optimal single-path routing algorithm in the datacenter network dpillar. arXiv preprint arXiv: 1509.01746
56. Fang W, Lu Z, Wu J, Cao Z (2012) Rpps: a novel resource prediction and provisioning scheme in cloud data center. In: Services Computing (SCC), 2012 IEEE Ninth International Conference on. IEEE, pp 609–616
57. Fang W, Liang X, Li S, Chiaraviglio L, Xiong N (2013) Vmplanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers. Computer Networks 57(1):179–196
58. Farahnakian F, Ashraf A, Pahikkala T, Liljeberg P, Plosila J, Porres I, Tenhunen H. Using ant colony system to consolidate vms for green cloud computing
59. IEEE Transactions on Services Computing 8(2), 187– 198 (2015)
60. Farahnakian F, Pahikkala T, Liljeberg P, Plosila J, Tenhunen H (2014) Multi-agent based architecture for dynamic vm consolidation in cloud data centers. In: 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, pp 111–118
61. Farahnakian F, Pahikkala T, Liljeberg P, Plosila J, Tenhunen H (2015) Utilization prediction aware vm consolidation approach for green cloud computing. In: 2015 IEEE 8th International Conference on Cloud Computing, pp 381–388
62. Farahnakian F, Liljeberg P, Plosila J (2013) Lircup: Linear regression based cpu usage prediction algorithm for live migration of virtual machines in data centers. In: Proceedings of the 39th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp 358–364
63. Farahnakian F, Pahikkala T, Liljeberg P, Plosila J (2013) Energy aware consolidation algorithm based on knearest neighbor regression for cloud data centers. In: Proc. IEEE/ACM 6th Int. Conf. Utility Cloud Comput., pp 256–259, Dec. 2013
64. Fatema K, Emeakaroha VC, Healy PD, Morrison JP, Lynn T (2014) A survey of cloud monitoring tools: Taxonomy, capabilities and objectives. Journal of Parallel and Distributed Computing 74(10):2918–2933
65. Feller E, Rilling L, Morin C (2011) Energy-aware ant colony based workload placement in clouds. In: Grid Computing (GRID), 2011 12th IEEE/ACM International Conference on, pp 26–33
66. Feller E (2012) Autonomic and energy-efficient management of large-scale virtualized data centers, Ph.D. thesis. Universit´e Rennes, p 1
67. Fischer A, Botero JF, Beck MT, de Meer H, Hesselbach X (2013) Virtual network embedding: A survey. IEEE Communications Surveys Tutorials 15(4): 1888–1906
68. Fu X, Zhou C (2015) Virtual machine selection and placement for dynamic consolidation in cloud computing environment. Frontiers of Computer Science 9(2):322–330
69. Gao Y, Guan H, Qi Z, Hou Y, Liu L (2013) A multiobjective ant colony system algorithm for virtual machine placement in cloud computing. Journal of Computer and System Sciences 79(8):1230–1242
70. Ge C, Sun Z, Wang N (2013) A survey of power-saving techniques on data centers and content delivery networks. IEEE Communications Surveys Tutorials 15(3):1334–1354
71. Ghribi C, Hadji M, Zeghlache D (2013) Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms. In: Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on, pp 671–678
72. Ghribi C (2014) Energy efficient resource allocation in cloud computing environments, Ph.D. dissertation, Evry, Institut national des telecommunications
73. Gong Z, Gu X, Wilkes J (2010) Press: Predictive elastic resource scaling for cloud systems. In: 2010 International Conference on Network and Service Management, pp 9–16
74. Goudarzi H, Pedram M (2012) Energy-efficient virtual machine replication and placement in a cloud computing system. In: Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE, pp 750–757
75. Guan X, Choi BY, Song S (2014) Topology and migrationaware energy efficient virtual network embedding for green data centers. In: Computer Communication and Networks (ICCCN), 2014 23rd International Conference on, pp 1–8

76. Gutierrez-Aguado J, Alcaraz Calero JM, Diaz Villanueva W (2016) Iaasmon: Monitoring architecture for public cloud computing data centers. Journal of Grid Computing 14(2):283–297
77. Hameed A, Khoshkbarforoushha A, Ranjan R, Jayaraman PP, Kolodziej J, Balaji P, Zeadally S, Malluhi QM, Tziritas N, Vishnu A et al (2014) A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. Computing:1–24
78. Hameed A, Khoshkbarforoushha A, Ranjan R, Jayaraman PP, Kolodziej J, Balaji P, Zeadally S, Malluhi QM, Tziritas N, Vishnu A et al (2016) A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. Computing 98(7):751–774
79. Hanai M, Suzumura T, Ventresque A, Shudo K (2014) An adaptive vm provisioning method for large-scale agentbased traffic simulations on the cloud. In: Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on, pp 130–137
80. Hertzmann A, Fleet D (2014) Machine Learning and Data Mining Lecture Notes. Department of Computer and Mathematical Sciences, University of Toronto Scarborough
81. Horri A, Mozafari MS, Dastghaibyfard G (2014) Novel resource allocation algorithms to performance and energy efficiency in cloud computing. The Journal of Supercomputing 69(3):1445–1461
82. Hu B, Lei Z, Lei Y, Xu D, Li J (2011) A time-series based precopy approach for live migration of virtual machines. In: Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on. IEEE, pp 947–952
83. Huang J, Wu K, Moh M (2014) Dynamic virtual machine migration algorithms using enhanced energy consumption model for green cloud data centers. In: High Performance Computing Simulation (HPCS), 2014 International Conference on, pp 902–910
84. IBM, I (2014) IBM ILOG CPLEX optimization studio. IBM Corporation
85. Islam S, Keung J, Lee K, Liu A (2012) Empirical prediction models for adaptive resource provisioning in the cloud. Future Generation Computer Systems 28(1):155–162
86. Ismaeel S, Miri A, Al-Khazraji A (2016) Energyconsumption clustering in cloud data centre. In: 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), pp 1–6
87. Ismaeel S, Al-Jebory K (2001) Adaptive fuzzy system modeling. Eng. Technology 20(4):201–212
88. Ismaeel S, Al-Khazraji A, Al-delimi K (2017) Fuzzy information modeling in a database system. IAES International Journal of Artificial Intelligence (IJ-AI) 6(1):1–7
89. Ismaeel S, Miri A (2015) Using ELM techniques to predict data centre VM requests. In: Proceedings of The 2nd IEEE International Conference on Cyber Security and Cloud Computing (CSCloud 2015). IEEE, New York, pp 80–86
90. Ismaeel S, Miri A (2016) Multivariate time series ELM for cloud data centre workload prediction. In: Proceedings, Part I, of the 18th International Conference on HumanComputer Interaction. Theory, Design, Development and Practice - Volume 9731. SpringerVerlag New York, Inc., New York, pp 565–576
91. Ismaeel S, Miri A, Chourishi D, Dibaj SR (2015) Open source cloud management platforms: A review. In: Proceedings of The 2nd IEEE International Conference on Cyber Security and Cloud Computing (CSCloud 2015). IEEE, New York, pp 470–475
92. Ismail, U.: Comparing 7 monitoring options for docker, http://rancher.com/comparing-monitoring-options-for-docker-deployments/. Accessed Dec 2017
93. Jennings B, Stadler R (2015) Resource management in clouds: Survey and research challenges. Journal of Network and Systems Management 23(3): 567–619
94. Jheng JJ, Tseng FH, Chao HC, Chou LD (2014) A novel VM workload prediction using grey forecasting model in cloud data center. In: Information Networking (ICOIN), 2014 International Conference on. IEEE, pp 40–45
95. Jiankang D, Hongbo W, Shiduan C (2015) Energy performance trade-offs in IAAS cloud with virtual machine scheduling. China Communications 12(2): 155–166
96. Jin X, Han J (2010) Partitional Clustering. Springer US, Boston, p 766
97. Karim R, Ding C, Miri A (2015) End-to-end performance prediction for selecting cloud services solutions. In: Service-Oriented System Engineering (SOSE), 2015 IEEE Symposium on, pp 69–77
98. Karim R, Ding C, Miri A, Rahman MS (2015) End-toend QoS prediction model of vertically composed cloud services via tensor factorization. In: Cloud and Autonomic Computing (ICCAC), 2015 International Conference on, pp 158–168
99. Karim R, Ismaeel S, Miri A (2016) Energy-efficient resource allocation for cloud data centres using a multiway data analysis technique. In: Proceedings, Part I, of the 18th International Conference on Human-Computer Interaction. Theory, Design, Development and Practice - Volume 9731. Springer-Verlag New York, Inc., New York, pp 577–585
100. Kashyap R, Chaudhary S, Jat P (2014) Virtual machine migration for back-end mashup application deployed on openstack environment. In: Parallel, Distributed and Grid Computing (PDGC), 2014 International Conference on, pp 214–218
101. Khan A, Yan X, Tao S, Anerousis N (2012) Workload characterization and prediction in the cloud: A multiple time series approach. In: Network Operations and Management Symposium (NOMS), 2012 IEEE. IEEE, pp 1287–1294
102. Kong F, Liu X (2014) A survey on green-energy-aware power management for datacenters. ACM Computing Surveys (CSUR) 47(2):30
103. Kousiouris G, Menychtas A, Kyriazis D, Gogouvitis S, Varvarigou T (2014) Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms. Future Generation Computer Systems 32:27–40
104. LaCurts, K.L.: Application workload prediction and placement in cloud computing systems. Ph.D. thesis, Massachusetts Institute of Technology (2014)
105. Larumbe F, Sansò B (2017) Elastic, on-line and network aware virtual machine placement within a data center. In: Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on. IEEE, pp 28–36
106. Lee B, Oh KH, Park HJ, Kim UM, Youn HY (2014) Resource reallocation of virtual machine in cloud computing with mcdm algorithm. In: Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2014 International Conference on. IEEE, pp 470–477
107. Lee CA, Sill AF (2014) A design space for dynamic service level agreements in openstack. Journal of Cloud Computing 3(1):1–13
108. Li S, Wang Y, Qiu X, Wang D, Wang L (2013) A workload prediction-based multi-vm provisioning mechanism in cloud computing. In: Network Operations and Management Symposium (APNOMS), 2013 15th AsiaPacific. IEEE, pp 1–6
109. Li X, Garraghan P, Jiang X, Wu Z, Xu J (2017) Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. IEEE Transactions on Parallel and Distributed Systems
110. Liang TY, Wang SY, Wu IH (2008) Using frequent workload patterns in resource selection for grid jobs. In: AsiaPacific Services Computing Conference, 2008. APSCC '08. IEEE, pp 807–812
111. Liao Y, Yin D, Gao L (2010) Dpillar: Scalable dual-port server interconnection for data center networks. In: Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN). IEEE, pp 1–6
112. Liu XF, Zhan ZH, Du KJ, Chen WN (2014) Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach. In: Proceedings of the 2014 conference on Genetic and evolutionary computation. ACM, pp 41–48
113. Liu X, Gu H, Zhang H, Liu F, Chen Y, Yu X (2016) Energy-aware on-chip virtual machine placement for cloud-supported cyber-physical systems. Microprocessors and Microsystems
114. Lòpez-Pires F, Barán B (2013) Multi-objective virtual machine placement with service level agreement: A memetic algorithm approach. In: Utility and Cloud Computing (UCC), 2013 IEEE/ACM 6th International Conference on, pp 203–210
115. Lòpez-Pires F, Barán B (2015) Virtual machine placement literature review. CoRR abs/1506.:01509
116. Lòpez-Pires F, Barán B, Bentez L, Zalimben S, Amarilla A (2018) Virtual machine placement for elastic infrastructures in overbooked cloud computing datacenters under uncertainty. Future Generation Computer Systems 79:830–848
117. Lotfalipour MR, Falahi MA, Bastam M (2013) Prediction of CO2 emissions in Iran using grey and ARIMA models. International Journal of Energy Economics and Policy 3(3):229–237
118. Luo JP, Li X, Chen MR (2014) Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers. Expert Systems with Applications 41(13):5804–5816
119. Mann ZA, Szabó M (2017) Which is the best algorithm for virtual machine placement optimization? Concurrency and Computation: Practice and Experience 29(10)
120. Marian T, Weatherspoon H, Lee KS, Sagar A (2012) Fmeter: Extracting indexable low-level system signatures by counting kernel function calls. In: Middleware 2012. Springer, pp 81–100

121. Masdari M, Nabavi SS, Ahmadi V (2016) An overview of virtual machine placement schemes in cloud computing. Journal of Network and Computer Applications 66:106–127
122. Masoumzadeh SS, Hlavacs H (2013) An intelligent and adaptive threshold-based schema for energy and performance efficient dynamic VM consolidation. In: Energy Efficiency in Large Scale Distributed Systems. Springer, pp 85–97
123. Masoumzadeh S, Hlavacs H (2013) Integrating vm selection criteria in distributed dynamic VM consolidation using fuzzy q-learning. In: Network and Service Management (CNSM), 2013 9th International Conference on, pp 332–338
124. Masoumzadeh S, Hlavacs H (2015) Dynamic virtual machine consolidation: A multi agent learning approach. In: Autonomic Computing (ICAC), 2015 IEEE International Conference on, pp 161–162
125. Mastelic T, Fdhila W, Brandic I, and Rinderle-Ma S (2015) Predicting resource allocation and costs for business processes in the cloud. In World Congress on Services, New York City, NY, USA, pp 47–54.
126. Mastelic T, Oleksiak A, Claussen H, Brandic I, Pierson JM, Vasilakos AV (2015) Cloud computing: survey on energy efficiency. ACM Computing Surveys (CSUR) 47(2):33
127. Mazumdar S, Scionti A, Kumar AS (2017) Adaptive resource allocation for load balancing in cloud. In: Cloud Computing. Springer, pp 301–327
128. Minarolli D, Mazrekaj A, Freisleben B (2017) Tackling uncertainty in long-term predictions for host overload and underload detection in cloud computing. Journal of Cloud Computing 6(1):4. https://doi.org/10.1186/s13677-017-0074-3
129. Mohan A, Shine S (2013) Survey on live VM migration techniques. International Journal of Advanced Research in Computer Engineering and Technology 2(1):155–157
130. Monil M, Rahman R (2015) Implementation of modified overload detection technique with VM selection strategies based on heuristics and migration control. In: Computer and Information Science (ICIS), 2015 IEEE/ACIS 14th International Conference on, pp 223–227
131. Monil MAH, Rahman RM (2016) VM consolidation approach based on heuristics, fuzzy logic, and migration control. Journal of Cloud Computing 5(1):8. https://doi.org/10.1186/s13677-016-0059-7
132. Monil M, Rahman R (2015) Fuzzy logic based energy aware VM consolidation. In: Di Fatta G, Fortino G, Li W, Pathan M, Stahl F, Guerrieri A (eds) Internet and Distributed Computing Systems, Lecture Notes in Computer Science, vol. 9258. Springer International Publishing, pp 31–38
133. Moreno I, Garraghan P, Townend P, Xu J (2014) Analysis, modeling and simulation of workload patterns in a large-scale utility cloud. IEEE Transactions on Cloud Computing 2(2):208–221
134. Navaridas J, Stewart IA (2015) An efficient shortest-path routing algorithm in the data centre network dpillar. In: Combinatorial Optimization and Applications: 9th International Conference, COCOA 2015, Houston, TX, USA, December 18-20, 2015, Proceedings. vol. 9486. Springer, p 209
135. Ortigoza J, López-Pires F, Barán B (2016) Dynamic environments for virtual machine placement considering elasticity and overbooking. arXiv preprint arXiv:1601.01881
136. Parker, H.: Energy efficient data centres (2013), http://www.alliancetrustinvestments.com/
137. Pires FL, Barán B (2015) A virtual machine placement taxonomy. In: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp 159–168
138. Portaluri G, Adami D, Gabbrielli A, Giordano S, Pagano M (2017) Power consumption-aware virtual machine placement in cloud data center. IEEE Transactions on Green Communications and Networking 1(4):541–550
139. Prevost J, Nagothu K, Jamshidi M, Kelley B (2014) Optimal calculation overhead for energy efficient cloud workload prediction. In: World Automation Congress (WAC), 2014, pp 741–747
140. Prevost JJ, Nagothu K, Kelley B, Jamshidi M (2011) Prediction of cloud data center networks loads using stochastic and neural models. In: Proceedings of the 2011 6th International Conference on System of Systems Engineering (SoSE), pp 276–281
141. Qu G, Fang Z, Zhang J, Zheng SQ (2015) Switch-centric data center network structures based on hypergraphs and combinatorial block designs. IEEE Transactions on Parallel and Distributed Systems 26(4):1154–1164
142. Quang-Hung N, Son NT, Thoai N (2017) Energy-saving virtual machine scheduling in cloud computing with fixed interval constraints. In: Transactions on LargeScale Data-and Knowledge-Centered Systems XXXI. Springer, pp 124–145
143. Rajan, S S.: Dynamic scaling and elasticity -windows azure vs amazon EC2 (2010)
144. Ramezani F, Lu J, Hussain F (2013) An online fuzzy decision support system for resource management in cloud environments. In: Proceedings of the 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS). IEEE, pp 754–759
145. Rasheduzzaman M, Islam M, Islam T, Hossain T, Rahman R (2014) Task shape classification and workload characterization of google cluster trace. In: Proceedings of the 2014 IEEE International Advance Computing Conference (IACC), pp 893–898
146. Reiss C, Wilkes J, Hellerstein JL (2011) Technical Report. In: Google clusterusage traces: format+ schema. Google Inc, Mountain View, CA, USA
147. Rochman Y, Levy H, Brosh E (2014) Efficient resource placement in cloud computing and network applications. ACM SIGMETRICS Performance Evaluation Review 42(2):49–51
148. Rossigneux F, Lefevre L, Gelas JP, Assuncao D, Dias M (2014) A generic and extensible framework for monitoring energy consumption of openstack clouds. In: Proceedings of The 4th IEEE International Conference on Big Data and Cloud Computing (BdCloud). IEEE, pp 696–702
149. Saaty TL (2016) The analytic hierarchy and analytic network processes for the measurement of intangible criteria and for decision-making. In: Multiple Criteria Decision Analysis. Springer, pp 363–419
150. Sahi S, Dhaka V (2015) Study on predicting for workload of cloud services using Artificial Neural Network. In: Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom). pp. 331–335 (March 2015)
151. Sahni S, Varma V (2012) A hybrid approach to live migration of virtual machines. In: Cloud Computing in Emerging Markets (CCEM), 2012 IEEE International Conference on. pp. 1–5 (Oct 2012)
152. Sarikaya R, Isci C, Buyuktosunoglu A (2010) Runtime workload behavior prediction using statistical metric modeling with application to dynamic power management. In: Workload Characterization (IISWC), 2010 IEEE International Symposium on. pp. 1–10 (Dec 2010)
153. Sarji I, Ghali C, Chehab A, Kayssi A (2011) Cloudese: Energy efficiency model for cloud computing environments. In: Proceedings of The International Conference on Energy Aware Computing (ICEAC). pp. 1–6 (Nov 2011)
154. Sedaghat M, Wadbro E, Wilkes J, De Luna S, Seleznjev O, Elmroth E (2016) Diehard: reliable scheduling to survive correlated failures in cloud data centers. In: Cluster, Cloud and Grid Computing (CCGrid), 2016 16th IEEE/ACM International Symposium on. IEEE, pp 52–59
155. Semeraro G, Magklis G, Balasubramonian R, Albonesi DH, Dwarkadas S, Scott ML (2002) Energyefficient processor design using multiple clock domains with dynamic voltage and frequency scaling. In: HighPerformance Computer Architecture, 2002. Proceedings. Eighth International Symposium on, pp 29–40
156. Shidik GF, Mustofa K et al (2015) Evaluation of selection policy with various virtual machine instances in dynamic vm consolidation for energy efficient at cloud data centers. Journal of Networks 10(7):397–406
157. Shim YC (2016) Performance evaluation of static VM consolidation algorithms for cloud-based data centers considering inter-vm performance interference. International Journal of Applied Engineering Research 11(24):11794–11802
158. Shoaib Y, Das O (2014) Performance-oriented cloud provisioning: Taxonomy and survey. arXiv preprint arXiv 1411:5077
159. Song A, Fan W, Wang W, Luo J, Mo Y (2013) Multi-objective virtual machine selection for migrating in virtualized data centers. In: Pervasive Computing and the Networked World. Springer, pp 426–438
160. Song F, Huang D, Zhou H, Zhang H, You I (2014) An optimization-based scheme for efficient virtual machine placement. International Journal of Parallel Programming 42(5):853–872
161. Duong-Ba T, Tran T, Nguyen T, Bose B (2018) A Dynamic virtual machine placement and migration scheme for data centers. IEEE Transactions on Services Computing
162. Song W, Xiao Z, Chen Q, Luo H (2014) Adaptive resource provisioning for the cloud using online bin packing. IEEE Transactions on Computers 63(11):2647–2660
163. Sotiriadis S, Bessis N, Amza C, Buyya R (2016) Elastic load balancing for dynamic virtual machine reconfiguration based on vertical and horizontal scaling. IEEE Transactions on Services Computing

164. Strunk A (2013) A lightweight model for estimating energy cost of live migration of virtual machines. In: Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on, pp 510–517
165. Svard P, Hudzia B, Walsh S, Tordsson J, Elmroth E (2015) Principles and performance characteristics of algorithms for live vm migration. ACM SIGOPS Operating Systems Review 49(1):142–155
166. Taieb SB, Bontempi G, Atiya AF, Sorjamaa A (2012) A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. Expert systems with applications 39(8): 7067–7083
167. Tan PN, Steinbach M, Kumar V et al (2006) Cluster analysis: basic concepts and algorithms. Introduction to data mining 8:487–568
168. Tang M, Pan S (2014) A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. Neural Processing Letters 41(2):211–221
169. Tarighi M, Motamedi SA, Sharifian S (2010) A new model for virtual machine migration in virtualized cluster server based on fuzzy decision making. arXiv preprint arXiv 1002:3329
170. Tseng FH, Chen CY, Chou LD, Chao HC, Niu JW (2015) Service-oriented virtual machine placement optimization for green data center. Mobile Networks and Applications 20(5):556–566
171. Uddin M, Darabidarabkhani Y, Shah A, Memon J (2015) Evaluating power efficient algorithms for efficiency and carbon emissions in cloud data centers: A review. Renewable and Sustainable Energy Reviews 51:1553–1563
172. Varia, J.: Amazon white paper on cloud architectures (2008)
173. Vazquez C, Krishnan R, John E (2015) Time series forecasting of cloud data center workloads for dynamic resource provisioning. Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA) 6(3):87–110
174. Wang B, Qi Z, Ma R, Guan H, Vasilakos AV (2015) A survey on data center networking for cloud computing. Computer Networks 91:528–547
175. Wang CN, Phan VT (2014) An improvement the accuracy of grey forecasting model for cargo throughput in international commercial ports of Kaohsiung. International Journal of Business and Economics Research 3(1):1–5
176. Wang T, Su Z, Xia Y, Hamdi M (2014) Rethinking the data center networking: Architecture, network protocols, and resource sharing. IEEE Access 2:1481–1496
177. Weingärtner R, Bräscher GB, Westphall CB (2015) Cloud resource management: A survey on forecasting and profiling models. Journal of Network and Computer Applications 47:99–106
178. Xia Q, Lan Y, Zhao L, Xiao L (2014) Energy-saving analysis of cloud workload based on K-means clustering. In: The IEEE Computing, Communications and IT Applications Conference (ComComAp), pp 305–309
179. Xu D, Yang S, Luo H (2013) A fusion model for CPU load prediction in cloud computing. Journal of Networks 8(11):2506–2511 Yang, T., Lee, Y.C., Zomaya, A.Y.: Energy-efficient data center networks planning with virtual machine placement and traffic configuration. In: Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on. pp. 284–291 (2014)
180. Yue W, Chen Q (2014) Dynamic placement of virtual machines with both deterministic and stochastic demands for green cloud computing. Mathematical Problems in Engineering (2014)
181. Zhan ZH, Liu XF, Gong YJ, Zhang J, Chung HSH, Li Y (2015) Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Computing Surveys (CSUR) 47(4):63
182. Zhang L, Li Z, Wu C (2014) Dynamic resource provisioning in cloud computing: A randomized auction approach. In: INFOCOM, 2014 Proceedings IEEE. IEEE, pp 433–441
183. Zheng Q, Li R, Li X, Shah N, Zhang J, Tian F, Chao KM, Li J (2015) Virtual machine consolidated placement based on multi-objective biogeography-based optimization. Future Generation Computer Systems
184. Zhu YH, Chen D, Zhuang Y (2016) Virtual machine scheduling algorithm based on energy-aware in cloud data center. Computer and Modernization 4:017