

Energy-Efficient Resource Allocation for Cloud Data Centres Using a Multi-way Data Analysis Technique

Raed Karim^(✉), Salam Ismaeel, and Ali Miri

Department of Computer Science, Ryerson University,
350 Victoria Street, Toronto, ON M5B 2K3, Canada
{r2karim, salam.ismaeel, ali.miri}@ryerson.ca

Abstract. Cloud Data Centres (CDCs) are facilities used to host large numbers of servers, networking and storage systems, along with other required infrastructure such as cooling, Unsupervised Power Supplies (UPS) and security systems. With the high proliferation of cloud computing and big data, more and more data and cloud-based service solutions are hosted and provisioned through these CDCs. The increasing number of CDCs used to meet enterprises' needs has significant energy use implications, due to power use of these centres. In this paper, we propose a method to accurately predict workload in physical machines, so that energy consumption of CDCs can be reduced. We propose a multi-way prediction technique to estimate incoming workload at a CDC. We incorporate user behaviours to improve the prediction results. Our proposed prediction model produces more accurate prediction results, when compared with other well-known prediction models.

Keywords: Workload prediction · Cloud Data Centres · Tensor factorization · Energy Efficiency

1 Introduction

In CDCs, Physical Machines (PMs) use virtualization to host multiple Virtual Machines (VMs), where a wide range of applications (data-intensive and compute-intensive) are deployed and run [1]. Servers and storage systems in CDCs are used to host and run applications, and to process, store and provision data and contents to consumers in a client/server computing architecture. CDCs are equipped with different PMs brands (e.g. IBM, HP, Dell, etc.) with different compute resource specifications such as CPU cores with levels of performance, memory sizes, storage capacities and network bandwidths. It has been estimated that data centre energy consumption will reach 140 billion kilowatt-hours annually by 2020, costing US businesses \$13 billion annually in electricity bills and emitting approximately 100 million metric tons of carbon pollution per year. This represents a significant increase from only 0.6 % of the global carbon emission in 2008 to 2.6 % in 2020 [2].

In CDCs, users often request compute resources to perform different IT-related tasks. However, not all of requested resources are used. According to Google [3], only a small segment of the provisioned VM instances are used during deployment. Lack of

knowledge of future resources needed by a CDC can lead to over-provisioning or under-provisioning problems [4].

To address the above problem, we propose a prediction model that estimates future incoming workload to a CDC. Our model predicates workload based on users' requirements, and in particular identifies required number and types of VMs, represented by vCPU and memory specifications. To test our model, we have utilized the available historical workload data collected from Google traces over a period of 29 days. The collected data represent Google compute cells. The tracelog contains over 25 millions tasks, submitted by 930 users who (previously) requested different types of VMs over different time slots. The number of recorded requests is 3295896. We will show how our model can improve workload prediction over this set of data using a Multi-Way Data Analysis (MWDA) approach that incorporates users' behaviour.

The rest of the paper is organized as follows. Section 2 provides an overview of the workload prediction process. Section 3 describes the proposed prediction model. Section 4 presents the experimental work. Section 5 discusses the related work. Section 6 concludes the paper and suggests some future work.

2 Related Work

Research on computing resource prediction and virtualization techniques in CDCs has gained lots of interest over the past few years, with a number of different techniques proposed in the literature to tackle the machine workload prediction problem. In [5], Qazi *et al.* used an autoregressive moving average technique (ARMA), whereas Dabbagh *et al.* [6] proposed the use of a weighted average of previous observations. Machine learning techniques, such as ELM [7] have also been used to predict future workload. These methods have a number of possible shortcomings because they do not consider of all the key inputs for obtaining accurate predications. In [5], both user behaviour and actual usage of CPU and memory were not considered. In [6], user behaviour was not considered during data processing and prediction computation. The main shortcoming of machine learning techniques is that adding user clusters (behaviours) as inputs to the prediction process increases the error in estimating the number of VM requests in each cluster. This is because adding more variables (users) to a nonlinear process can negatively affect estimation. This implies that traditional machine learning techniques cannot handle multi-dimensional problem domains.

The main consideration of our work is the optimal use of the available computing resources of PMs in CDCs. Our proposed model has the ability to capture multiple variables in a multi-dimensional environment. Since user behaviours have a large impact on improving prediction results, we incorporate users as one of the key model variables in order to improve prediction accuracy. Prediction accuracy, in turn, enables us to reduce the number of required PMs and hence to obtain better energy conservation.

3 Proposed Workload Prediction Process

As discussed in the previous section, our goal is to accurately estimate the right number and size of the required VMs based on future users' needs. This can in return result in reduction of power consumption in CDCs. To achieve this goal, we propose using the following three phases – data clustering (steps 1–3), data filtering (step 4) and workload prediction (step 5):

– Data clustering

Step 1: We cluster VMs based on the calculated workload into different clusters. In our work and in line with other work [6, 7], we consider a case study of four VM clusters to demonstrate the effectiveness of our prediction model in which actual resources are utilized for the prediction process. We label clusters with ranges of workload percentages, as follows. The very big VMs cluster contains workload of 75 % and up, the big VMs cluster contains [50 %–74.9 %], the medium VMs cluster contains workload of [25 %–49.9 %] and the small VMs contains workload of [0.1 %–24.9 %]. We put users reported in the data set into different clusters based on the number of VM requests made in the past. Each cluster is characterized by request density. Based on the experimental work we have done [7], we observed that increasing the number of user clusters produces more accurate results. However, at a certain point no further improvement can be observed. We have performed the clustering process on different data sizes, and we have found that the number of clusters between 25 and 30 gives the best results.

Step 2: According to the clustering outcome, we count the number of requests submitted by different users under each VM cluster. The calculated numbers represent historical workload data that we utilize during the prediction computation process.

Step 3: We arrange the VMs' historical workload data in a tensor of three dimensions of users U_i , VM clusters V_j and time intervals t_n . Each entry of the tensor denotes the number of VMs of a particular VM cluster that a user has used within a specific time.

– Data filtering

Step 4: We analyze recorded (historical) workload data by discovering their patterns and relationships. We consider user behaviours when used VMs in the past by calculating the linear correlations (dependencies) between users' clusters.

– Workload prediction

Step 5: We employ a multi-way (tensor) technique to predict the incoming workload of VMs for a future time interval (i.e. the number of required VMs of each cluster based on users' requests). We sum up the predicted number of VMs of each cluster for all users to obtain the total number of VMs under each cluster for the future time interval.

3.1 Analyzing Workload Data for the Prediction Process

Our objective is to uncover hidden patterns (features) and dependencies among the available VMs, which are hosted on heterogeneous PMs, and dependencies among their users with respect to their workload information. In our previous work on web service domain [7], we have proven that learning hidden features can improve the prediction results.

In order to improve the prediction results, we analyze user behaviours which have used VMs in the past. Our analysis is done by calculating the correlation degrees among users using their workload values in series of time intervals. The correlation could be either positive or negative. We only consider users with positive correlation meaning that they had similar historical experiences in terms of workload they have applied on a same set of VMs. In this work and based on the available information, users' similarity is determined by two criteria: if users are geographically located close to each other and/or they have similar trends. User trends are determined by request characteristics (how often users request VMs, workload intensity, peak season time and off season time). We look into their historical invocations of the VMs, and then we calculate their correlations over the history with respect to request characteristics. We consider workload data of users with the highest degree of correlations. We denote the space of similar users as a user neighborhood. Users with negative correlations are those whose past experiences are dissimilar. Usually we are not interested in this type of users since they represent noise to the prediction computation process. We calculate the users' correlations using Pearson Correlation Coefficient (PCC) technique [8]. The neighborhood contains local information of workload data. In our proposed model, the local information is integrated in the global information (workload data of the whole tensor) during the prediction process.

4 Proposed Prediction Model (MWDA)

To address this problem, we propose a multiway low-rank Tensor Factorization (TF) model [9, 10]. In our model, we integrate three vectors (user-specific, VM-specific and time-specific) into a workload matrix. The TF factorizes the workload matrix and hence makes accurate prediction. Our goal is to map VMs and users information within sequential time intervals to a cooperative latent feature space of a low dimensionality, such that VM-time interactions can be captured as inner products in that space. The premise behind a low-dimensional TF technique is that there are only a few hidden features affecting the VM-time interactions, and a user's interactive experience is influenced by how each feature affects the user. TF can discover features underlying interactions between VMs as well as between users. It balances the overall information from all VMs (global information) and users, and information associated with users with similar behaviours (local information). We verify our proposed approach by conducting experiments. We use data of usage traces of a Google compute cell which is a set of PMs packed into racks in a data centre. We specifically extract the Task Event table that contains PMs ID, tasks sent to VMs, user ID and workload data represented by resource request for CPU and memory [3].

Let R denote the workload tensor. As we mentioned, R contains workload values based on different users requesting VMs of different time intervals. The future workload are predicted by minimizing the objective function as follows:

$$\frac{1}{2} \| R - \hat{R} \|_F^2 \quad (1)$$

where \hat{R} denotes the predicted workload tensor; $\| \cdot \|_F^2$ denotes the Frobenius form which is calculated as the square root of the sum of the absolute squares of $R - \hat{R}$.

Since R is very sparse, only VMs with recorded workload values are factorized. The tensor factorization term is minimized by applying the following function:

$$\min_{U, V, A} \zeta(R, U, V, A) = \frac{1}{2} \sum_{e=1}^i \sum_{f=1}^j \sum_{g=1}^n I_{efg} (R_{efg} - \hat{R}_{efg})^2 \quad (2)$$

where I_{efg} is an indicator function that is equal to 1 if a VM is used, and a workload value is available; otherwise it is equal to 0.

$$\min_{U, V, A} \zeta(R, U, V, A) = \frac{1}{2} \sum_{e=1}^i \sum_{f=1}^j \sum_{g=1}^n I_{efg} (R_{efg} - \hat{R}_{efg})^2 \quad (3)$$

Considering the original tensor factorization term, unknown workload values (the number of required VMs of each cluster) are predicted by learning the latent features of all known workload values through factorizing the user-specific, VM-specific and time-specific matrices. The main drawback for using only this term is that the prediction accuracy might be poor since workload values of all users are considered some of which could have caused noise into the prediction computation process [9]. To overcome this drawback, we propose to add an additional regularization term to the tensor factorization model. The new term considers the information of similar users in predicting future workload values. The premise is that neighbours have similar interactive experience when using VMs. This is due the fact that users within the same geographical locations and have similar workload patterns are more likely to have similar VM requests in the future. We incorporate the new regularization term into our tensor model as follows:

$$\begin{aligned} \min_{U, V, A} \zeta(R, U, V, A) = & \frac{1}{2} \sum_{e=1}^i \sum_{f=1}^j \sum_{g=1}^n I_{efg} (R_{efg} - \hat{R}_{efg})^2 \\ & + \frac{\sigma}{2} \sum_{e=1}^i \sum_{f=1}^j \sum_{g=1}^n \| \hat{R}_{efg(e)} - \sum_{k \in K(e)} R_{efg(k)} P_{ek} \|^2, \end{aligned} \quad (4)$$

where $R_{efg(k)}$ denotes workload of similar users to V_e ; $K(e)$ is a set of *top k* similar users and P_{ek} is the similarity weight of a similar users, and it is calculated as follows:

$$P_{ek} = \frac{sim(e, k)}{\sum_{k \in K(e)} sim(e, k)}, \quad (5)$$

Where $sim(e, k)$ is calculated using the PCC method.

A local minimum of the objective function in (9) can be found by performing the gradient descent algorithm in U_e , V_f and A_g as follows:

$$\begin{aligned} \frac{\partial \zeta}{\partial U_e} &= \sum_{f=1}^j \sum_{g=1}^n I_{efg} (\hat{R}_{efg} - R_{efg}) (V_f^T A_g) + \sigma (\hat{R}_{efg(i)} - \sum_{k \in K(e)} R_{efg(k)} P_{ek}) (V_f^T A_g) \\ \frac{\partial \zeta}{\partial V_f} &= \sum_{e=1}^i \sum_{g=1}^n I_{efg} (\hat{R}_{efg} - R_{efg}) (U_e^T A_g) + \sigma (\hat{R}_{efg(i)} - \sum_{k \in K(e)} R_{efg(k)} P_{ek}) (U_e^T A_g) \\ \frac{\partial \zeta}{\partial A_g} &= \sum_{e=1}^m \sum_{f=1}^n I_{efg} (\hat{R}_{efg} - R_{efg}) (U_e^T V_f) + \sigma (\hat{R}_{efg(i)} - \sum_{k \in K(e)} R_{efg(k)} P_{ek}) (U_e^T V_f) \end{aligned} \quad (6)$$

5 Experiments

In the experiments, we have used Google traces of CPU and memory data that are recorded for a period of 29 days. The data are recorded with timestamps in microsecond and it describes machines used and tasks requested by different users' requests. We specifically used the Task event table that contains time stamps, user information, CPU, memory and local disk resources requested by users. To demonstrate the effectiveness of our proposed prediction model, we have used a slice of the data trace of 24 h (1440 min) with a time interval of 5 min. We mapped the recorded CPU and memory workload data into multiple VM clusters so that each user request for a VM is mapped to a specific cluster. During the time frame, there were 3295896 requests as inputs to the clustering process. The number of VM clusters that we have selected is 4 which correspond to four VM categories (Small, Medium, Big and Very Big) according to our proposed prediction process described in Sect. 2. On the other hand, 426 users have been recorded within the specified time frame. We clustered the users based on their historical usages of requested VMs (i.e. the number of request users have made to VMs). We have used the fuzzy c-mean clustering algorithm. In our previous work [7], we demonstrated the efficiency of the fuzzy c-mean clustering algorithm compared to the traditional k-mean clustering algorithm. In this work, we used 25 clusters which produced the best results (the lowest error rate). The premise for clustering the users is to improve the efficiency of the correlation computation process described in Sect. 3.1.

5.1 Evaluation and Discussion

Our objective in conducting the experiments was to evaluate the prediction accuracy of our proposed model by comparing its results with the following well-known prediction algorithms available in the literature: (1) *Mean*: this method considers the average value

of historical workload data. (2) *ARMA*: this method calculates the auto-regression moving average of the training data [11]. (3) *Weiner*: this method was proposed by [6]. It calculates the weighted average of the training data. (4) *Latest*: this method takes the training data as an input and returns the latest observation [7]. (5) *nUTF*: this method is a different version of our implemented algorithm that we used in our prediction model. It computes the tensor factorization without considering users' correlations (behaviours). (6) *MWDA*: this is our proposed prediction algorithm in this paper. We used a three dimensional multi-way technique to compute the prediction. We have calculated user clustering, and incorporated user correlations (behaviours) during the prediction process.

We have used the Mean Absolute Error (MAE) method to measure the prediction accuracy of each of the prediction algorithms including our proposed algorithm by computing the average absolute deviation of the predicted values from the actual data. The smaller MAE values indicate higher prediction accuracy. The MAE is defined as follows:

$$\text{MAE} = \frac{\sum_{m,n,c} |\hat{R}_{efg} - R_{efg}|}{L}, \quad (7)$$

where, m , n , c denote the number of the user clusters, timestamps and VM components; R_{efg} denotes the actual workload value; \hat{R}_{efg} denotes the predicted workload value; L is the number of the predicted values.

In this work, our objective is to estimate the future workload (the number of VMs of each type) based on historical workload values. Therefore, for the purpose of our experiments we removed the data of the future time interval (the next five minutes) from the tensor R . The remaining values are used for the learning purpose to predict the removed ones. We used the cross validation method during the MAE calculation process to obtain reliable error calculations. Table 1 shows the MAE values of the compared prediction methods. The observation was that our MWDA model outperformed all other models in terms of the accuracy of workload prediction results as it produces the lowest MAE values. The prediction accuracy is an important factor that determines how many VM needed for the next time frame and the types of these VMs. The better prediction results the better knowledge that is required to plan ahead of time for optimal placements of incoming VMs onto PMs in CDCs. Eventually, we can accurately estimate the number of PMs which can be turned off or used for other tasks. As a consequence, a considerable amount of energy can be conserved. Relying on users' knowledge or using poor prediction models can make the estimation of the number of unused PMs far from being accurate. Thus, it leads to a large percentage of energy waste or failing to meet users' QoS requirements. In our approach, we attempt to build a knowledge base that is dynamically updated and relies on the actual usages of computing resources. By training the historical workload data using a reliable prediction algorithm we can accurately estimate future workload. Accurately predicting workload can improve not only CDCs' providers' energy consumption but also users' QoS experience, which heavily rely on the adequacy of compute resources.

Table 1. Workload prediction comparisons (lower MAE values indicate better prediction accuracy).

	Mean	ARMA	Wiener	Latest	Minimum	nUTF	MWDA
MAE	13.92	7.41	7.38	9.70	9.81	6.28	3.42

6 Conclusions

In this paper, we proposed a model for predicting incoming workload in CDCs. Our prediction model solves the problems of machine overloading (a possible violation of users' QoS requirements) and underloading (unused computing resources lead to energy waste) by accurately predicting the number and the types of VMs based on user requirements. Using our model, we can accurately estimate the number and types of VMs required for the incoming workload. Hence, we can free up unused clusters that can be turned off or used for new VMs predicted by our model. Overall, the amount of energy consumed in CDCs is reduced for environmental and economy advantage. To the best of our knowledge, this is the first technique that incorporates user behaviours in a multi-way technique to improve the prediction of future incoming workload for energy saving purposes in CDCs. As an extension of this work, we plan to develop a placement mechanism that takes our prediction results as an input in order to optimally place the predicted VMs onto PMs in CDCs.

References

1. Dutta, S., Gera, S., Verma, A., Viswanathan, B.: SmartScale: automatic application scaling in enterprise clouds. In: Proceedings of IEEE Conference on Cloud Computing, pp. 221–228 (2012)
2. Delforge, P.: America's data centre consuming and wasting growing amounts of energy. <http://www.nrdc.org/energy/data-centre-efficiency-assessment.asp>
3. Reiss, C., Wilkes, J., Hellerstrin, J.L.: Google cluster-usage traces: format + schema. Google Inc. Technical report (2011)
4. Li, X., Qian, Z., Lu, S., Wu, J.: Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data centre. *Math. Comput. Model.* **58**(5), 1222–1235 (2013)
5. Qazi, K., Li, Y., Sohn, A.: Workload prediction of virtual machines for harnessing data centre resources. In: Proceedings of IEEE International Conference on Cloud Computing, pp. 522–529 (2014)
6. Dabbagh, M., Hamdaoui, B., Guizani, M., Rayes, A.: Energy-efficient cloud resource management. In: Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM), pp. 386–391 (2014)
7. Ismaeel, S., Miri, A.: Energy-consumption clustering in cloud data centre. In: Proceedings of IEEE MEC International Conference on Big Data and Smart City (2016)
8. Karim, R., Ding, C., Miri, A.: End-to-end performance prediction for selecting cloud service solutions. In: Proceedings of IEE Symposium On Service-oriented System Engineering, pp. 69–77 (2015)

9. Karim, R., Ding, C., Miri, A.: End-to-end QoS prediction of vertically composed cloud services via tensor factorization. In: Proceedings of IEEE Cloud and Autonomic Computing, pp. 229–236 (2015)
10. Acar, E., Yener, B.: Unsupervised multiway data analysis: a literature survey. *IEEE Trans. Knowl. Data Eng.* **21**(1), 6–20 (2009)
11. Roy, N., Dubey, A., Gokhale, A.: Efficient autoscaling in the cloud using predictive models for workload forecasting. In: Proceedings of IEEE International Conference on Cloud Computing, pp. 500–507 (2011)