Power consumption estimation in data centers using machine learning techniques

Emmanouil Karantoumanis[1[0000-0003-4929-9886]] and Nikolaos Ploskas[1[0000-0001-5876-9945]⋆]

Department of Electrical & Computer Engineering, University of Western Macedonia, Kozani 50100, Greece {dece00034,nploskas}@uowm.gr

Abstract. Large data centers consume large amounts of electricity. Estimating the energy consumption in a data center can be of great importance to data centers administrators in order to know the energy-consuming tasks and take actions for reducing the total energy consumption. Smart workflow mechanisms can be built to reduce the energy consumption of data centers significantly. In this paper, we are investigating the factors that affect the energy consumption of scientific applications in data centers. We also use eight machine learning methods to estimate the energy consumption of multi-threaded scientific applications. Extensive computational results on a computer with 20 cores show that the CPU usage is the most important parameter in the power consumed by an application. However, better results can be obtained when the CPU utilization is combined with other parameters. We generate various regression models that predict the energy consumption of an application with an average accuracy of 99%. Simpler models with one and two parameters can achieve comparable accuracy with more complex models. We also compare various machine learning methods for their ability to obtain accurate predictions using as few parameters as possible.

Keywords: Data centers · Energy consumption · Machine learning · Cloud computing.

1 Introduction

Cloud computing has evolved in the last decade to become the technological backbone for most modern enterprises. With the increase of the data hosted in data centers and the applications ported in them, new larger data centers are needed to meet the demands of the users. However, data centers consume large amounts of electricity. For example, Google's data center used about 2.26 million MW hours of electricity to run its operations in 2010 [11]. Energy consumption increased by 90% from 2000 to 2005, but only by 4% from 2010 to 2014, and this is due to the optimization of energy consumption that most data centers apply [6]. In addition, the total carbon dioxide emissions of the Information and Communication Technology (ICT) sector keep increasing. The carbon dioxide

⋆ Corresponding author

emissions of the ICT sector are equal to those of the aviation sector [1]. An average data center consumes energy equivalent to $25,000$ households and the environmental impact of data centers was estimated at 116.2 million tons of carbon dioxide in 2006 [13,15]. Therefore, new solutions to minimize the energy consumption of data centers are of great importance.

Various works have studied the energy consumption problem in data centers. Some researchers focused on the workload and consumption prediction [9,10,14], while others use methods to estimate the energy consumption in virtual machines or servers [2,12,16]. Various input parameters have been used to model the energy consumption: (i) CPU, (ii) cache, (iii) disk, (iv) DRAM, (v) network, and (vi) maximum number of open sockets. In addition, linear regression models, neural networks, and Gaussian mixture models have been utilized to model power consumption.

In this paper, we are interested on estimating the energy consumed by a scientific application running in a data center. Various works [2,3] have shown that the CPU usage is the most important parameter that affecting the power consumption of a computer. In this paper, we aim to improve the prediction accuracy by investigating whether or not other input parameters (e.g., memory usage, memory size, disk size, etc.) can be used to predict the power consumption. Being able to estimate the energy consumption of an application, task scheduling mechanisms can be built to reduce the total energy consumption of a data centers. We are investigating the factors that affect the power consumption of scientific applications. Various machine learning methods are compared in terms of their ability to obtain accurate predictions of the power consumed by an application. Simpler regression models with one and two parameters are proposed.

2 Computational results

The experiments were performed on a computer with an Intel Xeon CPU E5-2660 v3 (2 CPUs - 10 cores each) and 128 GB of main memory, a clock of 2.6 GHz, an L1 code cache of 32 KB per core, an L1 data cache of 32 KB per core, an L2 cache of 256 KB per core, and an $L3$ cache of 24 MB, running under Centos 7 64-bit.

In this work, we aim to investigate the parameters that affect the power consumed by an application. The parameters that we considered are the following:

- number of cores (nc)
- CPU usage (cu)
- memory size (ms)
- memory usage (mu)
- disk size (ds)
- total number of transfers per second (dt)
- total amount of data written to devices in blocks per second (dw)
- total number of network requests (nn)
- total number of kilobytes received per second (nr)

– total number of kilobytes transmitted per second (nt)

In order to estimate the energy consumption of an application, we used the stress-ng [8] tool to stress CPU, RAM and disk, and the ab [4] tool to stress network. We utilized sar [5] to collect the values of the aforementioned ten input parameters in each second and powerstat [7] to collect the power consumed (Watts). A total of $1,000$ runs were performed with different combinations for the number of threads, the memory usage, the disk usage and the network consumed by the application (in this simulation, the application is the stress-ng and the ab tools that stress the CPU, RAM, disk and network). Each experiment was run for 80 seconds and the instantaneous value of each input and output parameters was stored in each second. Afterwards, we eliminated the first ten and the last ten values and we calculated the average of the remaining 60 values.

We used eight regression methods from scikit-learn to estimate the power consumption:

1. Ordinary least squares linear regression (LinearRegression)
2. Lasso regression (Lasso)
3. Ridge regression (Ridge)
4. Epsilon-support vector regression (SVR)
5. Decision tree regression (DecisionTreeRegressor)
6. Random forest regression (RandomForestRegressor)
7. Regression based on k-nearest neighbors (KNeighborsRegressor)
8. Multi-layer Perceptron regression (MLPRegressor)

We use 70% (700 samples) of data to train each model, and the rest 30% (300 samples) for testing the model. We use 10-fold cross validation to test the accuracy of each model. To evaluate the performance of our model, we use R-squared (R^2, coefficient of determination) that provides an estimate of the strength of the relationship between a regression model and the dependent variable (output). Table 1 presents the results of all regressors using (i) all input parameters (ten parameters), (ii) only the CPU usage (cu) as input (one parameter), and (iii) all combinations of the CPU usage parameter with all other parameters (two parameters). The second column shows the R^2 scores that each regressor achieved with all ten parameters. The third column shows the R^2 scores that each regressor achieves with the cu parameter as single input, while the fourth column shows the best R^2 score with two parameters, one of which is always the CPU usage parameter. All regressors, except from SVR and MLPRegressor, achieve high accuracy when using all parameters as input. The best performing regressor is the RandomForestRegressor with a score of 99.44%. Equation 1 is the best model that was obtained from the three linear regression methods using ten input parameters. As it is obvious, the CPU usage is the most important parameter.

$$\begin{aligned} Watts = (cu \times 1.14) + (ms \times -1.04e^{-01}) + (ds \times 4.30e^{-01}) \\ +(dw \times 9.68e^{-06} + (nn \times 2.20e^{-05}) + (nr \times -5.17e^{-05}) + 65.04 \end{aligned} \tag{1}$$

Therefore, we investigate the accuracy that can be obtained when using only the CPU usage as input parameter. The accuracy obtained by all regressors, except from SVR and MLPRegressor, is lower than their performance when using all input parameters. However, an accuracy of 97.33% can be obtained. Equation 2 is the best model that was obtained from the three linear regression methods using only the CPU usage as input parameter.

$$Watts = cu \times 1.17 + 64.3 \tag{2}$$

Finally, we used as input the CPU usage with all other input parameters (nine combinations). The last column in Table 1 shows that very accurate estimations can be made using only two parameters. The disk size (ds) and the total number of transfers per second (dt) are the second most important parameters in predicting the power consumption of an application. Equation 3 is the best model that was obtained from the three linear regression methods using two input parameters.

$$Watts = (cu \times 1.11) + (ds \times 0.49) + 66 \tag{3}$$

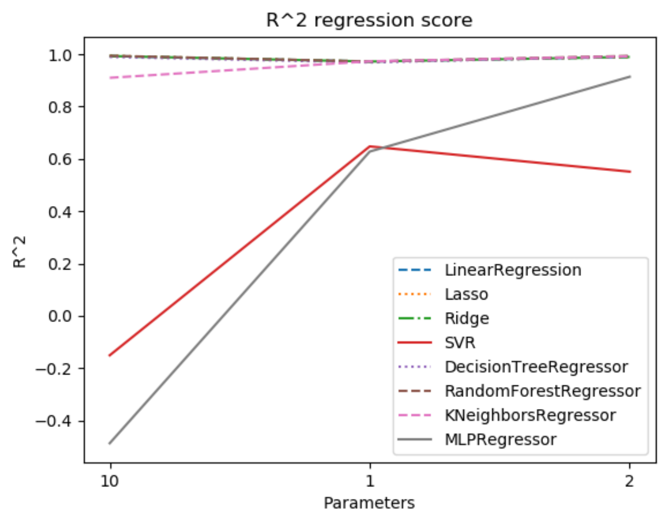
When using only the CPU usage as input parameter, the results for SVR and MLPRegressor are better than when using ten parameters as input. In addition, the rest of the scores is reduced to a very small degree, so CPU usage is the most important input parameter. In most regressors, the error when using one parameter increases by only 2% relative to the error when using ten parameters. When using two parameters, the error increases only by less than 1%. This means that simpler models with one or two parameters can be built to predict the power consumed by an application.

Table 1. R^2 scores for regressors using all (ten), one, and two input parameters.

Regressor	**All parameters**	**One parameter (cu)**	**Two parameters**
LinearRegression	0.9932	0.9725	0.9899
Lasso	0.9939	0.9724	0.9897
Ridge	0.9934	0.9725	0.9899
SVR	-0.1508	0.6479	0.5514
DecisionTreeRegressor	0.9900	0.9689	0.9922
RandomForestRegressor	0.9944	0.9726	0.9938
KNeighborsRegressor	0.9102	0.9733	0.9922
MLPRegressor	-0.4871	0.6281	0.9142

In Figure 1, we present the scores that each regressor can achieve with different input parameters. The y axis shows the R^2 scores of the regressors and the x axis shows the number of parameters used. Most regressors have similar patterns and their lines overlap because their scores are very close. The KNeighborsRegressor method has a lower accuracy than other regressors when using ten input parameters, but it has a good performance when using one and two input parameters. The SVR and MLPRegressor methods are the worst performers. However, their accuracy scores are significantly improved when using one and two parameters.

Fig. 1. Accuracy achieved by each regressor using one, two, and ten input parameters.



3 Conclusions

In this paper, we use eight regression methods to predict the power consumption of an application on a computer with 20 cores. Extensive computational results show that the CPU usage is the most important parameter for the energy consumption prediction. We also investigated the accuracy that can be achieved using simpler models. Most regressors are able to achieve a high accuracy score when using one and two parameters. Therefore, simpler models can be utilized to predict the power consumed by an application. In future work, we plan to collect data from different computers and confirm whether or not the models generated on a specific machine can be also used to predict the energy consumption on other machines. We also aim to stress servers with various applications that will be executed concurrently and validate the application of the proposed models. Finally, we will also experiment with tuning the parameters of each regressor in order to further improve their accuracy.

References

1. Avgerinou, M., Bertoldi, P., Castellazzi, L.: Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency. Energies **10**(10), 1470 (2017)
2. Economou, D., Rivoire, S., Kozyrakis, C., Ranganathan, P.: Full-system power analysis and modeling for server environments. International Symposium on Computer Architecture-IEEE (2006)
3. Fan, X., Weber, W.D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. In: ACM SIGARCH computer architecture news. vol. 35, pp. 13–23 (2007)
4. Foundation, A.S.: ab - apache http server benchmarking tool. `https://manpages.ubuntu.com/manpages/disco/en/man1/ab.1.html`, last accessed on January 30, 2020
5. Godard, S.: sar - collect, report, or save system activity information. `https://manpages.ubuntu.com/manpages/precise/man1/sar.sysstat.1.html`, last accessed on January 30, 2020
6. Google: Google environmenta report 2019. `https://services.google.com/fh/files/misc/google_2019-environmental-report.pdf` (2019), last accessed on January 30, 2020
7. King, C.: powerstat - a tool to measure power consumption. `http://manpages.ubuntu.com/manpages/xenial/man8/powerstat.8.html` (2015), last accessed on January 30, 2020
8. King, C.: stress-ng - a tool to load and stress a computer system. `http://manpages.ubuntu.com/manpages/xenial/en/man1/stress-ng.1.html` (2016), last accessed on January 30, 2020
9. Kumar, J., Singh, A.K.: Workload prediction in cloud using artificial neural network and adaptive differential evolution. Future Generation Computer Systems **81**, 41–52 (2018)
10. Marino, D.L., Amarasinghe, K., Manic, M.: Building energy load forecasting using deep neural networks. In: IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society. pp. 7046–7051. IEEE (2016)
11. Miller, R.: Google's energy story: High efficiency, huge scale. `http://www.datacenterknowledge.com/archives/2011/09/08/googles-energy-story-high-efficiency-huge-scale`, last accessed on January 30, 2020
12. Öztürk, M.M.: Tuning stacked auto-encoders for energy consumption prediction: a case study. International Journal of Information Technology and Computer Science **2**, 1–8 (2019)
13. Rao, G.K.V., Premchand, K.: Scheduling virtual machines across data centres in accordance to availability of renewable sources of energy. International Journal of Engineering and Computer Science **5**(10), 18372–18376 (2016)
14. Sinha, A., Chandrakasan, A.P.: Dynamic voltage scheduling using adaptive filtering of workload traces. In: VLSI Design 2001. Fourteenth International Conference on VLSI Design. pp. 221–226. IEEE (2001)
15. Uchechukwu, A., Li, K., Shen, Y.: Energy consumption in cloud computing data centers. International Journal of Cloud Computing and Services Science **3**(3), 145–162 (2014)
16. Yu, X., Zhang, G., Li, Z., Liangs, W., Xie, G.: Toward generalized neural model for vms power consumption estimation in data centers. In: ICC 2019-2019 IEEE International Conference on Communications (ICC). pp. 1–7. IEEE (2019)