Final Project - Resource system utilization in Datacenters

Javier Galindos Vicente, 214373IV

***Abstract*—Business-critical workloads web servers, mail servers, app servers, etc. are increasingly hosted in virtualized datacenters acting as Infrastructure-as-a-Service clouds (cloud datacenters). Understanding how business-critical workloads demand and use resources is key in capacity sizing, in infrastructure operation and testing, and in application performance management. However, relatively little is currently known about these workloads, because the information is complex, large- scale, heterogeneous, shared-clusters and because datacenter operators remain reluctant to share such information. Large-scale and long-term workload traces corresponding to requested and actually used resources in a distributed datacenter servicing business-critical workloads are collected. This work will present a baseline to predict system resources utilization using ARIMA models and additionally the time series will be clustered to find different operation zones and find outliers within the data.**

I. Introduction

Datacenters and Supercomputers have to distribute a huge pool of hardware resources efficiently across users and applications. The utilization of those resources can vary significantly depending on various factors such as the workload and time of day. Being able to accurately predict the demand for hardware resources in the future is very important to ensure uninterrupted operation. Spurred by a rapid development of hardware and of resource management techniques, cloud datacenters are hosting an increasing number of application types. Over a billion people access daily a diverse collection of free or paid cloud utilities, from search to financial operations, from online social gaming to engineering. To continue the adoption of cloud datacenters, and to improve the ability of the datacenter operators to tune existing and to design new resource management techniques, understanding of the workload characteristics and the underlying datacenters is key for both datacenter operators and for cloud service providers [1].

II. Dataset collection

The dataset [2] contains the performance metrics of 1,750 virtual machines (VMs) from a distributed datacenter from Bitbrains, which is a service provider that specializes in managed hosting and business computation for enterprises. Customers include many major banks (ING), credit card operators (ICS), insurers (Aegon), etc.

Each file contains the performance metrics of a VM. These files are organized according by traces: fastStorage and Rnd.

The first trace, *fastStorage*, consists of 1,250 VMs that are connected to fast storage area network (SAN) storage devices. The second trace, *Rnd*, consists of 500 VMs that are either connected to the fast SAN devices or to much slower Network Attached Storage (NAS) devices. The *fastStorage* trace includes a higher fraction of application servers and compute nodes than the *Rnd* trace, which is due to the higher performance of the storage attached to the *fastStorage* machines. Conversely, for the Rnd trace we observe a higher fraction of management machines, which only require storage with lower performance and less frequent access.

The format of each file is row-based, each row represent an observation of the performance metrics. The variables of the dataset are:

- Timestamp: number of milliseconds since 1970-01-01.
- CPU cores: number of virtual CPU cores provisioned.
- CPU capacity provisioned (CPU requested): the capacity of the CPUs in terms of MHZ, it equals to number of cores x speed per core.
- CPU usage: in terms of MHZ.
- CPU usage: in terms of percentage
- Memory provisioned (memory requested): the capacity of the memory of the VM in terms of KB.
- Memory usage: the memory that is actively used in terms of KB.
- Disk read throughput: in terms of KB/s
- Disk write throughput: in terms of KB/s
- Network received throughput: in terms of KB/s
- Network transmitted throughput: in terms of KB/s

For the collection of the traces, the monitoring and management tools provided by VMware, such as vCloud suite are used. For each trace, the vCloud Operation tools record 10 performance metrics per VM, sampled every 5 minutes. The data is collected between August and September 2013. Combined, the traces accumulate data for 1,750 nodes, with over 5,000 cores and 20 TB of memory, and operationally acumulate over 5 million CPU hours in 4 operational months; thus, they are long-term and large-scale time series.

III. Data pre-processing

The dataset presented in the previous section is a very large-scale time series. Thus, the analysis of the whole dataset is beyond the scope of this work. One VM of the *fastStorage* trace is selected for analysis. Among the performance metrics (variables or features) the most critical ones are selected. First, the CPU utilization in terms of percentage. Second, a combination of two variables Memory provisioned and Memory usage are used to get the memory utilization in terms of percentage. Specifically, the new feature is:

$$memoryUtilization = 100 \cdot \frac{Memory\,usage}{Memory\,provisioned} \quad (1)$$

Thus, the analyzed dataset are two univariate time-series: the CPU and memory utilization in terms of percentage of a VM within a cloud datacenter.

IV. Descriptive Statistics

In this section, an exploratory data analysis is performed. The CPU utilization over time is presented in Fig. 1 and the memory utilization over time is presented in Fig. 2.

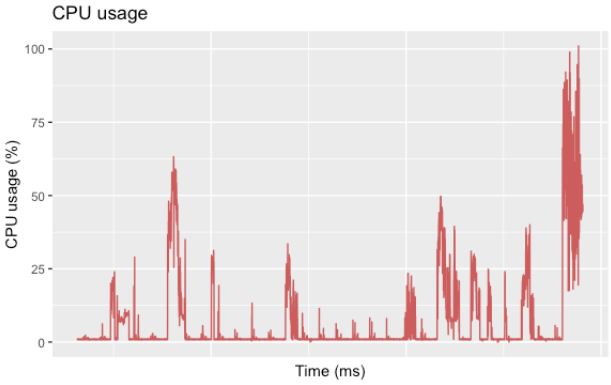

Fig. 1: CPU utilization over time.

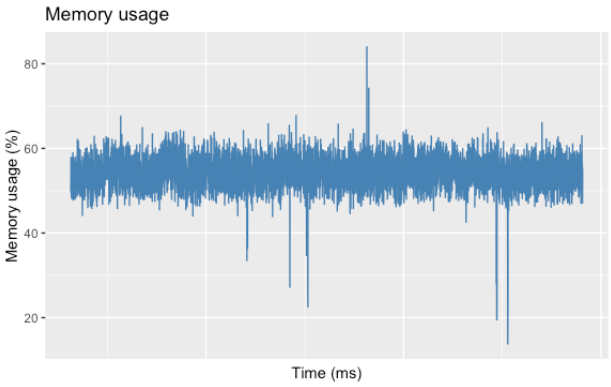

Fig. 2: Memory utilization over time.

A summary of the descriptive statistics is presented in Table I. One should observe how the CPU utilization is close to 0 during most part of the time and there are peaks of high utilization. The mean of the CPU usage is 7 % and the median is 0.93 %. In addition the maximum is over 100 %, which is an anomaly. On the other hand, the memory is used around 50 % most part of the time as could be observed in Fig. 2. The mean and the median are around 54 %.

TABLE I: Descriptive statistics of the dataset

	CPU utilization	**Memory utilization**
Min.	0.00	13.79
1st Qu.	0.80	51.93
Median	0.93	54.11
Mean	7.00	54.15
3rd Qu.	2.60	56.31
Max.	101.06	83.97

The aforementioned results are also supported through the histograms presented in Fig. 3 and Fig. 4. The histogram of the CPU utilization shows how most of the time the usage is close to 0. On the flip side, the histograms of the memory utilization could be approximated to a Gaussian distribution with the mean close to 55 %.

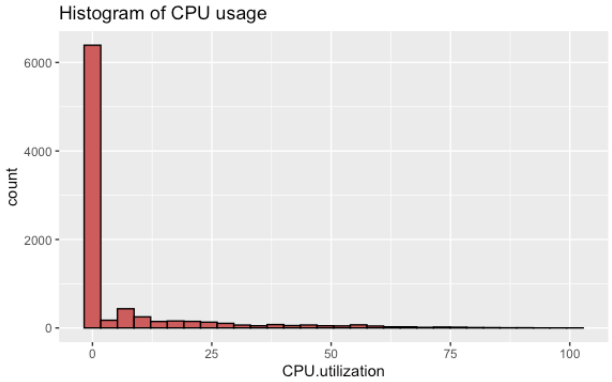

Fig. 3: CPU utilization histogram.

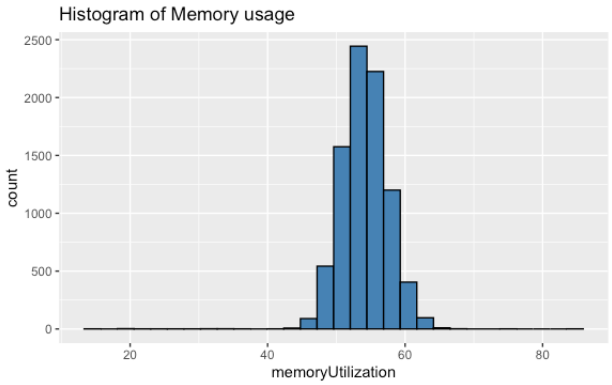

Fig. 4: Memory utilization histogram.

V. ARIMA

In this section, an Auto-regressive Moving Average model (ARIMA) is going to be fitted to the time-series in order to be able to forecast future values. Within an $ARIMA(p, d, q)$, the auto-regressive $AR(p)$ takes into account linear combination of the previous p timestamps. The moving average $MA(q)$ model takes into account the error in the previous q timestamps. Finally, the $I(d)$ component indicates the degree of differentiation necessary to make the series stationary in mean. The first step is to split the time-series into two subsets to validate the model, one for training and another one for testing. The forecasting horizon is going to be 100 timestamps. Thus, the training set contains all the timestamps but the last 100 and the testing subset if composed by the last 100 timestamps.

A. CPU utilization

Observing the time-series presented in Fig. 1, one should note that the series in not stationary in mean. Thus, a differentiation is needed. Now, the series is approximately stationary in mean and variance, except during the peaks of utilization. To validate this assumption, an Augmented Dickey-Fuller Test is performed. The p-value is smaller than 0.05, thus the null hypothesis is rejected and the series is likely to be stationary in mean.

To identify the order of the $ARIMA(p,d,q)$ model, analyzing the ACF and PACF (see Fig. 5) of the differentiate series is needed. One should try to fit the simplest model to avoid over-fitting. The ACF present two lags statistically significant in the first few timestamps and the PACF presents 6 lags statistically significant. To fit the simplest model, the function $auto.arima$ is utilized. The function returns the best ARIMA model according to either AIC, AICc or BIC value.

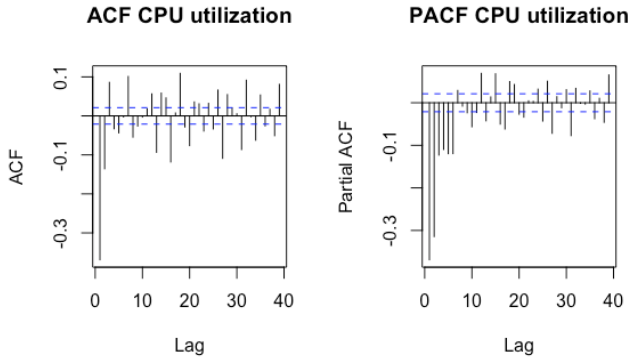

Fig. 5: ACF and PACF of Diff CPU utilization.

The optimal model of the CPU utilization time series is an $ARIMA(1,1,2)$. The summary of the model is presented in Table II.

TABLE II: CPU utlization ARIMA model

	AR1	MA1	MA2
Coefficients	-0.2294	-0.3025	-0.2452
S.E.	0.1208	0.1185	0.0696

The diagnosis of the model presented in Fig. 6 shows that the residuals do not follow a white noise distribution and the ACF of the residuals is not clean. The complexity of the time-series exhibits that an ARIMA model is not capable to model the shape of the time-series and a more complex model may be needed.

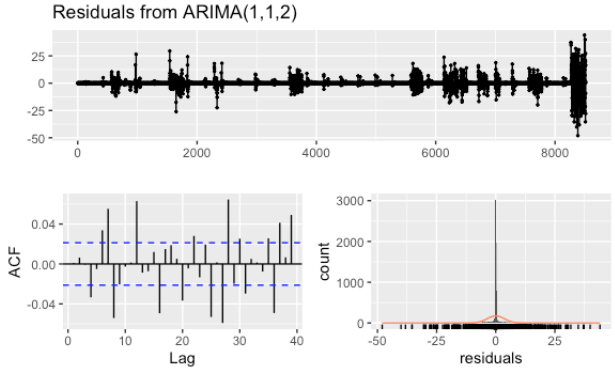

Fig. 6: Diagnosis CPU utilization model.

To validate the results, several metric are analyzed. Scale-dependent errors, such as mean error (ME) mean percentage error (MPE), mean absolute error (MAE) and root mean squared error (RMSE) are based on a set scale, and cannot be used to make comparisons that are on a different scale [3]. Percentage errors, like MAPE, are useful because they are scale-independent, so they can be used to compare forecasts between different data series, unlike scale-dependent errors. The disadvantage is that it cannot be used in the series has zero values (as the CPU utilization). Scale-free errors (e.g., Mean Absolute Scaled Error (MASE)) were introduced more recently to offer a scale-independent measure that doesnt have many of the problems of other errors like percentage errors [4].

TABLE III: Validation errors CPU utilization

	ME	RMSE	MAE	MPE	MAPE	MASE
Training	0.018	3.59	1.20	Inf	Inf	0.96
Testing	-5.11	14.56	11.06	-17.55	24.84	8.84

It is noteworthy to observe the differences between the training and the testing set, the infinite values for percentage errors due to the presence of 0s within the time series and the poor results in the testing set. A more complex model is needed for this time-series.

1) Outlier detection: The procedure to find outliers decomposes the time series into trend, seasonal and remainder components ($y_t = T_t + S_t + R_t$). The seasonal component is optional, and it may containing several seasonal patterns corresponding to the seasonal periods in the data. The idea is to first remove any seasonality and trend in the data, and then find outliers in the remainder series, R_t. Once the trend and seasonal components are eliminated, we could look for outliers. If $Q1$ denotes the 25th percentile and $Q3$ denotes the 75th percentile of the remainder values, then the interquartile range is defined as $IQR = Q3 - Q1$. Observations are labelled as outliers if they are less than $Q1 - 3 \times IQR$ or greater than $Q3 + 3 \times IQR$ [5]. Any outliers identified in this manner are replaced with linearly interpolated values using the neighbouring observations, and the process is repeated.

The result of the aforementioned procedure in the CPU utilization time-series is presented in Fig. 7.

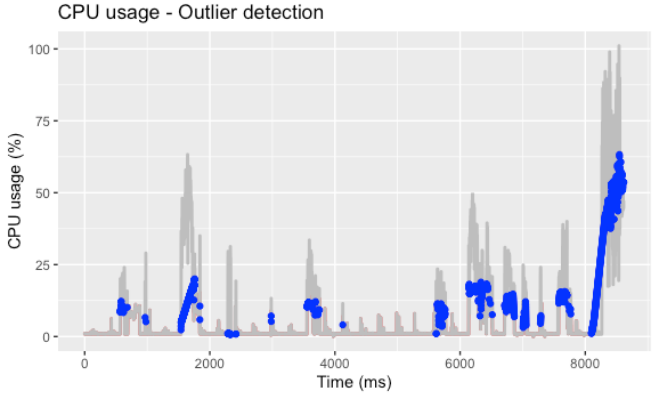

Fig. 7: Outliers CPU utilization model.

B. Memory utilization

The methodology used to analyze the memory utilization is analogous to the one used for the analysis of the CPU utilization and due to the lack of space, only the results and conclusions are presented.

The series presented in Fig. 2 is stationary in mean and variance. Analyzing the ACF and PACF of the series and using the auto.arima function, the optimal model found is an $ARIMA(2,0,1)$ with non-zero mean. The summary of the model is presented in Table IV.

TABLE IV: Memory utlization ARIMA model

	AR1	AR2	MA1	Mean
Coefficient	1.2452	-0.2537	-0.9702	54.1598
S.E.	0.0127	0.0114	0.0064	0.1228

The diagnosis of the model is presented in Fig. 8 shows that the residuals follow a white noise distribution and the ACF of the residuals is clean. Thus, the model is capable of capturing the characteristics of the time series.

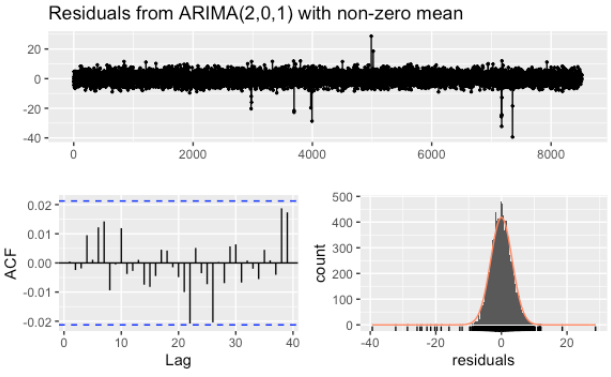

Fig. 8: Diagnosis memory utilization model.

The validation errors are presented in Table V. The results are similar in the training and testing set and the performance is significantly better than the obtained in the CPU utilization model.

TABLE V: Validation errors memory utilization

	ME	RMSE	MAE	MPE	MAPE	MASE
Training	0.001	3.25	2.48	-0.41	4.69	0.78
Testing	-0.96	3.01	2.39	-2.09	4.55	0.75

VI. Clustering

Both time-series are clustered using EM and k-means algorithms. The Hopkins statistics is used to test the probability that the data is generated by a uniform random distribution and it is often as a proxy measure for clustering tendency. The Hopkins statistic is 0.748 and 0.78 for the CPU utilization and memory utilization, respectively. Thus, both series are highly clusterable.

To determine the optimal number os clusters, the silhouette coefficient is used. In both time-series, the silhouette coefficient is maximal for two clusters. Thus, the optimal number of clusters for both series is two.

A. CPU utilization

The clusters determined by the EM algorithm are presented in Fig. 9 and the ones depicted by k-means are presented in Fig. 10. The results of both algorithms are similar, but the EM output seems more coherent and accurate. There are two clearly distinguish clusters, one when the CPU utilization is 0 or very close to 0, and the other one when the CPU is utilized more intensively.

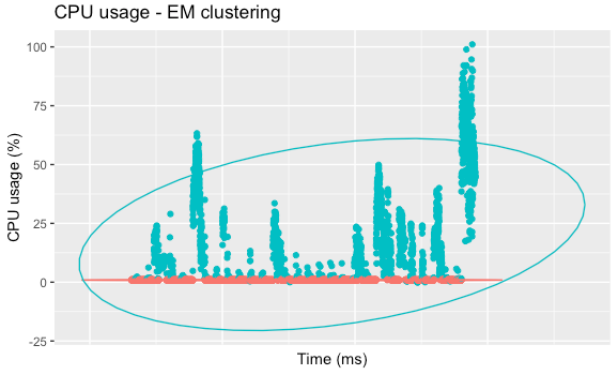

Fig. 9: EM clustering CPU utilization.

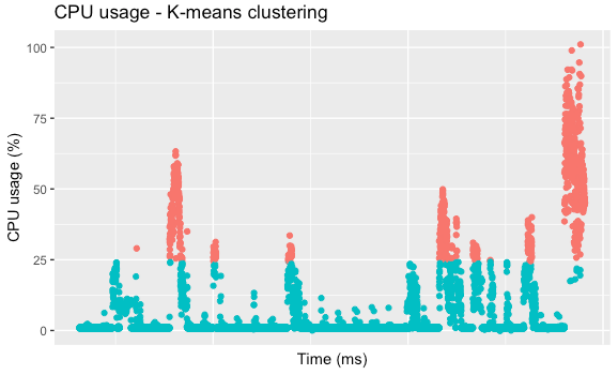

Fig. 10: K-means clustering CPU utilization.

B. Memory utilization

The clusters determined by the EM algorithm are presented in Fig. 11 and the ones depicted by k-means are presented in Fig. 12. Again, the EM algorithm performs better than k-means. According to the EM output, there are two clusters, one when the memory is used in percentages similar to the mean and other when the memory utilization differ from the mean and could be consider as an outlier. The k-means output shows two clusters, one when the memory utilization is higher than a threshold of around 55 % and another when it is lower.

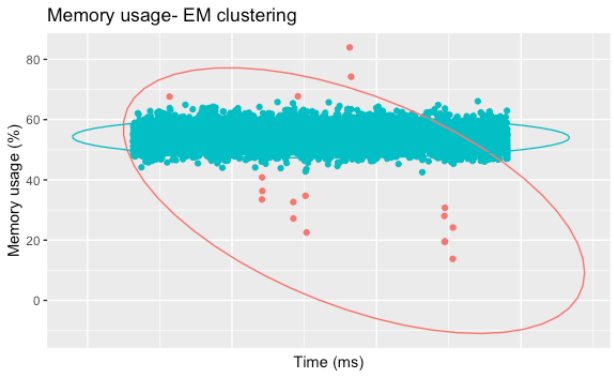

Fig. 11: EM clustering memory utilization.

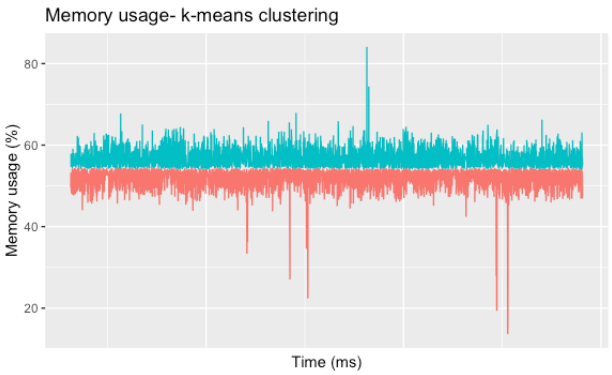

Fig. 12: K-means clustering memory utilization.

REFERENCES

[1] S. Shen, V. Van Beek, and A. Iosup, "Statistical characterization of business-critical workloads hosted in cloud datacenters," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2015, pp. 465–474.

[2] 2022. [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains

[3] 2022. [Online]. Available: https://towardsdatascience.com/measures-performance-for-a-time-series-model-ets-or-arima-18b0a3e91e83

[4] P. M. Maçaira and F. L. Cyrino Oliveira, "Another look at ssa.boot forecast accuracy," *International Journal of Energy and Statistics*, vol. 04, no. 02, p. 1650008, 2016.

[5] 2022. [Online]. Available: https://robjhyndman.com/hyndsight/tsoutliers/