

Simulation Modeling of Virtual Machine Based Services in Multi-Cloud Shared Access Centers

Petr Polezhaev

*Department of Computer Safety
Orenburg State University
Orenburg, Russia
newblackpit@mail.ru*

Leonid Legashev

*Science Library
Orenburg State University
Orenburg, Russia
silentgir@gmail.com*

Alexander Shukhman

*Department of Geometry & CS
Orenburg State University
Orenburg, Russia
shukhman@gmail.com*

Irina Bolodurina

*Department of Applied Mathematics
Orenburg State University
Orenburg, Russia
prmat@mail.osu.ru*

Abstract—This paper addresses the problem of performance improvement of multi-cloud centers for shared access to virtual resources. Statistical study of Azure workload data was performed in Jupyter Notebooks using SciPy, StatsModels and Facebook Prophet libraries. It should be noted relatively high underutilization of the virtual resources of the Azure cloud platform. As a result of experiments the best way to simulate end-user request stream is to use the empirical cumulative distribution functions and probability mass functions calculated from the sample data. To produce high quality forecasts for time series data Prophet framework for Python and SARIMAX prediction model of StatsModels library were used. The model based on Prophet framework showed the best prediction accuracy by the Mean Absolute Error metric. Prophet framework can be used to forecast number of created and deleted virtual machines in ‘Interactive mode’, which in turn could be used in the implementation of the multi-cloud shared access center simulator.

Keywords—simulation modeling, Azure, DaaS, statistical analysis

I. INTRODUCTION AND RELATED WORKS

The development of cloud technologies, as well as virtualization and containerization technologies, makes it possible to organize the sharing of a common virtual resource between end users in the most efficient way. Solving the problem of predicting the characteristics of the flow of user requests for access to resources and the problem of efficiently scheduling virtual environments will lead to the implementation of a unified system for the efficient use of virtual resources by cloud providers with support for scaling and load balancing virtual machines.

Simulation modeling is widely used in relation to cloud systems in order to create an adequate and reliable model for generating end-user requests for access to virtual resources (IaaS, DaaS, SaaS) and assessment the performance of cloud system.

In paper [1] researchers present the CloudNetSim++ cloud environment simulator, which can be used to evaluate such cloud components as: processing elements, storage, service level agreement (SLA), scheduling algorithms, consolidation algorithms and virtual machine migration.

Experimental simulations were conducted on the data from the Center for Computational Research of State University of New York at Buffalo datacenter. Researchers in their paper [2] studied a lot of modern simulators of cloud environments in terms of architecture and key features, and also presented a framework for designing components of a cloud simulator.

Analytical and simulation modeling are used in relation to the complexity of cloud computing systems and accurate assessment of resource performance, as it shown in [3]. The models developed by researchers include such parameters as: batch arrival of end-user requests, resource virtualization, servicing stages, allowing to obtain important performance indicators. The paper [4] proposes an approach focused on modeling the design of a cloud system, which takes into account the goals of stakeholders. The authors conduct experimental studies of the proposed approach for the private cloud deployment model using the example of a smartcard technology for the public transport system. Researchers from IBM in their work [5] describe a modular framework for simulating cloud performance, taking into account time and resource constraints. The modular approach allows you to clearly separate the simulation of hardware and software components. This paper demonstrates the usage of a framework to simulate the performance of a disk image deployment for OpenStack managed clouds. Researchers Garg S. K. and Buyya R. in their work [6] point out the critical importance of modeling tools for cloud computing environments. The authors present a cloud simulation environment with a scalable network and message-passing parallel applications model that allows scheduling and provisioning of virtual resources to be assessed in order to optimize the performance of the cloud infrastructure.

The paper [7] describes the extensive workload characteristics of virtual machines (VMs) in Microsoft Azure cloud platform, including the distribution of the lifetime of virtual machines, the size of the VM deployment, and resource consumption. The authors represent the Resource Central (RC) client library, a system that collects VMs telemetry, periodically examines the behavior of VMs in stand-alone forecasting models, and provides online predictions of VMs behavior for various resource management systems. Based on the analysis of real data from

Azure VMs workload, it is possible to select probability distributions and distribution coefficients for the characteristics of the flow of end-user requests. An open-source web application Jupyter Notebook [8] was used to perform analysis on Python using SciPy, StatsModels and Prophet libraries.

II. STATISTICAL ANALYSIS OF AZURE DATACENTER WORKLOADS

The first step is to join and select data from several csv-files downloaded from the Azure workload dataset, the resulting CSV table contains the following characteristics: `vm_id` – VM identifier; `subscription_id` – subscription identifier; `deployment_id` – deployment identifier; `vm_created` – VM creation time in seconds; `vm_deleted` – VM deletion time in seconds; `max_cpu` – maximum CPU load; `avg_cpu` – average CPU load; `p95_cpu` – 95th percentile of CPU load; `vm_category` – VM workload category (Interactive, Delay-insensitive or Unknown); `vm_virtual_core_count` – number of VM virtual cores; `vm_memory` – the amount of virtual RAM in GB; `first_vm_created` – first VM creation time; `count_vms_created` – number of created VMs in request; `deployment_size` – number of required VMs.

At the second stage, we study 1000000 user requests including distributions of their characteristics.

Distribution of VMs creation time is shown in Fig 1. The shown interval of end-user requests arrival is [0; 45] minutes of the whole 30 days interval. For 7.9% of user requests, VMs creation time occurred at zero time. Virtual machines running at zero time intervals must be from set of all the virtual machines that were in the Azure data center at this point.

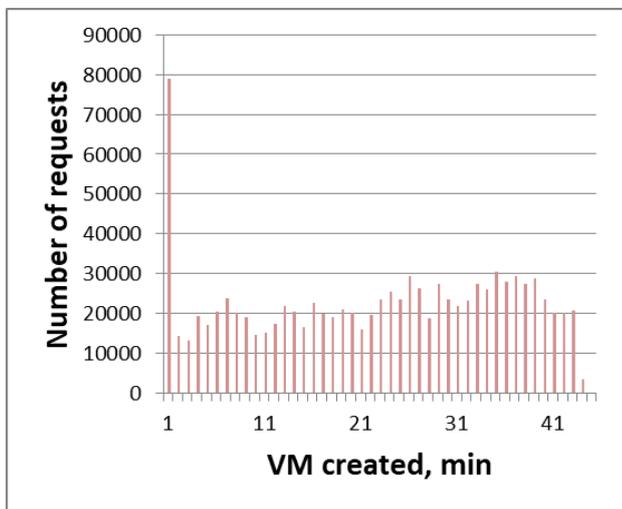


Fig. 1. Distribution histogram of 'vm_created'.

The distribution of the average CPU load is shown in Fig. 2. For 10.7% of user requests, the average CPU load is less than 1%. The distribution of the number of virtual CPU cores of user requests is shown in Fig. 3. 42% of requests use virtual machines with 1 core, 23.7% of requests use virtual machines with 2 cores, 13.8% requests use virtual machines with 3 cores, 20.3% – 8 cores and 0.0063% – 16 cores. The distribution of the number of virtual CPU cores can be defined in simulator by probability mass function (PMF).

The distribution of the number of VMs in end-user requests is shown in Fig 4. The interval of the number of VMs is [1; 1814].

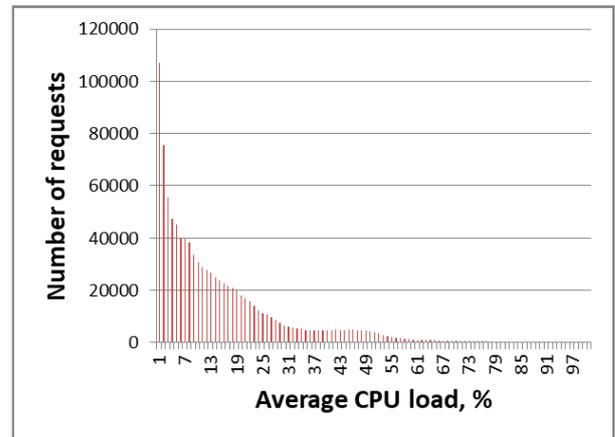


Fig. 2. Distribution histogram of 'avg_cpu'.

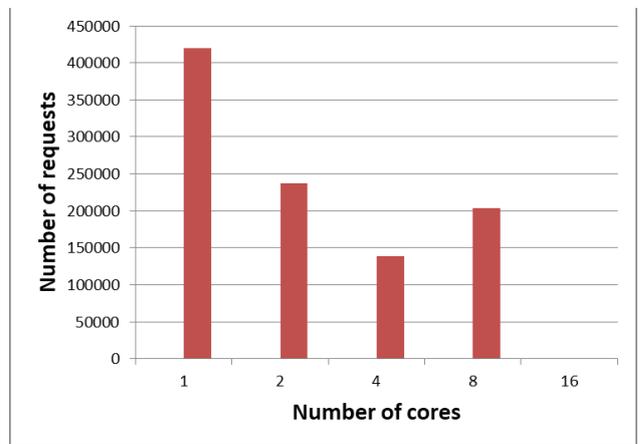


Fig. 3. Distribution histogram of 'vm_virtual_core_count'.

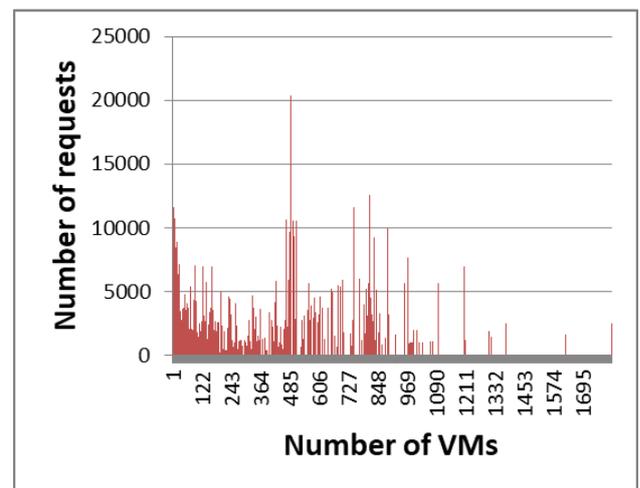


Fig. 4. Distribution histogram of 'deployment_size'.

The distribution of virtual RAM consumption for end-user requests is shown in Fig. 5. In 39.9% of user requests 1 Gb of RAM is used, in 23.6% – 3 Gb, in 14.1% – 14 Gb, in 13.7% – 7 Gb, in 6.2% – 56 Gb and in 2% of user requests less than 1 Gb is used. The distribution of virtual RAM consumption also can be defined in simulator by PMF.

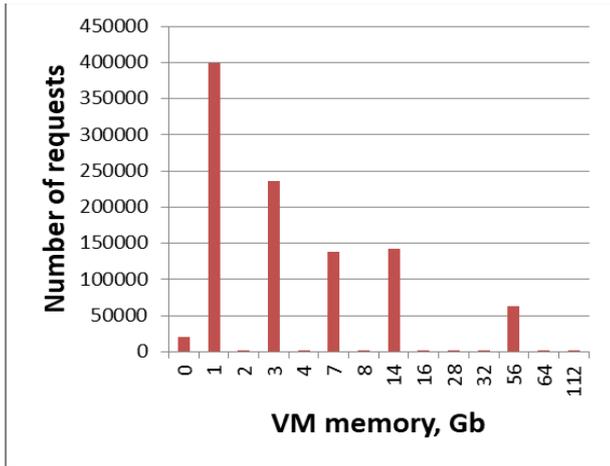


Fig. 5. Distribution histogram of 'vm_memory'.

We also researched the lifetime of the VMs in the 'Interactive' mode. The distribution of VMs lifetime of 13571 end-user requests is shown in Fig. 6. The average life is 985152 seconds.

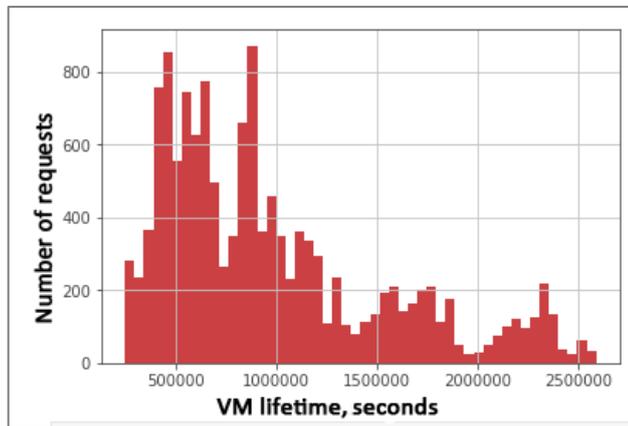


Fig. 6. Distribution of lifetime for VMs in the 'Interactive' mode.

It is worth noting the relatively high underutilization of the virtual resources of the cloud system.

We tried out Chi-squared and Kolmogorov-Smirnov tests to fit request parameters to all distribution laws with the best parameters from StatsModels library, but these criteria rejected all of our hypotheses. We also tried different transformations to parameters without any success. Thus, the best way to simulate end-user request stream is to use the empirical CDFs (Cumulative Distribution Functions) and PMFs calculated from the sample data.

III. VMS DATA FORECASTING WITH PROPHET AND SARIMAX

To produce high quality forecasts for time series data we use Prophet framework [9] for Python. The first step is to set prediction model parameters as follow: minutes_per_interval = 20 – prediction interval length; prediction_size = 7 * 24 * 60 – the size of the Hold-out test part representing the last elements of the time series (the rest first part of time series is used to train the model); total_num = int(prediction_size * 1.2) – total number of values to show. Descriptive statistics of dataset of 'Interactive' VMs is shown in Table I.

TABLE I. PREPARED DATASET OF INTERACTIVE VMs

	'vm_created'	'vm_deleted'
count	60682	60682
unique	3591	3162
top	2017-01-01 00:00:00	2017-01-30 23:55:00
freq	47111	45688
first	2017-01-01 00:00:00	2017-01-04 12:25:00
last	2017-01-26 16:10:00	2017-01-30 23:55:00

The distribution of created and deleted VMs in 'Interactive' mode is shown in Fig. 7.

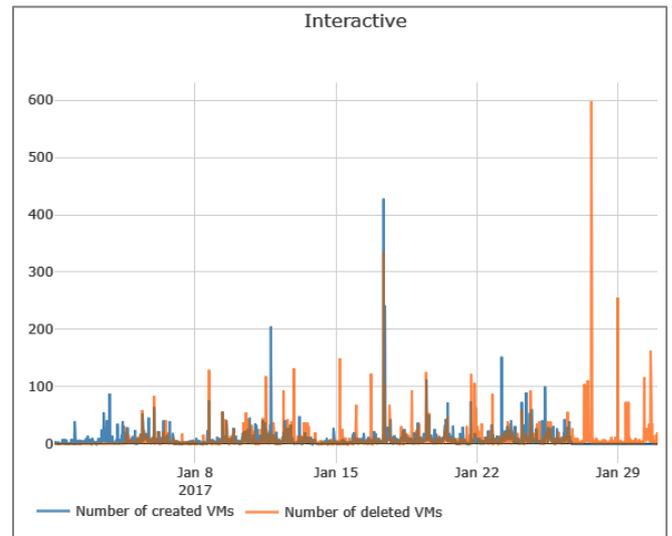


Fig. 7. Distribution histogram of created/deleted VMs.

Prophet forecast for number of created VMs in 'Interactive' mode is shown in Fig. 8. Black dots denote actual values, dark blue line – forecast, blue area refers to upper and lower bound of forecast. Components for number of created VMs in 'Interactive' mode are shown in Fig. 9. They include the trend, weekly seasonality, and daily seasonality of the time series.

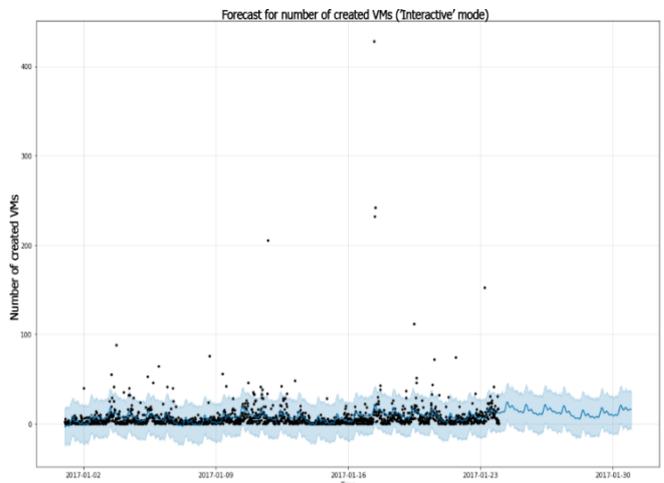


Fig. 8. Forecast for number of created VMs in 'Interactive' mode.

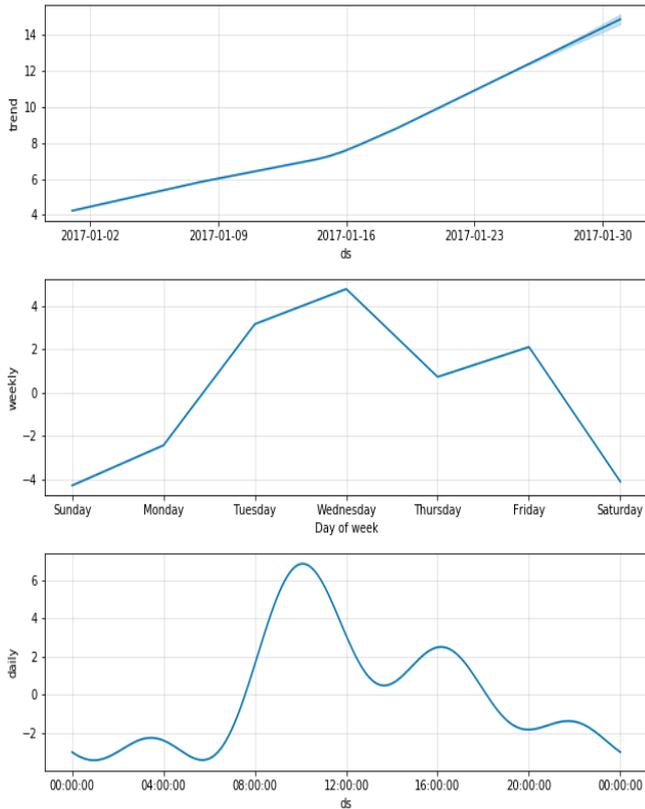


Fig. 9. Components for number of created VMs in ‘Interactive’ mode.

The next step is to perform quality estimation of prediction model. Quality metrics for ‘Interactive’ mode VMs and columns are presented in Table II. It includes Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Weighted Absolute Percentage Error (WAPE) and Mean Squared Error (MSE).

TABLE II. QUALITY METRICS FOR INTERACTIVE VMs

	‘vm_created’	‘vm_deleted’
Category	Interactive	Interactive
MAPE	inf	inf
MAE	11.419437	10.986289
WAPE	1.149005	1.083745
MSE	243.651502	451.604650

With Prophet it is also possible to calculate cross validation performance metrics. Mean values of metrics are presented in Table III.

TABLE III. CROSS VALIDATION METRICS FOR INTERACTIVE VMs

	‘vm_created’	‘vm_deleted’
Category	Interactive	Interactive
MAPE	inf	inf
MAE	5.138880	7.791051
MSE	46.804447	470.000004

Cross validation metric MAE for number of created and deleted VMs in ‘Interactive’ mode is depicted in Fig. 10 and Fig. 11 respectively.

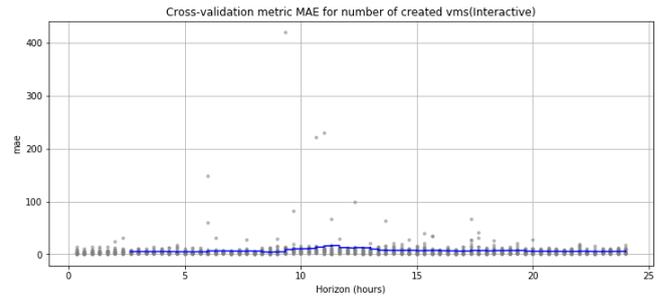


Fig. 10. Cross validation metric MAE for number of created VMs in ‘Interactive’ mode.

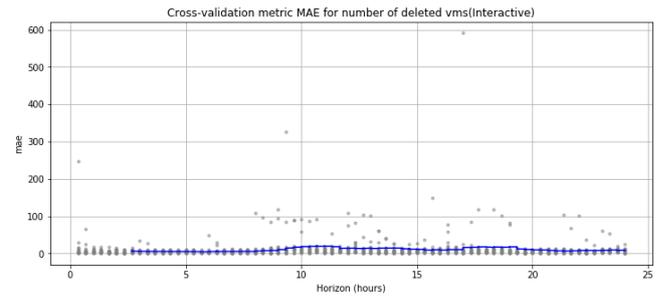


Fig. 11. Cross validation metric MAE for number of deleted VMs in ‘Interactive’ mode.

Predicted values for number of created and deleted virtual machines in ‘Interactive’ mode are depicted in Fig 12 and Fig. 13 respectively.

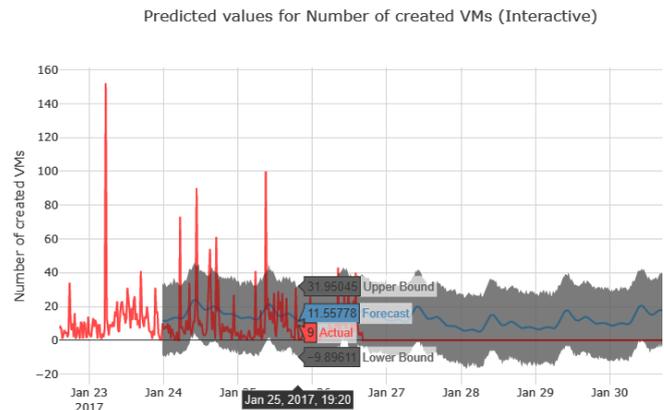


Fig. 12. Predicted values for number of created VMs in ‘Interactive’ mode.

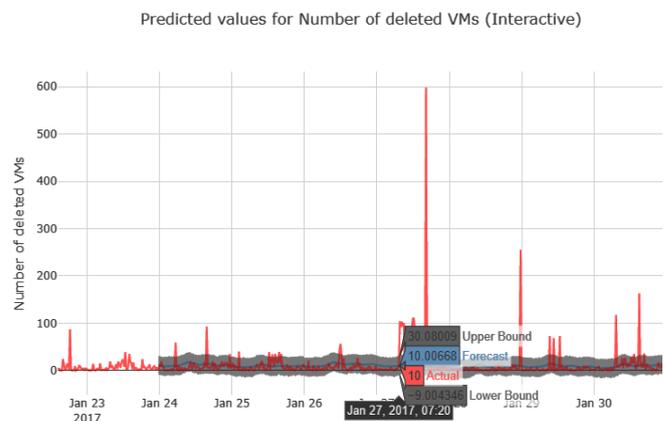


Fig. 13. Predicted values for number of deleted VMs in ‘Interactive’ mode.

It can be concluded that Prophet tool can be used to forecast number of created and deleted virtual machines in ‘Interactive mode’ since the value of MAE metric is pretty low.

Another way of producing forecasts for time series data is to use SARIMAX [10] prediction model of StatsModels library in Python. The first step is to perform a seasonal decomposition of time series by Loess (STL) to divide time series into seasonal, trend and irregular components. STL decomposition for number of created and deleted VMs in ‘Interactive’ mode is shown in Fig. 14 and Fig. 15 respectively.

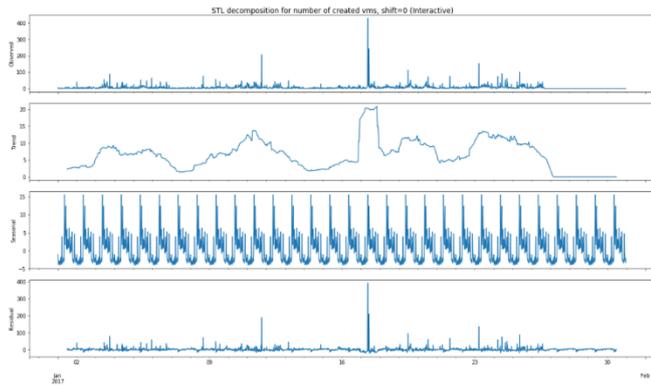


Fig. 14. STL decomposition for number of created VMs in ‘Interactive’ mode.

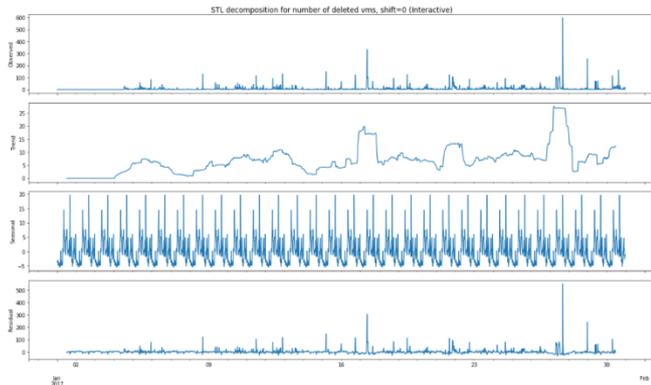


Fig. 15. STL decomposition for number of deleted VMs in ‘Interactive’ mode.

Autocorrelation and partial autocorrelation for number of created and deleted VMs in ‘Interactive’ mode are shown in Fig. 16 and Fig. 17 respectively. Blue area refers to the level below which the correlation is not significant. The seasonal shift value (shift = 577) is calculated in a way to reduce the seasonality and non-stationarity of data. To do these we calculated two seasonal differences, which correspond two one week and to one day (in terms of the number of 20 minute time intervals).

The next step is to identify parameter search space for SARIMAX models and to search for best SARIMAX models. The best parameters (p, d, q) x (P, D, Q, s) and AIC (Akaike Information Criterion) values for top 5 models out of 140 for number of created and deleted virtual machines in ‘Interactive’ mode are shown in Table IV.

TABLE IV. PARAMETERS AND AIC VALUES FOR TOP 5 MODELS OUT OF 140 FOR NUMBER OF CREATED AND DELETED VMs IN ‘INTERACTIVE’ MODE

Model id	Parameters	AIC
65	(1, 0, 3) x (2, 2, 2, 4)	17859.755887
92	(2, 0, 2) x (2, 2, 2, 4)	17859.913324
22	(0, 0, 2) x (2, 2, 2, 4)	17867.255283
127	(3, 0, 2) x (2, 2, 2, 4)	17868.656433
30	(0, 0, 3) x (2, 2, 2, 4)	17869.643598

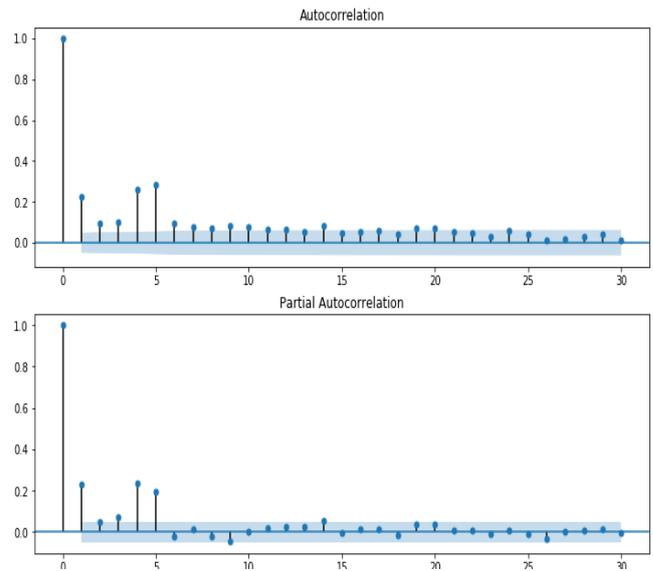


Fig. 16. Autocorrelation and partial autocorrelation for number of created VMs in ‘Interactive’ mode.

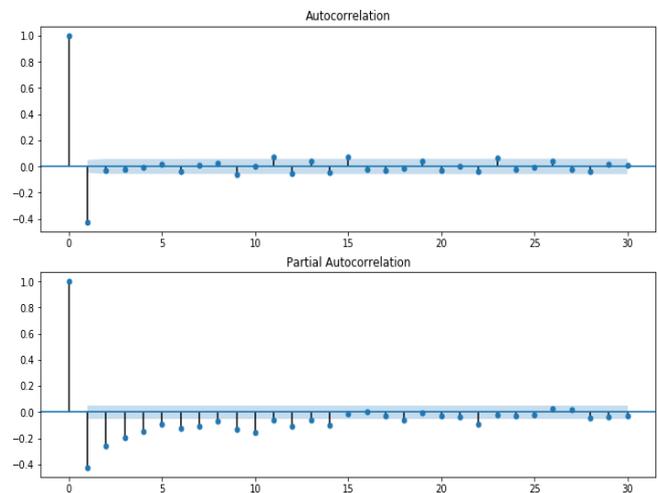


Fig. 17. Autocorrelation and partial autocorrelation for number of deleted VMs in ‘Interactive’ mode.

Actual and forecast values for number of created VMs in ‘Interactive mode’ are shown in Fig 18. ‘1 day’ forecast for number of created VMs in ‘Interactive mode’ is shown in Fig 19.

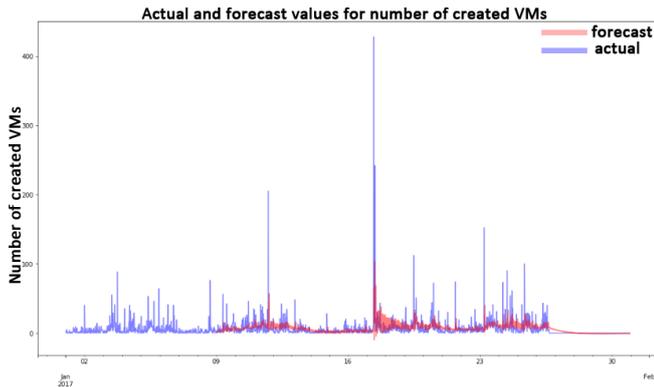


Fig. 18. Actual and forecast values for number of created VMs in 'Interactive' mode.

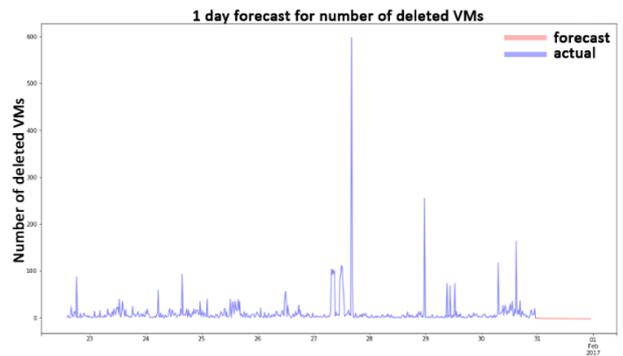


Fig. 21. '1 day' forecast for number of deleted VMs in 'Interactive' mode.

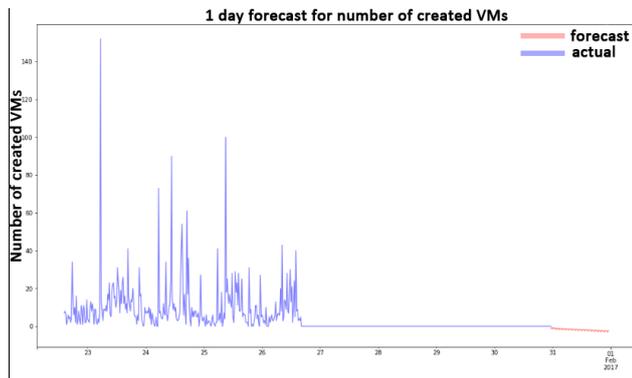


Fig. 19. '1 day' forecast for number of created VMs in 'Interactive' mode.

Quality metrics for different VM categories and columns are presented in Table V. Actual and forecast values for number of deleted VMs in 'Interactive mode' are shown in Fig 20. '1 day' forecast for number of created VMs in 'Interactive mode' is shown in Fig 21.

TABLE V. QUALITY METRICS OF INTERACTIVE VMs

	'vm_created'	'vm_deleted'
Category	Interactive	Interactive
MAPE	inf	inf
MAE	7.951340	12.216388
WAPE	inf	1.223140
MSE	145.2403	982.5438

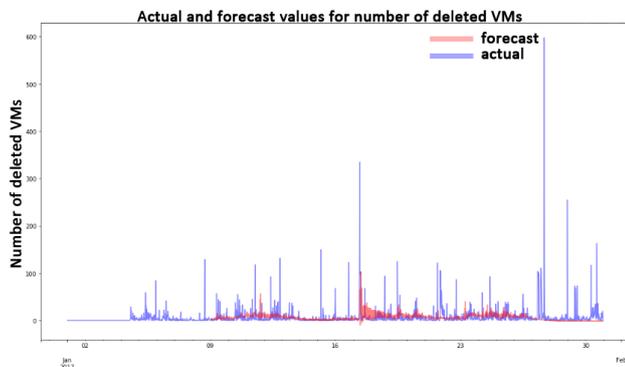


Fig. 20. Actual and forecast values for number of deleted VMs in 'Interactive' mode.

IV. CONCLUSION

Judging from values of MAE metric, it can be concluded that Prophet tool shows the best results and can be used to forecast number of created and deleted virtual machines for the multi-cloud shared access center simulator. Our next goal is to perform more in-depth statistical analysis of Azure VMs dataset in order to design an adequate and reliable cloud system model for generating end-user requests. This model will be used to create an algorithm for efficient scaling and distribution of load on virtual machine pools used to host containers with virtual desktop environments taking into account the predicted characteristics of the flow of end-user requests.

ACKNOWLEDGMENT

THIS WORK WAS SUPPORTED BY THE RUSSIAN FOUNDATION FOR BASIC RESEARCH, PROJECTS NO. 18-37-00460 MOL_A, 16-29-09639, 18-07-01446

REFERENCES

- [1] Malik A. W. et al. "CloudNetSim++: A GUI based framework for modeling and simulation of data centers in OMNeT++" IEEE Transactions on Services Computing. – 2017. – V. 10. – №. 4. – pp. 506-519.
- [2] J Sharkh M. A. et al. "Building a cloud on earth: a study of cloud computing data center simulators" Computer Networks. – 2016. – V. 108. – pp. 78-96.
- [3] Khazaei H., Mistic J., Mistic V. B. "A fine-grained performance model of cloud computing centers" IEEE Transactions on parallel and distributed systems. – 2013. – T. 24. – №. 11. – pp. 2138-2147.
- [4] Chung L. et al. "A goal-oriented simulation approach for obtaining good private cloud-based system architectures" Journal of Systems and Software. – 2013. – V. 86. – №. 9. – pp. 2242-2262.
- [5] Altevoigt P., Denzel W., Kiss T. "The IBM performance simulation framework for cloud" IBM Research Technical Paper <http://domino.research.ibm.com/library/cyberdig.nsf/papers>. – 2013. – pp. 1-9.
- [6] Garg S. K., Buyya R. "An environment for modeling and simulation of message-passing parallel applications for cloud computing" Software: Practice and Experience. – 2013. – V. 43. – №. 11. – pp. 1359-1375.
- [7] Cortez E. et al. "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms" Proceedings of the 26th Symposium on Operating Systems Principles. – ACM, 2017. – pp. 153-167.
- [8] Project Jupyter – URL: <https://jupyter.org/>
- [9] GitHub - facebook/prophet: Tool for producing high quality forecasts for time series data that has multiple seasonality with linear or non-linear growth. URL: <https://github.com/facebook/prophet>
- [10] statsmodels.tsa.statespace.sarimax.SARIMAX URL: <https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html>