Public IaaS Cloud Simulator

1st Berat SERT
Software Engineering Dept.
Bogazici University
Istanbul, Turkey
beratsert41@gmail.com

***Abstract*—As a final project of the SWE579 course in Software Engineering Master's program, conductşng experiments in Public IaaS Cloud Simulator has been chosen. In the beginning of the report, it is possible to find the general information about the cloud simulators and properties of the PICS. Then, there are requirement, usage and parameters sections as a guidance to warm up the simulator. At the final part, the results of the experiments with visualization and their brief explanation are placed.**

***Index Terms*—cloud computing, cloud providers, cloud simulator, PICS, Public IaaS Cloud Simulator**

I. Introduction

Public IaaS (Infrastructure as a Service) Cloud Simulator is a software tool that allows users to simulate the use of a public cloud infrastructure. It provides a virtual environment that emulates the functionality and features of a public cloud platform, such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform [1].

Using cloud systems is essential for tech companies. Although cloud providers have their own cost plan for their services, the clients are in need of cost calculation for different scenarios. With a public IaaS cloud simulator, users can choose relevant parameters for the simulation and create a system with predefined workload in a simulated environment, without incurring the costs and complexities of using a real cloud infrastructure. The simulator allows users to create and manage virtual resources, such as virtual machines, storage, and networking, in a way that is similar to how they would do so in a real cloud environment.

There are several public IaaS cloud simulators available, each with its own features and capabilities. Some popular examples include CloudSim, OpenStack, and CloudSim Plus. There are a few key requirements that a public IaaS cloud simulator should meet in order to be effective:

- Compatibility with real cloud platforms: A good public IaaS cloud simulator should be able to simulate the APIs and services offered by popular public cloud platforms, such as AWS, Azure, and GCP. This allows developers to test and develop their cloud-based applications and services in a familiar environment.
- Scalability: Public IaaS cloud simulators should be able to scale up or down as needed to meet the demands of the applications and services being tested. This is important because real cloud environments can be highly dynamic, with workloads fluctuating based on demand.
- Flexibility: A public IaaS cloud simulator should offer a range of configuration options, including the ability to customize network topologies, storage options, and security settings. This allows developers to test their applications and services in a variety of different scenarios.
- Ease of use: A public IaaS cloud simulator should be easy to set up and use, with a user-friendly interface and clear documentation. This makes it easier for developers to get started and quickly test their applications and services.

Overall, PICS provides a realistic and flexible environment for testing and simulating cloud-based applications and services without deploying the actual cloud application, while also being easy to use and scalable.

A. Capabilities

PICS allows its users to assess various aspects of cloud services such as cost, job response time and resource allocation. After running the simulation, the simulation generates time-series reports for tracking simulations. Users can use this reports to evaluate their experiments.

Also, it is possible to say that dynamic workload configurability is important for flexibility and scalability. Simulations in PICS can be adjusted to have dynamic job arrival patterns and SLA requirements.

In cloud computing, horizontal scaling (also known as "scaling out") refers to the practice of adding more instances of a service or application to handle an increase in workload. This is done by adding more virtual machines, containers, or other types of compute resources to the system.

Vertical scaling (also known as "scaling up") refers to the practice of increasing the capacity of a single instance of a service or application by allocating more resources to it. This is done by increasing the amount of CPU, memory, or storage available to the instance.

Auto scaling is a feature of cloud computing platforms that allows users to automatically scale their applications and services up or down based on predefined rules or triggers. For example, a user may set up an auto scaling rule to add more instances of an application if the average CPU utilization exceeds a certain threshold.

Horizontal auto scaling involves adding or removing instances of an application or service based on workload demands. This is useful for applications that can handle an increase in workload by adding more instances, such as those designed to run in a distributed or parallel manner.

Vertical auto scaling involves increasing or decreasing the capacity of a single instance of an application or service based on workload demands. This is useful for applications that require more resources to handle an increase in workload, such as those with a large memory footprint or those that perform computationally intensive tasks.

Overall, horizontal and vertical scaling are both useful techniques for managing the performance and cost of cloud-based applications and services, and auto scaling can help automate this process. PICS provides a realistic support to simulate auto scaling. Also, there are entities to schedule custom job policies and job failure cases.

Another good capability of the PICS is support for the various types of cloud resource types such as virtual machine instances and storage services. Also, it can be found that their own unique billing models and performance uncertainty parameters can be adjusted inside the simulations.

B. Validation Results

The validation results of PICS refer to the effectiveness of the simulator in accurately reproducing the behavior and performance of a real public IaaS cloud platform. In other words, it measures how closely the simulated environment mimics the real-world conditions of a public cloud platform.

In [1], the researchers used the same horizontal test setups in AWS and PICS. They used the same resources and aim is to compare time series costs data of the simulation with calculated cost by AWS. As can be seen from Equations 1 & 2, the researchers calculated overall costs and error of the different workloaded scenarios with using these equations. As a result, the average error of the simulation results is only 2.6% compared to the actual results in AWS. Also, as can be seen in Fig 1, the costs are correlated.

$$Cost = \sum_{i=1}^{n} cost_{VM_i} \quad (1)$$

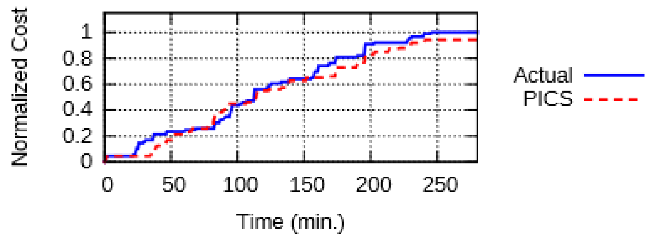
$$\%SimulationError = \left| \frac{cost_{actual} - cost_{simulation}}{cost_{actual}} \right| \times 100 \quad (2)$$

II. Requirements

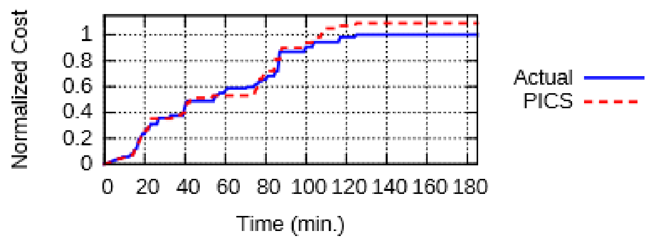
Firstly, you need to clone the official repository in your local computer. PICS is written in Python 2.7 and includes couple of outdated libraries. It is strongly recommended to download Python 2.7 in virtual environment and creating a requirement file with the libraries used in the project with the corresponding versions as can be seen in Table I.

TABLE I
REQUIREMENTS

Packages	*Version*
Python	2.7
Numpy	1.16
Pandas	0.24
Statsmodels	0.9



(a) VM Cost Trace for WL #13



(b) VM Cost Trace for WL #16

Fig. 1. Cost Traces for Vertical Scaling Cases. [1]

III. Usage

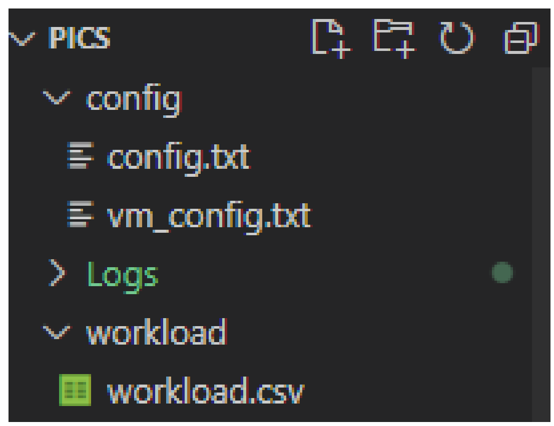


Fig. 2. Folder Structure of PICS.

Once installed the PICS and the virtual environment; as can be seen in Fig 2, there are “config.txt“ and “vm_config.txt“ under config folder. It can be used these files to create and configure virtual resources, such as virtual machines, storage volumes, network configurations and resource management policies. Then, these resources are used to simulate your cloud-based applications and services.

PICS typically offers a range of configuration options, allowing you to customize network topologies, storage options, billing models, startup process and fob failures. This allows you to test your applications and services in a variety of different scenarios.

As can be seen in Fig 2, there is an “workload.csv“ which enables to create workload transfer from outside to cloud or from cloud to cloud to simulate scenarios. Workload configuration allow you to simulate high levels of traffic or activity on your virtual resources, allowing you to evaluate the performance and scalability of the applications and services in terms of cost, storage and network.

Once the config and workload files are prepared to run simulation, running the "run_simulation.py" in the virtual environment created for the PICS would start the simulation. After the simulation completes all the workload, the simulation ends and logs the simulation in Logs folder which enables to access generated reports for the simulation.

IV. Parameters

You can find classified config parameters and their brief description in this section.

A. Simulation Configurations

TABLE II
SIMULATION CONFIGURATION PARAMETERS

Parameter	***Description***	***Default Value***
SIM_TRACE_INTERVAL	Simulation Trace Interval in seconds	60
WORK_LOAD_FILE	Workload File Path	workload.csv
VM_CONFIG_FILE	VM Configuration File Path	vm_config.txt

B. Public IaaS Configurations

TABLE III
PUBLIC IAAS CONFIGURATION PARAMETERS

Parameter	***Description***	***Default Value***
VM_BILLING_TIME_UNIT	Set IaaS Pricing Model (Hour or Minbased)	BTU_HOUR
VM_BILLING_TIME_PERIOD	Set Billing Time Period (Int and 0)	1
MIN_STARTUP_LAG	Minimum Startup LagTime for Creating a new VM	30
MAX_STARTUP_LAG	Maximum Startup LagTime for Creating a new VM	60

C. Cloud Storage Configurations

TABLE IV
CLOUD STORAGE CONFIGURATION PARAMETERS

Parameter	***Description***	***Default Value***
MAX_CAPACITY_OF_STORAGE	Max Volumn of Cloud Storage (Hour or Minbased)	10240000
STORAGE_UNIT_COST	Storage Usage Cost ($) for Gigabytes/Month	0.1
STORAGE_BILLING_TIME_UNIT	Storage Billing Time Unit (Second, 0)	1

D. Network Configurations

TABLE V
NETWORK CONFIGURATION PARAMETERS

Parameter	***Description***	***Default Value***
PERF_DATA_TRANSFER_CLOUD	Network Bandwidth for Data Transfer (Unit: MB/s)	3.0
PERF_DATA_TRANSFER_IN	Bandwidth from Incoming Traffic	2.0
PERF_DATA_TRANSFER_OUT	Bandwidth for Outgoing Traffic	1.0
COST_DATA_TRANSFER_CLOUD	Network Cost from Cloud to Cloud)	0.01
COST_DATA_TRANSFER_IN	Network Cost for Incoming Traffic	0.02
COST_DATA_TRANSFER_OUT	Network Cost for Outgoing Traffic	0.03

E. Job Management Configurations

TABLE VI
JOB MANAGEMENT CONFIGURATION PARAMETERS

Parameter	***Description***	***Default Value***
JOB_ASSIGNMENT_POLICY	Job Scheduling Configuration	EDF
PROB_JOB_FAILURE	Probability for Job Failure Occurrence	0.05
JOB_FAILURE_POLICY	Job Failure Recovery Policy	JF-POLICY-01

F. VM Management Configurations

TABLE VII
VM MANAGEMENT CONFIGURATION PARAMETERS

Parameter	***Description***	***Default Value***
VM_SELECTION_METHOD	VM Selection Policy for VM Scaling-up	VM-SEL-COST
MAX_NUM_OF_CONCURRENT_VMS	Maximum Number of Concurrent VMs	UNLIMITED
VM_SCALE_DOWN_POLICY_NAME	VM Scale Down Policy	SD-IM
ENABLE_VERTICAL_SCALING	Vertical Scaling	No
VERTICAL_SCALING_OPERATION	Vertical Scaling Options	VSCALE-BOTH

For further details about the parameters, you can review the official document of PICS.

V. Simulations and Results

In this section, there will be scenarios corresponding parameters and simulation setup, simulation results visualized and their assessments.

A. Auto-scaling

While reading documentation of the PICS and as you can see from the configuration parameters, it is important to observe the effect of auto-scaling in the simulator. With using the same workload, enabling vertical scaling is the only controlled parameter in this simulation. Cost minimizing algorithm has been preferred in vertical scaling. As can be seen from Figure 3, although there is no disadvantages in terms of performance since they completed the tasks at the same time, vertical scaling out weights the default simulation. The vertical scaling would lead to 20% cost reduction.

B. Startup Cost

In this simulation, the auto-scaling for horizontal and vertical scaling is disabled. Pair of small and high workload is defined. The expected behaviour of the small workload to trigger the setup of the VM instances, while the machines with high workload to work constantly. As you can see from Figure 4, the VM instances of the simulation with small workload become idle which triggers to stop the instances and then start up again when there is a workload again. The effect of the start up costs on total cost can be seen Figure 5, which means the startup cost can be main source of the costs so the clients should pay attention while using the cloud computing services.

C. Network Bandwidth

In this simulation, the network bandwidth is the controlled parameter. The network bandwidth was doubled to the effect on simulation. As you can see from the Figure 6, increasing network bandwidth would lead to decreasing in cost. Since there is a bottleneck in the receiving input from the the outer

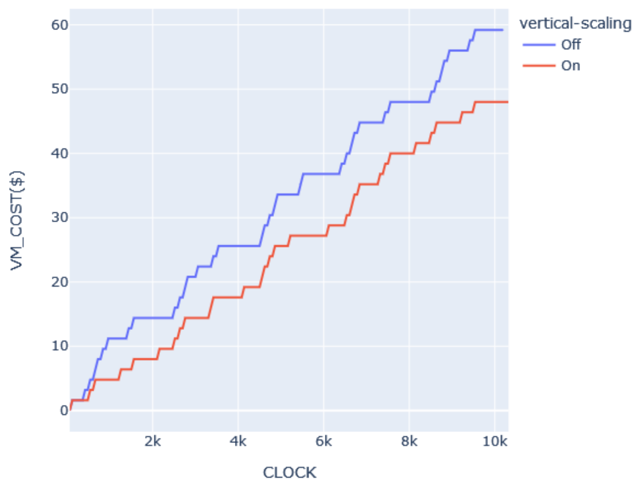


Fig. 3. the Effect of Vertical Scaling on Cost.

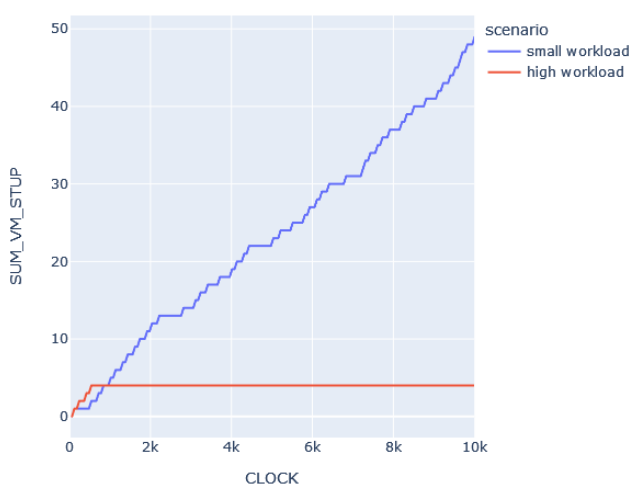


Fig. 4. the Effect of Workload on the count of Start up VM instances.

sources for cloud services. The significant capacity of the VM instances are not used. Also, it is possible to mention that the duration for completing the same workload is decreased as well.

VI. Conclusion

PICS is a tool that aims to mimic the functionality of a cloud computing platform, such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP). PICS allows developers to test and assess cloud-based applications and services in a controlled environment, without incurring the costs associated with using a real cloud platform.

There are several reasons why developers might choose to use these simulators:

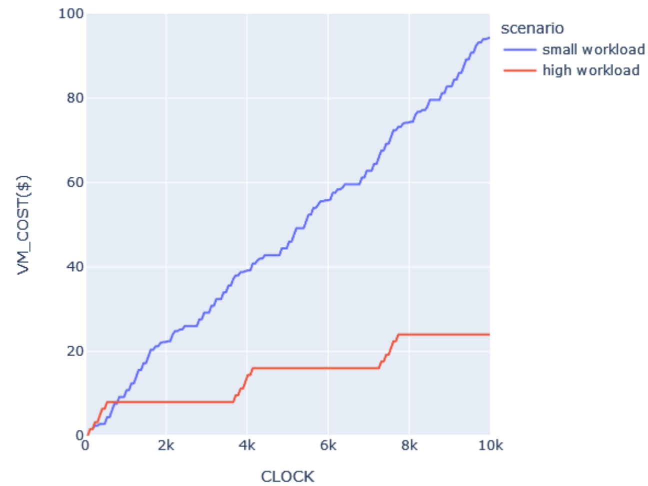


Fig. 5. the Effect of Start up VM instances on Cost.

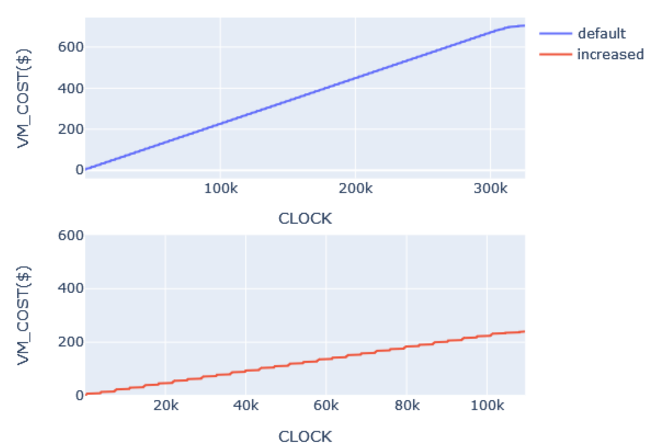


Fig. 6. the Effect of Network Bandwidth on Cost.

- Cost savings: Using PICS can help developers save money on cloud computing costs. Because the simulator is running locally, developers do not need to pay for cloud resources, such as virtual machines or storage, while testing and assessing their applications and services. Then, they would have understanding of the factors effecting to cost.
- Controlled environment: PICS allows developers to test their applications and services in a controlled environment, where they have complete control over the resources and settings. This can be helpful when testing new features or trying out different configurations. On the other hand, developers will not be able to control

easily all of the resources and the settings if they desire to conduct an experiment in real environment.

- Familiarity: PICS aims to mimic the APIs and services offered by popular cloud platforms, such as AWS, Azure, and GCP. This allows developers to test and assess their applications and services in a familiar environment, which can increase the learning curve and improve efficiency.
- Scalability: PICS offers the ability to scale up or down as needed to meet the demands of the applications and services being tested. This allows developers to test the scalability of their applications and services, and identify any potential bottlenecks or issues, which helps to have understanding about scalability of their applications.

Overall, PICS can provide developers with a cost-effective, controlled, and familiar environment for testing and assessing cloud-based applications and services. They can also help developers improve the scalability and performance of their applications and services, ultimately leading to better quality software.

References

[1] Kim, I. K., W. Wang and M. Humphrey, "Pics: A public iaas cloud simulator", 2015 IEEE 8th International Conference on Cloud Computing, pp. 211–220, IEEE, 2015.