Proceedings of the 52nd Hawaii International Conference on System Sciences | 2019

Statistical Analysis and Modeling of Heterogeneous Workloads on Amazon's Public Cloud Infrastructure

Frederick Nwanganga
University of Notre Dame
fnwanganga@nd.edu

Nitesh V. Chawla
University of Notre Dame
nchawla@nd.edu

Gregory Madey
University of Notre Dame
gmadey@nd.edu

Abstract

Workload modeling in public cloud environments is challenging due to reasons such as infrastructure abstraction, workload heterogeneity and a lack of defined metrics for performance modeling. This paper presents an approach that applies statistical methods for distribution analysis, parameter estimation and Goodness-of-Fit (GoF) tests to develop theoretical (estimated) models of heterogeneous workloads on Amazon's public cloud infrastructure using compute, memory and IO resource utilization data.

1. Introduction

As more organizations migrate their technology services from local data centers to public cloud infrastructure, the need to minimize cost while maintaining Quality of Service (QoS) guarantees becomes critically important. However, due to the opaqueness, heterogeneity, scale and tenancy issues with public cloud environments, the development of resource management policies for optimal workload resource allocation is difficult. The evaluation of these policies is even more challenging because of the difficulty in reproducing and controlling the environment in which they can be evaluated. As a result, cloud workload resource allocation is often a heuristic process. This approach increases the risk of over-provisioning and under-provisioning of resources which consequently result in increased Total Cost of Ownership (TCO) and Service Level Agreement (SLA) violations, respectively.

Rather than relying on heuristics for the allocation of workloads to cloud resources, the adoption of an analytic approach is likely to yield better long term results. There are two critical requirements for this to be successful. The first is that the environment requires extensive and continuous analysis in order for its characteristics to be understood and quantified. The second is that the quantified parameters have to be exploited in order to develop simulation models which accurately represent operational conditions.

2. Problem Definition & Importance

Several prior efforts have attempted to address workload analysis and modeling by developing mechanisms to characterize workload patterns in public cloud infrastructure [1], [2], [3]. However, most of these efforts are predominantly based on coarse-grained statistics [4] over a short period of time [5] and simply focus on analyzing cloud workload diversity by classifying tasks using critical characteristics [6], [7], [8]. Most of the approaches lack a comprehensive model that has sufficient detail about the parameters derived from the analyses.

The first objective of this paper is to present an approach for the in-depth empirical analysis and modeling of public cloud workloads. Building on the previous work done by [9] and [10], our effort uses metrics gathered over a 14 month period from an Amazon Web Services (AWS) Virtual Private Cloud (VPC) to develop statistical distributions of the workload patterns, estimate the parameters of the distributions in order to create a target function (model) and validate the simulated theoretical model against the empirical data. An important contribution of the simulated models is that they are not a simple replay of the collected data but rather models with random patterns based on realistic parameters. This allows for a dynamic simulation of the operational environment in order to understand the impact of proposed optimization policies so they can be validated before being implemented in real world environments.

The second objective of this research is the development of a pipeline for the continuous extraction, transform and load (ETL) of utilization metrics for the cloud workloads.

The final objective of this research is the practical application of the models developed in this work as input to our previous work [11]. In our previous effort,

URI: https://hdl.handle.net/10125/59626
ISBN: 978-0-9981331-2-6

we developed an Integer Linear Programming (ILP) model for resource allocation based on the location (mean) and spread (standard deviation) of the workload data. The simplifying assumption in that effort was that the workload parameters of CPU, Memory and IO are normally distributed. However, in this research, we use a more fine-grained approach to modeling workload data by fitting them against theoretical distributions and estimating their parameters. This allows us to generate simulations that help validate the resource management policies generated by our ILP model before making them operational.

The remainder of this paper is organized as follows: In the rest of this section we discuss some of the challenges and importance of workload modeling in the Cloud. Section III presents some of the previous work on cloud workload modeling. Section IV describes our data set and the relevant metrics used in our analysis. In sections V, VI and VII we discuss our methodology for cluster and distribution analysis. Our results are presented in section VIII, followed by our conclusion and future work in sections IX and X, respectively.

2.1. Challenges of Workload Modeling in the Cloud

Workload modeling and analysis is especially challenging in a heterogeneous, large and highly dynamic environment such as a public cloud data center for several reasons. Some of these reasons include:

- **Opaqueness:** While users are able to interact with their services in AWS, they have no visibility or access to the physical infrastructure that hosts them. As a result, a user has limited knowledge about which specific servers, network components and storage devices service their workloads. This lack of visibility can lead to inaccurate assumptions about the capacity of compute, memory or IO when attempting to develop resource management or allocation policies.

- **Tenancy:** Amazon, like most public cloud providers, makes use of an over-subscription model [12]. With this approach, the underlying hardware is shared by multiple service subscribers which in turn, can result in competition for shared resources (such as storage and networking). Users have no knowledge of the behavioral patterns or resource needs of the other workloads on the shared platform. While over-subscription is useful for cloud providers in order to leverage under utilized capacity, it can lead to overload conditions which have adverse impact on subscribers [12]. Multi-tenancy can result in unexplained or unpredictable behaviors in workload performance patterns.

- **Scale:** As a leader in the IaaS (Infrastructure as a Service) space [13], AWS operates at a massive scale in terms of the number of physical data centers and devices managed. An empirical failure-analysis of large-scale cloud computing environments shows that with an increase in the number components managed, the failure characteristics for workloads and servers are highly variable [14]. This leads to an issue of differential observability, which is a key component of gray failure [15]. Differential observability implies that due to scale, system failures may not be apparent to consumers even if their services are being affected by the failures. This can manifest as unpredictable performance patterns in services.

- **Heterogeneity:** Workloads can be broadly classified according to their consumption of CPU, memory and IO resources [16]. Workloads can also be classified based on their interaction with end-users as either latency-sensitive or batch [17]. Cloud workloads can be submitted at any time and have different characteristics and profiles with different resource needs. This high degree of variability makes it difficult to predict patterns over time.

2.2. Importance of Workload Modeling in the Cloud

As an abstraction of reality, cloud workload modeling increases the understanding of the behavioral patterns of workloads, consequently leading to more informed decision making with regards to resource allocation policies [8], [4]. As mentioned previously, workload modeling also allows for the simulation and evaluation of resource management policies without deploying them in production environments.

Modeling permits sensitivity analysis, which allows a focus on the assumptions and parameters which impact performance in a meaningful way, and not on those that do not.

In order to generate realistic models, it is critical that the data used to create the workload models are curated from real-world cloud metrics. This helps to capture the intrinsic diversity, dynamism and nuanced interaction between the components of the system, within the limits of the challenges mentioned in the previous section.

Table 1. Summary of related work

Authors	Trace Size	Modeling Approach	Workload Type	Model Parameters	Parameters Analyzed
Kavulya [4]	10 months	Distribution analysis	MapReduce	Yes	Duration
Zhang [5]	30 days (5 day sample)	Coarse-grain	MapReduce	Yes (partially)	CPU, Memory and Disk
Mishra [6]	4 days	Cluster centroids	MapReduce	No	CPU, Memory and Disk
Aggarwal [7]	24 hours	Cluster centroids	MapReduce	No	Disk Usage
Moreno [8][9]	29 days	Cluster centroids & distribution analysis	MapReduce	Yes	CPU, Memory, Length
Maghalhaes [10]	-	Distribution analysis	Web	Yes	CPU, Memory, Disk, Response time
Smith [18]	7 hours	Distribution analysis	MapReduce	No	Duration
Our Approach	**14 months** **(3 month sample)**	**Distribution analysis**	**Heterogenous**	**Yes**	**CPU, Memory and Disk IO**

3. Related Work

Several prior efforts have engaged in the analysis and modeling of workload patterns on cloud infrastructure from different perspectives. In this section, we will discuss the most relevant approaches to our work, as well as their limitations and gaps.

Based on the analysis of the first version of the Google tracelog, both [1] and [19] classify jobs and tasks by duration using statistical data from cloud computing workloads on the Hadoop ecosystem. This work is limited in application to the study of timing problems and unsuitable to the analysis of cloud resource utilization patterns.

The statistical analysis of 10 months of MapReduce traces done by [4] presents a description of distributions as they relate to job completion times. Though the work presents the statistical characteristics of the data as it relates to resource utilization, job patterns and source of failures, it fails to go into enough detail in exploring other factors that impact resource consumption such as capacity constraints and user demand patterns.

In [5], the authors present a study which evaluates the suitability of the mean values for task waiting time, CPU, memory and disk consumption to the representation of the performance characteristics of real traces. The work is based on historical trace data from six Google compute clusters spanning five days of operations. While the work shows that overall task usage can be described using mean values of runtime tasks, it does not address how the boundaries for task classification were made.

The authors of [6] use task resource consumption data generated by five Google clusters over four days to develop cloud computing workload classifications. The work proposes an approach that identifies workload characteristics, constructs the task classification, identifies the qualitative boundaries of each cluster and merges adjacent clusters to reduce the number of clusters. While very useful in classifying tasks, this work does not perform an analysis of the actual clusters formed.

Using data from one of Yahoo's production clusters over a 24 hour period, the work done by [7] describes an approach to characterize Hadoop jobs. The main objective of this effort was to use clustering techniques to group jobs with similar characteristics. The work is limited to a focus on the storage system and neglects other critical resources such as CPU and memory.

[8] and [9] provide an approach for characterizing cloud workloads in the context of both users and tasks using Google tracelog. While this work does use real production data curated from Google tracelogs [8], it is limited to a two day sample and [9] ignores storage resource utilization.

The work done by [10] presents an approach to model resource utilization for interactive applications such as a web server. The work makes use of the RUBiS [20] benchmark to evaluate the impact of users on resource consumption patterns. While the model developed by the work defines distributions and parameter estimates, it is based on a simulation of user behavior on a private cloud environment.

From the analysis of the related work, it is clear that previous analyses present gaps and limitations that need to be addressed in order to achieve more realistic workload patterns. Table 1 presents a summary of the related work and how each compares to our approach.

4. Data

4.1. CloudWatch Metrics

Amazon CloudWatch [21] is a service for monitoring AWS resources such as Amazon Elastic Cloud Compute (EC2) instances, Amazon Elastic Block Store (EBS) volumes, Load Balancers, and Amazon Relational Database Service (RDS) instances. By default, CloudWatch provides metrics such as CPU utilization, latency, and request. Alarms can also be set using CloudWatch and it can also be used to automatically react to predefined conditions.

For our research, we collected and curated Amazon CloudWatch metrics for both Windows and Linux EC2 instances on a weekly basis. These metrics are extracted from CloudWatch into an S3 bucket and then transformed and downloaded using a combination of Python and R scripts. For the Linux-based instances, a locally installed monitoring perl script was used to collect memory, swap, and disk space utilization data, which was then remotely reported to CloudWatch as custom metrics [22]. *EC2Config* was used for the Windows systems, to collect and send memory and disk metrics to CloudWatch Logs [23].

Our working data set includes CloudWatch metrics collected from 522 Windows and Linux servers running individually unique workloads over a 14 month period. From the overall data set, we analyze subsets of data which represent metrics for the previous one-week period, previous two-week period, previous one-month period and the previous three-month period.

The EC2 CloudWatch metrics collected and used in our work are as follows:

- **EC2 Compute Utilization:** The average percentage compute units that are used by an EC2 instance for a defined period (every 30 minutes). This is the processing power required to run an application or workload on a particular instance [24]. Over the last 14 months, we have collected 8,070,886 records for this metric. We normalize this metric for our work as the variable **CPU**.
- **EC2 Memory Utilization:** The average percentage memory capacity used by an EC2 instance for a defined period (every 30 minutes). This is the memory consumed to run an application or workload on a particular instance [24]. Over the last 14 months, we have collected 5,173,491 records for this metric. We normalize this metric for our work as the variable **MEM**.

Similar to the EC2 CloudWatch metrics, we also collect data for each of the EBS volumes attached to an EC2 instance. EBS metrics are reported to CloudWatch only when the volume is attached to an instance and active. The EBS CloudWatch metrics collected and used in our work are as follows:

- **EBS Disk Read Operations:** The average number of completed read operations from an EBS device for a defined period (every 5 minutes). We can determine the average read I/O operations per second (IOPS) from this metric by dividing by 300 [25]. This metric is only available for instances and instance types that use instance store volumes. Over the last 14 months, we have collected 15,363,265 records for this metric.
- **EBS Disk Write Operations:** The average number of completed write operations to an EBS device for a defined period (every 5 minutes). We can determine the average write I/O operations per second (IOPS) from this metric by dividing by 300 [25]. This metric is only available for instances and instance types that use instance store volumes. Over the last 14 months, we have collected 15,363,258 records for this metric.

For our work, we use the sum of these two metrics at each collection period as the variable **IOPS**.

4.2. AWS Price List Service

Amazon provides an API (AWS Price List Service) for users to get up-to-date data on pricing and other attributes of specific AWS services [26]. Using this service, we extracted the following EC2 and EBS attributes: **Instance Family**, **vCPU**, **Clock Speed**, **Elastic Compute Units (ECU)** and **Memory Capacity**.

5. Methodology

Reliable performance evaluations require the use of representative workloads [27]. As we mention in section II, this is a difficult task because of the variations and complexity in user workloads and the great number of interrelated attributes and structures of workloads. Our objective is to present an approach for capturing and modeling the behavioral patterns of different application workloads on production public cloud infrastructure. Our effort can be described in three phases: data collection, data exploration and preparation, and data modeling and evaluation (see Figure 1). In the next two sections of our paper, we will focus on the last two phases of our process, which deal with cluster and distribution analysis.

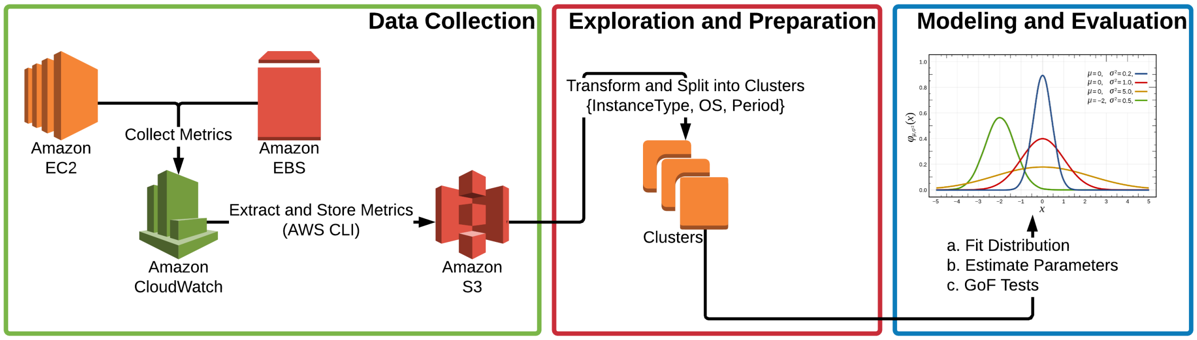

Figure 1. Data collection, analysis, modeling and evaluation workflow.

6. Cluster Analysis

The cluster analysis phase is focused on describing the characteristics and behavior of the data clusters that we analyze. This involves a study of the statistical properties of each of the parameters (CPU, Memory and IO) within each cluster. Each cluster is made up of metrics for workloads of the same instance type (or capacity), operating system and observational period. In order to access the variability or stability of our workloads over time, we partitioned the clusters using observational periods of one week each and generated summary statistics for each of them. Our results were as expected - using the most recent week as the baseline, the farther away the observed window is from the baseline, the more the workload behavior differed from the baseline (as illustrated in Figure 2). While there wasn't a meaningful change in the variance for CPU between periods, the farther back we go, we observe an appreciable shift in the location of each cluster's distribution. Both Memory and IO (to a lesser degree) metrics exhibit the same pattern.

Due to the observed variability in the behavior and characteristics of workloads over time, we decided to use four overlapping observational periods instead: a one-week period, a two-week period, a one-month period and a three-month period. This resulted in 168 different clusters. A correlation analysis between the metrics of the clusters showed a moderate to high correlation between CPU and Memory, as well as between CPU and IOPS. However, the correlation between Memory and IOPS was consistently low.

For illustrative purposes, we will limit our discussion to one of the clusters - an m4.large Windows workload for the one-week observational period. We will use this cluster to illustrate our process for the remainder of the paper, albeit for different observational periods. For our sample cluster, the correlation coefficient between CPU and Memory of 0.45 shows that as CPU resource consumption goes up so does Memory consumption. The same relationship exists between CPU and IOPS, which show a correlation of 0.55. This can be explained by the fact that as Memory demand increases, systems tend to start writing to disk more often to compensate

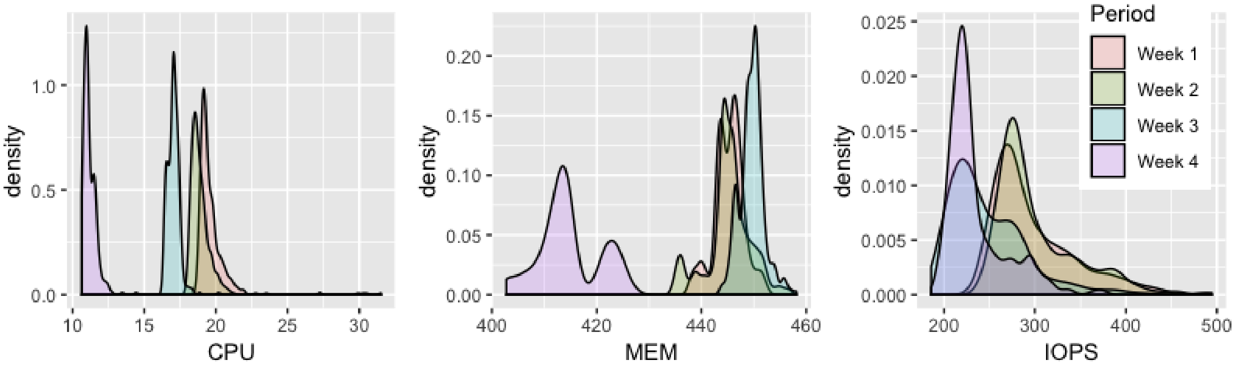

Figure 2. Change in the weekly distribution for CPU, Memory and IO over four time periods (current week, one week prior, four weeks prior and twelve weeks prior)

for resource constraints. This increase in read and write activity consequently results in an increase in CPU utilization. There is little to no direct correlation between Memory and IOPs (-0.02). While we do observe some correlation between our measures, to simplify our model, we decided to apply a naive independence to them. Therefore, we limit ourselves in the distribution analysis phase to univariate theoretical distributions.

7. Distribution Analysis

This phase consists of analyzing the data distributions for each of the 3 measures of each cluster and fitting the data to a theoretical distribution using a statistical Goodness-of-Fit (GoF) test [28]. From the parameters of the distribution we develop a Probabilistic Distribution Function (PDF), which is used as a representative target function (or model) for the data distribution of each cluster.

Building on the work done by [10], [29] and [27], the analysis was done in three major steps:

1. Statistical analysis of the data characteristics to determine the candidate distributions to represent the model;

2. Parameter estimation to set the parameters of the model using estimation methods against the selected distribution from the sample data;

3. GoF tests to evaluate whether the selected distributions and their respective parameters approximate the empirical data adequately.

We evaluated the usefulness of 21 different theoretical univariate distributions against our data. Some of the distributions evaluated include: Lambda, Generalized Lambda (GLD) [30], Burr, Kappa, 4-parameter Kappa (4P-Kappa), Generalized Extreme Value (GEV) [31], Generalized Weibull (GWD) [32], 3-parameter Error distribution (3P-Error) [33], and 6 types of the Pearson distribution system.

7.1. Parameter Estimation

Building on the work done by [34], we used seven different parameter estimation methods: Maximum Log-Likelihood (MLE), Histogram Fitting (HIST), Quantile Matching (QM), Probability Weighted Moments (PWM) [35], Method of Moments (MoM), Method of L-Moments (MoL) and Maximum Product of Spacing Estimator (MPS) [36]. Here we discuss four of the methods we used:

- **Method of Moments (MoM)**
  We start by using the Method of Moments parameter estimation approach to fit a distribution to the CPU utilization data (empirical data). We then compare the four moments of the fitted distribution with those of the empirical data.

- **Method of Maximum Likelihood (MLE)**
  According to [37], the method of maximum likelihood is the preferred method for providing definitive fits to data using distributions such as GLD. Using this approach against our data, we also get the four moments of the fit.

- **Method of L-Moments (MoL)**
  Another promising parameter estimation approach introduced in [38] and [39] is the method of L-moments. According to [39], this is a more suitable technique when dealing with heavy-tailed distributions.

- **Quantile Matching (QM)**
  We also apply quantile matching in our parameter estimation process, with the goal of numerically minimizing the sum of squared differences between our observed and theoretical quantiles.

7.2. Goodness of Fit

After we identify a potential representative distribution and its parameters, we then test if the model fits the data. The goal is to verify that the empirical and theoretical data belong to the same distribution. We use three tests: the Q-Q plots technique (graphical method), the Kolmogorov-Smirnoff (KS) test as described in [27] and the Anderson-Darling (AD) test.

- **Kolmogorov-Smirnoff (KS) Test**
  We use the Kolmogorov-Smirnoff (KS) resample test as described in [27]. This test is based on the sample statistic Kolmogorov-Smirnoff Distance (D) between the data in the sample and the fitted distribution. The null hypothesis is that the sample data is drawn from the same distribution as the fitted distribution. We run two different tests based on this method. The first function we use counts the number of times (out of a 1,000) that the KS test p-value exceeds 0.05 for the null hypothesis. The second function runs a one-sample KS test and returns the test statistic (D) as well as the p-value.

- **Graphical Comparison**
  To further evaluate goodness of fit, we look at the

Quantile-Quantile (QQ) plots and the Probability Density (PDF) plots derived from the different estimation methods. By overlaying the PDF curve of the fitted distribution on top of the histogram of the CPU distribution data we can see which estimation methods perform better than others.

- **Anderson-Darling (AD) Test**
  The Anderson-Darling test [40] is a modification of the KS test and gives more weight to tails than the KS test. While the KS test is distribution free, the AD test makes use of the distribution being tested to calculate critical values. This makes it a more sensitive test and better suited for skewed distributions with long tails such as ours. Using the AD test, we select the theoretical distribution with the lowest AD-value to represent the data distribution of each of the parameters in each cluster.

8. Results

Table 2 shows the best-fit distributions, parameter estimates and corresponding AD values based on our distribution analysis and fitting process. From the results, we can see significant diversity in the characteristics of the workload as the observation window expands or contacts. We see that the best-fit distribution for CPU varies from the Pearson type IV distribution, to the 4-parameter Kappa distribution, to a Log Gamma distribution and to a 3-parameter Gamma distribution. We see a similar level of heterogeneity with Memory as well. However, we see a homogeneous pattern with regards to the disk IO (IOPS) parameter.

The second observation we make is that both CPU and IOPS are right-tailed as can be seen in both Figures 3 and 4, while Memory is more normally distributed. This indicates that both CPU and IO resource consumption occurs in bursts followed by periods of low activity. Memory resources, on the other hand, are consumed at a more gradual pace. This can likely be explained by the operating system's memory management mechanism which tends to cache data in memory in order to limit expensive IO. These mechanisms tend to allocate and hold on to memory blocks in anticipation that it will be used in future compute cycles.

Figure 3 shows the Cumulative Distribution Function (CDF) comparing the theoretical distributions to the empirical distributions for CPU, Memory and IO for the one-week observation period. We can see from visual inspection the similarity between the two distributions for all the parameters. Similarly, Figure 4 shows the theoretical Probability Density Functions generated from our parameter estimates in comparison to the empirical PDFs of our data set.

Table 2. Best Fit distributions and parameter estimates

		Distribution	Parameters	AD Value			Distribution	Parameters	AD Value
1 Week	CPU	Pearson IV	$m = 2.1757$, $\nu = -18.5381$, $\lambda = 18.5246$, $\alpha = 0.1376$	0.4739	1 Month	CPU	Log Gamma	$\alpha = 2141.1955$, $\lambda = 735.1686$	5.7394
	Memory	Burr	$\alpha = 1.9436$, $\gamma = 248.0608$, $\theta = 0.0022$	2.9577		Memory	Burr	$\alpha = 0.4558$, $\gamma = 219.5190$, $\theta = 0.0022$	3.4414
	IOPS	Pearson IV	$m = 1.2933$, $\nu = -3.4923$, $\lambda = 243.9343$, $\alpha = 14.7534$	0.8082		IOPS	Pearson IV	$m = 1.1889$, $\nu = -0.7064$, $\lambda = 261.3269$, $\alpha = 29.6874$	3.7380
2 Weeks	CPU	4P Kappa	$\alpha = -0.2684$, $\tau = -1.5195$, $\xi = 0.3380$, $\lambda = 19.1737$	1.5477	3 Months	CPU	3P Gamma	$\alpha = 28.4381$, $\tau = 0.9699$, $\theta = 2.0320$	24.1220
	Memory	Pearson VII	$n = 3.0010$, $\lambda = 445.4054$, $s = 2.3671$	3.1084		Memory	3P Gamma	$\alpha = 5.2173$, $\tau = 11.2417$, $\theta = 384.3967$	13.5013
	IOPS	Pearson IV	$m = 1.2639$, $\nu = -2.9577$, $\lambda = 248.7673$, $\alpha = 15.6189$	1.4966		IOPS	Pearson IV	$m = 1.6784$, $\nu = -33.1663$, $\lambda = 172.6298$, $\alpha = 4.45019$	24.9011

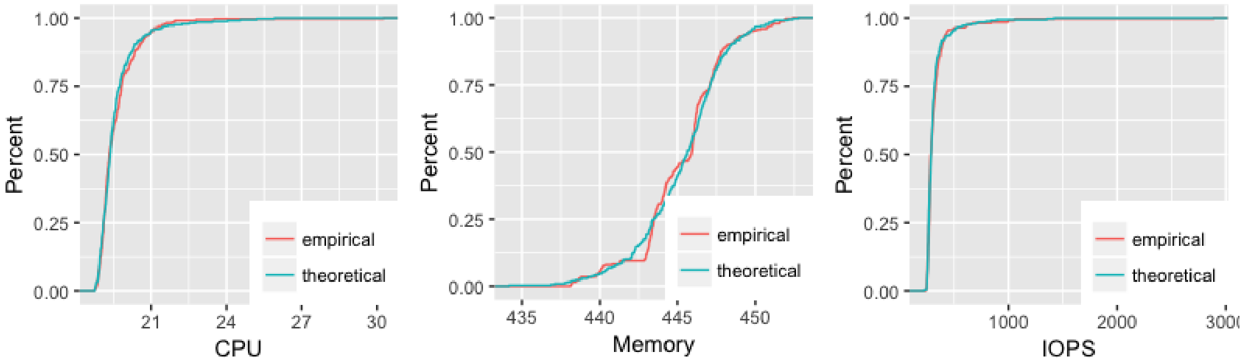

Figure 3. Empirical and theoretical CDF of CPU, Memory and IO for the one-week observation period.

In order to validate the accuracy and usefulness of our simulated models, we need to compare the data generated by our model to the real data. This requires that we compare the simulated distributions from each of our models against the corresponding data for each of the observation periods our models are based on. To do this, we use the Wilcox Mann-Whitney statistical test (WMW) [41] [42]. This test is commonly applied instead of the two-sample t-test to non-gaussian distributions, which is the case with our dataset. The WMW test is based on the null hypothesis (H0) that the distributions of two unspecified populations are equal against the hypothesis that the distributions have the same shape but are shifted. So if the p-value is greater than 0.05, then we can accept the hypothesis H0 of statistical equality of the data distributions.

From the WMW test results in Table 3, it can be observed that the simulated parameters for CPU, Memory and IOPS follow the distributions of the real data. The exception to this are the Memory parameters of the 3-month observation period, for which we have no statistical evidence to support the WMW null hypothesis. This is as a result of the multimodal nature of the 3-month memory distribution. Fitting it with a single theoretical distribution is unsuitable.

9. Conclusion

This paper presents an approach for the analysis and modeling of workloads on production public cloud infrastructure. It describes the challenges that are inherent with such an effort as well as the importance. With Amazon Web Services as the IaaS service provider, the work explains the CloudWatch metrics and additional data relevant to successfully quantifying workload behavior. Using different overlapping observational period data clusters, this paper presents a detailed process for developing a representative model of a workload with the use of distribution analysis, parameter estimation and GoF tests.

From the observations made and the results obtained, we can make some reasonable conclusions. The first is that workload patterns vary significantly across observation periods. Secondly, CPU and IO behavior are more sensitive to user patterns than Memory. The third conclusion is that developing representative simulation models that mimic actual cloud workload

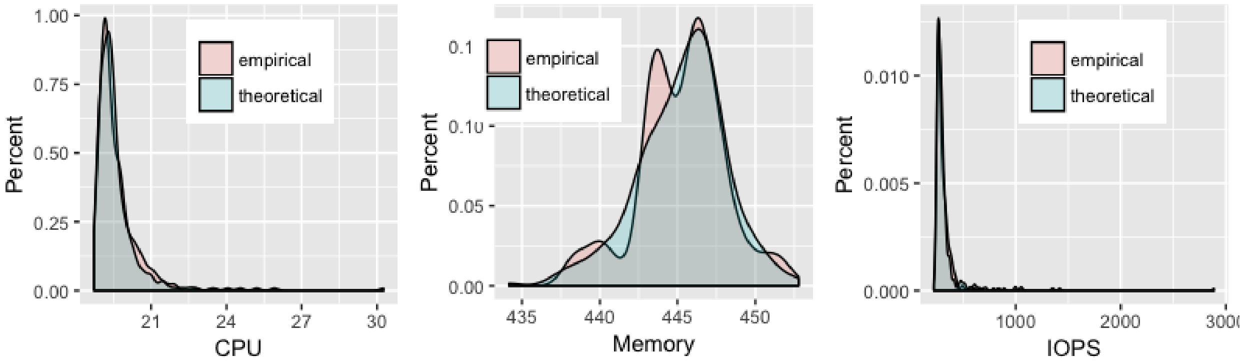

Figure 4. Empirical and theoretical PDF of CPU, Memory and IO for the one-week observation period.

behavior within a one-month window is possible. This supports the evaluation of new operational resource management policies before they are deployed in production environments.

Table 3. Wilcox Mann-Whitney Test: Empirical vs Theoretical Distributions

	WMW Test	CPU	Memory	IOPS
1 week	W	57696	55506	58075
	p-value	0.6199	0.7081	0.5179
2 weeks	W	225720	226910	227030
	p-value	0.9918	0.8747	0.8622
1 month	W	922290	888160	911880
	p-value	0.3419	0.4555	0.6652
3 months	W	7976400	$1.6e+07$	7819200
	p-value	0.8196	$<2.2e-16$	0.8009

10. Future Work

As we observed in our model validation results, the multimodal nature of the Memory distribution presents some challenges to the approach used in this paper which tries to fit the data to a single theoretical distribution. Future work will include looking at the use of multi-peak histogram analysis for region splitting [43] in order to fit the derived dataset sub-regions to new parametric distributions.

We also intend to continue building on our previous work [11] by implementing a workflow where we simulate and evaluate the resource allocation recommendations from our Integer Linear Programming (ILP) mechanism by using the techniques described in this paper. This helps support better decision making in the allocation of resources to workloads in the Cloud environment.

Besides the usefulness of distribution fitting and parameter estimation in the simulation and modeling of future workloads, the work presented in this paper also serves as a foundation for the classification of workloads in order to better forecast future behavior patterns when capacity constraints change. This work is in progress.

Acknowledgments

The authors would like to thank Brian Perri from the Office of Information Technologies at the University of Notre Dame for his assistance in the extraction and curation of the AWS CloudWatch metrics used in this research.

This research was supported in part by the National Science Foundation (NSF) Grant IIS-1447795.

References

[1] Google, "Google Cluster Data V1," 2010. [Online]. Available: http://code.google.com/p/googleclusterdata/wiki/TraceVersion1

[2] ——, "Google Cluster Data V2," 2011. [Online]. Available: http://code.google.com/p/googleclusterdata/wiki/ClusterData2011_1

[3] Yahoo, "Yahoo! M45 Supercomputing Project." 2007. [Online]. Available: http://research.yahoo.com/node/1884

[4] S. Kavulya, J. Tan, R. Gandhi, and P. Narasimhan, "An analysis of traces from a production mapreduce cluster," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. IEEE Computer Society, 2010, pp. 94–103.

[5] Q. Zhang, J. L. Hellerstein, R. Boutaba *et al.*, "Characterizing task usage shapes in googles compute clusters," in *Proceedings of the 5th international workshop on large scale distributed systems and middleware*. sn, 2011, pp. 1–6.

[6] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das, "Towards characterizing cloud backend workloads: insights from google compute clusters," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 4, pp. 34–41, 2010.

[7] S. Aggarwal, S. Phadke, and M. Bhandarkar, "Characterization of hadoop jobs using unsupervised learning," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 748–753.

[8] I. S. Moreno, P. Garraghan, P. Townend, and J. Xu, "An approach for characterizing workloads in google cloud to derive realistic resource utilization models," in *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*. IEEE, 2013, pp. 49–60.

[9] ——, "Analysis, modeling and simulation of workload patterns in a large-scale utility cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 208–221, 2014.

[10] D. Magalhães, R. N. Calheiros, R. Buyya, and D. G. Gomes, "Workload modeling for resource usage analysis and simulation in cloud computing," *Comput. Electr. Eng.*, vol. 47, no. C, pp. 69–81, Oct. 2015. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2015.08.016

[11] F. Nwanganga, M. Saebi, G. Madey, and N. Chawla, "A minimum-cost flow model for workload optimization on cloud infrastructure," in *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*, June 2017, pp. 480–487.

[12] S. A. Baset, L. Wang, and C. Tang, "Towards an understanding of oversubscription in cloud," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, ser. Hot-ICE'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 7–7. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228283.2228293

[13] L. Lydia *et al.*, "Gartner, magic quadrant for cloud infrastructure as a service." Gartner Research, 28 May 2014. [Online]. Available: http://www.gartner.com/technology/reprints.do?id=1-1UKQQA6&ct=140528&st=sb

[14] P. Garraghan, P. Townend, and J. Xu, "An empirical failure-analysis of a large-scale cloud computing environment," in *Proceedings of the 2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*, ser. HASE '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 113–120. [Online]. Available: http://dx.doi.org/10.1109/HASE.2014.24

[15] P. Huang, C. Guo, L. Zhou, J. R. Lorch, Y. Dang, M. Chintalapati, and R. Yao, "Gray failure: The achilles' heel of cloud-scale systems," in *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, ser. HotOS '17. New York, NY, USA: ACM, 2017, pp. 150–155. [Online]. Available: http://doi.acm.org/10.1145/3102980.3103005

[16] V. Vasudevan, D. Andersen, M. Kaminsky, L. Tan, J. Franklin, and I. Moraru, "Energy-efficient cluster computing with fawn: Workloads and implications," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*. ACM, 2010, pp. 195–204.

[17] T. N. B. Duong, X. Li, R. S. M. Goh, X. Tang, and W. Cai, "Qos-aware revenue-cost optimization for latency-sensitive services in iaas clouds," in *Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*. IEEE Computer Society, 2012, pp. 11–18.

[18] J. W. Smith and I. Sommerville, "Workload classification & software energy measurement for efficient scheduling on private cloud platforms," *arXiv preprint arXiv:1105.2584*, 2011.

[19] G. Wang, A. R. Butt, H. Monti, and K. Gupta, "Towards synthesizing realistic workload traces for studying the hadoop ecosystem," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*. IEEE, 2011, pp. 400–408.

[20] Rubis, "Rice University Bidding System," 2013. [Online]. Available: http://rubis.ow2.org/

[21] Amazon Web Services, Inc, "Amazon CloudWatch," 2016. [Online]. Available: https://aws.amazon.com/cloudwatch/

[22] ——, "Monitoring memory and disk metrics for amazon ec2 linux instances," 2016. [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html

[23] ——, "Sending performance counters to cloudwatch and logs to cloudwatch logs using ec2config," 2016. [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/send_logs_to_cwl.html

[24] ——, "Amazon ec2 metrics and dimensions," 2016. [Online]. Available: http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ec2-metricscollected.html

[25] ——, "Amazon ebs metrics and dimensions," 2017. [Online]. Available: http://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/ebs-metricscollected.html

[26] ——, "Using the AWS Price List Service," 2017. [Online]. Available: http://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/price-changes.html

[27] D. G. Feitelson, *Workload Modeling for Computer Systems Performance Evaluation*, 1st ed. New York, NY, USA: Cambridge University Press, 2015.

[28] N. Sematech, "Engineering statistics handbook," *NIST SEMATECH*, 2006.

[29] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson, "Statistics-driven workload modeling for the cloud," in *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*. IEEE, 2010, pp. 87–92.

[30] C. Hastings Jr, F. Mosteller, J. W. Tukey, and C. P. Winsor, "Low moments for small samples: a comparative study of order statistics," *The Annals of Mathematical Statistics*, pp. 413–426, 1947.

[31] A. F. Jenkinson, "The frequency distribution of the annual maximum (or minimum) values of meteorological elements," *Quarterly Journal of the Royal Meteorological Society*, vol. 81, no. 348, pp. 158–171, 1955.

[32] H. Pham and C.-D. Lai, "On recent generalizations of the weibull distribution," *IEEE transactions on reliability*, vol. 56, no. 3, pp. 454–458, 2007.

[33] A. M. Mineo, M. Ruggieri *et al.*, "A software tool for the exponential power distribution: The normalp package," *Journal of Statistical Software*, vol. 12, no. 4, pp. 1–24, 2005.

[34] Y. Chalabi, "New directions in statistical distributions, parametric modeling and portfolio selection," 2012.

[35] J. A. Greenwood, J. M. Landwehr, N. C. Matalas, and J. R. Wallis, "Probability weighted moments: definition and relation to parameters of several distributions expressable in inverse form," *Water Resources Research*, vol. 15, no. 5, pp. 1049–1054, 1979.

[36] R. Cheng and N. Amin, "Estimating parameters in continuous univariate distributions with a shifted origin," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 394–403, 1983.

[37] S. Su *et al.*, "Fitting single and mixture of generalized lambda distributions to data via discretized and maximum likelihood methods: Gldex in r," *Journal of Statistical Software*, vol. 21, no. 9, pp. 1–17, 2007.

[38] J. Karvanen and A. Nuutinen, "Characterizing the generalized lambda distribution by l-moments," *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 1971–1983, 2008.

[39] W. H. Asquith, "L-moments and tl-moments of the generalized lambda distribution," *Computational Statistics & Data Analysis*, vol. 51, no. 9, pp. 4484–4496, 2007.

[40] M. A. Stephens, "Edf statistics for goodness of fit and some comparisons," *Journal of the American statistical Association*, vol. 69, no. 347, pp. 730–737, 1974.

[41] D. Brown, P. Rothery *et al.*, *Models in biology: mathematics, statistics and computing*. John Wiley & Sons Ltd., 1993.

[42] A. Gold, "Understanding the mann-whitney test," *Journal of Property Tax Assessment and Administration*, vol. 4, no. 3, p. 55, 2007.

[43] S. Pal and P. Bhattacharyya, "Multipeak histogram analysis in region splitting: a regularisation problem," *IEE Proceedings E-Computers and Digital Techniques*, vol. 138, no. 4, pp. 285–288, 1991.