2015 IEEE International Conference on Cloud Engineering

Cloud Desktop Workload: a Characterization Study

Emiliano Casalicchio*, Stefano Iannucci*† and Luca Silvestri†
*Dep. of Civil Engineering and Computer Science
University of Rome Tor Vergata
Email: casalicchio@ing.uniroma2.it
†Grep s.r.l., Rome, Italy
Email: {s.iannucci, l.silvestri}@grepsrl.it

***Abstract*—Today the cloud-desktop service, or Desktop-as-a-Service (DaaS), is massively replacing Virtual Desktop Infrastructures (VDI), as confirmed by the importance of players entering the DaaS market. In this paper we study the workload of a DaaS provider, analyzing three months of real traffic and resource usage. What emerges from the study, the first on the subject at the best of our knowledge, is that the workload on CPU and disk usage are long-tail distributed (lognormal, weibull and pareto) and that the length of working sessions is exponentially distributed. These results are extremely important for: the selection of the appropriate performance model to be used in capacity planning or run-time resource provisioning; the setup of workload generators; and the definition of heuristic policies for resource provisioning. The paper provides an accurate distribution fitting for all the workload features considered and discusses the implications of results on performance analysis.**

***Keywords*-cloud computing; workload characterization; performance evaluation; cloud desktop; desktop-as-a-service; capacity planning; monitoring**

I. Introduction

Resource management is a key issue in cloud computing and is central to guarantee elasticity, high availability and performance. Moreover, an optimal management of resources allows to maximize the provider revenue and to guarantee SLAs. Offline capacity planning and runtime resource provisioning (autonomic resource provisioning) are two approaches typically used for the optimal management of resources in computer systems (and cloud computing) and both methods require a deep understanding of the system workload characteristics [1], [2].

Cloud workload characterization is an open challenge as stressed in [3]. State of the art research results can be summarized as follows: cloud workload exposes an high variability [4]; the Markovian Arrival Processes (MAP) and related MAP/MAP/1 model is a candidate tool for performance prediction of servers deployed in the cloud [5]; and that Hidden Markov Modeling and regression methods are useful techniques to characterize the temporal correlation and therefore to predict future workload [6], [7].

What lacks in literature is a statistical characterization of cloud workload features, mainly because strictly dependent on the type of cloud service.

In this paper we focus our attention on the workload characterization of cloud desktops services that, today, are massively replacing Virtual Desktop Infrastructures (VDI) as confirmed by the importance of some players entering the DaaS market (e.g. Amazon with AWS Workspace, Dell with Workspace-as-a-Service and VMware with Desktone).

Three distinguishing features characterize DaaS from a performance perspective. First, although DaaS is offered to the Consumers market and to the Enterprises market, the latter segment of customers is the larger. Offering the service to enterprises implies that pools of infrastructure resources are dedicated to a set of users (employees/members of the same organization) sharing goals, data, information and applications. Second, users typically consume resources over long persistent sessions (e.g. 8h) and over this time period the DaaS provider must guarantee continuous high performances. Even if the DaaS provider offer is characterized by *bundles* there is often the possibility to install on the desktop different kinds of applications or to integrate the desktop with other kinds of services, e.g. web applications, mail services, storage services. Therefore, users of the same organization can work with different applications on their desktop generating heterogeneous workloads (compared to other members of the same organization). One more interesting reason to study DaaS workload is that, often, cloud desktops are deployed as private clouds. In this context the proper provisioning of resources has an higher impact on costs and performances, and workload knowledge is very important.

Today there are very few studies on DaaS performances. In [8] the authors proposed a synthetic benchmark that, however, is not based on real workload observation or statistical properties. In [9] the authors characterize a VDI system accessing only web mail application. In [1] is presented a detailed study of desktop and workstation workload running on single nodes (pc or servers). Finally, VMware [10] proposed a capacity planning methodology for VMware VDI that is based on synthetic workload.

Our contribution to the literature is a detailed characterization study of the DaaS workload. We monitored a DaaS provider [11], offering services to SMEs, for three months, from Mar. 23 to Jun. 22, 2014. From the analysis of data

978-1-4799-8218-9/15 $31.00 © 2015 IEEE
DOI 10.1109/IC2E.2015.25

we extract important informations on:

- the characteristics of users behavior;
- the statistical properties of the CPU load of Virtual desktop; and
- the statistical properties of the read/write rate interesting the infrastructure storage systems.

We concentrate our attention on the CPU load of the VMs running virtual desktops and on the storage systems because these are the more critical and more stressed components of the infrastructure. Servers memory usage and network usage are not considered in this study for two reasons. First, real system observations show that memory usage is almost constant or shows slight variability. Second, despite the virtual desktop is a network sensible application, real system observations highlight that bandwidth consumption is not a critical factor for desktop running office automation applications. Therefore we decided to not consider network traffic characterization.

The main results of our characterization are that the CPU load and read/write rate are long-tail distributed and fit with a lognormal distribution. Parameters were accurately estimated for all the considered cases. Moreover, we observed also that the CPU load distribution is invariant to the number of concurrent desktop activated by the client. Indeed, the number of concurrent working sessions influences only the scale parameter of the distribution but not the distribution type.

The paper is organized as follow. Related works are described in Section II. The system architecture is presented in Section III. Section IV describes the workload model and Section V presents the workload analysis. Concluding remarks are in Section VI.

II. Related Works

In literature there are few works focused on the characterization of cloud workloads [4]–[7], [12], [13] and very few characterizing DaaS or VDI workloads [8]–[10].

Cloud workloads, as illustrated in [4], are more variable and difficult to characterize and to predict than traditional workloads. To highlight the differences in terms of workload between modern cloud systems and Grid/HPC systems, in [4] is proposed a study of the workload of a production data center at Google. Comparing Grid and HPC systems, the authors found that Google jobs are usually shorter and are submitted at higher frequency, leading to a finer resource allocation granularity. Therefore, hosts load in cloud data centers has an higher variance than in Grids. The characterization of cloud workload using Google traces has been done also in [12], where a reusable workload generation model based on real operational data extracted from a 30 day tracelog from Google Cloud has been proposed. The model considers the workload composed by two principal elements: tasks (defined as the basic units of computation performed in the cloud) and users (i.e., the actors responsible for creating and configuring the volume of tasks to be computed). In [5] MAP and the MAP/MAP/1 queueing model are used as tools for performance prediction of servers deployed in the cloud. In the paper a maximum likelihood method for fitting a MAP to the web traffic measurement collected in HTTP web server traces is presented. Moreover, a methodology that supports the handling of short traces during the modeling and simulation activities, and the different request types in HTTP workloads is presented to parametrize the MAP/MAP/1 model for web server performance prediction. The authors of [6] present a method to characterize and predict workload in a cloud environment. This method searches for repeatable workload patterns by exploring cross-VM workload correlations resulting from the dependencies among applications running on different VMs that belong to a cloud customer. A co-clustering technique is developed to identify groups of VMs that frequently exhibit correlated workload patterns. This method allows to predict individual VM's workload based on the groups identified in the clustering phase. In [7] workload characterization is performed using the information available at the virtual machine monitor (VMM) level. After identifying a set of canonical workloads (i.e., CPU, memory, disk read, disk write, network receive and network transmit), regression algorithms are used to express a target workload as a linear combination of the activity of the canonical workload set. For each workload in the canonical set, low-level data available at the VMM level are collected and processed to produce a set of features that are provided as input to the regression algorithms. A qualitative model of the workload behavior is obtained as output using multiple linear least-square regression.

The authors of [13] use statistical models to predict resource requirements for data intensive applications in the cloud. The execution time of MapeReduce jobs is described and evaluated using Kernel Canonical Correlation Analysis (KCCA), that allows to simultaneously predict multiple performance metrics using a single model. MapReduce optimization is evaluated using a statistics driven workload generator synthesizing realistic workloads using the models developed in the KCCA framework.

Except the results presented in [5], [13], none of the above works gives a statistical characterization of the workload that can be used in capacity planning. However Pacheco-Sanchez et al. tested their methodology only on web server load and Ganapathi et al. consider only MapReduce workloads.

Concerning the characterization of Cloud desktop workloads, in [10] VMware use two synthetic workloads (heavy and light) to test a methodology to determine the server capacity needed for a VDI deployment. Nevertheless, this study is specific for the virtual desktop solution proposed by VMware and does not provide any statistical characterization of the VDI workload.

Another study focused on remote desktop systems is

conducted in [9]. The authors model a remote desktop system through the case study of an office application: email. The proposed workload model, based on discrete time Markov chain, although allowing to improve resource management, is limited to a single application and cannot be used on VDI or DaaS systems running different concurrent applications. Focused on DaaS is the work presented in [8] where the authors present a human-centric reference architecture for modeling and assessment of objective user Quality of Experience within virtual desktop clouds. The authors propose a VDI benchmark that is not parameterized using statistical properties of real VDI workloads.

III. Reference architecture

The cloud desktop computing platform we consider is composed by a number of Microsoft Hyper-V Failover clusters which replicate each other through the Replica Broker Hyper-V functionality. Since each cluster has the same internal architecture, without loss of generality in the remaining of this section we will focus on a single Hyper-V cluster.

A. Physical architecture

A single cluster in the DaaS computing platform is composed by a number of physical servers, usually no more than ten in order to keep it simple to manage and to isolate potential threats or bottlenecks.

Figure 1 illustrates the components of the cluster. The core nodes are the N servers, which are connected to an *internal* and to an *external* network. The external network is used for internet connectivity and it is where provided services are exposed, while the internal network is used for inter-server communications and disk access. We make use of a fully redundant network topology in order to tollerate both single NIC server failures and a complete switch failure.

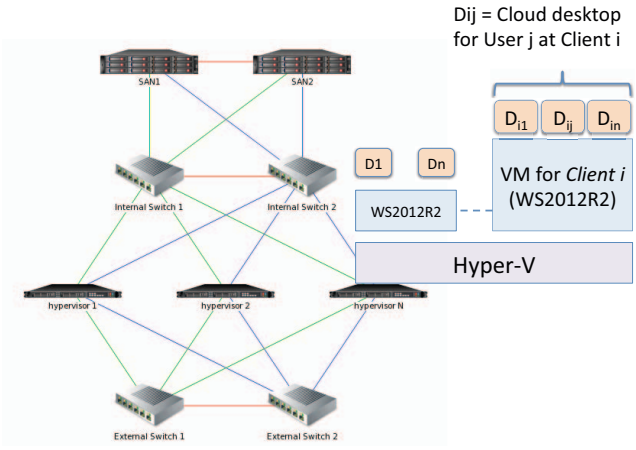


Figure 1. Architecture of a single failover cluster in the cloud-desktop computing platform

On the top of the physical architecture we have built a network virtualization layer to isolate different kinds of network traffic.

B. Storage, distributed filesystem and virtual machines

The compute infrastructure is based on Microsoft Windows Server 2012R2 and VMs run using Microsoft Hyper-V technology. Though the DaaS infrastructure is based on Microsoft Windows, it can host VMs that run any Hyper-V supported operating system. We currently run Microsoft Windows, Linux and FreeBSD VMs; however, most of the VMs we run provide the Virtual Desktop service, i.e. they are Microsoft Windows Servers that host multiple desktop sessions.

Each customer, or client hereafter, run a set of cloud desktops on one or more VMs. In the default setting each Client is associated to and isolated into a VMs (see Fig. 1). However, large clients (e.g. with more than 50 dektops) are partitioned on two or more VMs.

Another core component of a failover cluster is the cluster storage, implemented by two high-available Storage Area Network (SAN) configured with two RAID-6 arrays. The two SANs store the disks of the VMs that cannot reside on servers local storage, otherwise a server failure could affect the execution of all the VMs it stores. The SANs are also used to host the storage space dedicated to the cloud storage service (see Sec.IV).

C. Monitoring system

We monitor the entire system using the opensource product Zabbix [14], [15]. This software is composed by three modules: Server, Agent and Proxy. The first one is the daemon that collects monitoring data that are sent by the agents installed on the machines we want to monitor. Since we have a complex network architecture with many isolated subnets, we also use the Zabbix Proxy. This component has a twofold role: it acts as a server collecting data from clients, but it also behave as a client sending back all the collected data to the real server. Usually we deploy the Zabbix proxy on the virtual firewalls associated with every customer. Zabbix is able to monitor thousands of metrics for both physical and virtual machines, as well as network appliances supporting SNMP.

IV. Workload model

The global system load of the provider under study is produced by different applications: cloud desktops, cloud storage, e-mail, web servers, web applications, office management tools, helpdesk tools, and maintenance tasks. Considering that all the *bundles* of the DaaS service include the cloud desktop and cloud storage services, the two workload considered as the more rapresentatives are:

- the cloud desktop service workload (W_1), is the core service of the provider. The service consists of virtual

desktops running on the cloud infrastructure and hosting office automation applications
- The cloud storage service workload (W_2), is a classical cloud storage service allowing file sharing inside and outside the client private network. Storage service is accessible using WebDav clients or HTTP clients.

As previously introduced, the DaaS customers are referred as *clients*. Clients generate the system workload and each client C_i aggregates a set of user, $C_i = \{u_{i,1}, \dots u_{i,p_i}\}$, sharing the same objectives, consuming the same set of resources and generating the same workload(s). In our analysis we propose the following characterization of DaaS clients:

- The size S_i, that is the maximum number of cloud desktops that can be activated concurrently by client C_i and it coincides with the number of cloud desktops purchased.
- The workload set $\mathcal{W}_i$, that is the set of workloads generated by client C_i.
- The working time T_i, is the time interval Client i is active and it depends on the working location of the users and on their working habits. T_i is defined by the tuple $(T_i^{start}, \Delta T_i)$, where T_i^{start} is the hour of the day the working time starts and ΔT_i is the number of working hours per day. For example $T_i = (9am, 8)$ means that Client's users are active from 9am to 5pm.

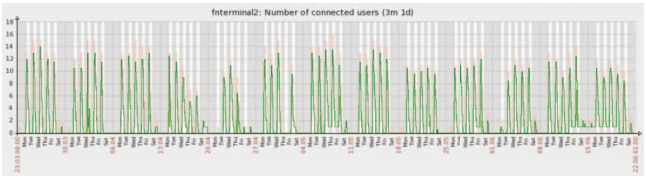
Considering the pay-per-use nature of the DaaS service, the features S_i and $\mathcal{W}_i$ can change over time. In this work we consider six clients anonymised as *Client 1* - *Client 6* characterized by: $\mathcal{W}_i = \{W_1, W_2\}$; $T_i = (9am, 9)$ i.e. client i is active from 9am to 6pm; and different sizes $\{S_1, \dots, S_6\} = \{16, 13, 4, 5, 10, 6\}$ (measured at Mar. 22, 2014).

A. Cloud desktop service session model

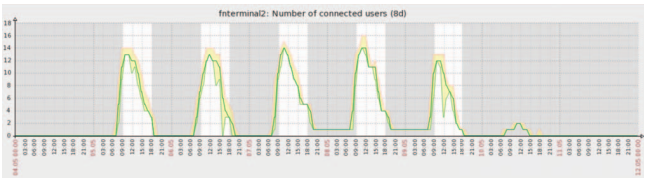
The access pattern of client C_1 is reported in Figure 2. We chose this client as example because the larger, but we observe that all the clients show the same behavior. Each Client generates a number of concurrent working sessions, or active sessions, bounded by S_i. As defined above, in our case study, the client activity is concentrated from 9am to 6pm, even if a marginal number of working sessions are observed during the night, at early morning and during the weekend. While early morning sessions bring with them real workload, the sessions left open during the nights and the weekends are inactive. Figure 2b shows the details, over one week, of the timeseries of the daily access profile measured in number of active sessions.

A working session is characterized by three phases (see Figure 3):

1) Start - in this phase a Windows desktop is started, and a working session begins;
2) Use - in this phase office automation applications are executed;



(a) Mar. 23, 2014 – June 22, 2014



(b) Daily access profile

Figure 2. Access pattern for Client 1

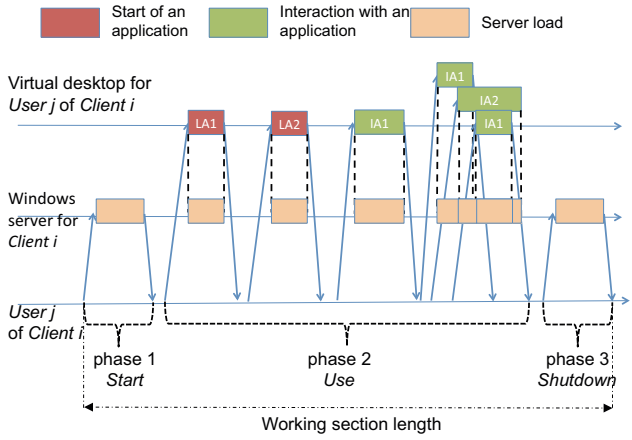


Figure 3. DaaS workload model: anatomy of a working session

3) Shutdown - in this phase a desktop session is closed.

Working sessions have a variable length depending on the duration of Phase 2 - Use. Start and Shutdown phases can be considered of constant length and their duration is assumed negligible respect to the length of Phase 2.

The average length $\bar{L}_i$ of a working session can be modeled as described in the following. Let us define:

- $a_{i,j}$ the j-th sample of the number of concurrent active sessions, e.g. the cloud desktops currently active, for client i-th.
- Δt, the sample period length (usually measured in seconds);
- N_{T_i}, the number of samples, of a_i, in the working time period T_i. $N_{T_i} = \Delta T_i \times \Delta t$.

The cumulative length of all the cloud desktop sessions in T_i is given by

$$\mathcal{L}_i = \sum_{j=1}^{N_{T_i}} a_{i,j} \times \Delta t$$

Therefore, the average session lenght $\bar{L}_i$ is given by

$$\bar{L}_i = \mathcal{L}_i / S_i$$

Since a client can decide to add or remove desktops as it needs, the size of the client changes over time and this behaviour has a non negligeable impact on $\bar{L}_i$. To consider this workload feature we introduce two different, and alternative, metrics for the client size:

- $S_{i,j}$ the size of client i measured at observation j; and
- $S_{i,max} = max_{j \in [1, N_{T_i}]}\{S_{i,j}\}$ the maximum value of the client size over the observation period.

Depending on the metric used we have two new expressions for the average session lenght:

$$\bar{L}_i = \sum_{j=1}^{N_{T_i}} \frac{a_{i,j} \times \Delta t}{S_{i,j}}$$

or

$$\bar{L}_{i,max} = \mathcal{L}_i / S_{i,max},$$

V. Workload Characterization

In this workload characterization study we are interested in the following metrics:

- Session length;
- CPU load;
- Disk read and write (r/w) rate;

The session length model and metrics have been introduced in the previous section. We remark that a careful characterization of the session length is core for the design of workload generators and to define system performance models.

The *CPU load* is measured as the average number of active jobs in the system in the last minute. CPU load is measured for each client i and is defined as: $CPULoad_i = \frac{Njobs_i}{\Delta t_{CPU}}$ where $Njobs_i$ is the numebr of jobs submitted by client i and Δt_{CPU} is the sampling period for the CPU Load. This metric is useful to understand whether or not there is a correct sizing of the number of virtual CPU ($vCPU$) assigned to a VM running cloud desktops. In the actual setting a VM is allocated to each client, therefore, if $CPULoad_i \leq vCPU_i$, where $vCPU_i$ is the number of $vCPU$s assigned to client C_i, we have an over-provisioned system because every time a process starts it finds a free processor. On the contrary, processes could get queued before execution and the client will experiment a performance degradation. The case of underprovisioning must be carefully studied analazying the cumulative distribution of $CPULoad_i$. An example is provided in Section V-B.

Finally, the *disk read/write (r/w) rate* is measured in Bytes/sec and represents the workload submitted to the SAN hosting the virtual disk of the VM. This metric, can be measured for each VM (i.e. each Client) or aggregated for all the VMs mapped on a storage system (e.g. SAN1). The *disk r/w rate* is important to plan both the input-output operations per second (iops) of the storage system that can be sustained and to characterise the rate of r/w requests toward the storage system.

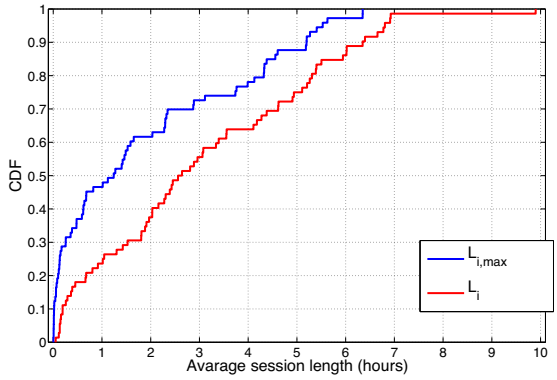


Figure 4. CDF of the average session length $\bar{L}_{i,max}$ and $\bar{L}_i$ for Client 1

A. Session length analysis

To show an example of the average session length distribution, and the impact of the session length model adopted, we report (see Fig. 4) the CDF of $\bar{L}_i$ and $\bar{L}_{i,max}$ for client C_1. The results obtained using $\bar{L}_i$ are more realistic than the results obtained with $\bar{L}_{i,max}$. In the first case we have working sessions longer than 5 hours in the 25% of the observations and 50% of the sessions are shorter than 2.5 hours. The $\bar{L}_{i,max}$ metric obviously sets a lower bound for the average session length. Indeed, considering this metric only the 12% of the sessions are longer than 5 hours and about 50% of the sessions are shorter than 1 hour.

From a capacity planning perspective is important to know the probability distribution of the session length. Using the Maximum Likelihood Estimator (MLE) method [1] we determine, here and hereafter, what type of distribution best fits the observations. (Table I reports the probability distribution functions used in the fitting).

Figure 5 shows the fitting of the empirical distribution of $\bar{L}_i$ with the Exponential and Weibull distributions. The fitting is almost the same but the MLE selects the Exponential with parameter $\mu = 3.076$.

B. CPU Load Analysis

First, we analyze the CPU load for *Clients 1 - 6*. Figure 6 shows the log-log plot of the frequency observed for the CPU load during working hours The client size S_i has direct impact on the intensity of the workload, that spans over three order of magnitude. Client 1 and Client 2 have a tail of the CPU load distribution ranging from 10 to 40 active jobs. The other clients (3-6) have a tail of the distribution ranging from 0.1 to 10 jobs.

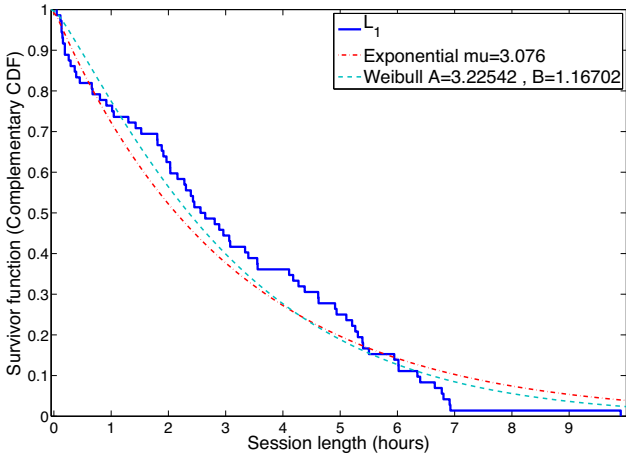


Figure 5. LLCD plot of session length $\bar{L}_i$

Figure 7 shows the LLCD plot of the empirical distribution for the observed data. The plot clearly shows that Client 4 - 6 have a CPU load less than 1 in more than the 99% of observations. $CPULoad_3 \leq 2$ in the 99% of the observations, and $CPULoad_3 \leq 1$ in the 90%. For Client 1 and 2 CPU load can reach, with a probability less than 10^{-4}, the value of 40 and 25 respectivelly. However for this two clients we observe that $CPULoad_1 \leq 4$ in the 95% of observations and that $CPULoad_2 \leq 2$ in the 99% of observations.

Considering that in the actual sizing each VM uses 4 $vCPU$s, a first result of the $CPULoad$ characterization is a resizing of the VMs obtaining an optimal setting. For example, using 4 vCPUs for client C_1 allows to keep the *CPU Load* below 1 job per vCPU in the 95% of the observations. For clients C_2 and C_3 2 vCPU are enough to obtain good performances, but for clients $C_4 - C_6$ 1 vCPU is adequate.

The workload characteristics pointed out in Fig. 7 is well known as the long tail nature of a probability distribution. This feature can be discovered with a simple visual inspection of the curves [1]. Hoverver, to be more precise and to provide a useful modeling tool to the research community, in what follows we assess the probability distributions that best model the $CPULoad$ for clients C_1 and C_2. As above demonstrated, although the $CPULoad$ for clients $C_3 - C_6$ shows a long tail nature, these clients are of small size and generate a negligible load.

Figure 8 shows the LLCD plot for the empirical distribution of the CPU load and for the candidate Weibull and Lognormal distributions. Both clients C_1 and C_2 $CPULoad$s fit best with a Lognormal distribution, as confirmed by the values of the Log-Likelihood estimator reported in Table II.

Table I
FITTING DISTRIBUTIONS PARAMETERS FOR CPU LOAD

Distribution	Pdf	Params
Exponential	$f(x) = \mu^{-1} e^{-x/\mu}$	μ
Lognormal	$f(x) = \frac{1}{x\sigma\sqrt{2\ pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$	μ, σ
Weibull	$f(x) = \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-(x/\beta)^{\alpha}}$	$\alpha > 0$, $\beta > 0$
Generalized Pareto	$f(x) = \frac{1}{\sigma}\left(1 + k\frac{x-\theta}{\sigma}\right)^{-1-\frac{1}{k}}$	$\theta < x$ if $k > 0$ or $\theta < x < \theta - \sigma/k$ when $k < 0$

C. Storage system read/write rate analysis

The storage system is the most important component of the cloud platform: it stores all the VM images, user profiles and user data. Considering a Cloud desktop working session, the storage system is accessed during the first phase to load the user profile needed to activate the desktop. During Phase 2 the storage system is continuously accessed by the applications and by the OS (for management tasks). Finally, in Phase 3 changes to the user session profile/configurations are written back along with the content of the memory/buffers. During the night hours and during the weekend the storage system is backed-up and other management operations are carried on.

From a detailed analysis of the clients behavior it emerges that, even if all the bundles includes both cloud desktop and cloud storage services, the users heavily use the local storage space (i.e., the storage associated to the cloud desktop) and not the cloud storage space (W_2), mainly used only to temporarily share documents between users. This means that the storage system workload is primarily generated by W_1 and only marginally by W_2.

As previously introduced, the analysis of the storage system workload aims at identifying:

- the statistical properties of the aggregated *disk read/write rate* measured directly at SAN1 and SAN2 storage systems; and
- the statistical properties of the *disk read/write rate*

Table II
FITTING DISTRIBUTIONS PARAMETERS FOR CPU LOAD

Client	Distribution	Params	Max. Log Likelihood estimator
Client 1	Weibull	$\alpha = 0.427$ $\beta = 0.585$	-12205.6
Client 1	Lognormal	$\mu = -1.703$ $\sigma = 1.657$	**-8122.42**
Client 2	Weibull	$\alpha = 0.145$ $\beta = 0.714$	23996.1
Client 2	Lognormal	$\mu = -2.594$ $\sigma = 1.258$	**28989.5**

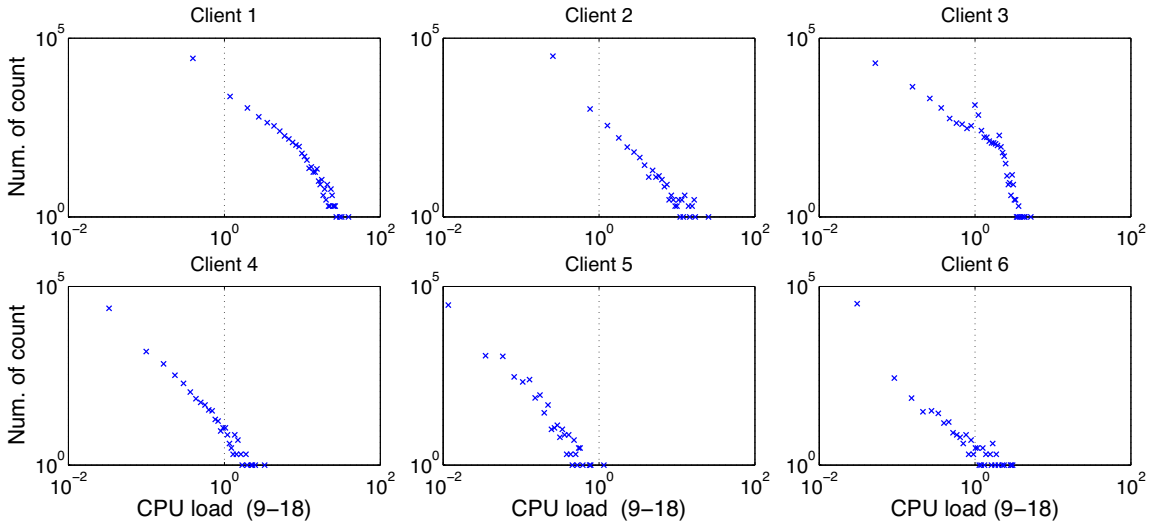


Figure 6. Log-Log plot of the Frequency of the *CPU Load* (1 min. avg) for working hours (9am - 6pm)

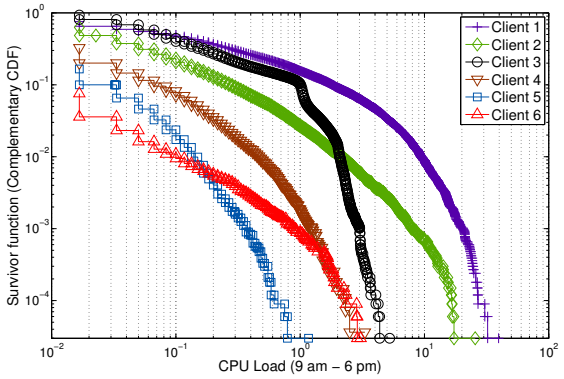


Figure 7. LLCD plot of the *CPU Load* for working hours (9 am - 6pm). The shape of the survivor function show that this workload attribute is long-tail distributed

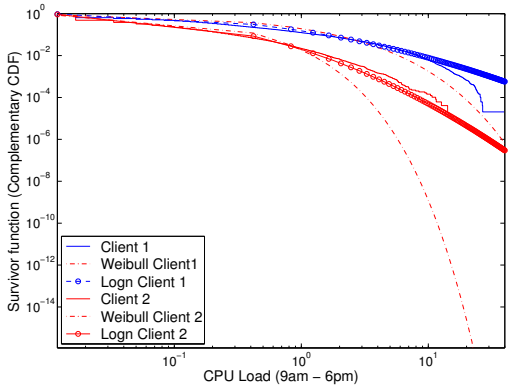


Figure 8. LLCD plot of the *CPU Load* for clients C_1 and C_2.

component of workload W_1.

The log-log plot of the frequency for SAN1 and SAN2 *disk read/write rate* is reported in Figure 9. This graph gives important hints: first, there are few differences in the rate over the 24h and during the working hours. That is, the backup and management operations marginally influence the scale of the distribution but not its shape. Second, the *disk read rate* is more intense than the *disk write rate* and confirms the 70/30 r/w ratio, typical of enterprise workload. Third, the workload is almost balanced between SAN1 and SAN2. Finally, the distribution of the *disk read and write rate* is long-tail.

Since in normal working conditions SAN1 and SAN2 workload are balanced, in what follows we will analyze the distribution fitting taking as reference SAN1. Figure 10 shows the LLCD plot of byte read and write versus three distributions: Lognormal, Weibull and Generalized Pareto (see Tab. I). The plot shows that the Lognormal distribution (selected as the best fitting based on the Log Likelihood estimator - see Table III) best fits the body of the distribution (for both read and write). For *disk read rate* the Weibull best fits the tail of the distribution, while for *disk write rate* the Pareto and Lognormal fit is almost the same for the tail of the distribution.

For what concern the load generated by client C_1, also in this case the read/write activities during the 24h and during the working hours are similar. For the single client too there is a difference in scale but not in term of shape of the distribution, as shown by the log-log plot of the frequency (see Figure 11).

The fitting of the empirical distribution confirms that also the single client workload attribute *byte red/write rate* is long-tail distributed. Indeed, Figure 12 shows that the log-

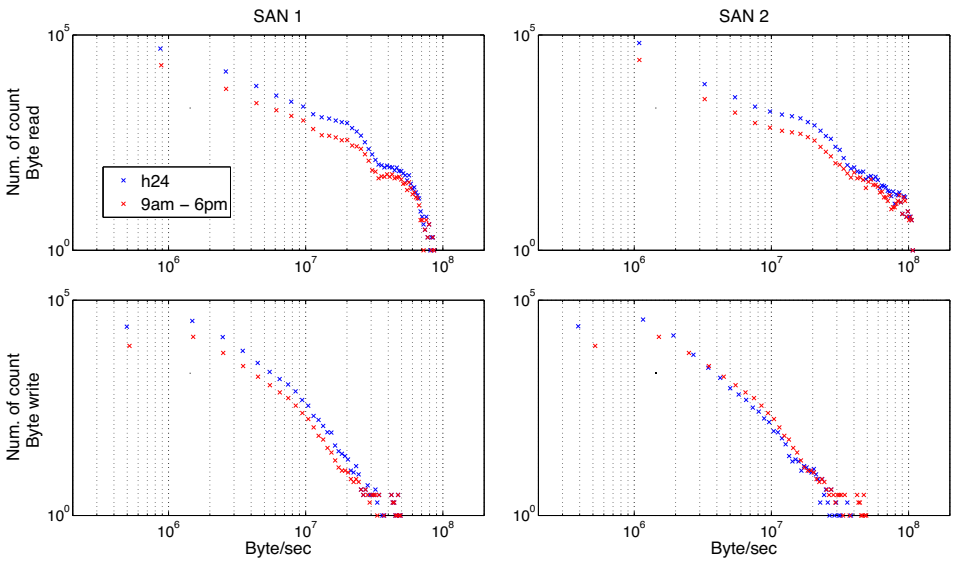


Figure 9. Log-log plot of the frequency of *disk read/write rate* for the storage systems SAN1 and SAN2.

Table III
FITTING DISTRIBUTIONS PARAMETERS FOR SAN1 DISK R/W RATE

Byte read per second

Distribution	Params	Max. Log Likelihood estimator
Weibull	$\alpha = 3.19587e+06$ $\beta = 0.679$	-1.42371e+06
Lognormal	$\mu = 14.209$ $\sigma = 1.533$	**-1.41819e+06**
Generalized Pareto	$\sigma = 1.68518e+06$ $\theta = 10$, $k = 0.741$	-1.42026e+06

Byte write per second

Distribution	Params	Max. Log Likelihood estimator
Weibull	$\alpha = 2.38074e+06$ $\beta = 1.236$	-1.3744e+06
Lognormal	$\mu = 14.3042$ $\sigma = 0.719$	**-1.35976e+06**
Generalized Pareto	$\sigma = 2.16243e+06$ $\theta = 1 + e4$, $k = 0.011$	-1.37769e+06

normal distribution is the candidate for optimal fitting. Also if the maximum likelihood estimator selects the lognormal as the optimal fit, a visual analysis of the plot puts in evidence that, while the lognormal best fits the body, the Weibull best fits the tail. Table IV shows the distribution parameters.

VI. CONCLUDING REMARKS

Summarizing, this study on cloud desktop workload allows to find out important features useful for capacity planning and for the design of heuristic algorithms for dynamic resource provisioning.

First, the session length, that is central to design workload generators and performance models, can be modelled with an exponential distribution.

Second, the CPU Load generated by the cloud desktop workload (W_1) is long-tail distributed, as well as the Disk load (r/w rate). This information drives the selection of the right queue model, e.g. the M/G/1, G/G/1 or GI/G/1. Moreover, the analysis of the CPU load enables us to empirically set the proper size of VMs for the running infrastructure and to instrument autonomic reconfiguration thresholds. It is important to remark that from the workload study emerges that peaks in *CPU Load* or *disk read/write rate* will exceed the average value by two (or more) order of

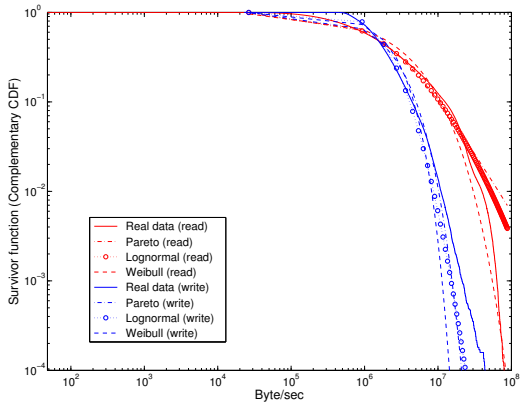


Figure 10. Fitting of SAN1 *disk r/w rate* versus Weibull, Pareto and Lognormal distributions (LLCD plot)

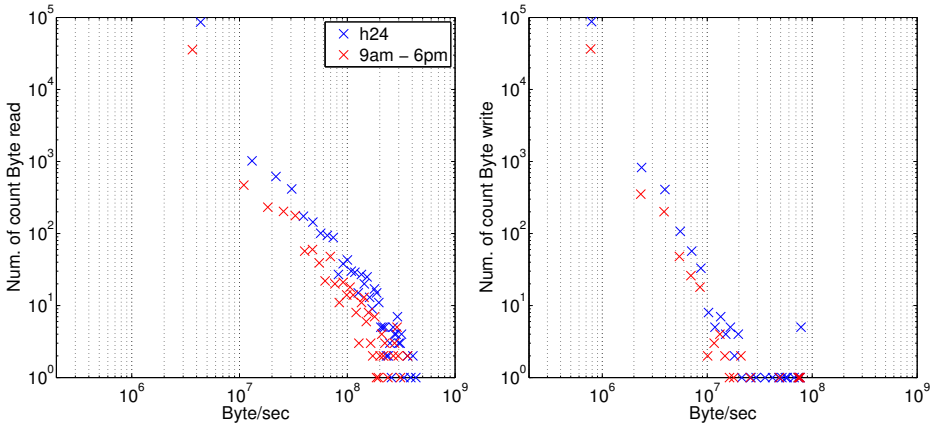


Figure 11. Log-log plot of the frequency for client C_1 read (left) and write (right) rate.

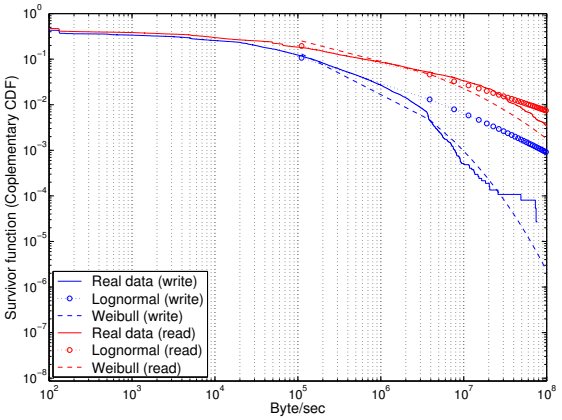


Figure 12. Fitting of client C_1 empirical distribution versus the Lognormal and Weibull distributions (LLCD plot)

Table IV
FITTING DISTRIBUTIONS PARAMETERS FOR CLIENT C_1 DISK R/W RATE

Byte read per second

Distribution	Params	Max. Log Likelihood estimator
Weibull	$\alpha = 26112.5$ $\beta = 0.223$	-310712
Lognormal	$\mu = 7.914$ $\sigma = 4.308$	**-307221**

Byte write per second

Distribution	Params	Max. Log Likelihood estimator
Weibull	$\alpha = 8168.61$ $\beta = 0.272$	-280671
Lognormal	$\mu = 7.117$ $\sigma = 3.621$	**-277578**

magnitude. Therefore forecasting mechanism and proactive provisioning techniques must be implemented.

Finally, the detailed analysis of the disk load allow to discover that workload W_2 is marginal and that the cloud storage service is not used by the users. This information activated a more detailed analysis to asses the end-user satisfaction level, the usability of the service and the customer requirements. Probably a re-definition of the bundles is needed.

ACKNOWLEDGMENT

The authors would like to thank the Grep/Rainbow Cloud Desktop Lab team for technical support in this research and the Grep s.r.l. for economic support.

REFERENCES

[1] D. G. Feitelson, *Workload Modeling for Computer Systems Performance Evaluation*. Cambridge University Press, 2015.

[2] D. A. Menasce and V. Almeida, *Capacity Planning for Web Services: Metrics, Models, and Methods*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[3] G. Aceto, A. Botta, W. de Donato, and A. Pescapè, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093 – 2115, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128613001084

[4] S. Di, D. Kondo, and W. Cirne, "Characterization and comparison of cloud versus grid workloads," in *Cluster Computing (CLUSTER), 2012 IEEE International Conference on*. IEEE, 2012, pp. 230–238.

[5] S. Pacheco-Sanchez, G. Casale, B. Scotney, S. McClean, G. Parr, and S. Dawson, "Markovian workload characterization for qos prediction in the cloud," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 147–154.

[6] A. Khan, X. Yan, S. Tao, and N. Anerousis, "Workload characterization and prediction in the cloud: A multiple time series approach," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, 2012, pp. 1287–1294.

[7] F. Azmandian, M. Moffie, J. G. Dy, J. A. Aslam, and D. R. Kaeli, "Workload characterization at the virtualization layer," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*. IEEE, 2011, pp. 63–72.

[8] Y. Xu, P. Calyam, D. Welling, S. Mohan, A. Berryman, and R. Ramnath, "Human-centric composite-quality modeling and assessment for virtual desktop clouds," *ZTE Communications*, vol. 11, no. 1, pp. 27–36, 2013.

[9] V. Talwar, K. Nahrstedt, and D. Milojicic, "Modeling remote desktop systems in utility environments with application to qos management," in *Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on*. IEEE, 2009, pp. 746–760.

[10] VMware, "Vdi server sizing and scaling - wmware infrastructure."

[11] Grep, "The grep/rainbow cloud computing solution."

[12] 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.

[13] 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.

[14] Zabbix, "The enterprise-class monitoring solution for everyone."

[15] A. D. Vacche and S. K. Lee, *Mastering Zabbix*. PACKT publishing, 2013.