Characterizing Private Clouds: A Large-Scale Empirical Analysis of Enterprise Clusters

Ignacio Cano *
University of Washington
icano@cs.washington.edu

Srinivas Aiyar
Nutanix Inc.
sriniva.aiyar@nutanix.com

Arvind Krishnamurthy
University of Washington
arvind@cs.washington.edu

Abstract

There is an increasing trend in the use of on-premise clusters within companies. Security, regulatory constraints, and enhanced service quality push organizations to work in these so called private cloud environments. On the other hand, the deployment of private enterprise clusters requires careful consideration of what will be necessary or may happen in the future, both in terms of compute demands and failures, as they lack the public cloud's flexibility to *immediately* provision new nodes in case of demand spikes or node failures.

In order to better understand the challenges and trade-offs of operating in private settings, we perform, to the best of our knowledge, the first extensive characterization of on-premise clusters. Specifically, we analyze data ranging from hardware failures to typical compute/storage requirements and workload profiles, from a large number of Nutanix clusters deployed at various companies.

We show that private cloud hardware failure rates are lower, and that load/demand needs are more predictable than in other settings. Finally, we demonstrate the value of the measurements by using them to provide an analytical model for computing durability in private clouds, as well as a machine learning-driven approach for characterizing private clouds' growth.

Categories and Subject Descriptors C.2.4 [*Computer Communication Networks*]: Distributed Systems; D.4.8 [*Software*]: Operating Systems—Performance

Keywords Private clouds, Measurements, Performance and Reliability.

* This work was done while the author was interning at Nutanix Inc.

SoCC '16, October 05-07, 2016, Santa Clara, CA, USA.
© 2016 ACM. ISBN 978-1-4503-4525-5/16/10...$15.00.
DOI: http://dx.doi.org/10.1145/2987550.2987584

1. Introduction

There is growing use of on-premise computing clusters in enterprises, commonly called "private clouds". Retaining control of their infrastructure, having greater security, greater proximity to other enterprise resources, and stronger SLA guarantees, among others, lead organizations to operate in internal cloud environments. On the other hand, private clouds require enterprises to provision sufficient compute resources to deal with worst-case demand spikes and hardware failures. Therefore, in their choice of private clouds, enterprises have to trade-off the greater control and security with the lack of flexibility to add or drop capacity in response to load changes/failures.

In order to understand this trade-off, we need measurement data regarding hardware reliability of enterprise-grade equipment, the resource demands in such settings, and other application characteristics of enterprise workloads. While there have been many measurement studies characterizing some of these issues in the context of desktop environments and public clouds [12, 15, 22], it is unclear whether these previous results carry over to private cloud settings due to a number of reasons. First, private cloud deployments use enterprise-grade hardware, which has greater reliability than the consumer-grade hardware used in desktops and many large-scale public clouds. Second, the private cloud workloads contain long-running enterprise applications that might have different characteristics than short-running jobs deployed on public clouds. Finally, private clouds represent the long tail of enterprise needs, which likely differ from the computational needs of much-studied large-scale enterprises such as Google or Facebook.

In this work, we perform an extensive measurement study of private cloud infrastructure, namely the installations of Nutanix[1] private clouds at various enterprises. In particular, we analyze data from approximately two thousand clusters. Our measurement study provides data on various aspects of private clouds ranging from hardware reliability to application characteristics. First, we provide measurement data on

[1] Nutanix is a provider of enterprise clusters. For more details refer to http://www.nutanix.com.

the failure rates of hardware components. In addition to focusing on enterprise-grade hardware, our work differs from other measurement studies in its analysis of the failure characteristics of all critical components in a cluster node. We also include measurements of virtualization-related failures, e.g., hypervisor bugs and misconfigurations. Furthermore, we provide measurements on the storage and compute requirements of enterprise clusters. We characterize the overall dataset and working set sizes that are managed by the clusters, and how these storage needs change over time. Similarly, we provide measurements regarding the CPU and I/O needs of the clusters and how they vary over time.

Among our main findings, we show that hard disk failures account for less than 20% of the total hardware failures, and that virtualization-related failures are almost as high as hardware ones. Further, we show that load/demand needs in private settings are predictable. While compute/storage elasticity is great to have, private clouds try to provision most resources upfront, and elasticity requirements are not as extreme as what is expected of public cloud workloads.

We then demonstrate the value of this measurement data by using it to quantify some of the trade-offs associated with the use of private clouds and how they should be provisioned. We first provide an analytical model of how the durability needs of the enterprise can be met given the failure characteristics of the hardware. We also provide a model for predicting when enterprises would opt to add more nodes to their clusters in response to the demands they observe.

1.1 Contributions

We present the first large-scale measurement study of enterprise private clouds that enables a better understanding of the challenges and trade-offs of operating in private settings. Summarizing, our main contributions are:

- We provide an extensive characterization of failures in on-premise clusters equipped with enterprise-grade hardware. We present measurements of all critical components in a node as well as virtualization-related problems (e.g., hypervisor bugs and misconfigurations).
- We characterize the storage, compute, and I/O needs in private clouds, and how these needs vary over time. Further, we analyze the profile of workloads that typically run in these clusters.
- Finally, we use the collected data in an analytical model for characterizing durability in private clouds. We also formulate a machine learning-driven approach for predicting cluster growth, using the measurement data to train our model.

The remainder of this paper presents these findings as follows: §2 discusses the background and related work, §3 describes the measurement methodology, and §4 gives an overview of the enterprise cluster profiles. We describe the failure analysis in §5, and workload characteristics (storage, compute, I/O) in §6. Further, we present our modeling strategy in §7. We describe the durability analysis in §7.1, and the cluster growth predictive model in §7.2. Finally, we conclude in §8.

2. Background and Related Work

The related work can be roughly categorized into two dimensions. One dimension is the setting: public clouds, enterprise settings, and consumer PCs. The other dimension is what is studied: e.g., hardware reliability and application workload characteristics such as storage and compute requirements. We focus our discussion along the second dimension and within each of its categories we discuss the setting in which the measurement work was performed. We note that most of the prior measurement work is on settings related to public clouds and consumer PCs, with very little measurement of private clouds.

2.1 Hardware Reliability

Prior work on understanding the characteristics of hardware infrastructure exists in the literature. Most previous work focuses on understanding the reliability aspect of individual components [2, 6, 18–20], whereas our study attempts to provide a comprehensive characterization of all components in a node and how their failure rates relate to each other.

Most hardware failures characterizations were done in the context of consumer PCs [12], networked systems [24], Internet services [13], high-performance computing systems [17], and some, in public clouds settings [14, 22]. Even in the latter, since public clouds typically employ consumer-grade equipment, there is not much measurement data regarding the reliability of enterprise-grade hardware. In our work, we include a hardware failure analysis of enterprise-grade equipment in the context of private clouds.

2.2 Storage Measurements

Although plenty of studies have been done in the storage measurement space, many of them target desktop settings. Agrawal et al. [1] presents a study of file system metadata from over 60K consumer-grade Windows machines in a large corporation. They study temporal changes in file size, age, directory size, namespace structure, storage capacity, and consumption. Another study [5] of consumer workstations analyzes the I/O behavior of home workloads, mainly productivity and multimedia applications running on Apple Desktop computers. In contrast to these studies, our work targets enterprise applications of several organizations in different customer segments, where workloads characteristics can vary significantly among clients. Similar to previous work, we analyze storage capacity and usage, and how they change over time. Further, we provide measurements for data transformations, namely compression and dedupe, and how they impact the overall storage capacity. We also contrast our findings to that of studies that measure deduplication savings in desktop settings [8].

There is some related work that pertains to storage measurements of enterprise workloads. The paper by Leung et al. [7] provides measurements of two enterprise-class file servers. In their work, they analyze how their workloads compare to previously studied traces. While they provide an in-depth look at access, usage and sharing patterns, our focus is to provide a higher-level characterization of workloads by hiding the details of the underlying file system implementation. Our work also aims to provide measurements on many private clouds deployments, with a diversity of application workloads, focusing on describing the working set sizes and the storage requirements needed by those enterprise clusters.

2.3 Compute Measurements

Google Cluster Data [16, 23] has enabled research on a broad set of topics, from workload characterizations [10] to new algorithms for machine assignment [15]. Their workloads mainly consist of short running jobs, although most resources are consumed by a few non-continuous tasks with long durations that have large demands for CPU and memory. There is not much data on storage needs in their traces. As opposed to their predominant short tasks, we mainly measure long-running enterprise applications in our study.

Mishra et al. [10] proposes a characterization of workloads based on Google's data. They group workloads of similar resource needs, which helps both for capacity planning and task scheduling. Similarly, in our work, we provide a fine-grained analysis of distinctive workloads and applications characteristics that typically run in private clouds.

Reiss et al. [15] analyzes Google's trace data to improve schedulers. They observe a high degree of heterogeneity in the resource types (cores, RAM per node) and their usage (duration, resources needed) patterns. Our work also examines the heterogeneity in application workloads, but does that in the context of continuously running enterprise workloads used in private clouds.

3. Measurement Methodology

Our measurement study uses data from various Nutanix cluster installations at different enterprises. In this section, we first describe the Nutanix cluster platform and how instrumented data from Nutanix clusters can provide a comprehensive view of private cloud characteristics. Then, we describe the different data sources we use in our study, ranging from cluster instrumentation to maintenance tickets. Further, we provide a set of caveats regarding the data collection process.

3.1 Nutanix Cluster Architecture

Nutanix is a well-known provider of enterprise cloud platforms. Their clusters provide virtualized execution of (legacy) applications, typically packaged as virtual machines (VMs), and virtualized and highly available storage that can be accessed through legacy filesystem interfaces. The cluster manager software performs various tasks to provide this abstraction. For example, it will automatically migrate VMs to cope with higher than expected resource loads, migrate data across nodes to both preserve locality as well as achieve good load balance between them, respond to both temporary and long-term failures by copying data to live nodes in order to meet a desired replication level, and so on.

Crucially, the cluster management and storage software has a comprehensive view regarding cluster state and thus can be instrumented to provide valuable measurement data. We can collect data regarding resource utilization on different nodes (e.g., CPU, memory, storage), the number of VMs running on a node, the I/O operations performed (since all data access is mediated by Nutanix's storage layer), and cluster health attributes. We use these types of data in our study, collected from different layers of the Nutanix cluster architecture.

Data Source	Sub-category	From
Metrics	Config	2014-11
	Storage & IOPS	2015-08
	CPU	2016-04
Customers	Accounts	2011-03
	Assets	2015-02
	Cases	2011-12
Repair/Maintenance	-	2013-08

Table 1: Data Sources Summary

3.2 Data sources

We derive our datasets from the following sources:

- *Metrics*: a database that contains different metrics uploaded by customer clusters. The periodicity varies from 60 seconds to per day. We extract performance related metrics (e.g., CPU usage, I/O load), cluster configuration, and storage usage data.
- *Customers*: a database that contains customer related information. We extract customer accounts, associated assets and their configurations, hardware dispatches, etc. We further obtain case information corresponding to various issues encountered in customer clusters. This data is created on demand, i.e., whenever a new hardware dispatch is performed, a new database entry is created, and similarly, whenever a new contract is signed, a new customer is added to the database.
- *Repair/Maintenance*: a service that provides information related to hardware component return rates, such as hard disk drives (HDDs), solid-state drives (SSDs), and memory, as well as product level metrics. This data is updated on a daily basis.

Configuration	Storage		Compute		Memory (GB)
	SSD (GB)	HDD (TB)	Cores	Clock Rate (GHz)	
Config-1	1600	8	24	2.5	384
Config-2	800	4	12	2.4	128
Config-3	800	30	16	2.4	256

Table 2: Node Configurations

3.3 Data Caveats

As most empirical studies, potential threats such as selection bias and representativeness of the data may risk the validity of our work. To that end, we make no attempt to cherry-pick any particular data point. Although we examine data from around 2K clusters, we have no means of verifying nor guaranteeing the generality of these results.

The data sources mentioned in §3.2 were originally meant for different purposes and were not necessarily aimed at tracking the particular quantities we are interested in. Further, their levels of granularity differ, e.g., some metrics are provided at a cluster level while others at a node or even at a disk level. We face the challenge of combining these disparate sources and aggregating them in different ways.

Moreover, we have access to different time frames of the data, and those time frames might even vary between the different sub-categories within the same data source. Table 1 provides a summary of the start times we have access to for each of the data sources and its sub-categories (if different).

Further, we extract information related to failures from the *Customers* data source, in particular *Cases*. This source consists of trouble tickets that are filed for hardware and software incidents. We trust the human operator on correctly annotating these fields, otherwise we may be underestimating the total number of failures as well as inaccurately characterizing their types.

4. Cluster Profiles

We focus on characterizing enterprise private clouds deployed across many customers around the globe. To that extent, in this section, we provide an overview of the general cluster profiles we study.

4.1 Clusters

We examine around 2000 clusters of different clients from a wide variety of industries, from retail to automobile, health care to financial services. Note that this represents only a fraction of Nutanix's overall cluster population that collect and report diagnostic measurement data[2]. We take a snapshot on April-2016 from a subset of the clusters that push information into Nutanix backends.

[2] For confidentiality reasons, we cannot disclose the overall numbers.

Summary Statistics	Value
# of Clusters	2,168
# of Nodes	13,394
# of Disks	70K-90K
Cluster Sizes	3-40

Table 3: Summary Statistics of Clusters in our study

Table 3 shows some summary statistics of our sample. We study deployments of sizes ranging from 3 to 40 nodes, with a total number of nodes of 13,394, spread across 2,168 clusters, which gives an average of 6.18 nodes per cluster, and a disk population of around 70K to 90K. This further highlights the uniqueness of our dataset, where we have many relatively small clusters, as opposed to prior work with few large ones [3, 9, 22]. One of the reasons this happens is that many of the clusters in our sample are deployed in remote office/branch office (RoBo) configuration. Also, we find that some customers create clusters within each line of business.

4.2 Configurations

Understanding the primary configurations of the different nodes in the clusters is important to characterize the types of workloads they can run. Among our sample, we identify three main node configurations, shown in Table 2. We observe that *Config-1* refers to compute-heavy nodes (24 cores at 2.5 GHz), whereas *Config-3* is more storage-heavy (around 30 TB of storage). *Config-2* is a baseline version.

As opposed to prior studies on public clouds, where one of the most notable characteristics of clusters is the heterogeneity of machines [15], in the case of private cloud clusters, we notice a much more homogeneous pattern in terms of node configurations within a cluster.

4.3 Virtual Machine Usage

In addition to the typical node configurations, the number of virtual machines per node (and their sizes) can also help in characterizing the type of applications that execute in each cluster. To that extent, we group VMs into three main buckets:

- *small*: VMs that have 1 vCPU assigned.
- *medium*: VMs that have between 2 and 4 vCPUs.

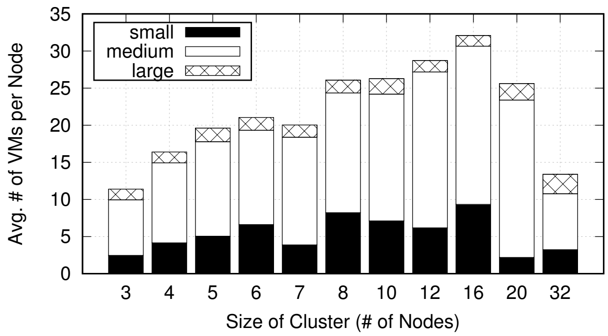


Figure 1: Distribution of VMs per Node

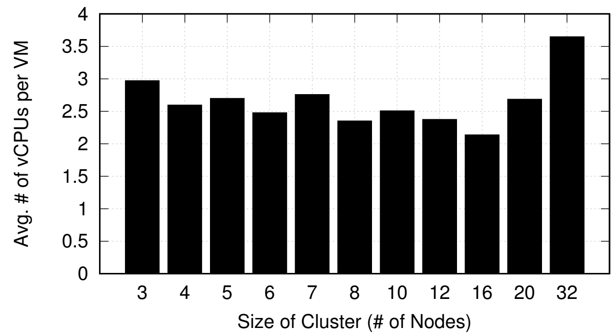


Figure 2: Average # of vCPUs per VM by cluster size

- *large*: VMs with 5 or more vCPUs. Within this category, we notice that most VMs lie in between 5 and 8 vCPUs.

Figure 1 shows the average number of VMs per node, together with their size distribution, among clusters of varying number of nodes. We observe a median of around 21 VMs per node. In particular, 3 node clusters have the lowest density of VMs, followed by 32 and 4 node clusters. On the other hand, 16 node clusters have on average 32 VMs per node. Further, we note that the vast majority of VMs in each of the clusters are *medium* size, which is also highlighted in Figure 2.

Figure 2 depicts the average number of vCPUs per VM for varying cluster sizes. We observe an overall average of $\sim$ 2.6 vCPUs/VM, with 32 node clusters having the maximum of 3.6 vCPUs/VM on average.

4.4 Workloads

We conduct a survey of Nutanix customers from our cluster sample, and identify the following four main workload categories, which typically run in more than 90% of the clusters:

- *VDI*: refers to Virtual Desktop Infrastructure, the practice of hosting a desktop operating system within a VM running on a centralized server. Nodes running VDI workloads typically have high compute and low storage usage. *Config-1* is the main node configuration we observe in this category.
- *SERVER*: refers to server-like workloads, such as database servers, web servers, etc. *Config-2* and *Config-3* are the dominant node configurations in this type of workloads. In this case, *Config-2* nodes are mainly used for light server virtualization ($\sim$ 5-10 VMs), whereas *Config-3* nodes focus more on denser server virtualization ($\sim$ 15-20 VMs), backups and file servers.
- *BIG DATA*: refers to workloads that analyze massive amounts of data, such as Splunk and Hadoop. *Config-3* nodes are the most typical in this scenario.
- *OTHERS*: encompass the rest of the workloads, e.g., IT infrastructure, custom applications, Sharepoint, Lync, etc. Here, there is no clear dominant configuration but is rather a mix.

Table 4 presents a summary of the common workloads seen among our cluster sample, together with few sample applications.

Workloads	Example Applications
VDI	Citrix XenDesktop, VMware Horizon/View
SERVER	SQL Server, Exchange
BIG DATA	Splunk, Hadoop
OTHERS	IT Infrastructure, Custom Apps

Table 4: Workloads and sample applications

As an interesting finding, we observe that most big enterprises have separate clusters for different applications or different classes of applications (i.e., workloads). Further, few of them create clusters within each line of business, so cluster boundary becomes the multi-tenancy boundary where they can track the growth and charge-back model for each sub division. On the other hand, small and medium-sized businesses are more on the consolidation-side, i.e., all applications deployed in one cluster.

5. Failure Analysis

We study private cloud clusters built from enterprise-grade commodity components, each of which may fail with some probability, potentially leading to unavailability, performance degradation or data loss. The possible damage caused by such failures provides a strong motivation for understanding their characteristics. In this section, we analyze failure data along various dimensions, e.g., component return rates, time to repair, etc. In particular, we provide details on two types of failures: hardware (§5.1) and virtualization (§5.2). Analyzing other types of failures, e.g., the transient failures that are automatically handled by Nutanix software, is left to future work.

5.1 Hardware Failures

Figure 3 shows the distribution of hardware failures in our private cloud environments. The data is based on *Cases* information (Table 1).

We observe that less than 20% of the total hardware failures are due to HDD problems, followed by a 16% of memory issues. SSDs and power supply units (PSUs) are the other two components that make it to the top four.

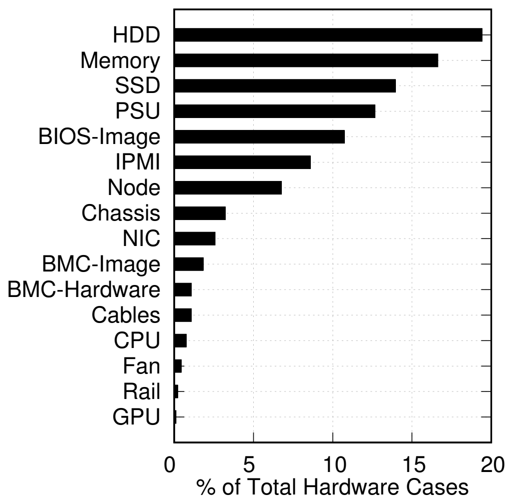


Figure 3: Distribution of Hardware Failures

Hardware component manufacturers typically specify the reliability of their products in terms of annualized failure rate (AFR), which gives the estimated probability that a component will fail during a full year of use. As pointed out by Schroeder et al. [18], the definition of a "faulty component" that a customer uses does not necessarily match the definition used by a manufacturer to make the reliability projections. As an example, they mention that a disk vendor reported they found no problems in 43% of the disks returned by customers.

Therefore, strictly speaking, in this work we report the annual return rate (ARR) seen from component replacements reported by customers.

Table 5 provides ARRs for the four hardware components that account for more than 60% of the total hardware cases that occur in private cloud settings. We observe that these values are lower than typical industry standards. For example, HDD replacements rates have been reported to be in the range of 2-9% [14, 18], whereas SSDs replacement rates range from 4 to 10% in a four year period [19].

There is one main reason that explains the goodness of these numbers. We focus our study in enterprise-grade hardware components, which is in the high-end portion of commodity hardware. Other studies report numbers based on consumer-level commodity hardware [14]. Regarding HDDs, our numbers might also be better off because they are not being used at the full capacity (§6.1).

Component	ARR (%)
HDD	0.7558
Memory	0.2075
SSD	0.7232
PSU	0.9120

Table 5: Annual Return Rates

5.2 Virtualization Failures

Besides hardware failures, we are also interested in analyzing software-related problems. Including Nutanix-specific software bugs would conflict our attempt to characterize private clouds in general. Therefore, in this section we only include virtualization-related issues.

We refer to virtualization failures as either software bugs or misconfigurations in hypervisors. In accordance to previous studies [4, 11, 25], operator/customer mistakes are also a common cause of system unavailability in private settings. Here are a few misconfiguration examples the operations team observed. Many customers attempted to implement resource pools in VMware (aggregated physical compute hardware – CPU and memory, as well as other components – allocated to virtual machines) but mistakenly configured some parameters, and experienced problems in their clusters. Also, some people tried to migrate from standard to distributed virtual switch in VMware, and they incorrectly handled the configuration.

Besides the configuration-related problems, virtualization failures also include hypervisor software bugs. One of the anecdotes the operations team recall was regarding a customer that experienced repeated disconnections between vCenter Server, a centralized cluster management software, and ESXi hypervisor running on a host because of the host daemon process crashing and restarting. Later on, they found out the crash was due to a software bug. Specifically, the host daemon process crashed when responding to *esxcli network vswitch dvs vmware list* command. The issue was not a direct cause of data loss or other severe issue, but could have triggered them. It was alleviated with a work-around until the actual fix was provided by the hypervisor vendor. Basically the operations team suggested disabling a cluster health mechanism that periodically executed that command.

Failure Type	Percentage
Hardware	59.32
Virtualization	40.68

Table 6: Hardware vs. Virtualization Cases

Having introduced virtualization-related problems, Table 6 shows a comparison between virtualization and hardware failures. We notice that from the total number of

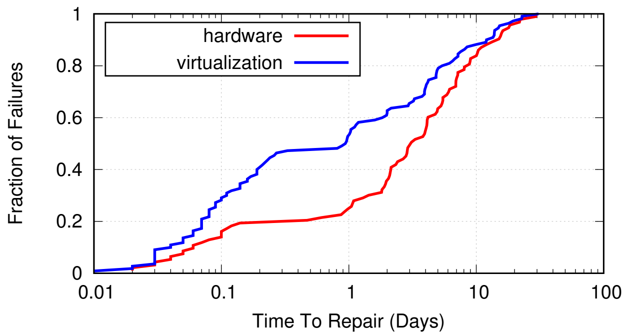


Figure 4: Time To Repair

hardware and virtualization cases reported, around 60% are due to hardware problems, whereas 40% are due to virtualization-related issues, either hypervisor bugs or misconfigurations. Surprisingly, virtualization failures are as high as hardware failures.

5.3 Time To Repair

Although most of the failures seen in our private clouds are self-healed, i.e., human operators do not intervene and the problems are resolved automatically, we only consider cases that do involve manual intervention. Many of the hardware-related cases involve shipments of some component, thus, as an extra caveat, the numbers provided here include operational latency (e.g., time to ship). Regarding virtualization failures, in case they are not customer misconfigurations, and no work-around is available, we depend on the vendor's bug resolution latency.

Figure 4 shows the cumulative distribution function (CDF) of virtualization and hardware cases time to repair (TTR). We observe that around 25% of hardware cases require at most 1 day to repair, whereas in the case of hypervisor failures, 50% of the cases are resolved in the same period of time. One of the reasons of this difference is related to the fact that many hypervisor cases are misconfigurations, i.e., remote troubleshooting by support teams can quickly solve the issue. Even more, in the case it is an actual software bug, it may be "solved" by a work-around and the ticket closed, as was the case of the bug mentioned in §5.2. Finally, we also notice from Figure 4 that 50% of hardware cases are closed in $\leq$ 3 days, as they usually involve actual shipments.

Summary: Our measurement data helps characterizing the failure rates of all components used in a private enterprise cluster. Hard-drives failures, memory, SSDs, and PSUs correspond to the bulk of the hardware failures. As expected, the failure rates of the enterprise-grade hard-drives and SSDs are significantly lower than that of the consumer-grade equipment characterized by previous studies [12]. Interestingly, virtualization failures, which include software bugs and configuration errors of the virtualization software, constitute a significant fraction of the observed failures.

6. Workload Characteristics

We have described so far private cloud clusters profiles, and their failure characteristics. In this section, we first focus on storage-related measurements, and then provide a characterization of the typical compute and I/O needs observed in our private clouds. Lastly, we present a fine-grained analysis of workload characteristics, using some of the most common applications we observe in enterprise clusters.

6.1 Storage Measurements

Measurements from Nutanix clusters provide data on enterprise workloads that are likely different from consumer workloads (e.g. desktops[1]) or transient jobs executed on public clouds [16, 23]. Further, since storage devices are accessed through a virtualization layer, the instrumentation of this layer provides valuable data regarding diverse aspects of the storage workload. We provide a characterization of storage requirements, working set size, and workload-specific data transformation savings (§6.1.1). Further, we look at predictability in storage usage among private cloud clusters (§6.1.2).

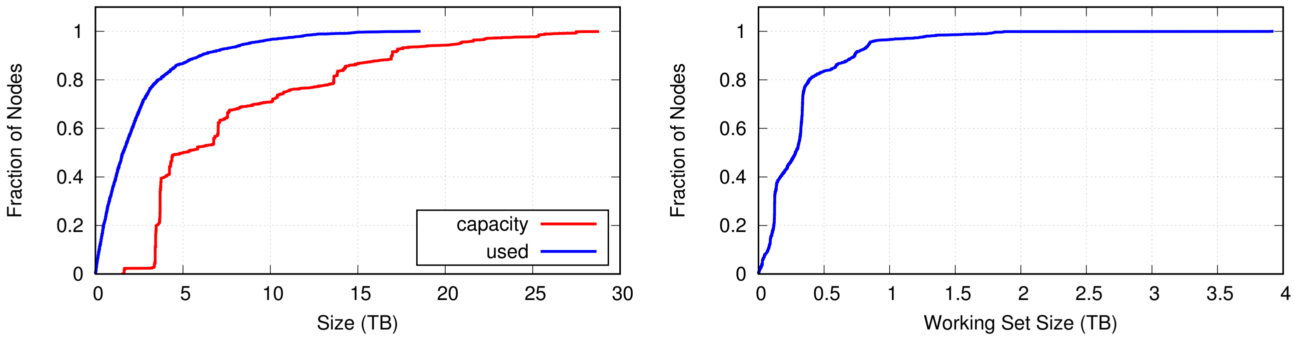
6.1.1 Storage Requirements and Data Savings

Figure 5a shows the CDF of the total storage capacity and used capacity per node in our cluster sample. We observe that around 50% of the nodes have 5 TB or less of storage capacity, and that 70% have less than or equal to 10 TB. Further, about 80% of the nodes use a storage capacity that is less than 4 TB, and about a third of the nodes use less than 1 TB.

The storage layer keeps track of the application workload's working set with the intention of caching elements of the working set in SSDs when possible (in order to lower the access latency). We collect data on the working set sizes across nodes in all the clusters, and we depict them in Figure 5b at a node-level. We observe that around 80% of the nodes have a working set size of 500 GB or less. This value indicates that a typical 800 GB SSD available on the cluster machines is sufficient to absorb the I/O generated by the workloads in most of the clusters.

In many of the clusters in our sample, the storage system is configured to perform various data transformations, such as compression and deduplication, in order to reduce storage overheads. We want to understand the impact of compression and dedupe on the storage usage in the context of enterprise applications. To that end, we analyze clusters with only compression enabled ($\sim$ 50% of the sample), only dedupe enabled ($\sim$ 25% of the sample), and the ones with both compression and dedupe enabled ($\sim$ 16% of the sample). The transformations are performed at the logical block-level of 1 MB to 4 MB chunks, as the Nutanix storage layer operates below the level of host file systems.

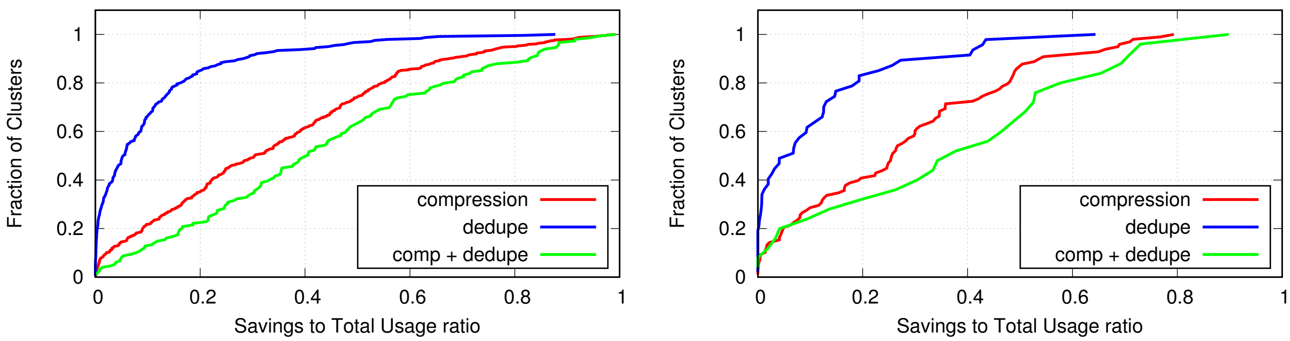
Figure 6 illustrates the savings to total storage usage ratio per workload for the different clusters. We plot only clusters



(a) Total Storage and Used Capacity Per Node

(b) Working Set Size per Node

Figure 5: Storage Requirements



(a) SERVER Relative Savings

(b) VDI Relative Savings

Figure 6: Relative Data Transformation Savings per Workload

with SERVER and VDI workloads, i.e. a fraction of the 50%, 25%, and 16% mentioned above. In both cases, we observe that compression gives more savings than dedupe.

Figure 6a indicates that 60% of the clusters with compression enabled running mainly SERVER workloads save up to 40% of the total used space. This means, for example, that if the total usage with compression is 10 TB, without compression it could have been 14 TB. Figure 6b shows the VDI counterpart.

We see that SERVER compression savings are higher than VDI savings. The 95^{th} percentile for SERVER is 0.8, whereas for VDI is 0.7. Regarding dedupe savings, we do not see a big difference between the workloads. The 95^{th} percentile is around 0.4 for both. It is worth noting that we observe dedupe savings of 20% or more in less than 20% of the clusters. This is significantly less than the dedupe savings obtained from a study that measured deduplication across a collection of desktop file systems [8].

Further, we observe that clusters with both compression and dedupe enabled provide more savings than having just one of the transformations activated, but the bulk of the savings comes from compression. For example, 50% of the clusters running SERVER or VDI workloads can save up to 40% of the total used space when both mechanisms are running, while compression alone can provide about 25-30% of savings in the median case.

Overall, the measurements of the dedupe savings and the total storage usage in clusters have the following implications. When application data is stored on disks, given their low overall utilization, the relatively low amount of savings obtained from deduplication might not be worth the overhead associated with performing the transformation. However, storage savings provide greater head room for recovery as a failed node's data has to be newly replicated on live ones. Furthermore, when frequently accessed application data is being cached in the SSD tier, the savings from compression and dedupe will allow more of the application-level data to be stored in the faster SSD layer, thus improving overall performance.

6.1.2 Predictability

We are also interested in characterizing how the storage usage evolves over time in order to understand whether the storage demands are somewhat predictable in enterprise settings.

Figure 7 illustrates the CDF of the mean, standard deviation, and 95^{th} percentile of the total usage for different clus-

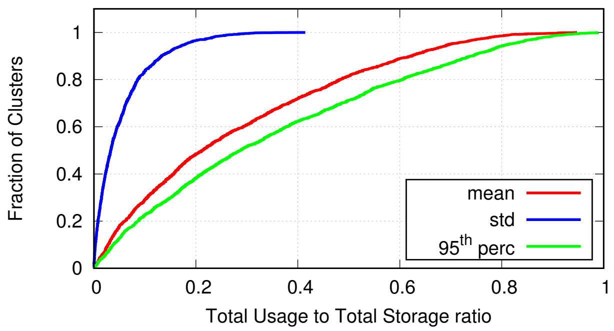


Figure 7: Total Usage to Total Storage ratio

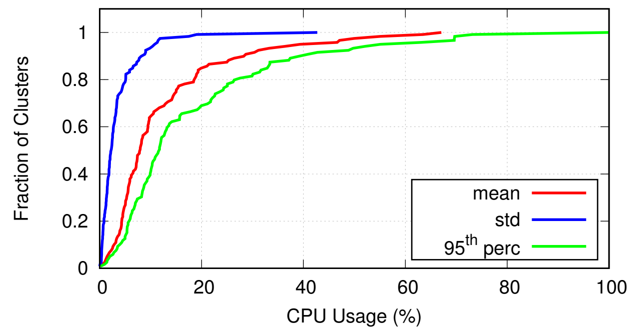


Figure 8: Cluster CPU usage

ters using eight months of measurement data. We observe that for 50% of the clusters, the mean storage usage is at most 20% of the total capacity, with a standard deviation of at most 3%. Similarly, 80% of the clusters have an average storage usage of at most 50% of the total storage capacity, with a standard deviation of $\leq$ 8%. In the case of the 95^{th} percentile, we observe that 80% of the clusters use 60% of the total capacity.

These results show that customers tend to fill the nodes in a non-chaotic manner, and the usage is quite predictable.

6.2 CPU and I/O Measurements

In this section we describe the compute and I/O demands we observe in our sample clusters.

We monitor CPU utilization across clusters and time. Figure 8 shows the CDF of the average, standard deviation, and 95^{th} percentile CPU utilization of clusters based on one month of measurement data with minute-level granularity[3]. The cluster CPU utilization is computed as the average of its nodes' CPU usage. We see a stable pattern, where around 80% of the clusters use 20% or less CPU on average, with a standard deviation of at most 5%. It is interesting to note that CPU utilization is moderately high only in some cases, e.g., 90% of clusters at the 95^{th} percentile have at most 40% CPU usage. Even at the 99^{th} percentile (not shown in the figure), the CPU usage is at most 60% for 90% of the clusters. Overall, similar to storage usage, CPU usage also seems to be stable over time, which indicates that the enterprise workloads run on these clusters have limited need for elasticity.

We also measure the number of I/O operations per second (IOPS) performed by the enterprise workloads, as it is another metric of application activity observed by the cluster software. Figure 9 illustrates the CDF of the mean, standard deviation, and 95^{th} percentile of the node IOPS over time. Around 60% of the nodes perform on average 1K or less I/O operations per second, with a standard deviation of less than 450. Although the standard deviation seems high compared to the mean, these numbers are still far from the maximum number of IOPS supported by the nodes, which simplifies performance capacity planning tasks in private settings. We further observe that 95% of the nodes perform at most 6K IOPS at the 95^{th} percentile.

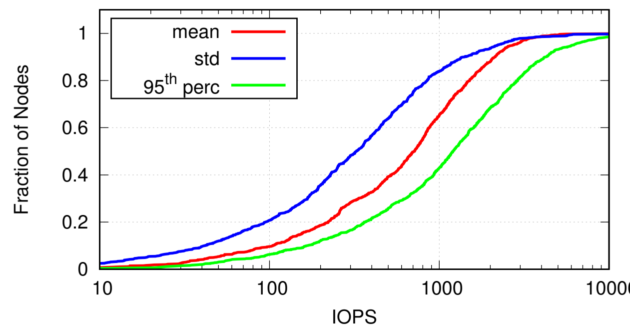


Figure 9: Node IOPS

6.3 Workload Analysis

We introduced the typical workloads that run in our sample clusters in §4.4. Here, we present a fine-grained analysis of some of their distinctive characteristics. The properties highlighted in this section serve as indicators for calculating typical IOPS requirements, vCPUs, size of VMs, etc.

Figure 10 shows the average number of VMs per node for different workloads by VM size (§4.3). We observe that VDI workloads (e.g., Citrix XenDesktop and VMware Horizon/View) have on average a higher number of VMs than the other workloads, around 26, as opposed to $\sim$ 19. Further, we see that Linux virtual servers run relatively more of the larger VMs than Windows does. The latter runs more of the smaller ones. In general, we see that typical SERVER workloads (e.g., MySQL, Web Servers) run more larger VMs than the VDI-like applications. *Medium* size VMs dominate in every application, which is further evidenced in Figure 11, where we observe an average number of vCPUs per VM of around 2.5. From §4.3, we know that *medium* VMs have 2-4 vCPUs.

Figure 11 also shows that VDI workloads have on average less number of vCPUs per VM, which follows from the fact

[3] We use a subsample of $\sim$ 100 clusters, as we could only retrieve fine-grained CPU usage from a smaller fraction of the overall sample.

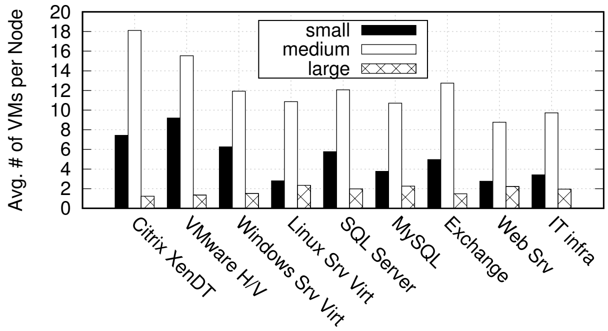


Figure 10: Average # of VMs per Node for the different workloads

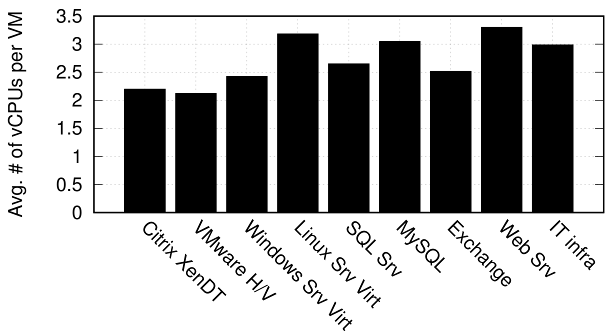


Figure 11: Average # of vCPUs per VM for the different workloads

that they typically have more VMs per node (Figure 10), thus, less vCPUs are left for each VM.

Figure 12 depicts the average number of IOPS per node for the different applications. In general, IOPS tends to be a big consideration when designing a VDI environment. It is interesting to see that VMware Horizon/View performs on average 3.5K I/O operations per second, whereas another VDI workload, Citrix XenDesktop, executes only 2.5K. We believe this happens because Citrix uses Intellicache as a caching proxy to reduce the overall I/O load on the backend.

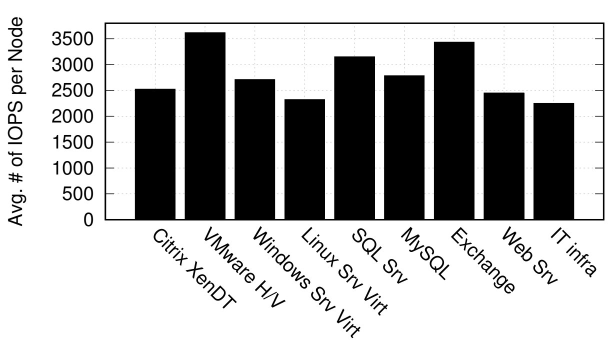


Figure 12: Average # of IOPS per Node for the different workloads

Summary: We describe storage, compute and I/O needs in private settings. Among our main findings, we note that working sets of most applications are small enough to fit into SSDs, which means caching/tiering is feasible. Further, we notice there is significant predictability in terms of CPU load, I/O demand, and storage requirements. Lastly, we show that application workloads have distinctive features, some of which are number of VMs and the vCPUs allocated to each one of them.

7. Modeling

In previous sections we provided a characterization of private clouds clusters along many different aspects, from failures rates to compute/storage requirements, from common workloads to typical cluster configurations, etc. In this section, we build upon those measurements to answer some of the questions that are pertinent to the use and management of enterprise clusters. We first provide an analytical model for durability based on failure rates and storage characteristics (§7.1). Then, we provide a predictive model that intends to provide a better understanding of the reasons behind cluster growth in private settings (§7.2).

7.1 Durability Analysis

We are interested in characterizing data storage durability. We build on measurements presented in earlier sections in order to estimate the probability of successive failures wherein additional nodes fail before the data on the original failed node can be re-replicated on other nodes in the system. This analysis can help in determining the level of replication necessary to meet a desired guarantee on data durability.

7.1.1 Analytical Model

In the analysis below, we assume that the replication factor used in cluster c is RF2[4]. Further, we assume that the replication is performed randomly; that is, with RF2, every piece of data stored on a node is replicated on one other *random* node in the cluster.

Let n denote the number of nodes in cluster c. Further, let d be the storage usage of any node in cluster c, and r the rate of data transfer in the network. Then, the time Δt required to create a new replica with data d when a node goes down in cluster c is given by:

$$\Delta t = \frac{d}{(n-1)\,r} \tag{1}$$

Further, let $p(\Delta t)$ be the probability of a node failure in Δt time. We decompose the overall period over which we want to provide the durability guarantee into a sequence of intervals each of length Δt. We define a data loss event Q in a given interval as the event where a node failure f_1 occurs, and a subsequent failure f_2 happens within Δt time of f_1,

[4] RF2 means the data is stored in two locations, RF3 in three locations, and so on. The analysis generalizes to other replication factor values.

i.e., the data could not be replicated[5]. Then, the probability that there is no data loss in a given interval has the following upper bound:

$$P(\neg Q,\Delta t) \leq (1-p(\Delta t))^n + np(\Delta t)(1-p(\Delta t))^{n-1}(1-p(\Delta t))^{n-1} \quad (2)$$

The first term in the summation indicates that there are no failures in the interval. The product in the second term has two components: the first refers to the probability that exactly one node fails in the given interval, whereas the second indicates that the remaining $n-1$ nodes do not fail within Δt of the first failure.

In order to characterize durability on a yearly-basis, we need to consider all Δt intervals within a year. Thus, the probability of no data loss over a one year period becomes:

$$P_{durability} = P(\neg Q,\Delta t)^{N(\Delta t)} \quad (3)$$

where $N(\Delta t)$ is the number of Δt intervals in a year.

7.1.2 Data Loss in Private Clouds

We now apply the model above to characterize data loss in private clouds. We use the following measurements observed in our clusters to perform the durability calculations: (a) failure rates, (b) I/O replication rate[6], and (c) the used storage capacity. Figure 13 shows the CDF of the data loss probability (in log-scale) of our sample clusters. We notice that the probability of data loss over a period of 1 year for 60% of the clusters is $\leq 10^{-6}$ for RF2, which gives a durability of six 9's or more. Instead, in the case of RF3, nine 9's durability or more is achieved by most of the clusters. As a rule of thumb, each additional replica provides an additional five 9's of durability given the failure model and the storage characteristics.

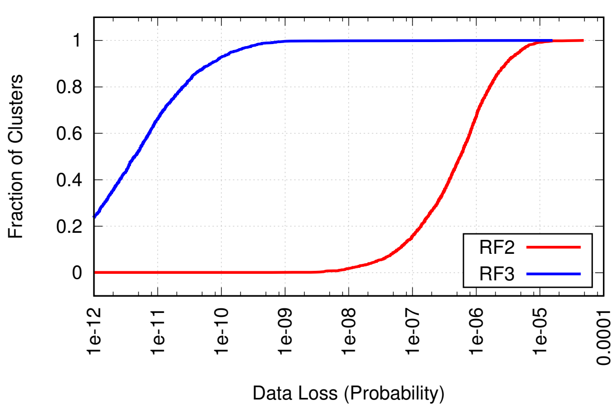


Figure 13: Data Loss in Log-Scale for different RFs

[5] Note that this formulation allows for the second failure to be in a following interval.

[6] We use r = 40Mbps in our calculations, as this is the typical transfer rate we observed in the network.

7.2 Cluster Growth Analysis

We find that customers periodically add nodes to their existing clusters. Given that cluster growth is part of the deliberate capacity planning activities of an enterprise, it would be interesting to understand what drives such growth. In doing so, we resort to the machine learning-driven approach explained below.

We frame this analysis as a binary classification problem, where the goal is to predict whether a cluster would increase its size or not. To that extent, we use a linear classifier, more specifically, we train a logistic regression model. As we are more concerned with the interpretability of the results (i.e., getting the most important predictive features) rather than achieving high accuracy, we induce sparsity by adding an l_1 regularizer [21].

7.2.1 Data

For this modeling exercise, we use a subsample of clusters C to build a dataset D of N examples. Each example in D consists of a tuple (c_i,t_i,x_i,y_i) where $c_i \in C$ denotes the cluster identifier, t_i the timestamp, $x_i \in \mathbb{R}^d$ the features, and $y_i \in \{0,1\}$ the label of example i. Note that every cluster c is associated with several examples. In particular, we consider around 200 clusters over a period of 8 months, which gives a total of 15K examples, split into training (70%), validation (10%), and testing (20%) samples. We only consider clusters that grew at least once during that period of time, which provides some positive instances (i.e., grow the cluster) to feed into our classifier.

7.2.2 Features and Methods

We construct three sets of basic features, based on general cluster size characteristics (F^c), based on storage-related information (F^s), and based on performance metrics (F^p). A summary of the features we use is presented in Table 7.

Cluster Features F^c	
$n(nodes)$	discretized # of nodes
$n(vms)$	# of vms per node
Storage Features, F^s	
$r(ssd)$	ssd usage to ssd capacity ratio
$r(hdd)$	hdd usage to hdd capacity ratio
$r(store)$	storage usage to total capacity ratio per node
Performance Features, F^p	
$n(vcpus)$	# of virtual cpus
$n(iops)$	# of iops per node

Table 7: Features for Predicting Cluster Growth

We consider two main categories of models, *Memory* and *Memory-less*. The former takes into account previous instances to help predict the current one, whereas the latter only uses current information. The order in *Memory* models is given by the number of steps we look back in the past[7].

7.2.3 Evaluation and Results

We evaluate our model using Area Under Curve (AUC), suitable for binary classification problems with unbalanced datasets. To measure the contribution of each feature set in the context of all the other features, we perform an ablation study, where we train the model using different feature combinations. We observe that F^c features perform quite well on their own, which seems to indicate that cluster size information helps in predicting cluster growth. Further, we note that storage-related features (F^s) seem to be better predictors than their performance counterparts (F^p). In general, *Memory* models perform better, especially 2^{nd} order models, i.e., when we look two steps back in the past. Our best model achieves an AUC of 0.6430, and combines F^c and F^s features, with 2^{nd} order information. Going beyond two previous timestamps does not seem to help.

7.2.4 Discussion

We are interested in understanding the rationale behind cluster growth. Given that we now have a classifier to predict such growth, we can inspect its internals (i.e., weights) to better explain its decisions.

Table 8 lists the top three most important features of our best model together with their weights. Although the exact weights are not important for the discussion that follows, we include them for completeness. We are only interested in the relative ordering of the feature weights, the higher the weight, the more important the feature. According to our classifier, the cluster size is one of the most important predictors of cluster growth. In this particular dataset, we observe that most of the upgrades are done from small-sized clusters, thus the high importance of this feature. Nevertheless, it is also an intuitive result. The chances that you increase the size of small-sized clusters is higher than if you already have big clusters.

HDD usage ratio is also a top feature. By manually inspecting the positive and negative instances in our dataset, we find that on average the HDD usage ratio is much lower in negative instances than in positive ones, which indicates that growth decisions are also driven by an increase in storage demands. This also matches with the boost in accuracy seen when we include storage-related features in our models (§7.2.3). Finally, we observe that the number of VMs per node from a previous example of the cluster also contributes to the growth decision. Similar to the HDD case, we manually inspect the dataset and find that on average the number of VMs in negative examples is lower than in positive ones. Therefore, according to our model, the third factor that drives cluster growth in private clouds is given by the need of more VMs. Interestingly, after controlling for the cluster and storage features, the performance features related to number of vCPUs or IOPS load do not add much predictive accuracy in modeling growth.

Feature	Weight
$n_t(nodes)$	0.317107669518
$r_t(hdd)$	0.176336623724
$n_{t-1}(vms)$	0.0591175856085

Table 8: Most Important Features for predicting Cluster Growth

[7] We use subscript t to denote the timestamp of each feature.

8. Conclusions

The adoption of on-premise enterprise clusters within companies is growing at a fast pace. In this work, we present the first large-scale measurement study of enterprise private clouds, namely Nutanix deployments at various organizations, which entails a full characterization of failures, and a comprehensive description of the storage, compute and I/O demands that are typically observed in these settings.

Among our main findings, we show that hardware failures are fewer than with consumer hardware, and notice a surprisingly high percentage of virtualization-related problems. We further describe the characteristic applications that run in these environments. Given that private clouds tend to provision most of the resources upfront and that the usage behavior observed is quite stable, we highlight that the elasticity requirements in these clusters are not as essential as in public clouds.

Finally, we build on these measurements and propose an analytical model of durability, and a predictive model for analyzing cluster growth. We believe that the observations arising from our work will enable a better understanding of the challenges and trade-offs associated with the use of private clouds, as well as provide useful guidance for their design and management.

Acknowledgments

We are grateful to our reviewers and colleagues for their help and comments on earlier versions of this paper. Special thanks to Timothy Isaacs from the Nutanix Product Management team, Parag Kulkarni, Manoj Thirutheri, Owen Richter, Stephen Parson, Stijn Vander Maelen, and Ranga Rathnam from the Serviceability team, and all the wonderful folks in Nutanix Engineering, Karan Gupta, Varun Arora, Rishi Bhardwaj, Manosiz Bhattacharya, Chern Cheah, and Rahul Singh for providing very insightful information. This work was supported by the National Science Foundation (CNS-1318396 and CNS-1420703). One of the authors was also supported by the Argentine Ministry of Science, Technology and Productive Innovation with the program BEC.AR.

References

[1] N. Agrawal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch. A five-year study of file-system metadata. *Trans. Storage*, 3(3), Oct. 2007.

[2] D. Ford, F. Labelle, F. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan. Availability in globally distributed storage systems. In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation*, 2010.

[3] P. Gill, N. Jain, and N. Nagappan. Understanding network failures in data centers: Measurement, analysis, and implications. In *Proceedings of the ACM SIGCOMM 2011 Conference*, SIGCOMM '11, pages 350–361, New York, NY, USA, 2011. ACM.

[4] J. Gray. Why do computers stop and what can be done about it? In *Symposium on Reliability in Distributed Software and Database Systems*, pages 3–12. IEEE Computer Society, 1986.

[5] T. Harter, C. Dragga, M. Vaughn, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau. A file is not a file: Understanding the I/O behavior of apple desktop applications. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 71–83, New York, NY, USA, 2011. ACM.

[6] W. Jiang, C. Hu, Y. Zhou, and A. Kanevsky. Are disks the dominant contributor for storage failures?: A comprehensive study of storage subsystem failure characteristics. *Trans. Storage*, 4(3):7:1–7:25, Nov. 2008.

[7] A. W. Leung, S. Pasupathy, G. Goodson, and E. L. Miller. Measurement and analysis of large-scale network file system workloads. In *USENIX 2008 Annual Technical Conference*, ATC'08, pages 213–226, Berkeley, CA, USA, 2008. USENIX Association.

[8] D. T. Meyer and W. J. Bolosky. A study of practical deduplication. *Trans. Storage*, 7(4), 2012.

[9] J. Meza, Q. Wu, S. Kumar, and O. Mutlu. A large-scale study of flash memory failures in the field. In *Proceedings of the 2015 ACM SIGMETRICS Conference*, SIGMETRICS '15, pages 177–190, New York, NY, USA, 2015. ACM.

[10] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das. Towards characterizing cloud backend workloads: Insights from google compute clusters. *SIGMETRICS Perform. Eval. Rev.*, 37(4):34–41, Mar. 2010.

[11] K. Nagaraja, F. Oliveira, R. Bianchini, R. P. Martin, and T. D. Nguyen. Understanding and dealing with operator mistakes in internet services. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 5–5, Berkeley, CA, USA, 2004. USENIX Association.

[12] E. B. Nightingale, J. R. Douceur, and V. Orgovan. Cycles, cells and platters: An empirical analysis of hardware failures on a million consumer pcs. In *Proceedings of the Sixth Conference on Computer Systems*, EuroSys '11, pages 343–356, New York, NY, USA, 2011. ACM.

[13] D. Oppenheimer, A. Ganapathi, and D. A. Patterson. Why do internet services fail, and what can be done about it? In *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, USITS'03, pages 1–1, Berkeley, CA, USA, 2003. USENIX Association.

[14] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies*, FAST '07, pages 2–2, Berkeley, CA, USA, 2007. USENIX Association.

[15] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch. Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In *Proceedings of the Third ACM Symposium on Cloud Computing*, SoCC '12, pages 7:1–7:13, New York, NY, USA, 2012. ACM.

[16] C. Reiss, J. Wilkes, and J. L. Hellerstein. Google cluster-usage traces: format + schema. Technical report, Google Inc., Mountain View, CA, USA, Nov. 2011.

[17] B. Schroeder and G. A. Gibson. A large-scale study of failures in high-performance computing systems. In *Proceedings of the International Conference on Dependable Systems and Networks*, DSN '06, pages 249–258, Washington, DC, USA, 2006. IEEE Computer Society.

[18] B. Schroeder and G. A. Gibson. Disk failures in the real world: What does an mttf of 1,000,000 hours mean to you? In *Proceedings of the 5th USENIX Conference on File and Storage Technologies*, FAST '07, Berkeley, CA, USA, 2007. USENIX Association.

[19] B. Schroeder, R. Lagisetty, and A. Merchant. Flash reliability in production: The expected and the unexpected. In *14th USENIX Conference on File and Storage Technologies (FAST 16)*, pages 67–80, Santa Clara, CA, 2016. USENIX Association.

[20] B. Schroeder, E. Pinheiro, and W.-D. Weber. DRAM errors in the wild: A large-scale field study. In *Proceedings of the 2009 ACM SIGMETRICS Conference*, SIGMETRICS '09, pages 193–204, New York, NY, USA, 2009. ACM.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[22] K. V. Vishwanath and N. Nagappan. Characterizing cloud computing hardware reliability. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, pages 193–204, New York, NY, USA, 2010. ACM.

[23] J. Wilkes. More Google cluster data. Google research blog, Nov. 2011.

[24] J. Xu, Z. Kalbarczyk, and R. K. Iyer. Networked windows nt system field failure data analysis. *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*, 0:178, 1999.

[25] Z. Yin, X. Ma, J. Zheng, Y. Zhou, L. N. Bairavasundaram, and S. Pasupathy. An empirical study on configuration errors in commercial and open source systems. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 159–172, New York, NY, USA, 2011. ACM.