Failure Analysis of Virtual and Physical Machines: Patterns, Causes and Characteristics

Robert Birke, Ioana Giurgiu, Lydia Y. Chen, Dorothea Wiesmann, Ton Engbersen,
IBM Research Zurich Lab, Rüschlikon, Switzerland, Email: {bir,igi,yic,dor,apj}@zurich.ibm.com

***Abstract*—In today's commercial datacenters, the computation density grows continuously as the number of hardware components and workloads in units of virtual machines increase. The service availability guaranteed by datacenters heavily depends on the reliability of the physical and virtual servers. In this study, we conduct an analysis on 10K virtual and physical machines hosted on five commercial datacenters over an observation period of one year. Our objective is to establish a sound understanding of the differences and similarities between failures of physical and virtual machines. We first capture their failure patterns, i.e., the failure rates, the distributions of times between failures and of repair times, as well as, the time and space dependency of failures. Moreover, we correlate failures with the resource capacity and run-time usage to identify the characteristics of failing servers. Finally, we discuss how virtual machine management actions, i.e., consolidation and on/off frequency, impact virtual machine failures.**

***Index Terms*—Data centers, VM failures, failure root causes**

I. Introduction

Today's commercial datacenters hosting various services are typically composed of a large number of physical and virtual systems. To ensure high availability of hosted services, datacenters face stringent requirements on all aspects of system reliability, i.e, hardware, software, network and power. However, failure is more the norm than an exception in largescale datacenters with millions of components [1]. All in all, it is no mean feat to keep the systems up all the time especially when the scale of datacenters continuously grows, an issue compounded by the increasing system complexity due to virtualization technology. Therefore, a deep understanding of the patterns and causes of failures on both physical and virtual systems can lead not only to efficient solutions, which increase datacenter reliability, but also to a better fulfillment of the service objectives.

There exist significant research efforts on understanding hardware reliability, e.g., disk, CPU, and RAM, for personal desktops/laptops [2], cloud datacenters [3], and particularly HPC systems [4], [5]. As commercial workloads are highly dynamic and systems are typically highly distributed, collecting and analyzing failure-related data is more challenging in commercial datacenters than in HPC systems. For example, the very first hurdle failure analysis on commercial datacenters has to take is the gathering of server failure logs by mining a large number of distributed ticketing and performance databases. Consequently, how servers fail in commercial datacenters is little known, except for the recent study of [3]. Given the wide deployment of services on virtual machines (VMs) in cloud datacenters, it is surprising that existing work sheds no light on VM reliability, such as VM failure rates, (in)dependency of VM failures, and their correlation with resources and management.

In this paper, we conduct a large-scale analysis comparing failures of physical machines (PMs) and virtual machines, by means of a field collection of problem tickets and resource performance measures from commercial datacenters, covering five subsystems with around 10K hosts. Our main focus is to characterize the failure patterns and identify the factors that cause failures for both PMs and VMs, while highlighting their similarities and differences. We correlate failure rates with the capacity and usage of multiple resources, i.e., CPU, memory, disk, and network, as well as with VM resource management, i.e., consolidation and turning on/off. In addition to studying the failure rates by machine types, we also classify failures by root causes, i.e., hardware, network, reboot, power, and software. We present statistics that are straightforward to observe, such that one can easily develop an intuitive understanding of failure behaviors and causes in production datacenters. Overall, our study is limited by a few factors, namely, the lack of physical location information of the systems, asymmetric distribution of VM and PM populations, and, most importantly, inconsistent clarity and granularity across different data sources, including different time granularity relative to resource usage. However, our analysis is based on a large collection of failure events of many PMs and VMs and considers a comprehensive set of failure types.

Our study is composed of three parts: (1) the overview of PM/VM failure patterns, (2) the dependency of PM/VM failure rates on the resource capacity and usage, and (3) the correlation of VM failures with VM resource management. The main building blocks of our analysis are failure rates, random failure probabilities and recurrent probabilities in multiple time windows, i.e., days, weeks and months. For the failure patterns, we present the failure rates, distribution of inter-failure times per server, distribution of repair times, and, most importantly, the time and space dependency of subsequent failures. To identify the server characteristics that cause frequent failures, we compare failure rates across different resource capacities and usages. We focus particularly on the impact of the number of CPU processors, memory size [GB], disk capacity [GB], number of disks, CPU utilization [%], memory utilization [%], and network traffic [MB/S]. The last part of our analysis focuses on how VM-specific resource management actions, i.e., consolidation and on/off frequency,

affect the VM failure rate.

The contributions of this paper are twofold. To the best of our knowledge, this is the first extensive analysis of VM failure analysis in commercial datacenters in comparison with PMs. Moreover, our study considers not only a diverse set of failures, but also an extensive set of factors that correlate with PMs and VMs failures, i.e., multiple types of resource capacity and usage, as well as, VM resource management.

The outline of this work is as follows. Section II presents related work. Section III provides an overview of our dataset. The study of failure patterns and their dependency on resource capacity and usage are presented in Sections IV, and V, respectively. The impact of VM resource management is discussed in Section VI, followed by the conclusions in Section VII.

II. Related Work

System reliability is one of the foremost concerns in datacenters, as service unavailability directly results in significant business revenue loss [6]. As a result, considerable efforts are being invested in trying to understand the behavior of hardware failures and their root causes, based on large collections of failure logs. The related work to our study can be divided into work focusing on (1) individual hardware subsystems, especially disk failures [7]–[9], (2) HPC systems [4], [10], (3) laptop or desktop machines [2], and (4) cloud datacenters [3]. Motivated by the relevance of field analyses of failure data, there is another set of studies trying to enhance the failure diagnosis using static or dynamic approaches [11]. In the following, we summarize the key findings observed from failure data in hardware subsystems, HPC systems, and commercial systems.

Hardware Subsystems

There are many studies that focused on characterizing the reliability of hardware subsystems, i.e., CPU, DRAM [12] and, particularly, disks [7]–[9]. A common finding is that disks have the highest failure rates, in comparison with other hardware components. The average annual disk failure rate [9] is observed to be $2-4\%$, which is much higher than the suggested one on product datasheets. In addition, studies show that disk failures increase linearly with age, without a significant infant mortality effect. Moreover, there is no significant correlation between disk failure rates and high temperature or utilization [8], but the parameters of the self-monitoring facility in drives show strong correlations with the failure probability. Moreover, disk failure is the predominant root cause for storage systems.

HPC Systems

As HPC systems grow exponentially by increasing the area density and component counts, so do the failure rates [4]. Failure rates in HPC systems show a positive correlation with the number of processors as well as the type and intensity of workloads running on them. Statistical models are often applied to capture the distribution of times between failures and repair times [4], [13]. A common finding in various studies is that inter-failure times for a node or an entire system are not exponentially distributed and can be well captured by Gamma and Weibull distributions. The root causes are typically classified into six categories, namely, hardware, software, network, environment, human and unknown, and analyses show a high correlation among them [5]. In particular, power-related failures (classified under environment) induce a high probability of follow-in failure of any kind. Liang et al. [10] explore the correlation between the recurrence and the location of failures through an on-line predictive model.

Commercial Systems

In contrast to HPC systems, there are only few studies that focus on commercial systems, ranging from laptop/desktop machines [2] to datacenter servers [3]. A common finding for HPC and commercial systems is that failures are not memoryless, meaning that the probability of follow-on failures is usually much higher, e.g., two orders of magnitude, than that of random failures for laptops and desktops. In particular, Nightingale et al. [2] investigate the failure trends of different subsystems in laptops and desktops, i.e., CPU, disk and memory, and their dependency with the usage of other subsystems, such as increasing CPU speed or memory size. Their main findings are that overclocking the CPU speed increases the failure rates for CPU, memory and disk, and that brand-name systems have better reliability.

In commercial servers [3], disks are the components replaced most frequently, and server failure rates increase with the number of disks. The predominant factors in predicting failures are datacenter location, followed by manufacturer brand name, as opposed to server age and configuration. The common observations are that the server manufacturer has an important impact on the failure rate of different hardware components. However, both reliability studies on commercial systems focus solely on hardware-related failures and overlook the software- and environment-related failures, which account for a significant number of failures in both commercial and HPC systems.

In contrast to related work, our study captures the failure rates of not only PMs but also VMs. We consider a comprehensive set of failure types and identify relevant factors that correlate with the failure rate, i.e., resource capacity and usage as well as server age and consolidation. We summarize the comparison at the scope of our analysis and that of previous work in Table I. Note that the related work listed in Table I only considers a subset of resources in their capacity and usage studies. In terms of results, our findings on PM failures mirror similar observations from earlier work, and our findings on VM failures and their comparison with PM failures complements many existing hardware-reliability studies.

III. Data Collection Methodology

Our reliability study is based on data collected on five commercial datacenter subsystems from July 2012 to June 2013. Each subsystem consists of stand-alone non-virtualized PMs and VMs hosted on virtualized boxes. Throughout the analysis

TABLE I
COMPARISON BETWEEN OUR STUDY AND RECENT RELATED WORK.

		[4]	[5]	[2]	[3]	Ours
	System	HPC	HPC	Laptops	DC Servers	DC VM/PM
Failures	Hardware	√	√	√	√	√
	Software	√	√			√
	Power	√	√			√
Factors	Capacity			√	√	√
	Usage	√	√	√		√
	Age	√		√	√	√
	Repair time	√			√	√

TABLE II
SUMMARY OF DATASET STATISTICS.

	Sys I	Sys II	Sys III	Sys IV	Sys V
PMs	463	2025	1114	717	810
VMs	1320	52	1971	313	636
All tickets	7079	27577	50157	8382	25940
% crash tickets	6.9%	0.85%	2%	1.3%	3.3%
% crash tickets (PMs)	69%	100%	59%	63%	57%
% crash tickets (VMs)	31%	0%	41%	37%	43%

in this paper, we focus only on statistics related to PMs and VMs, excluding statistics on boxes, because of the limited data access. These machines span a wide ranges of architectures (e.g., HP, IBM[1], Dell), run major operating systems (e.g., Linux[1], Windows[1] or VMware) and vary in their hardware age. Any incidents occurring on them are either reported by users or automatically generated by monitoring tools, such as HP OpenView [14] or IBM Tivoli[1] Monitoring [15], and collected through the ticketing system. Out of the tens of thousands of problem tickets gathered, we extract crash tickets which are associated with the underlying PMs and VMs being unresponsive or unreachable. We refer to such incidents as server failures.

Our dataset consists of 2759 crash tickets spread across 4292 VMs and 5129 PMs. The detailed statistics across the five subsystems, namely, the number of PMs and VMs, the number of all problem tickets and the number of crash tickets, are summarized in Table II. In the following, we first describe the data collection and sanitization process, then present the measurements of interest that contribute to our analysis, and finally discuss the limitations of our study.

A. Data Collection Process

We faced several challenges in gathering and sanitizing the measurements of interests that correlate with server failures, such as the resource capacity and usage levels. First of all, we collect the data from various sources which spread across multiple databases, such as the ticket and server resource monitoring databases. Each source has its own coverage of servers and accurate logging of the fields of interest in various time granularities, as they are designed for disparate business purposes. For example, on the one hand the server resource monitoring database uses two-year observation periods with data recorded every 15 min, hourly, daily, weekly and monthly. On the other hand, the ticket database mainly uses one-year observation periods, with data recorded by events. Because of the vast data volume, it is not trivial to simply track a large number of measurements of interest [3] in commercial datacenters, such as the capacity of servers (e.g., number of CPU units, memory and disk size, server age), their resource usages (e.g., CPU/memory/disk utilization, and volume of network transfers) and problem tickets capturing crash events.

The process of obtaining the dataset used in our analysis consists of several steps. First, we need to identify the crash tickets among all tickets collected. Every ticket contains a text description, explaining the problem and possible causes, the resolution used by the service support staff to alleviate the problem, and the repair duration. The quality of the descriptions and resolutions may not be always consistent, meaning that not all tickets have the same level of clarity and accuracy. As a result, we apply manual labeling and k-means clustering on both the description and the resolution field of all tickets in a best-effort manner. After manually checking the classification of all tickets, our k-mean classification has an accuracy of 87%. As second step, we classify the crash tickets into six finer-grained classes based on their resolutions:

- Network-related failure – server failures that are caused by network issues and require a network fix.
- Hardware-related crashes – server failures that are caused by hardware malfunctions (e.g., faulty disk or battery, broken power supply) and require a hardware replacement or fix.
- Software-related failure – server failures that are caused by OS or application-level issues (e.g., hanging OS or critical service agent) and require a software fix.
- Power-outage-related failure – server failures that are caused by power outages and require an electrical fix.
- Reboot-related failure – server failures that are caused by unexpected reboots.
- Other failure – server failures that cannot be classified into any of the above classes, owing to less accurate ticket descriptions and resolutions.

Such a classification of crashes is essential to gain a deep understanding of the patterns and characteristics of server failures. Fig. 1 shows the failure distribution across the network, software, hardware, reboot and power failure classes, excluding the unclassified (other) failure class. The failures classified as other account for 53% of all tickets, distributed as follows across the systems: Sys I has 35%, Sys II has 68%, Sys III has 68%, Sys IV has 61%, and Sys V has 29%.

When not considering other failures, the most common failures are due to software and reboot issues, which account for 31% of all tickets. As for subsystems, we note that for Sys I-IV software problems are the major reason for server failures and account for 12-22% of the crashes. Reboots remain the

[1]IBM and Tivoli are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Windows is a trademark of Microsoft Corporation in the United States, other countries, or both. Other product or service names may be trademarks or service marks of IBM or other companies.

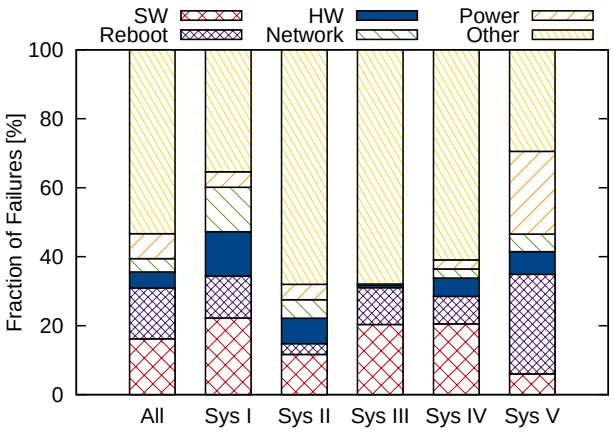


Fig. 1. Ticket distribution across the five types of failures: hardware, network, power, reboot, and software.

second most frequent cause of failures for all subsystems (8-29% of all), except for Sys II (only 3%). Hardware and network issues appear more frequently for Sys I and II, i.e., 26% and 13%, respectively, whereas for Sys II they account for less than 20 incidents throughout the observation period, i.e., 1%. On the one hand, power outages are generally not a major cause of failure, as they occur less frequently. For instance, Sys III experiences no power outages throughout the entire year, whereas for Sys I, II and IV the crashes due to power issues account for 4%, 4% and 3%, respectively. On the other hand, the servers corresponding to Sys V are affected by power outages in 29% of the cases, which leads us to conclude that they are not co-located in the same datacenter with any of the servers corresponding to the other 4 systems.

As third step, from all identified crash tickets we extract the server ids on which the crash events occurred. Finally, we collect the resource capacity and usage data for the extracted servers to map crash tickets to the server characteristics.

We note here that the ticket generation and resolution heavily involve human intervention, and thus will possibly include human induced errors. We try our best to present data that is sanitized by careful human intervention and interpretation of the raw data. In the remainder of this paper, we restrict the analysis to a smaller and consistent population, depending on the overlap of the measurements of interest across the aforementioned databases.

B. Measurements of Interest

As our aim is to identify the factors that determine the failures of PMs and VMs in commercial datacenters, the first measurement of interest is the *failure rate* of a single server and the subsystem in a granularity of day, week, and month over the one-year observation period. The failure rate is defined by the number of all failures divided by the number of servers. Additionally, we consider the *random failure* and the *recurrent failure* probability. The random failure probability is defined by the number of servers experiencing at least one failure divided by the total number of servers, whereas the recurrent failure probability is computed as the probability that given a server fails during the observation period, it will experience recurrent failures within 24 hours, a week, and a month. We further compute these failure probabilities per each subsystem, relative to PMs and VMs, as well as at the level of each failure class. We combine different categories and present subsets of results when deemed appropriate.

The second set of measurements of interest are related to server resource capacity, usage, age, and consolidation level. In particular for resources, we consider the CPU, memory, disk and network.

- VM age – We propose to consider a VM's first occurrence in the server resource monitoring database as its creation date. Given that records are kept over a two-year observation period, our approach suffers from the limitation that VMs can have been created prior to the earliest collection date. To increase the accuracy of our assumption, we filter out all VMs with creation dates that coincide with the initial observable data in the database. As a result, our analysis on the age impact focuses only on the VMs that have been created less than two years ago, which account for roughly 75% of the entire VM population. The VM age upon failure is defined by the time difference between the VM's creation date and the timestamp of the failure event.
- Resource capacity of PMs and VMs – For CPU, we ignore the architecture generation, but focus on the number of processors. We collect the memory size in terms of GB instead of the number of modules. For disks, we look into both the number of disks and the total storage volume. For lack of detailed information on the layout of each datacenter subsystems, our data lacks information on the network capacity of each datacenter, but contains the network demands expressed as volume of transfers in MB/s.
- Resource usage of PMs and VMs – In addition to the resource capacity, we study whether the workload intensity of each resource affects the server failure. We use the CPU utilization [%], memory utilization [%], disk utilization [%] and network bandwidth demands [MB/s], collected as weekly averages.
- VM consolidation level – Consolidation level refers to the number of VMs sitting on a hosting platform at a particular instance in time. We collect both the consolidation of VM failure instances and the average consolidation of VMs over the entire year.
- VM on/off frequency – Using the 15-min data of VM resource usages, we are able to track how frequently VMs are turned on and off in a two-month observation period, specifically March-April 2013.

C. Limitations

Our dataset does not contain information about the physical location of the servers, the hosting platforms for VMs, and the datacenters layouts. Thus, we are unable to provide a precise spatial dependency of server failures, especially across

different systems. Because of less accurate descriptions and resolutions for some of the tickets, there is an unbalanced distribution of tickets within the six crash categories, with those classified as "other" representing 53% of the dataset. Another weakness of this study is the bias selection, as we choose and analyze the datacenter subsystems that have the highest clarity and consistency in the description, and especially the resolution, of problem tickets. Finally, although the one-year observation period of our analysis is rather short, it provides reliable findings as our results match well with previous related studies.

IV. Overview on PM and VM Failure

The very first objective of this study is to answer simple questions, such as if VMs fail more often, take longer to repair, or exhibit different failure dependencies than PMs. To this end, we start by providing an overview on the server failure rates, distribution of times between failures, distribution of repair times, and failure dependencies over time, as well as over servers. In addition to the statistics computed over the entire population of servers and tickets, we also present the statistics relative to each system and the fine-grained failure classes.

A. Failure Rate

Our first step is to compare the frequency of VM and PM failures, using weekly and monthly failure rates over one year observation period. To compute the weekly failure rate of a certain system, we use the number of failures divided in a week by the number of servers belonged to that system. Fig. 2 summarizes the weekly failure rates of PMs and VMs, computed over the entire server population and subsystems. Each bar depicts the mean weekly failure rate and its 25^{th} and 75^{th} percentile. One can clearly see that PMs have higher failure rates for the entire observed population, as well as for most of subsystems, except Sys IV. Note that due to the low number of VMs associated with Sys II and the lack of crash tickets, no bar is drawn for its VM failure rate. When looking at the entire population (i.e., bars denoted by *All*), PMs have higher failure rates than VMs roughly by 40%, i.e., 0.005 vs. 0.003 . Since we only consider stand-alone PMs and exclude the hosting boxes of VMs, we rule out a certain dependency between VM and PM failures. In fact, if we would consider the failure of the hosting boxes as well, the failure rate of PMs would only further increase. Such a finding comes as a pleasant news that advocates VM deployment and the paradigm of cloud computing.

B. Inter-failure Times

Understanding the inter-failure times is crucial for reliability modeling and useful for the design of fault-tolerant systems. In this subsection, we are interested in understanding the inter-failure times from a single server's view, as well as from a datacenter operator's perspective. For a single server, we study only the time between failures that affect particular servers, i.e., PMs vs. VMs, while for operators we focus on failure that

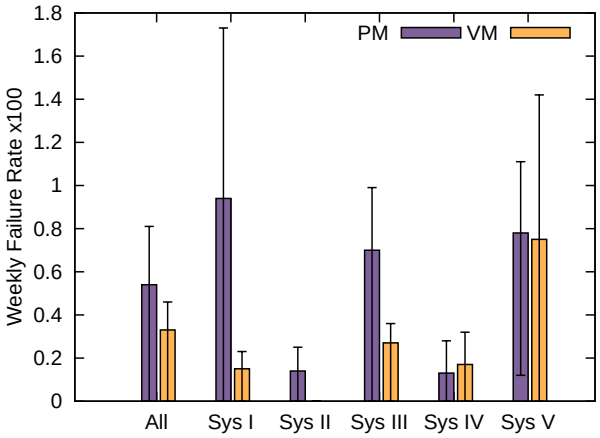


Fig. 2. Weekly failure rates for PMs and VMs over one year.

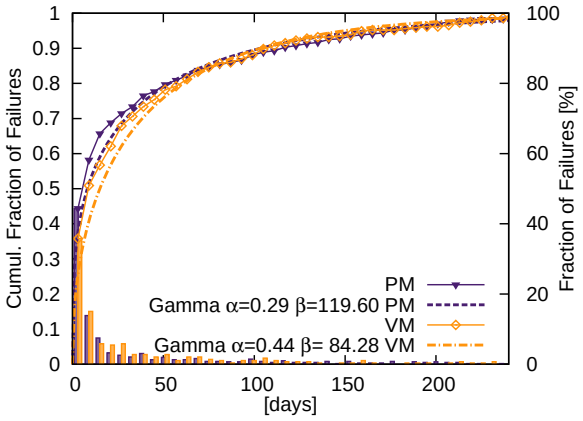


Fig. 3. CDF of inter-arrival times between failures of VMs and PMs.

affects any server in six failure classes. Note that we collect no inter-failure times for servers that only fail once.

We first present the distribution of inter-failure times of single VMs and PMs and plot their PDF/CDFs in Fig. 3. As an initial observation, we note that VMs and PMs have very similar distributions, as seen by the almost two overlapped lines. Zooming further, one can observe that roughly 80% of VMs have slightly higher inter-failure times than PMs, shown in the starting part of the CDF, i.e., roughly ranging between 0 and 100 days. This observation resonates well with our previous finding that VMs have lower failure rates than PMs. However, the tail part of the distribution corresponding to 20% of the VMs and PMs, i.e., beyond 100 days, shows that PMs have slightly longer inter-failure times. A possible explanation to such an inconsistent observation to the failure rate presented in the previous subsection is that roughly 60% of VMs have only single failure during the one year observation period, and thus no information about their inter-arrival times can be collected.

We also provide statistical fitting for both types of servers in Fig. 3. As shown by the long tails in the CDF, we choose a set of statistical distributions, i.e., Weibull, Gamma, and Log-normal, that are well known for describing the high variability

TABLE III
MEAN AND MEDIAN OF INTER-FAILURE TIMES IN DAYS, BY DIFFERENT ROOT CAUSES: OPERATOR'S VIEW V.S. SINGLE SERVER VIEW

Operator view: time between failures per class[days]						
	HW	Net	Power	Reboot	SW	Other
average	9.21	10.27	7.6	3.63	2.84	1.12
median	3.61	5.22	1.00	0.51	0.32	0.24
Signer server view: time between failures per server per class[days]						
	HW	Net	Power	Reboot	SW	Other
average	59.46	65.68	57.60	54.59	21.58	30.01
median	39.85	45.22	10.03	26.94	8.00	8.99

due to tails. Due to the lack of space, we only present the best fitting results, i.e., Gamma distribution. Similar to previous analysis for HPC systems [4], inter-failure times of PMs can be best captured by the Gamma distribution with parameters described in the figure. Most importantly, inter-failure times of VMs can be best fit by the Gamma distribution, as well, with the mean being 37.22 days.

To better understand how different root causes affect inter-failure times, we compute their respective mean and median across different resolution classes, as seen relative to particular failure classes and to a single server. We report the results in Table III (top and bottom, respectively). As expected, the inter-failure times of all classes seen by datacenter providers are much shorter than inter-failure time seen by servers. Both mean and median of software related failures are significantly lower than other classes by a factor of 2-3 times, indicating that software is less reliable from both the perspective of the datacenter provider and of the server. Failures caused by network have the highest inter-failure times, showing that datacenter providers and servers experience network failures roughly every 10 and 66 days, respectively. When looking at the failures caused by power outages, one can see that their inter-failure times are lower when compared to network and hardware crashes. This is explained by the fact that incidents classified as power failures do not only refer to unexpected power outages, but scheduled ones as well.

C. Repair Times

Another important failure behavior to characterize is the repair times. Let us take the time difference between the ticket issuing time and closing time as the repair time required to resolve failure events in tickets. We separate the repair times measured in hours between VMs and PMs and depict their distributions in Fig. 4. Note that the repair times represent actual down time, including the qeueing time, defined as the interval between the ticket generation and the start of repair. Usually, the queueing time in the case of server failure is very short, due to its urgency. As indicated by a lower CDF line, repair times of PMs are significantly higher compared to VMs, with mean repair times being 38.5 and 19.6 hours respectively. The reason behind is that a significant percentage of VM failures, i.e., roughly 35%, are caused by unexpected reboots, which take a short time to repair, as illustrated in the following paragraph. To facilitate the reliability modeling analysis, we fit the empirical distribution with several statistical distributions,

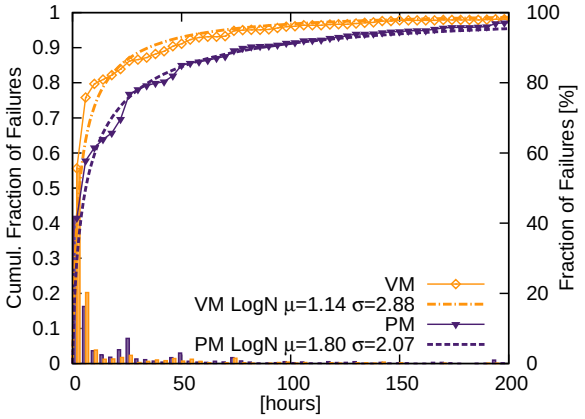


Fig. 4. CDF of repair time in hours relative to PMs and VMs.

TABLE IV
MEAN AND MEDIAN REPAIR TIMES IN HOURS FOR DIFFERENT FAILURE CLASSES.

	HW	Net	Power	Reboot	SW
mean	80.1	67.6	12.17	18.03	30.0
median	8.28	8.97	0.83	2.27	22.37

i.e., Gamma, Log-normal and Weibull. According to log likelihood of fitting, the repair times of PMs and VMs can be best described by the Log-normal distribution with specific parameters directly summarized in Fig. 4.

We further break down the repair time across the failure classes and summarize their mean and median values in Table IV. In all failure classes, the mean is much higher than the median, indicating a high variability in repair times. This is to be expected, since each of the five subsystems are serviced by different support groups. The shortest repair times are experienced for power related failures (0.83 hours for the 50% percentile), because in most cases: (1) their severity is critical, which means such incidents are immediately handled by the support teams, and (2) the resolution requires a simple electrical fix. As expected, reboot related failures take the second shortest repair times, as servers typically resume their services soon after the actual failure. Both hardware and network related failures require significantly longer repair times, as an extra delay may incur due to purchase of hardware/network components. Another observation worth noting is that software related failures have quite similar mean and median of repair times, indicating that their repair times have lower variation and thus coefficient of variation, compared to the other classes. This can be explained by the fact that failures caused by software tend to have a lower priority in the ticketing system than other failure classes and are serviced later in time.

D. Subsequent Failures

Motivated by the long tailed distribution of inter-failure times per server and previous hardware studies on non-independent failures [2], [5], [10], we study how subsequent PM/VM failures affect each other. To such an end, we use the recurrent failure probability within a day, week and month,

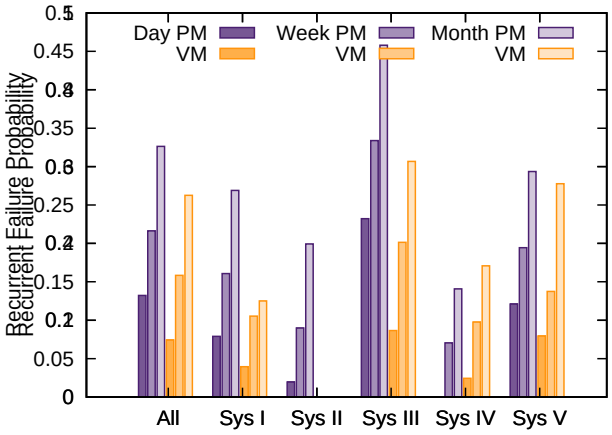


Fig. 5. Recurrent failure probabilities within a day, week, and month, for VMs and PMs.

TABLE V
COMPARISON BETWEEN WEEKLY RANDOM FAILURES AND RECURRENT FAILURES WITHIN A WEEK.

Physical Machines						
	All	Sys I	Sys II	Sys III	Sys IV	Sys V
Random	.0062	.015	.0020	.0090	.0028	0.0086
Recurrent	.22	.16	.09	.33	.07	.19
Ratio	35.5x	10.7x	45x	36.7x	25 X	10.5x
Virtual Machines						
	All	Sys I	Sys II	Sys III	Sys IV	Sys V
Random	.0038	.0023	0	.0030	.0032	.0094
Recurrent	.16	.11	0	.20	0.1	.14
Ratio	42.1x	47.8x	N.A.	66.7x	31.3x	16.7x

i.e., defined as given a server failure, what is the probability the server fails within a day, week, and month. Fig. 5 shows the recurrent failure probabilities for VMs, as well as PMs, computed for all three time windows. Overall, the recurrent probabilities of VMs are smaller than PMs. As expected, the recurrent probabilities grow with the time window, but not linearly with the window length. For example, the weekly recurrent probability is not 7 times higher than the daily recurrent one. This can indicate that subsequent failures tend to happen in a very close window, if at all.

To highlight the intensity of subsequent VMs' and PMs' failures, we propose to use the ratio between the recurrent failure probability within a week with the random weekly failure probability, which is defined as the probability that any server fails at least once within a week. Their values are summarized across all five subsystems and for each subsystem individually in Table V. One can straightforwardly see that the recurrent probability is higher for PMs, as well as VMs, roughly by a factor of 35X and 42X, respectively. In all subsystems, the intensity of recurrent failure ratios for VMs are visibly higher than for PMs. Our results show that both VM and PM failures show high intensities of recurrent failures, especially for VMs, though the absolute values of recurrent failure probabilities of PM are higher than VM.

E. Spatial (in)Dependency of Failures

To study how server failures affect each other at a given point of time, i.e., spatial dependency, we leverage the information of crash tickets by checking how many servers are affected by a single failure incident. The failure incidents can be, for instance, the power outage of a subset of servers or the crash of the underlying host platform which results in the corresponding VMs to fail at the same time. We first present the empirical distribution of how many servers are affected in a single failure incident, shown in Fig. VI. Roughly, 78% of all failure incidents involve only one server that could be a VM or a PM. The remaining 22% of all failures have a long-tailed distribution of the number of affected servers. The maximum number of servers involved in a failure incident is 34 and this is attributed to the undetermined (other) failure class.

To better see how such a spatial dependency propagates onto PMs and VMs, we formally propose a metrics describing the percentage of how many failure incidents involve dependent PM/VM failures. We define dependent PM/VM failures by considering failure incidents that include at least two PMs or VMs failures. Essentially, we compute this metric by the fraction of failure incidents affecting at least two VMs or PMs divided by the fraction of failure incidents affecting at least one VM or PM. Using the values listed in Table VII, we thus see that roughly 26% ($8\%/(30\% + 8\%)$) of the failure incidents involve dependent VM failures, whereas roughly 16% ($11\%/(57\%+11\%)$) failure incidents have dependent PM failures. This observation can be explained by the common practice of consolidating multiple VMs on a single hosting platform. As a result, we conclude that VMs show stronger spatial dependency than PMs.

To verify our conjecture on the degree of spatial dependency of server failures stemmed from root causes, we further breakdown the number of servers, across both PMs and VMs, by their classes. The average and maximum number of servers involved in failure incidents per different failure classes are summarized in Table VII. In terms of mean and maximum value, power failure indeed results into a higher number servers failing than other root causes. Judging from the absolute values related to power failures, we note that they occur at a local scale, rather than at the global datacenter level, because they affect only a small subset of servers. The spatial dependency of software failures is rather visible, and actually comes as the second highest after power failures. This is due to the fact that modern systems are composed of several distributed software services/modules, typically hosted on separate servers. A common example are 3-tier and enterprise applications. Although failures due to unexpected reboots involve a very low number of servers, they still have the second highest maximum value. We explain this by the fact that unexpected reboots are actually due to reboots of the underlying hosting platforms. Finally, we would like to note that the mean and maximum number of servers involved in unclassified failure incidents are 1.46 and 34 respectively. The overall values presented in Table VII are on the low

TABLE VI
PERCENTAGE OF FAILURE INCIDENTS INVOLVING ZERO, ONE AND EQUAL OR GREATER THAN TWO SERVERS, I.E., BOTH TYPES, PM ONLY, VM ONLY.

	0	1	≥ 2
PM and VM [%]	0	78	22
PM only [%]	62	30	8
VM only [%]	32	57	11

TABLE VII
MEAN AND MAX NUMBER OF SERVERS INVOLVED IN FAILURE INCIDENTS OF DIFFERENT CLASSES.

	HW	Net	Power	Reboot	SW
mean	1.2	1.5	2.7	1.1	1.7
max	10	9	21	15	10

side, because of the limitations explained in Section III. In particular, critical large scale failures can lead to the failure of the monitoring server, and thus leads to the missing generation of crash tickets. Out of 2300 tickets observed, 48 tickets report monitoring system failures.

F. Age Matters

The bathtub curve of age versus failures is well known for hardware component [16]. We try to find if such a curve also holds for virtual machines, i.e., old and young VMs fail more often than middle-age VMs. Due to the limitation of the data available to us, we are only able to trace VMs age dated back to January 2011. This accounts for roughly 75% of VMs under observation. In Fig. 6, we present the CDF and PDF of the number of failures with respect to different ages, that is defined as the time difference between the failure moment and the VM creation date. One can see that the CDF curve is very close to the diagonal line, indicating a close relationship with the uniform distribution. As for PDF, it shows a weak increasing trend with quite some fluctuation. The fluctuation of failure counts can be due to the fact that VMs are created in a batch manner and the underlying population of each age group is not evenly distributed. Consequently, we conclude that the relationship between VM failures and their age does not follow bathtub like function. Moreover, VM failures show a weak positive trend with age. This may lead us to suggest that periodically taking snapshots of existing VM images and creating new VM instances can reduce VM failures.

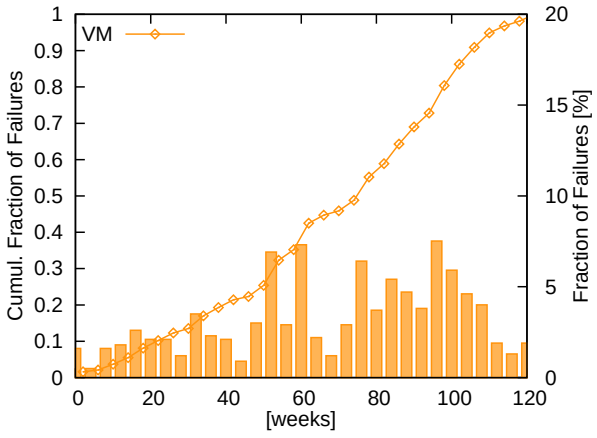


Fig. 6. The failure rate with respect to VM age.

V. IMPACT OF RESOURCES ON SERVER FAILURE

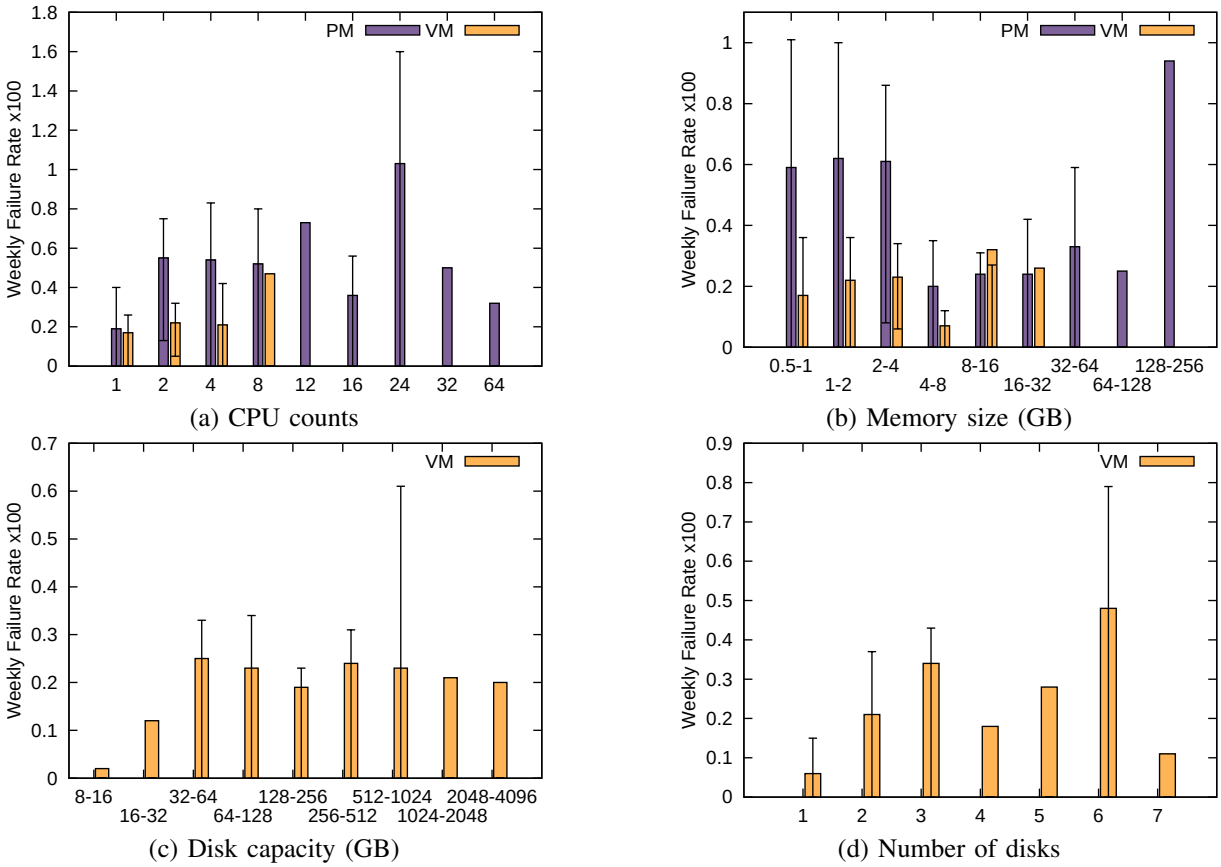
In this section, we study the impact of the server resources on servers' failure rate, in terms of resource capacity and usage. The objective is to find out if bigger and heavily loaded servers tend to fail more often. To this end, we start by considering capacity of multiple resources, i.e., the number of CPU processors, memory and disk size, as well as the number of attached disks, for both PMs and VMs. Second, we analyze the impact of the CPU, memory, disk and network utilization, collected from weekly averages over one year observation period. Our objective is to quantitatively differentiate the impact of resource capacity and utilization on VM failure rates from PM failure rates. We note that the following analysis is based on the weekly failure rate of servers with certain resource attributes, i.e., number of failure events divided by the number of servers, relative to the same range of resource attributes. Following the convention in Fig. 2, we present the average values as well as 25^{th} and 75^{th} percentiles with respective to different ranges of resource capacity and utilization attributes.

A. Resource Capacity

1) CPU Counts: To understand the impact of the number of CPU units a both PM and VM failures, we compute the weekly failure rate relative to the number of CPU ranging from 1 to 64 in Fig. 7(a). For PMs, we note that across different numbers of processors, the average failure rate increases from around 0.002 to 0.011 as the CPU count increases to 24 cores, accounting for a factor of $5.5X$. But, it decreases to below 0.005 for 32 and 64 CPUs, probably due to the higher reliability of such high-performance systems. Another worth noting observation is that the range between 25^{th} and 75^{th} percentiles increases with the number of CPU, due to the uneven distribution of the number of servers, i.e., 72% of servers have at most 4 processors. As for VMs, its average failure rate increases from 0.002 to 0.005 as the number of vCPUs goes from 1 to 8, showing an increment factor of $2.5X$. of In particular, the most of crash incidents occur on VMs with most 2 logical CPUs, which is the most popular configuration. Overall, the number of CPU processors has a positive impact on the failure rate for both PMs and VMs.

Comparing the failure rates of PMs and VMs, one can see that the number of CPU units has a more significant impact on PMs than on VMs, shown by a higher average failure rate. This can be possibly explained that while in the case of PMs a percentage of the failures can be caused by the actual processor failing, VMs do not have any access to the hardware and the impact of CPU counts translates only in shared CPU time.

2) Memory Size: Looking at the memory capacity shown in Fig. 7(b), we similar trends for both PMs and VM, i.e., failure rates show the trend of high - low - high with increasing



(a) CPU counts (b) Memory size (GB)

(c) Disk capacity (GB) (d) Number of disks

Fig. 7. Weekly failure rate across PMs and VMs relative to the CPU counts, memory size, disk capacity and number of disks.

memory size – a kind of bathtub shape. For PMs, the average failure rates are roughly 0.006 for the memory capacity up to 4 GB, stabilizes at around 0.002 for memory capacities ranging between 4 and 32 GB memory, and finally increases up to 0.01 as the memory size reaches 128 GB. The impact of memory size to PM failure rates is up to a factor of $5X$. In the case of VMs, the average failure rates are relatively flat at 0.002 when the memory capacity is between 256 MB and 4 GB, suddenly drop below 0.001 for the memory size between 4 GB and 8 GB, and increase to around 0.003 for VMs with up to 32 GB capacity. Comparing the lowest and highest average failure rates across different memory size, one can see an impact of $3X$ from the memory size. We note that most crash incidents happen on the majority of VMs equipped with 1-2 GB memory. We provide the following rationales to explain the higher average failure rates corresponding to (1) low memory capacities (256 Mb to 4 GB) are mostly due to software crashes, because of applications and services running out of memory, and (2) high memory capacities (256 GB for PMs and 32 GB for VMs) are mostly caused by hardware faults, since a higher number of memory modules increases the probability of faulty RAM. Overall, The decreasing and increasing trends of average failure rates are observed more dominantly for VMs as for PMs.

3) Disk Capacity: Next, we consider the impact of the disk capacity and the number of disks on weekly failure rate, illustrated in Fig. 7 (c) and (d). As opposed to the CPU and memory capacity, the data available to us does not contain any disk information for PMs, and thus we can only present results relative to VMs.

On the one hand, one can see the average failure rate increases steadily with the disk capacity from 0.00029 for 8 GB disks to 0.0025 for all bigger sizes, i.e., greater than 32 GB. Actually, such an increasing trend account for 15% of VMs, whereas the rest of VMs have disk capacity greater than 32 GB. Such an increasing trend of failure trend can be possibly attributed to using the constant size of disk physical platters to store different virtual data volume. The fact that currently large capacity disks use about the same sized platters as the small capacity ones, making the data density higher on the bigger disks. On the other hand, the average failure rates remain slightly above 0.0025 for VMs with disk size between 32 GB to 4 TB, i.e., roughly 85% of VMs. Combing with the observation of the distribution of disk configuration, we thus conclude that failure rates of VMs are quite steady around 0.0025, with respect to different size of disk capacities.

However, in contrast to the disk capacity , a different trend is observed for the number of disk, i.e., failure rates increase visibly with the number of disks. In particular, the average failure rates failure rate increase from 0.0005 for 1 disk to 0.005 for 6 virtual disks. This is almost a $10X$ increment in failure rates. Such different observations are due to the fact that

most of VMs have 2 disks covering a wide range of capacities. Actually, 83% of the failure events happen on the VMs with at most 2 disks. Moreover, the higher failure rates corresponding to more disks can be explained by that adding more disks to a server increases the chances of an independent disk failure. Similar findings are also reported by [3]. Comparing the two disk features, we note that the number of disks has stronger impacts on the VM failure rate, than the disk capacity.

Finally, we also want to draw the comparison of impact of different resource attributes on VMs' and PMs' average failure rates. To such an end, we look at the range of average failure rates, between a low provisioning to a high provisioning of a certain resource. In particular, we consider the increment factor of failure rates that are described earlier. We observe that the average failure rates of VMs increase by $2.5X$, $3X$, and $10X$ with respect to CPU counts, memory sizes, and disk counts. The average failure rates of PM increase by $5.5X$ and $5X$, due to CPU counts and memory sizes. This indicates that the number of disk has the highest impact on VM failures, while CPU counts and memory size are equivalently influential for PM failures.

B. Resource Usage

Next, we try to answer the question if increasing workloads on particular resources, i.e., CPU, memory, disk, and network, also increases the failure rates of both PMs and VMs.

1) CPU and Memory Usage: We look into the weekly failure rates with respect to CPU utilization and memory utilization, summarized in Fig. 8(a) and (b). Prior to analyzing the failure trend, we note that both the numbers of VMs and PMs decrease with CPU utilization, i.e., more than half of VMs and PMs is utilized at most 10%. As for the memory utilization, the majority of VMs is at most 10%, whereas the number of PMs surprisingly increases with memory utilization. When looking at CPU utilization ranging from 0 to 30%, accounting for majorities of VMs and PMs, one can see that VMs' average failure rates increase CPU utilization while PMs' average failure rate decrease with CPU utilization, by roughly an order of magnitude. A possible explanation for the increasing failure rate of VMs compared to PMs is the combination between the failure rate of the underlying hypervisor (e.g., due to overprovisioning reasons) and that of the actual VM (e.g., due to the workload). When considering the entire range of PMs' CPU utilization, the trend of average failure rates follows bath-tub curve, meaning moderately loaded PMs are more reliable.

In the case of memory, the average failure rates for both VMs and PMs follow an inverted bathtub curve, meaning that they first increase with memory utilization and then decrease. To achieve average failure rates below 0.005, PM's memory utilization should be less than 20% or above 70%. To achieve achieve failure rates below 0.0025, VM's memory utilization should be below 10% or above 50%. When comparing the relative differences between the max and min failure rate across all memory utilization, one can see that memory utilization has a stronger impact on PMs than VMs. Moreover, combining with previous finding that PMs with memory size between 4 to 128 GB experience lower failure rates, we conclude a reliable PM should equip a moderate size of memory and keep its utilization sufficiently high.

2) Disk and Network Usage: Finally, we look at the impact of disk space and network usage on failure rates. Since our dataset does not contain any PM-related disk and network information, we can only present results for VMs as shown in Fig. 8(c) and (d). The average failure rates increase with the disk utilization, from 0.001 (below 10% usage) to 0.003 (above 70% usage). The lower impact of disk usage on the VM failure rate, compared to CPU usage for instance, can be explained by the fact that disk space issues do not usually result in failures and can be easily solved by cleaning old files, to provide more free space. Such a finding resonances well to the fact that allocated disk spaces have a rather insignificant impact on the average VM failure rates. In the case of network, we quantify the usage as the network traffic in Kbps sent and received by the VM. Roughly, 45% of all VMs have an average bandwidth between 2 and 64 Kbps, and 34% transfer between 128 and 512 Kbps, while the remaining 21% totalize between 1024 and 8192 Kbps. The average failure rates increase from 0.001 to 0.005 for VMs with up to 64 Kbps. After that, the average failure rates decrease with network volume. Judging from the range of absolute values of average failure rates, our results show a weaker correlation between the transferred Kbps and the VM failure rates, compared to the CPU counts.

When comparing all resources by the relative differences between max and min average failure rates, we conclude that memory usage is the most dominating usage factor for PMs and CPU utilization is the key factor for VM's failure behavior. Moreover, as all average failure rates are observed below 0.005, memory utilization, the disk usage, and network transfers, appear to have similar degrees of influences in VM's failure rates.

VI. Impact of VM Management

In this section, we specifically discuss how two particular aspects of VM management, consolidation and turning on/off, affect VMs' failure rates.

A. VM Consolidation

Increasing the computation density by component counts rises the risk of hardware failures [4]. To overcome the limitations on parallel programming, consolidating VMs onto the same hosting platform serves as an alternative to increase computation density. We study if increasing the number of VMs leads to reliability issues, by comparing the trend of average failure rates computed across VMs associated with different consolidation levels. Specifically, we consider the average weekly failure rates. Since the consolidation level experienced by VMs changes over time due to VM turning-off and migrations, we propose to estimate it by the average monthly consolidation level of a VM, computed over one year observation period.

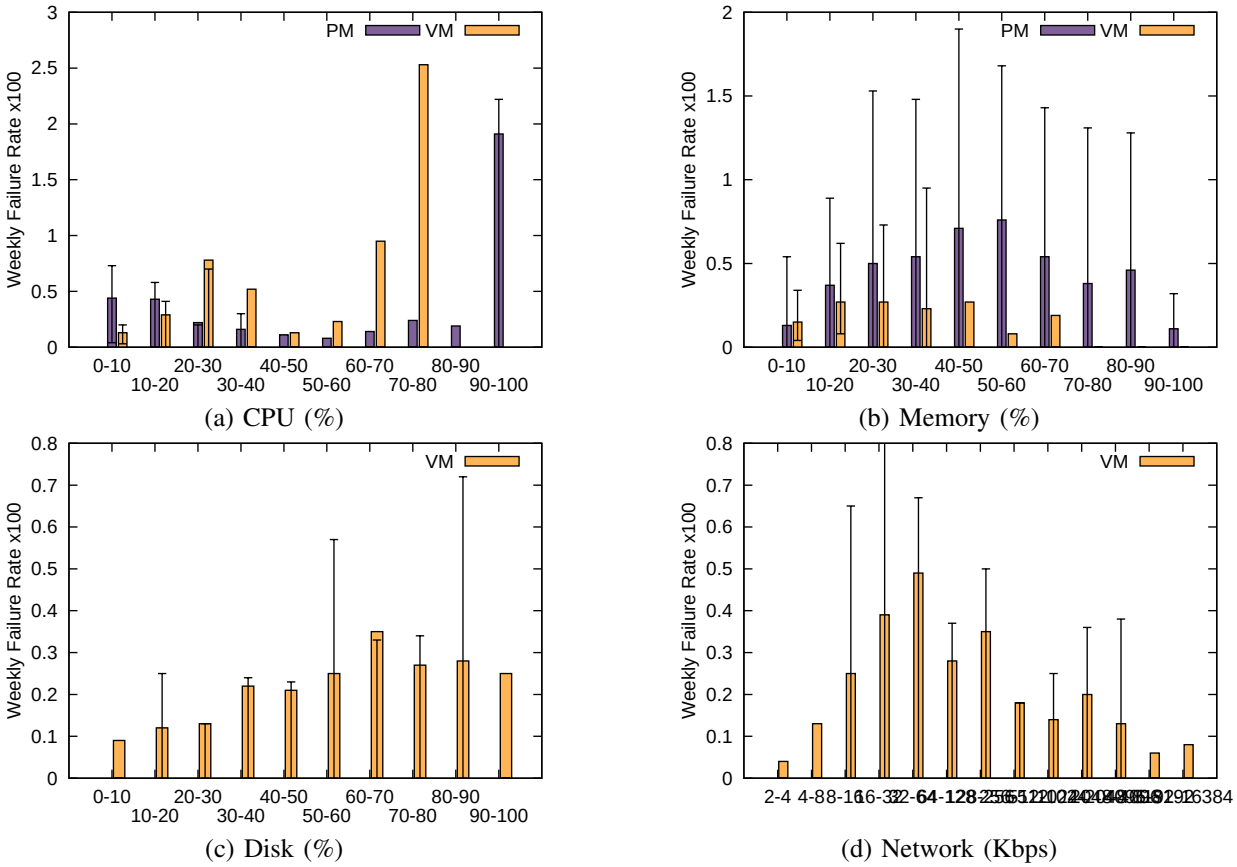


Fig. 8. Weekly failure rate across PMs and VMs relative to CPU, memory, disk and network usage.

First, we look at the VM and failure distributions across consolidation levels ranging from 1 to 32. We note that the number of VMs increases with the consolidation level, from 0.6% corresponding to 1 (i.e., VM is not sharing the hosting platform with other VMs) to 30% and 32% for 16 and 32, respectively. We depict the average failure rates and the corresponding 25^{th} and 75^{th} percentiles in Fig. 9. Clearly, we observe that the failure rate decreases significantly with the level of consolidation. This can be explained by the fact that the underlying machines that host more VMs, are high-end systems equipped with more reliable components and built-in fault tolerance features. This observation is also backed up by our previous finding on the relationship between PM's utilization level and failure rates. Furthermore, combining increasing failure rates in increasing VM's CPU utilization, we conclude that PMs hosting a decent number of VMs that are unfertilized result into low failure rates for both PMs and VMs. Our finding is a pleasant confirmation that virtualization can not only resolve the resource over-provisioning issues at today's datacenters but also potentially increase system reliability.

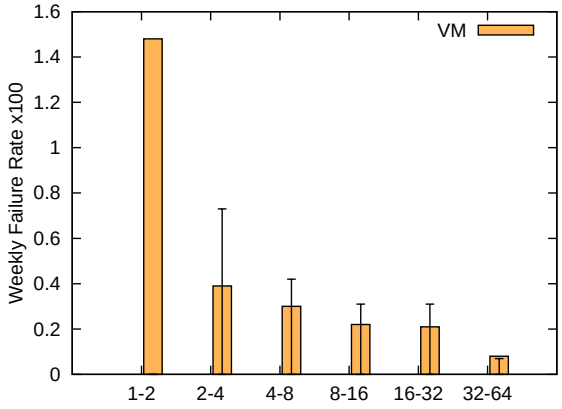


Fig. 9. Impact of VM consolidation level on weekly failure rate.

B. VM On/Off Frequency

The impact of wear and tear on hardware components is well known [16] and the frequency of turning PMs on/off can even deteriorate their lifespan. In contrast to PMs, VMs are designed to facilitate elastic resource provisioning and thus to be managed on demand, i.e., turning on/off when required by users. Indeed, VMs experience frequent on/off [17]. We are therefore interested in understanding how robust VMs are against frequent on/off in a similar way as for PMs.

To meet this end, we compare the weekly failure rates of VMs that experience different levels of weekly on/off frequency. For each VM, we collect its average weekly failure rate over one year observation period and average weekly on/off frequency, computed over a two month observation

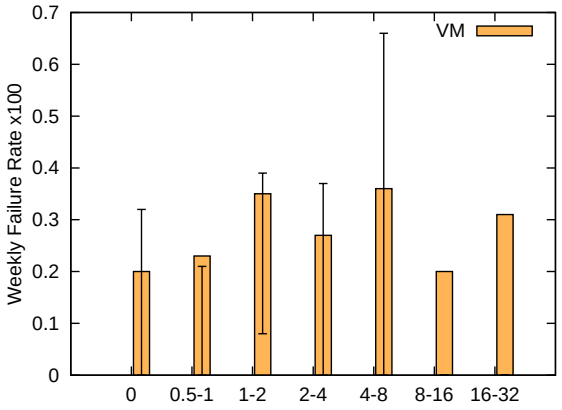


Fig. 10. Impact of monthly on-off frequency on weekly failure rate.

period. Note that to trace on-off frequency, we need to screen through fine-grained 15 minutes data points, for which we only have 2 month worth of data. Due to this limitation, our assumption is that the VMs on/off frequencies observed are consistent throughout the entire year.

In Fig. 10, we depict VM failure rates with respect to the average weekly on/off frequency. We first note that The number of VMs decreases with the on/off frequency, with 60% of all VMs being turned on/off at most once per month and only 14% of them are powered on/off 8 times per month. When focusing on the majority of VMs, i.e., with average on/off frequency less than 2 times per month, we can see an increasing trend of average failure rates from 0.002 to 0.0035. However, when considering the entire range of on/off frequency, one can observe that the average failure rates indeed vary, however without any obvious trend. . This observation implies that VM on/off frequency has a certain impact on the VM's reliability, especially from no on/off to 2 on/off per month. But, our findings dose not suggest that very frequent turning on/off VMs deteriorate VM's reliability.

VII. Summary and Conclusion

We summarize our key findings in the following:

- Differences in PM/VM failures – VMs have lower failure rates and lower recurrent failure probabilities than PMs. Inter-failure times of VMs are well captured by the Gamma distribution, showing a similar behavior as for PMs. Software inter-failure times are the shortest, compared to hardware/infrastructure related ones. The average repair time of VM failures is lower than for PM by almost a factor of two and follows the Log-normal distribution. Hardware failures take longest to repair. In addition to stronger time dependency, VM failures also show higher spatial dependency. The relationship between VM failures and their age does not follow a bathtub like function. Moreover, VM failures show a weak positive trend with age.
- Impact of resources usage and capacity on PM/VM failures – CPU units, memory size, and memory utilization are the most influential factors for PM failures. The key resource attributes affecting VMs' failure rates are the number CPU counts, the number of disk, and the CPU utilization, while the disk capacity has the least impact on VM failures.
- Impact of VM management on VM failures – VM failure rates decrease with the consolidation level. Systems consolidating a decent number of under-utilized VMs result into lower failure rates. Frequently turning on/off VMs do not seem to have a significant impact on VM failures.

We conduct a failure analysis on PMs and VMs hosted on commercial datacenters, using one year long data collected over 10K servers. Our analysis highlights the differences and similarities on PM and VM failure patterns, and correlation with resource capacity and usage. In addition to identifying the dominant resources affecting VM failures, we also shed light on the impact of VM resource management on VM reliability. Yet while some of our findings on PMs confirm those reported on physical systems by previous studies, some, particularly related to VMs, provide fresh perspectives and insights. Overall, VMs have lower failure rates than PMs, and show a surprising trend that increasing computation intensity by VM units does not deepen failure rate as for PMs.

References

[1] L. Barroso, J. Dean, and U. Hölzle, "Web search for a planet: The google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, Mar. 2003.
[2] E. B. Nightingale, J. R. Douceur, and V. Orgovan, "Cycles, cells and platters: An empirical analysisof hardware failures on a million consumer pcs," in *Proceedings of the Sixth Conference on Computer Systems*, ser. EuroSys '11, 2011, pp. 343–356.
[3] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *Proceedings of the 1st ACM Symposium on Cloud Computing*, ser. SoCC '10, 2010, pp. 193–204.
[4] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," in *DSN*, 2006, pp. 249–258.
[5] N. El-Sayed and B. Schroeder, "Reading between the lines of failure logs: Understanding how hpc systems fail," in *DSN*, 2013, pp. 1–12.
[6] "http://www.informationweek.com/data-center-outages-generate-big-losses/d/d-id/1097712?"
[7] 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," *TOS*, vol. 4, no. 3, 2008.
[8] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population," in *FAST*, 2007, pp. 17–28.
[9] B. Schroeder, S. Damouras, and P. Gill, "Understanding latent sector errors and how to protect against them," in *FAST*, 2010, pp. 71–84.
[10] Y. Liang, Y. Zhang, A. Sivasubramaniam, M. Jette, and R. K. Sahoo, "Bluegene/l failure analysis and prediction models," in *DSN*, 2006, pp. 425–434.
[11] D. Yuan, J. Zheng, S. Park, Y. Zhou, and S. Savage, "Improving software diagnosability via log enhancement," *ACM Trans. Comput. Syst.*, vol. 30, no. 1, p. 4, 2012.
[12] B. Schroeder, E. Pinheiro, and W.-D. Weber, "Dram errors in the wild: a large-scale field study," *Commun. ACM*, vol. 54, no. 2, pp. 100–107, 2011.
[13] R. K. Sahoo, A. Sivasubramaniam, M. S. Squillante, and Y. Zhang, "Failure data analysis of a large-scale heterogeneous server environment," in *DSN*, 2004, pp. 772–.
[14] HP OpenView, "http://support.openview.hp.com/."
[15] IBM Tivoli Monitoring, "http://ibm.com/software/tivoli/products/monitor/."
[16] K. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. Wiley, 2001.
[17] R. Birke, A. Podzimek, L. Y. Chen, and E. Smirni, "State-of-the-practice in data center virtualization: Toward a better understanding of vm usage," in *IEEE/IFIP Dependable System and Network (DSN)*, 2013.