Failure Process Characteristics of Cloud-Enabled Services

Besmir Tola, Yuming Jiang, Bjarne E. Helvik
Department of Information Security and Communication Technology
Norwegian University of Science and Technology
Email: {besmir.tola, jiang, bjarne}@ntnu.no

***Abstract*—The design of cloud computing technologies need to guarantee high levels of availability and for this reason there is a large interest in new fault tolerant techniques that are able to keep the resilience of the systems at the desired level. The modeling of these techniques require input information about the operational state of the systems that have a stochastic nature. The aim of this paper is to provide insights into the stochastic behavior of cloud services. By exploiting the willingness of service providers to publicly expose failure incident information on the web, we collected and analyzed dependability features of a large number of incident reports counting more than 10,600 incidents related to 106 services. Through the analysis of failure data information we provide some useful insights about the Poisson nature of cloud service's failure processes by fitting well known models and assessing their suitability.**

I. Introduction

As the cloud computing technology adoption enters its second decade, it continues to drive innovation for organizations and their customers. More and more new services, from customer service to financial transactions, from social networks to e-commerce, are continuously shifting to cloud models. Given this growth and our rising reliance on these cloud services, it become necessary that providers and customers themselves demand high level of a crucial aspect that is *dependability*, as the ability to deliver a service that can justifiably be trusted [1].

In addition to that, cloud services are provided through the interaction of different and complex building blocks. A cloud service relies on complex software, hardware and network infrastructure architecture that can be arbitrarily large in scale. That is why, when referring to their dependability, any of these blocks may fail and it becomes quite challenging to assess a cloud service reliability.

There have been different empirical studies performing failure analysis of cloud computing enviroments and services but while these prior works provide some very useful insights, they suffer from a few limitations that in our opinion leave a knowledge gap in understanding the failure process characteristics of such services. Prior studies have focused on specific infrastructure components [2]–[4]. For cloud operators and the dependability community it is important to understand not only specific component's reliability, but the reliability of the service as a whole. In this work we consider a diverse set of services where the collected data provide comprehensive information about the service reliability.

Other works have performed reliability and failure analysis of specific or limited set of cloud service providers (i.e. IaaS or SaaS) [5]–[7]. While, these studies have provided an in-depth analysis of the reliability of a few services they fail to capture a broader view of todays's cloud reliability. In a recent report, Skyhigh Networks [8] estimated that the average number of cloud-based applications usage has tripled in the past years reaching 1,427 distinct cloud services per company in 2016. That is why, we think that assessing a larger set of applications would be undoubtedly useful in helping operators to provide and design more dependable systems.

In order to overcome the above limitations, failure data about a large set of cloud services are mandatory. In this work, we take advantage of the fact that today's cloud service operators provide detailed failure incident information on custom web sites [9], [10]. Indeed, more and more service operators do promptly notify customers and media by reporting service incidents through several channels like, social media, corporate custom websites, third party monitoring tools or press releases. They publicly disseminate information about the current status of their services, usually including: 1) start and end times of an incident, 2) root cause, 3) failure impact on the services, 4) steps taken to repair a failure and 5) maintenance events.

We collected such publicly available incident reports published by a large number of cloud services with our own web crawling framework. Using this approach, we conduct the first large-scale measurement study of the dependability attributes and aggregated failure process characteristics for 106 cloud services, covering more than 10,600 failure incidents over a period of up to 3 years.

This paper is organized as follows: In section II we presents the most relevant related work and their main findings regarding failure process dynamics. In section III, we introduce the data sources used to collect the failure incident reports, the methodology used in collecting the data as well as the incident information regarding reliability data. In addition, we provide some high-level statistics about the data set characteristics. Section IV presents the various graphical and statistical techniques we utilized in modeling the failure process and estimating parameters for various well-known lifetime distributions. In Section V we show the results of the different methods we employed and finally in Section VI we conclude our work by summarizing the most important findings in this study and highlight some future investigations.

II. Related Works

Several recent works have analyzed and modeled dependability of cloud computing systems. In this section we cover the most relevant and closely related to our study.

Specific Reliability Studies: There have been a number of studies focusing on data center systems, specific components or service providers infrastructure dependability [2]–[4], [6], [11]–[15]. While studies are important to understand the characteristics of the different components and help in designing

new approaches to overcome their failures at a different level (e.g., through redundancy and fault-tolerant approaches), they are also limited because it is not clear how these individual component failures would affect a service's overall dependability. Not surprisingly, many of these studies originate from major cloud service providers because they have the means to conduct such studies. Examples are from Google [3], [6], Microsoft [13]–[15] or NetApp [11].

Di Martino *et al.* [7] study platform failures of a specific SaaS and expose some interesting insights as they found out that failure rates are directly proportional to the workload intensity but not to the workload volume (i.e., size of customer's data). The authors of [16] conduct a large scale analysis comparing and relating physical and virtual machine failures from commercial data centers. However, their study is limited due to an inconsistent clarity across different data sources they use.

Failure Process Characterization: Many of the previously mentioned studies have performed specific failure process characterization and here we report their findings so that the reader could have a better comparison with our findings.

The authors of [12] report that the number of failures per node are not Poisson processes and that normal and lognormal distributions are a better fit. They also report that time between failures (TBFs) are well modelled with Weibull distribution and repair times are with a lognormal distribution. Potharaju *et al.* [13], [15] show that network appliances (i.e., middleboxes) and network equipment failure inter-arrival times may be suitably modeled with heavy-tailed distributions using kernel density functions (in particular a mixture of lognormal), similarly the time to repair of middleboxes.

Di Martino *et al.* [7] reports that the distribution of TBFs of all failure types in a SaaS platform can be successfully fitted with a Fatigue life distribution whereas the distribution for all failure types except timeout errors may be well represented with an exponential distribution. The authors of [6] performed a failure analysis of a well known cloud provider and they found out that for server type failures, the Weibull distribution is the best model to represent the TBFs whereas, the lognormal and the loglogistic distributions are better models for the downtime duration. Instead, for task issues, they find that for different task priorities, various distributions are the best fits for TBFs, spanning from the Weibull with decreasing hazard rate to lognormal, loglogistic and gamma with shape less than one, i.e., decreasing rate. As for the time to repair, they report that the lognormal and 3-parameter loglogistic distributions are the best fits.

Birke *et al.* [16] suggest that virtual and physical machine TBFs have very similar distributions and the best fit is the gamma distribution with decreasing hazard rate, whereas the times to repair are well modelled with lognormal distribution. Instead, Viswanath *et al.* [14] shows that time between successive failures on the same machine fits well an inverse function model. Another work regarding failure analysis of cloud computing systems is performed by [17]. They report that outage and vulnerability incident intensity of various cloud services is best modeled by an exponential with intercept model. However, we are not sure how they have performed the data collection and more important the data consistency, as the service they refer to is unavailable at this time, i.e., cloutage.org.

III. Dependability Data

In this section, we present our measurement methodology including the data sources we used, how we collected the data, and the information contained in the collected data that we used to characterize the failure nature of the different cloud services. Moreover, we also present some high-level statistics of our data set and discuss the information trustworthiness.

In order to obtain data about failure events, we exploited the fact that cloud services continuously provide detailed incident reports publicly available on the web today. For example, the file sharing cloud application box.com reports such information about its incidents at https://status.box.com/history. This status page provides us rich data about failure events. Of course, not all cloud services release status information. While searching the web for such status pages, we noticed an interesting trend: many cloud services rely on a few popular frameworks that provide a *status page* for web services to publish incident reports (failures and maintenance tasks) as well as real-time health (i.e., response time) reports of the services. A few of these frameworks are statuspage.io, status.io, cachet.io, and statuscast.com.

After identifying such frameworks, we searched on the web to discovered a large number of cloud services that use these frameworks and identified 142 cloud services using statuspage.io and status.io. We cast those using cachet.io and statuscast.com when we later realized that many of the reports had missing information about the down time of failure incidents. In addition to these 142 cloud services [1], we further included 3 top cloud service providers, Amazon, Google Cloud and Azure, into our sample, even though they were not using any of the above four status publishing frameworks.

To obtain the reliability data reported on the status pages of the services, we developed ad-hoc web crawlers using Selenium [18]. We carried out the web crawls starting from October 31, 2016 to November 12, 2016. During the crawls, we scraped the entire incident reporting history (i.e., including incidents before our crawling start date) for each of the 145 cloud services. Therefore, for all cloud services in our data set, we have the complete historical information about their incidents starting from the begining of their respective reporting period until November 12, 2016. Not all services have the same reporting period and this could be due to different reasons, e.g., services have started adopting such frameworks in different time frames. The reporting period of the services composing our data set spans from a minimum of 3 months up to a maximum of 3 years for the different services. Furthermore, since maximum likelihood estimation (MLE), in our analysis, plays an important role in estimating lifetime distribution parameters, we apply a rule of thumb stating that MLE could be heavily bias if the data sample's size is lower than 10 [19]. For our study, in order to be on a safer side, i.e. better minimum estimation, we consider only the aggregated failure process that contain more than 15 events per each service. Applying this threshold, the number of services analyzed is reduced to 106.

To characterize the 106 cloud services, we additionally identified the cloud service model type. In our data set, we consider only Infrastructure-as-a-Service (IaaS) providers as

[1]The full list of the services may be found at: https://gitlab.com/Tola/Service_List/blob/master/service_list.txt

providers. Whereas, the two other cloud service models (i.e. SaaS and PaaS) have been considered as a single model, i.e., *applications*, due to the fact that distinguishing among them in today's cloud ecosystem is not straightforward. Thus, the overall data set consists in 91 *applications* and 15 *providers*. In addition, in order to mitigate any concern about possible bias in out data set we identified their categories as derived from Alexa's categories [2]. We observe that applications fall into 20 distinct categories and examples of these categories are *social networking*, *collaborative tools* and *communication services*. While not shown here due to lack of space, they are reported in the service list [1].

After the data collection, we parsed the scraped HTML pages to extract the useful information, i.e., start and end time of each incident, title and operator annotations about the incident's severity. The later one being operator's indication about the severity level of each incident. We noted that the status pages frameworks provide an interface for the operators to annotate the incident impact on the overall service from a small list of severity levels that include full disruption, partial disruption, critical, major, minor and a few other annotations indicating service degradation or informational message related to scheduled maintenance events. For the scope of this paper, in our analysis, we exclude the scheduled maintenance reports and separate the remaining incidents into two severity categories, 'major' and 'minor'. Major incidents include events related to service outages and minor incident those related to service degradation. This way, we assume that a service disruption/degradation has taken place only when the operators report so, by using the above annotations in each incident. We correlate these pieces of information with the external data we collected (i.e., application/provider) during our analysis.

Table I gives some high-level statistics of our collected incident reports. Consider that the number of cloud providers is small because we only consider IaaS cloud providers. It is interesting to observe that, on average, there are twice as many major incidents per provider compared to applications, indicating that despite various redundant systems utilized, providers experience more frequent service disruption.

TABLE I: **High-level data statistics.**

	Applications	**Providers**
#Cloud Services	91	15
#Incident Reports	8278	2357
#Major incidents	968	311
#Minor incidents	7310	2046

Our study is based on incidents reported by cloud services themselves. We rely on their trustworthiness, and we think that cloud service operators do not have an incentive to report wrong or falsified information about service failures to their customers; the whole point of releasing such information is to improve the transparency. Misleading failure reports can damage the reputation of cloud services and may even lead to significant financial losses if they lose customers. That is why, our aim is to only interpret the information that is publicly reported by operators themselves even though we are aware that this could lead to a limitation on our study.

[2]http://www.alexa.com/topsites/category

IV. Failure Process Characterization

In this section we introduce the procedure utilized to characterize the failure process that cloud enabled services experience on a service level basis. In our investigation, we assume a single service as a whole system made of different components that may fail and thus result or not to its unavailability. The information contained in the reports does not always provide insights on what kind of failures they experience (i.e. root cause) that is why we treat the system as a *black box* by not considering how the system 'looks inside'. Thus, each failure process consist in an aggregated process of different components that fail, and hence, leading to a service degradation or service interruption. In this sense, it is outside the scope of this paper to investigate what kind of failures such system components undergo.

In doing the assesment of service reliability, for each of them, we exclude the maintenance events and consider only events that have caused a service outage or a service degradation by regarding them as failures, just as stated by operators themselves. In addition to that, each service is considered as a repairable system [20], since these are systems where the components are mostly repaired rather than discarded every time they experience a failure.

The analysis of failure data from repairable systems generally follows a similar procedure: *A)* Identify the process type and model, *B)* Perform a model assessment and *C)* Choose the lifetime distribution that best fits the empirical time between failures. Such analysis can be performed using both, graphical and quantitative methods. The graphical techniques provide many advantages like, being quick and easy to use, require some simple calculations as well as achieve a visual test of model representation. On the other hand, they have the disadvantage of not being the most precise, are subjective to visual interpretation and can often be biased. These disadvantages may be overcome by the use of quantitative techniques. In performing the failure process analysis we make use of both of them.

A. Process Type and Model

Typically, the modeling of failure times of repairable systems regard the point process theory as the main tool used. The most typical used models for the failure process of such systems are the renewal process (RP), where the observations are independent and identically distributed (*i.i.d*), the homogeneous Poisson process (HPP), being it a special case of RP, and the non homogeneous Poisson process (NHPP) which handles inter-failure time trends through specific time dependent failure intensity functions.

When considering repairable systems, the rate at which failures occur during the normal operation is referred to as the rate of occurrence of failures (ROCOF) or 'failure intensity'. Let *N(t)* be a function that counts the number of failures from a starting observation time until *t*. This is a step function that increases by one every time a failure is experienced. Since we are considering repairable systems, i.e. more than one failure happens, the *failure intensity* is defined as the infinitesimal increase in the number of expected failures by time *t* as follows:

$$z(t) = \lim_{\Delta t \to 0} \frac{E\left[N(t+\Delta t)\right] - E\left[N(t)\right]}{\Delta t} = \frac{\mathrm{d}E\left[N(t)\right]}{\mathrm{d}t} \quad (1)$$

and the expected number of failures by time *t* is often referred to as the *cumulative failure intensity*, i.e.,

$$Z(t) = \int_0^t z(u)du = E\left[N(t)\right] \tag{2}$$

The simplest model for repairable systems is the HPP model where $Z(t) = \lambda t$, $z(t) = \lambda$ and the times between failures are *i.i.d* exponentially distributed with mean $1/\lambda$. The probability of having *n* number of failures within an interval T is given by the Poisson distribution with parameter λT. In this case, the cumulative distribution function (CDF) of inter-arrival time between failures is $F(t) = 1 - e^{-\lambda t}$ and the mean time between failures (MTBF) equals the reciprocal of the failure intensity λ.

As for the NHPP model, we have considered two commonly used models, the *power-law* [21], alternatively called Duane model, and the *exponential (or log-linear) law* model [22], sometimes called Cox-Lewis model, where the failure intensity in both cases are not constant in time and defined as:

$$z(t) = \alpha t^{-\beta}, \quad \alpha > 0, \beta < 1 \tag{3}$$

$$z(t) = \exp\left(\alpha + \beta t\right), \quad -\infty < \alpha, \beta < +\infty \tag{4}$$

respectively. The number of failures in any interval of length *T* is distributed as a Poisson distribution with parameter $Z(T)$. In the former, the failure intensity illustrated in (3) has a polynomial nature and can model both, increasing ($\beta < 0$) and decreasing ($0 < \beta < 1$) failure intensity. Whereas, the exponential law (4) models a decreasing failure intensity for $\beta < 0$ and an increasing intensity if $\beta > 0$. When $\beta = 0$, both models reduces to the HPP constant failure intensity model. For additional details, we suggest the reader may refer to [20].

A very important aspect of repairable system's data analysis is testing for a possible trend in inter-failure times. As previously stated, NHPP models are characterized by time dependent failure intensity functions governing inter-failure times. In order to identify the possible process type and consequently observe a possible time trend we make use of a graphical technique called scaled Total Time on Test (TTT) plot [23] and evaluate the statistical significance using various trend tests for the failure inter-arrival times. The scaled TTT plot is similar to a plot of the cumulative number of failures vs. the operating time but scaled to a unit square and with the axes interchanged. The absence of a trend is evidenced when the data are located close to the diagonal. The idea is that if $z(t)$ is constant, so that the processes are HPP, then the TTT plot is expected to lie near the diagonal. In case the plot shows a convex, concave or an S-shaped form it means that the failure intensity is decreasing, increasing or bath-tube shaped, respectively, and in such case the appropriate model could be a NHPP.

In reliability literature, there are various statistical trend tests based on different null hypothesis (H_0). Typical ones are those that consider the null hypothesis of 'the process is HPP' with the alternative being NHPP with a monotone intensity, and for sure the Laplace test and the Military Handbook 189 (MHB) are the most known ones. They are both optimal tests against the alternative of NHPP's with exponential-law intensity and power-law intensity, respectively [20]. On the other hand, the rejection of the null hypothesis means simply that the process is not a HPP but it could still, however, be a RP and thus still evidence a trend absence as the authors of [24], [25] showed. Using simulations, they proved that the Laplace and MHB may be misleading when used to detect trend departures from general renewal processes and this goes in contrast with [7]. To overcome this, several other tests have been proposed where the null hypothesis is a RP, like the Lewis-Robinson, the Mann test or variants of them [26].

However, since these tests are valid when the alternative hypothesis have a monotone trend, i.e., monotone increasing or decreasing failure intensity, we make use of a different test that is more powerful against both monotonic and nonmonotonic trends as proposed by Kvaløy *et al.* [26]. This test is a generalized Anderson Darling (GAD) test for RP null hypothesis where the null hypothesis is rejected at a 5% significance level if the test statistic is greater than 2.492.

B. Model Assessment

In assessing the HPP model assumption we perform a distribution fitting for the time between failures and check whether the exponential distribution is a reasonable fit as well as validate the fitting using a goodness of fit (GOF) statistical test. For the processes that may be modeled with a non homogeneous Poisson process, we make use of specific graphical methods to assess their suitability in modeling the time dependent failure intensity.

A quick and simple graphical technique in identifying whether a process may be modeled with a power-law model is the Duane plot [21]. It is a plot of the cumulative MTBF measures vs. cumulative failure times on a log-log graph. In case the data are consistent with a power-law model, the points in the graph will approximately follow a straight line with slope β and intercept -$\log_{10}\alpha$. Whereas, in the case of an exponential model failure intensity, a plot of the cumulative number of failures vs. failure times on a log-linear graph is used and the points should roughly follow a straight line with slope β and intercept $-\alpha \ln \beta$ [27]. In both cases, in addition to the plot we fit a cubic polynomial curve to the data, using a robust least-square method, in order to estimate α and β model parameters.

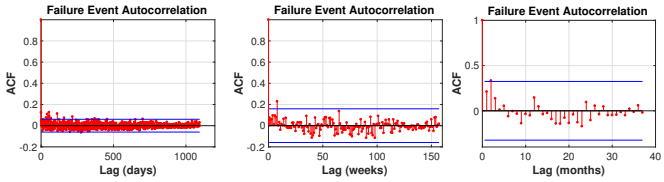
C. Time Between Failures Distribution Fitting

In order to identify which parametrized distribution is the best model that may be used for the times between failures of each service we apply the method of maximum likelihood estimation (MLE) for the parameters of the theoretical distribution that may best fit the data. In addition, we obtain the 95% confidence bounds and check whether the empirical cumulative distribution (CDF) is a suitable fit of the theoretical one and does not ecceed the confidence bounds. Then, we validate the outcome using the well-known Chi-Square GOF test [28]. This test determines if a data sample comes from a specified probability distribution, i.e, null hypothesis, with parameters estimated from the data.

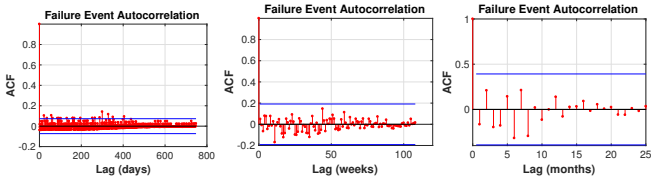
As possible theoretical distribution candidates we consider the exponential, gamma, Weibull and lognormal distribution. We compute the MLE for each of them and visually plot the comparisons.

V. Analysis Results

First, we start our analysis by investigating the failure process autocorrelation in order to identify whether the failure events have a temporal correlation. We utilize the autocorrelation function to measure the correlation of a random variable with itself at different time lags. The graphs in Figure 1 show the autocorrelation together with it's 95% confidence bounds, considering different time granularity, i.e., day, week and month, of only the first two services as mere examples. The obtained plots give a clear indication that the hypothesis of independence for the number of failures occurring in different time intervals cannot be rejected. Autocorrelation is stronger in a monthly granularity but still significantly low as it does not exceed the 95% confidence bounds. Moreover, we notice that even in a lower granularity, i.e., hours, and for all the services under analysis, that due to lack of space we will not present, the results are similar and show a low or no correlation of failure events in time, suggesting that the counting process *N(t)* has independent increments.

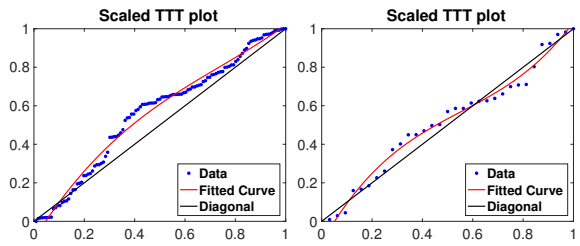


(a) Service 1

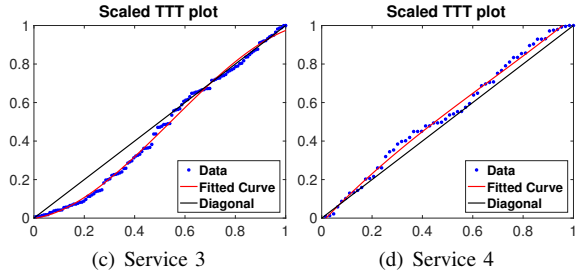


(b) Service 2

Fig. 1: **Failure process autocorrelation.**



(a) Service 1 (b) Service 2



(c) Service 3 (d) Service 4

Fig. 2: **Scaled TTT plots.**

Next, in Figure 5 we illustrate the scaled TTT plots for the first four services (service number is just for identification). Notice that the choice of showing just some service plots is just for illustration purposes. We observe that each of the four services shows different trends. Through a close inspection of the plots for all the services, we may exclude the trend presence for 56 services, i.e., data located close to the diagonal, 24 have a convex shape and 18 have a concave shape, indicating that processes show a monotonic decreasing and increasing failure intensity, respectively. Only 8 of them show an S-shaped form suggesting that such processes have a non monotonic intensity function.

Table II gives the result of the statistical trend test analysis. It shows the number of services together with their cloud model separation, i.e., applications and providers, that the various trend tests reject and do not reject their respective H_0. Observing the results from the GAD test we may conlcude with a 95% confidence level, that out of 106 services, 72 of them do not reject the H_0 of RP and 34 of them do. This means that, 72 service failure processes may be modeled as RP having no reliability improvement or deterioration due to a trend absence and 34 of them may be modeled with NHPP model having either a monotone or a nonmonotone failure intensity. Notice that being a RP means that the times between failures may be distributed according to any lifetime distribution [29], and only in case they are exponentially distributed the process is a HPP with constant failure intensity. In particular, when comparing the results of GAD vs. MHB test, we note that 64 out of 72 do not reject the H_0 corresponding to HPP processes with 95% confidence. The 8 additional services that MHB rejected the H_0 correspond to services that the test rejected as not being HPP but still having 'no trend' because the GAD does not reject them and thus being RP with *i.i.d* failure inter-arrival times. Specifically, they are the 8 services that show an S-shape failure intensity in the scaled TTT plots.

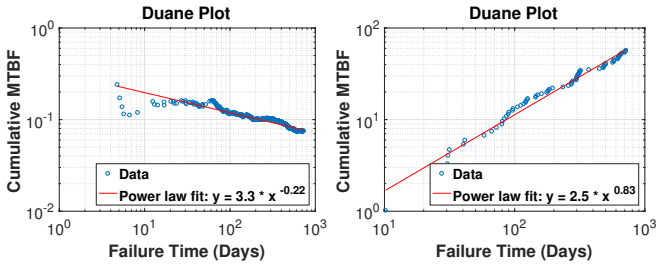
TABLE II: **Number of services (applications, providers) resulting from the different trend tests.**

	#Services (#Applications, #Providers)	
	Reject H_0	**Do not Reject H_0**
Laplace (HPP H_0)	43 (36,7)	63 (55,8)
MHB (HPP H_0)	42 (35,7)	64 (56,8)
GAD (RP H_0)	34 (27,7)	72 (64,8)

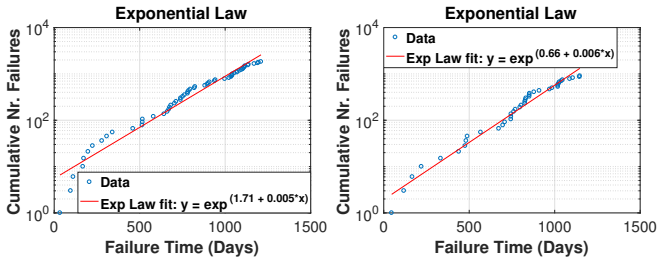
Within the set of 34 services that have either a monotonic or a non monotonic trend as resulted from the GAD test, we notice that 18 services may be reasonably well modeled with a NHPP power-law model as indicated from the fitting quality of the Duane plots. Though, only 12 of them may be modeled with an exponential-law model. The remaining 4 services does not fit either one of the models.

Figure 3 illustrate examples of the plots we have used to visually check the validity of the models. Specifically, Figures 3(a) and 3(b) are log-log plots including a fitted line that we have used to asses the fitting quality and estimate the model parameters. Whereas, Figures 3(c) and 3(d) show two of the services that are modeled according to an exponential model and we may notice that in both cases the points follow approximately a straight line.

In terms of failure intensity model parameters, Figure 4 shows the various model parameter values. The services having an intensity according to the power-law model yield a much more variable β compared to the exponential-law model. In particular, 9 services experience a deteriorating reliability (i.e.,



(a) Service 71 (b) Service 94



(c) Service 7 (d) Service 63

Fig. 3: **Power-law and exponential-law model plots.**

increasing failure intensity with $\beta < 0$) and 9 have a reliability growth (i.e., decreasing intensity with $0 < \beta < 1$). Whereas, all the services having a failure intensity modeled with an exponential-law experience a reliability deterioration. Note that β values in this case are relatively small but not equal to zero and such values are between 0.004 and 0.017.

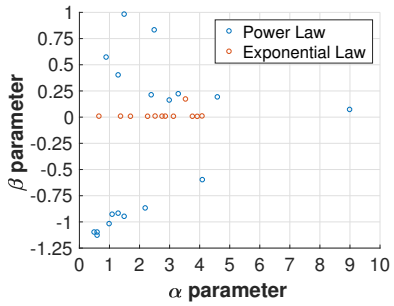
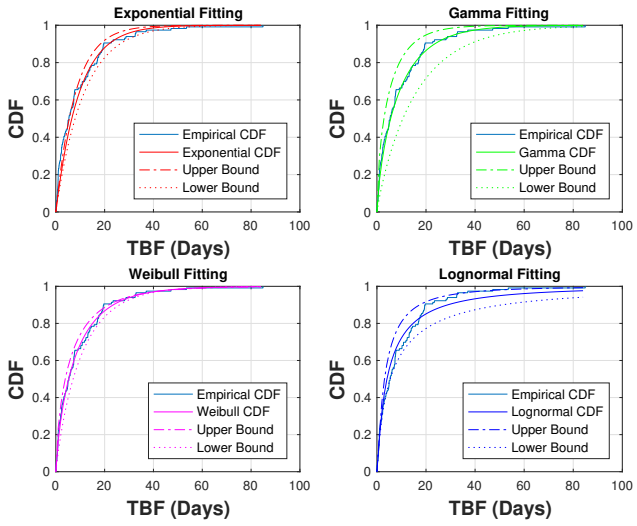


Fig. 4: **NHPP Power law and exponential law model parameters.**

Figure 5 shows the time between failures MLE computation for each of the theoretical distributions considered, together with their respective 95% confidence bounds, for service nr. 1. We report a single service just as illustration and due to space limitation. We observe that it is quite difficult to assess which of the distributions is the best fit. For example, take the exponential and gamma distribution, they both seem to be reasonably good fits of the empirical CDF. Moreover, we note that this difficulty is present in most of the services and thus we were not able to clearly decide which one was the best fit.

An additional confirmation to such observation comes from the computation of the GOF test. Specifically, when computing the Chi-Square test for a service, we notice that the null hypothesis (i.e., theoretical distribution) for the different considered candidates may not be rejected and thus we are still not able to choose the best fit for the empirical CDF. This issue results for many of the services and we may overcome this only by choosing the distribution that has the minimum



(a) Service 1

Fig. 5: **CDF fitting for time between failures.**

test statistic among those not being rejected. With this decision, we were able to select the best fit for the failure inter-arrival times of each service. Notice that this does not mean that only a single distribution, among those that we have taken into consideration, is a reasonable fit.

Applying the Chi-Square test, we discover that within the set of 64 services that the MHB do not reject the H_0 of HPP, 60 of them, at a 5% significance level, do not reject the exponential H_0 for the times between failures (53 Applications and 7 Providers). This test provides statistical significance that a HPP model may be a justifiable assumption for the majority of the services in our data set. Nonetheless, we still face the case of not rejection for some of the other distributions. Table III, shows the number of services, in brackets applications and providers, when we apply the minimization of the test statistic. For the 4 remaining services that MHB test suggests a trend absence and the additional 8 services that the GAD test does not reject the RP H_0, i.e. General RP in the table, the GOF test indicates that the majority of them may be modeled with a Weibull and lognormal distributions, 5 and 4 respectively, all being applications.

TABLE III: **Number of services resulting from the Chi-Square GOF test for inter-failure times distribution when minimizing the test statistic.**

	#Services (#Applications, #Providers)			
	Exp	**Gamma**	**Weibull**	**LogN**
General RP	2 (2,0)	1 (1,0)	5 (5,0)	4 (4,0)
NHPP Power	3 (3,0)	5 (5,0)	6 (5,1)	4 (3,1)
NHPP Exp	1 (1,0)	4 (1,3)	7 (7,0)	0

Among the services with a power-law failure intensity model, we find out that 3 of them have the exponential distribution as the best fit for time between failures, 5 have the gamma distribution and 6 of them Weibull, whereas the remaining 4 have the lognormal distribution. Instead, regarding those with an exponential model, as reported in table III, 7 of them have the Weibull distribution as the best fit, 4 have the gamma and only one has the exponential distribution. Furthermore, we observe that despite the different counting processes, the majority of application's time between failures

may be well modeled with an exponential distribution. In the cases of general RP and NHPP model failure intensity, the majority of applications have a Weibull distribution as the best fit for the time between failures followed by gamma, 18 and 10 services, respectively.

VI. Concluding Remarks

We presented the first large-scale study of cloud-enabled service aggregated failure process dynamics. By exploiting the willingness of cloud service operators to publicly report failure data we analyzed different reliability growth models and investigated their suitability in characterizing service failure processes. Our findings suggest that assuming a Poisson process for the number of failures in cloud environments is valid for more than half of the services we analyze, and thus assuming a memory less property is not a 'cardinal sin'. Moreover, we notice that in case service failure intensity is well modeled through time dependent functions there is not a single lifetime distribution that represents the best fit for the majority of the service's time between failures. On the contrary, we find that there are different statistical models that may properly fit the same failure process and we were able to decide a certain one only by making an additional decision, i.e. minimizing the GOF test statistics.

In our investigation we have considered common and rather simple models as possible fitting candidates. Having in mind the famous Box's quotes; "All models are wrong but some are useful" and "Overparameterization is often the mark of mediocrity", it would be interesting to examine the appropriateness of more refined and flexible failure intensity models (e.g., 3 parameter models) that could be derived through a combination of our considered models. Similarly, examining the pertinence of more adjustable life ageing distributions, i.e., Weibull-Poisson/Exponential-Poisson, and applying more choosy model selection criteria (e.g., Akaike Information Criterion) rather than GOF tests could lead to a more generalized model to describe cloud service failure processes.

VII. Acknowledgment

The work for this paper was performed in the context of the EU FP7 Marie Curie Actions project Grant Agreement No. 607584 (the Cleansky project).

References

[1] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11–33, 2004.

[2] 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 *FAST*, 2007, pp. 1–16.

[3] B. Schroeder, E. Pinheiro, and W.-D. Weber, "Dram errors in the wild: A large-scale field study," in *SIGMETRICS*, 2009.

[4] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: measurement, analysis, and implications," in *SIGCOMM*, 2011, pp. 350–361.

[5] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: Comparing public cloud providers," in *IMC*, 2010, pp. 1–14.

[6] P. Garraghan, P. Townend, and J. Xu, "An empirical failure-analysis of a large-scale cloud computing environment," in *High-Assurance Systems Engineering (HASE), 2014 IEEE 15th International Symposium on*. IEEE, 2014, pp. 113–120.

[7] C. Di Martino, Z. Kalbarczyk, R. K. Iyer, G. Goel, S. Sarkar, and R. Ganesan, "Characterization of operational failures from a business data processing saas platform," in *Companion Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 195–204.

[8] "Skyhigh Networks," https://www.skyhighnetworks.com/cloud-security-blog/12-must-know-statistics-on-cloud-usage-in-the-enterprise/.

[9] "Google Cloud platform status page," "https://status.cloud.google.com/summary".

[10] "Microsoft Azure status page," "https://azure.microsoft.com/en-us/status/".

[11] 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," *ACM Transactions on Storage (TOS)*, vol. 4, no. 3, p. 7, 2008.

[12] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," *IEEE Trans. Dependable Sec. Comput.*, vol. 7, no. 4, pp. 337–351, 2010.

[13] R. Potharaju and N. Jain, "When the network crumbles: An empirical study of cloud network failures and their impact on services," in *Proceedings of the 4th annual Symposium on Cloud Computing*. ACM, 2013, p. 15.

[14] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *SoCC*, 2010, pp. 193–204.

[15] R. Potharaju and N. Jain, "Demystifying the dark side of the middle: a field study of middlebox failures in datacenters," in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013, pp. 9–22.

[16] R. Birke, I. Giurgiu, L. Y. Chen, D. Wiesmann, and T. Engbersen, "Failure analysis of virtual and physical machines: patterns, causes and characteristics," in *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*. IEEE, 2014, pp. 1–12.

[17] L. Fiondella, S. S. Gokhale, and V. B. Mendiratta, "Cloud incident data: an empirical analysis," in *Cloud Engineering (IC2E), 2013 IEEE International Conference on*. IEEE, 2013, pp. 241–249.

[18] "Selenium Browser Automation ," http://www.seleniumhq.org/.

[19] "NIST/SEMATECH e-Handbook of Statistical Methods," "http://www.itl.nist.gov/div898/handbook/apr/section4/apr412.htm".

[20] H. Ascher and H. Feingold, "Repairable systems reliability: Modelling, inference, misconceptions and their causes," *Lecture Notes in Statistics*, vol. 7, 1984.

[21] J. Duane, "Learning curve approach to reliability monitoring," *IEEE Transactions on aerospace*, vol. 2, no. 2, pp. 563–566, 1964.

[22] D. Cox and P. Lewis, "The statistical analysis of series of events," 1966.

[23] J. T. Kvaløy and B. H. Lindqvist, "TTT-based tests for trend in repairable systems data," *Reliability Engineering & System Safety*, vol. 60, no. 1, pp. 13–28, 1998.

[24] G. Elvebakk, "Analysis of repairable systems data: Statistical inference for a class of models involving renewals, heterogeneity and time trends," *PhD dissertation, Norwegian University of Science and Technology, Dept. of Mathematical Sciences*, 1999.

[25] J. Lawless and K. Thiagarajah, "A point-process model incorporating renewals and time trends, with application to repairable systems," *Technometrics*, vol. 38, no. 2, pp. 131–138, 1996.

[26] J. T. Kvaløy, B. H. Lindqvist, and H. Malmedal, "A statistical test for monotonic and non-monotonic trend in repairable systems," in *Proc. European Conference on Safety and Reliability–ESREL*. Citeseer, 2001, pp. 1563–1570.

[27] C. Vallarino, "Fitting the log-linear rate to poisson processes," in *Reliability and Maintainability Symposium, 1989. Proceedings., Annual*. IEEE, 1989, pp. 257–261.

[28] P. E. Greenwood and M. S. Nikulin, *A guide to chi-squared testing*. John Wiley & Sons, 1996, vol. 280.

[29] B. Dhillion, *Reliability engineering in systems design and operation*. John Wiley & Sons, Inc., 1983.