Appears in the proceedings of the 27th IEEE International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2019)

Distribution Fitting and Performance Modeling for Storage Traces

Won Best Paper Award!

Muhammad Wajahat, Aditya Yele, Tyler Estro, Anshul Gandhi, and Erez Zadok
Department of Computer Science, Stony Brook University {mwajahat, ayele, testro, anshul, ezk}@cs.stonybrook.edu

***Abstract*—Understanding I/O workloads and modeling their performance is important for optimizing storage systems. A useful first step towards understanding the characteristics of storage workloads is to analyze their inter-arrival times and service requirements. If these characteristics are found to follow certain probability distributions, then corresponding stochastic models can be employed to efficiently estimate the performance of storage workloads. Such approaches have been explored in other domains using an assortment of distributions, including the Normal, Weibull, and Exponential. However, our analysis and others' past attempts revealed that none of those distributions provided a good fit for storage workloads. We analyzed over 200 traces across 4 different workload families using 20 widely used distributions, including ones seldom used for storage modeling. We found that the *Hyper-exponential* distribution with just two phases (H_2) was superior in modeling the storage traces compared to other distributions under five diverse metrics of accuracy, including metrics that assess the risk of over-fitting. Based on these results, we developed a Markov-chain-based stochastic model that accurately estimates the storage system performance across several workload traces. To highlight the applicability of our model, we conducted what-if analyses to investigate the performance impact of workload variability and garbage collection under various scenarios.**

***Index Terms*—Distribution fitting, storage traces, hyper-exponential, performance modeling.**

I. Introduction

Analyzing workload traces can provide useful insights into the characteristics of a system, helping to design better scheduling, caching, or service policies. Trace analysis can also help in the development of performance models that can enable useful what-if analysis, providing answers to questions such as "how will response time be affected if the arrival rate doubles?" or "how does workload variability impact performance?" Request-level traces, such as *inter-arrival times* (IATs) or *service times* (STs), are especially useful as they lend themselves to such performance modeling efforts and to the identification of system bottlenecks.

A popular approach to analyzing request-level traces is to *infer the distribution* of events (e.g., distribution fitting), such as the distribution of IATs [1], [2]. By fitting the empirical IATs to known distributions, such as the Normal distribution, one can leverage the various properties of the distribution to assess the traffic characteristics, such as burstiness and skew. Some of these distributions enable stochastic modeling of the performance of the system or device. For example, if the empirical IAT and/or ST traces can be shown to follow an Exponential distribution, then Markov Chain analysis or suitable queueing models can be developed to estimate the system performance [3]–[5]. The benefits offered by such distributions have encouraged many attempts to fit empirical data to these distributions or to simply assume that empirical data follows such distributions [6]–[8].

Unfortunately, storage workload characteristics are often too complex or skewed to be accurately modeled by simple Normal or Exponential distributions. Prior work has shown that storage workloads often exhibit long-tail latencies [9]–[13]; specifically, prior studies [2], [14] have found, via parametric fitting, that storage traffic IATs and access patterns are well modeled by the heavy-tailed Generalized Pareto distribution. These observations also extend to other workloads; for example, the IATs of web requests, grid computing workloads, and supercomputing workloads were found to be well approximated by the 2-parameter Weibull distribution [15]–[17]. Such complex distributions often have atypical properties that make them infeasible for practical analysis. For instance, the Pareto distribution can have an infinite variance [18]. Likewise, the generalized extreme value (GEV) distribution [19] and the log-logistic distribution [20], both of which we found can accurately model storage IAT traces (see Section V), can have infinite or undefined mean and/or variance. Thus, distributions that accurately model empirical request-level traces may not be helpful for analyzing storage workloads, say, for performance modeling, as we demonstrate in Section VI.

The goal of this paper is to analyze various request-level storage traces across multiple workloads and find distributions that (i) provide high distribution fitting accuracy *and* (ii) provide practical analytical properties across all traces. To the best of our knowledge, such practical and large-scale distribution fitting study has not been carried out for request-level storage traces. While request-level traces for various historical and modern storage systems already exist in the public domain (e.g., SNIA's trace repository [21]), prior studies that analyze such traces either did not focus on distribution fitting [22]–[24] or did not leverage the distribution fit to enable performance modeling [1], [2]; see Section VIII for a detailed discussion of related work.

The key takeaway of our analysis is that the flexible *Hyper-exponential* distribution is ideally suited for fitting and modeling request-level storage traffic. The Hyper-exponential distribution is a probabilistic mixture of several exponential distributions, or phases. In general, the Hyper-exponential distribution can model most heavy-tailed distributions by selecting the appropriate parameters [25]. Importantly, since it is a mixture of Exponential distributions, it is amenable to stochastic analysis, including queueing-theoretic modeling. Further, because the Hyper-exponential is expressed in terms of simpler Exponential distributions, it has finite and well-defined (closed-form) mean and variance, which are easy to compute (see Section III). The Hyper-exponential thus provides an opportunity to accurately model request-level traces while retaining the benefits offered by simpler distributions.

For our analysis, we use over 200 publicly available block-layer traces from different workload families (see Section IV). We used three different metrics and associated techniques to assess the accuracy of the distribution fit: (i) R^2 [26], which indicates the goodness of fit, (ii) the Jensen-Shannon divergence [27], which measures the similarity between two distributions, and (iii) the likelihood [28], which describes the plausibility of observing the empirical data given the fitted distribution. We also employed the Akaike and Bayesian information criteria (AIC and BIC) [29], [30], that assess over-fitting by evaluating the quality and simplicity of the fit.

We find that, compared to 19 widely used distributions (including Exponential, Generalized Pareto, Beta, Normal, Gamma, and Weibull), the Hyper-exponential distribution, with just two phases, provides better accuracy and lower risk of over-fitting across all traces we considered. For individual trace families, we found that Hyper-exponential is always among the top 3 distributions, and often the top distribution, for any metric of accuracy. While some distributions, like Burr and Pareto, do provide the best distribution fit in a few cases, their fit was poor in other cases (see Section V).

To highlight the importance of distribution fitting for workload traces, we developed a stochastic model based on using the Hyper-exponential distribution fit for IATs and STs that can estimate the performance of the storage system. Our model relies on the fact that the Hyper-exponential is a mixture of Exponentials, and is thus amenable to Markov chain modeling. Our resulting model accurately predicts the mean response time for workload traces. The median response time modeling error for our Hyper-exponential–based model was 17.5%; by contrast, the median error for other distributions was at least 2.7× larger than our error (see Section VI).

Finally, to illustrate the applications of our distribution fitting based performance model, we conducted two what-if analyses (Section VII). First, we investigated the impact on response time of an increase in workload traffic and/or increase in workload variability. We found that, at high arrival rates, doubling the workload variability can increase response time by as much as 66%. Second, we explored the performance degradation caused by garbage collection (GC), common in SSDs, as a function of various parameters, including the percentage of time spent in GC and its service rate slowdown. We found that GC can degrade average performance by as much as 2.8× even if it runs only 1% of the time. Without our performance model, the above analyses would require extensive experimentation, and might even be infeasible.

II. Background and Prior Work

To motivate the contributions of this paper and provide some context for our work, we next provide a brief overview of distribution fitting and then discuss related prior works that specifically focus on distribution fitting for storage traces. We discuss other related works later in Section VIII.

A. Significance of Distribution Fitting

Distribution fitting is the process of selecting a statistical distribution that best fits the target empirical data set. Distribution fitting is a popular tool for analyzing empirical data, with books and journals dedicated to the topic [18], [31]. We are specifically interested in Parametric fitting (or inference), where the empirical data is fit to a distribution with a known structure, but variable parameter values [32].

The key advantage of distribution fitting is that the many properties of the fitted distribution can now be directly applied to study the empirical data and possibly make predictions of future events. Further, appropriate statistical tests or hypothesis testing can be used to analyze the characteristics of the data. For example, if the service time (ST) of a storage workload is shown to follow a Pareto distribution, then the many moments of the distribution, as well as the tail probability (probability that a request takes longer than x seconds to complete), can be obtained in closed-form without any significant computational effort [18]. Likewise, if the inter-arrival time (IAT) of requests is shown to follow a Normal distribution, confidence intervals can be easily obtained for various measures of the data [5].

A more subtle but practical advantage of distribution fitting is the *performance models* that it enables. For example, if the request-level characteristics of a storage workload are shown to follow an Exponential distribution, its Markovian property can be used to track the evolution of the number of requests in the storage system as a continuous time Markov chain [4]. Likewise, based on the fitted distribution, various queueing-theoretic results can be applied to estimate the performance (e.g., response time) of the storage system.

Of course, the above advantages can only be realized if an *accurate* enough distribution fit is found. There are several techniques that exist in the literature for distribution fitting [18], [31]. Typically, there is an associated metric of accuracy that each technique aims to optimize for when deriving the parameters of a fitted distribution. Rather than using a single technique or metric, we employ the suggested practice [33], [34] of using *multiple techniques and metrics* to evaluate the distribution fitting; this avoids bias in results as a distribution may exhibit high accuracy for only one metric. We discuss our techniques and metrics in Section V-A.

B. Prior Work on Fitting Storage Traces

Prior work on distribution fitting for storage workloads is restricted to analyzing traces from a specific source. Gomez *et al.* [2] analyzed disk access patterns for HP-UX servers [35] and found that the spatial access pattern is well modeled by a

Pareto distribution. The authors employed parametric fitting to find the Pareto parameters but did not evaluate the accuracy of the fit. Gracia-Tinedo *et al.* [14] analyzed network traffic for the UbuntuOne cloud storage service and found that the IATs of some of the operations have long tails and are thus not well approximated by the Exponential distribution. Instead, the authors visually inspected the IATs and used a Pareto distribution fit. Birke *et al.* [36] analyzed storage workloads in an enterprise cloud and found that the VM-level storage capacity is well approximated by an Exponential distribution.

In general, heavy-tailed distributions have been used to model storage workload characteristics. However, we note that the above works rarely employ (one or more) statistical tests for evaluating the accuracy of the distribution fit. Further, prior work has not explored the performance models enabled by the fitted distribution.

III. The Hyper-Exponential Distribution and Related Performance Models

We now describe the Hyper-exponential distribution, which we find to be an accurate fit for the storage traces we analyze in Section V. We then discuss the performance models enabled by the Hyper-exponential distribution, and other distributions.

The k-phase Hyper-exponential distribution, denoted as H_k, is a probabilistic mixture of k Exponential distributions. The k-phase Hyper-exponential has $(2k-1)$ parameters, and can be expressed as:

$$H_k = \begin{cases} Exp(\lambda_1) & \text{with probability } p_1 \\ Exp(\lambda_2) & \text{with probability } p_2 \\ \vdots \\ Exp(\lambda_k) & \text{with probability } p_k, \end{cases} \tag{1}$$

where $p_1 + p_2 + \ldots + p_k = 1$. Since the Hyper-exponential is simply a mixture of Exponentials, its moments are finite and can be easily expressed in closed form. For example, the mean (first moment) of a k-phase Hyper-exponential is $\sum_{i=1}^{k} p_i/\lambda_i$.

In its simplest form, a 2-phase Hyper-exponential, or H_2, is a mixture of two Exponential distributions, say $Exp(\lambda_1)$ with probability p and $Exp(\lambda_2)$ with probability $(1-p)$. Since the number of parameters to be estimated for the k-phase Hyper-exponential distribution scales linearly with k, it is beneficial to use a small value of k for efficient distribution fitting. In Section V we show that the H_2 with $k = 2$ is already powerful enough to model the inter-arrival times (IATs) and service times (STs) of storage traces.

The Hyper-exponential distribution has been used for modeling metrics in other communities, such as modeling the amount of rainfall in a region [37], modeling the completion time in manufacturing systems [38], modeling the reliability of software [39], and even the modeling of network traffic [40]. However, to the best of our knowledge, the Hyper-exponential has not been applied to model storage workload characteristics.

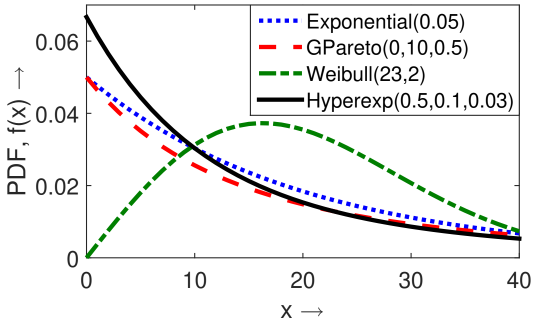


Fig. 1: Illustration of the probability distribution function (PDF) of various distributions, all with mean 20.

A. The Need for a Flexible and Heavy-Tailed Distribution

As discussed in Section II-B, storage workload characteristics often exhibit heavy-tailed behavior [18]; this is further evidenced by prior work that focuses on the long-tail latencies of storage workloads [9]–[13]. The Exponential distribution has a single parameter and is not heavy-tailed. In fact, a heavy-tailed distribution is often defined as one whose tail probability is heavier than that of an Exponential [41]. The Pareto and Weibull are heavy-tailed distributions that are often employed for distribution fitting of empirical data that exhibits long tails. There are several other heavy-tailed distributions that exist, such as the Lognormal, Burr, Loglogistic, etc.; we evaluated distribution fitting with many of these in our trace analysis in Section V. We find that, despite several statistical fitting techniques, the above distributions are *not* flexible enough to accurately fit the IATs and STs of storage workload traces obtained from different sources. That is, while a given heavy-tailed distribution fits a specific trace accurately, it does not fit other traces well. The 2-phase Hyper-exponential distribution, H_2, is flexible (3 parameters) and heavy-tailed. It has been shown that the k-phase Hyper-exponential can model most heavy-tailed distributions by selecting the appropriate parameters [25]. Like the Exponential distribution, the Hyper-exponential does have a decaying probability distribution function. We illustrate the probability density function (PDF) of the Hyper-exponential and other common distributions in Figure 1; here, we show the PDF for a 2-phase Hyper-exponential, or H_2.

B. Performance Models Enabled by the Hyper-Exponential

Since the Hyper-exponential has more parameters than the Exponential, it is more flexible than the Exponential distribution. However, the Hyper-exponential retains many of the analytical advantages of the Exponential distribution. Specifically, the memoryless or Markovian property of the Exponential allows us to model the evolution of events as a continuous time Markov chain [4]. If the IAT and ST are modeled as Exponentials, then we can model the storage system using an M/M/1 Markov chain, as shown in Figure 2. The Markov chain tracks the number of requests in the system as they dynamically increase due to arrivals and decrease due to service events. By solving for the steady-state probability of

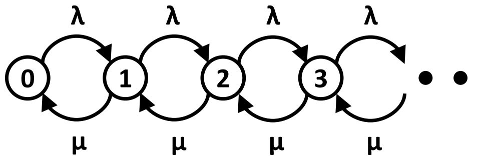


Fig. 2: Illustration of an M/M/1 Markov chain performance model with mean IAT = $1/\lambda$ and mean ST = $1/\mu$. The Markov chain tracks the number of requests in the storage system.

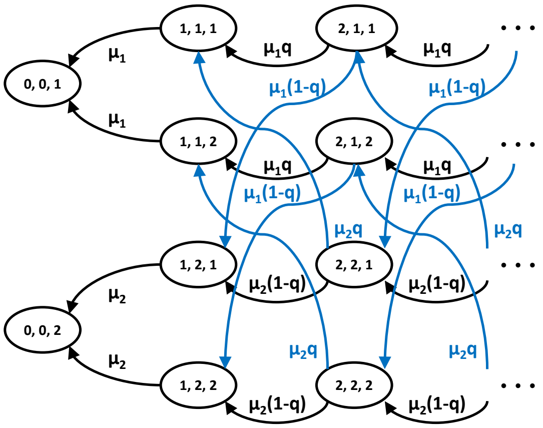


Fig. 3: Illustration of an $H_2/H_2/1$ Markov chain with the IAT modeled as an $H_2(p, \lambda_1, \lambda_2)$ and the ST modeled as an $H_2(q, \mu_1, \mu_2)$. We color-code some of the transitions and only show ST events for simplicity. The state space (i, j, k) refers to the number of requests in system, phase of the ST, and phase of the IAT, respectively.

the different states of the chain, we can derive the mean number of requests in the system [5]; the mean number in system can then be converted to the mean response time via Little's Law [42]. Fortunately, the Hyper-exponential is also amenable to Markov chain analysis due to its mixture of Exponential nature. Specifically, each phase of a Hyper-exponential can be modeled as a state in the Markov chain. However, the resulting chain is quite complex. Consider a workload whose IAT and ST are modeled as 2-phase Hyper-exponentials:

$$IAT = \begin{cases} Exp(\lambda_1) \text{ w.p. p} \\ \\ Exp(\lambda_2) \text{ w.p. (1-p)} \end{cases} \quad ST = \begin{cases} Exp(\mu_1) \text{ w.p. q} \\ \\ Exp(\mu_2) \text{ w.p. (1-q)} \end{cases}$$

Figure 3 shows the Markov chain for the $H_2/H_2/1$ system with the above IAT and ST distributions. Such chains have a repeating structure and can be solved, either numerically (e.g., using matrix analytic methods [43]) or analytically. Thus, the Hyper-exponential is a flexible and heavy-tailed distribution that also allows for exact performance models.

For other distributions, such as Pareto or Weibull, exact performance models are not known. However, approximations are available [44]; we leverage these models and approximations in Section VI to model the performance of a storage system with different distribution fits.

IV. Description of Traces and Workloads

For the distribution fitting analysis, we consider more than 200 different block-level traces from 4 different sources, including those from Flash-based devices and hard disks. We focus on the following request-level information in the traces, when available:

- *Inter-arrival time (IAT):* The IAT is defined as the time between successive requests. When analyzing IAT, we distinguish between reads and writes to better understand their individual characteristics.
- *Service time (ST):* The ST is defined as the time taken by the request for processing at the device, and does not include the queueing/waiting time at the device or at the upper layers. ST is often difficult to obtain as there is some queueing that happens within the device which cannot be easily tracked due to vendor-specific (proprietary) firmware [45].
- *Response time (RT):* The RT is the performance metric defined as the time taken by the request to complete service from when it first arrives at the block layer.

A. Florida International University traces

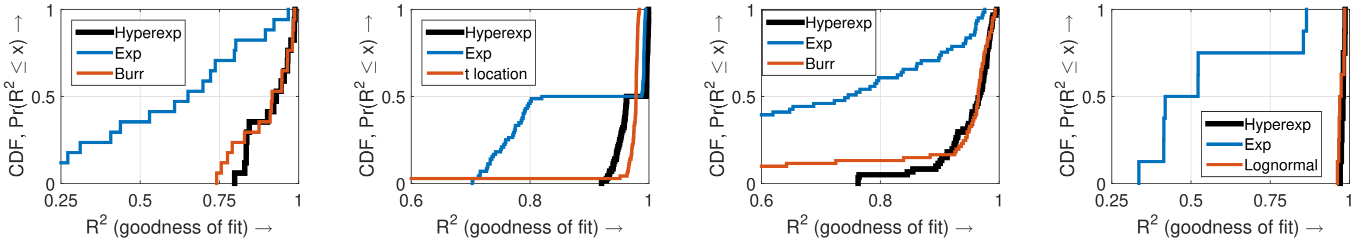
These are an assortment of 3-4 week-long block traces obtained from various HDD-based production systems in the Department of Computer Science at Florida International University (FIU) by Verma *et al.* [46]. The *home 1-4* workloads are 4 separate traces of the home directories of 4 different users in FIU's research group. The *mail* workload served the department's e-mail inboxes. The *online* workload is a web server hosting the department's course management system. The *webmail* workload is a web interface to the department's mail server. The *webusers* workload served the department members' websites. Lastly, the *webresearch* workload is an Apache server managing around 10 research projects. We analyzed the read and write IATs separately for each trace to better understand access type specific traffic, resulting in 18 total traces. ST information is not available for these traces.

B. Virtual Desktop Infrastructure (VDI) Traces

These are a collection of storage traffic traces from an enterprise virtual desktop infrastructure (VDI), obtained from Lee *et al.* [24]. The month-long traces contain I/O information for six different block storage devices (LUNs), with each device corresponding to one VDI server, which itself hosts about 50 VMs. We analyzed the read and write IATs separately for each of the 6 devices, resulting in 12 traces. ST information is not provided for these traces.

C. Mobile Storage Subsystem Traces

These 31 application-specific block-level I/O traces were collected on a Nexus 5 smartphone when running different mobile applications, as detailed in Zhou *et al.* [23]. The storage subsystem is a Flash-based (SanDisk iNAND) eMMC. I/O information is collected at the block layer and the eMMC driver layer, so we have both IAT and ST traces, and response times. We analyzed the read and writes IATs and STs separately,



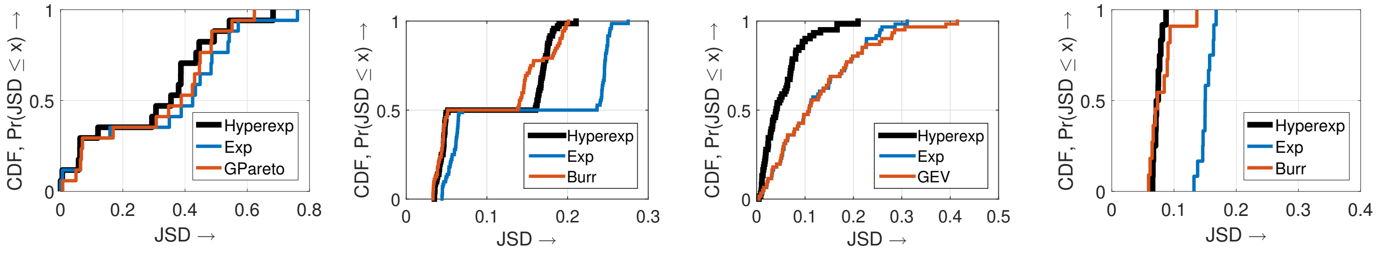
(a) FIU.

(b) Microsoft.

(c) Mobile.

(d) VDI.

Fig. 4: CDF of the R^2 metric (higher is better) for Hyper-exponential, Exponential, and the best alternative distribution fit.



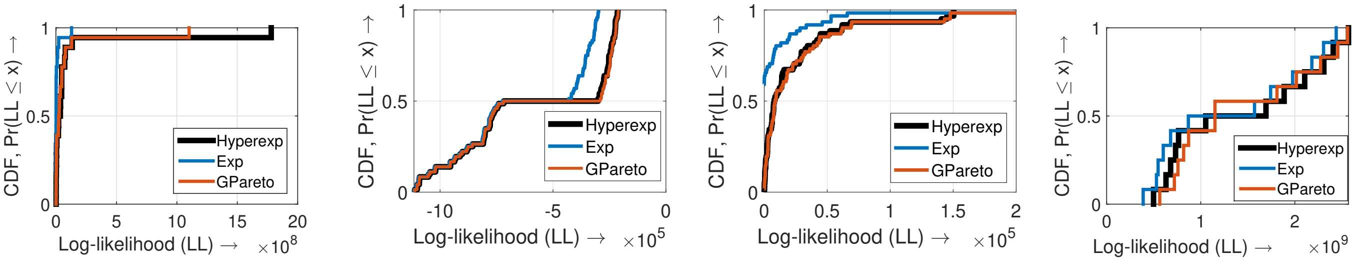
(a) FIU.

(b) Microsoft.

(c) Mobile.

(d) VDI.

Fig. 5: CDF of the JSD metric (lower is better) for Hyper-exponential, Exponential, and the best alternative distribution fit.



(a) FIU.

(b) Microsoft.

(c) Mobile.

(d) VDI.

Fig. 6: CDF of Log-likelihood (LL, higher is better) for Hyper-exponential, Exponential, and the best alternative distribution.

resulting in 62 traces each. We leveraged the response times to validate our performance models in Section VI.

D. Microsoft Production Server Storage Traces

These historical traces were collected on real production storage servers of Microsoft services, as described in Kavalanekar *et al.* [22]. We use the block-level IAT information for the storage metadata servers, separated by access type (reads and writes), resulting in 72 traces. ST information is not provided for these traces.

V. Request-Level Distribution Fitting

We now present our first contribution, analyzing the distribution fit for storage workload traces. We start with a description of the distribution fitting methods we employ and then present our results for IAT analysis and ST analysis.

Note that the focus of our study is analyzing the distribution fitting of the traces, and not the analysis of the traces themselves. The traces we analyze for distribution fitting have been studied before [22]–[24], [47], in other contexts, such as for deduplication and energy management.

A. Methods and Metrics for Distribution Fitting

We consider several widely used distributions for our distribution fitting analysis[1]. To find the parameters of a given distribution that result in the best fit to the empirical trace, we use three different techniques. Each of these techniques has its own designated metric of accuracy. We also evaluate the risk of over-fitting by reporting metrics that estimate the model quality. Using multiple techniques and metrics avoids any bias that a fitted distribution may have to a single metric [33], [34].

1) Least Squares Optimization to Maximize R^2

The coefficient of determination, R^2 [26], indicates the goodness of fit; that is, the closeness of the empirical data to the fitted distribution. It is often used as the first step in evaluating a fit. R^2 typically lies between 0 and 1, with higher values indicating a better fit, though negative values are possible when the fit is poor. We use the *least squares* approach to minimize the sum of the squares of the residuals between the

[1] The full list of distributions we consider includes Beta, Birnbaum-Saunders, Burr, Exponential, Extreme Value, Gamma, Generalized Extreme Value, Generalized Pareto, Half-normal, Hyper-exponential, Inverse Gaussian, Logistic, Loglogistic, Lognormal, Nakagami, Normal, Rayleigh, Rician, t-location scale, and Weibull.

empirical CDF and the CDF of the target distribution. Our global optimization heuristically chooses initialization points and then applies the interior-point method to find the best parameter values [48], [49]. At a high-level, the optimization tests the value of the objective function in the neighborhood of the initialization points, and moves in the direction that best improves the objective value [50]; eventually, the algorithm converges to the parameter values that result in the best value of the objective function [51].

It has been shown in prior statistical studies that the R^2 metric alone may be insufficient to evaluate the fit [52]; we thus make use of additional metrics as well.

2) Global Search Algorithm to Minimize Divergence

JSD, the Jensen-Shannon divergence [27], is a symmetric and smoothed version of the Kullback-Leibler divergence, and measures the similarity between two probability distributions. JSD is popularly used in information theory and coding theory to measure the relative entropy, or distance, between two distributions [53]. JSD typically lies in the $(0, 1)$ range, with lower values indicating higher accuracy. We use the global search algorithm [48] to find parameter values that minimize the JSD between the empirical PDF and the PDF of the target distribution. The algorithm is similar to the one described above for maximizing R^2, and uses a similar framework for initialization and convergence.

3) Expectation-Maximization to Maximize Likelihood

The likelihood objective function [28], the higher the better, is often used in Bayesian statistics to describe the probability of observing the empirical data given the target (fitted) distribution [54]. We use the popular expectation-maximization (EM) algorithm [28] to find the distribution parameters that maximize the expected log likelihood. EM is an iterative algorithm; we follow the suggested practice of using normally distributed values, with the same mean and variance as that of the empirical data, to generate our initial guesses [55], [56]. The best performing initialization is then chosen.

4) Akaike Information Criterion

AIC [29], the lower the better, is an estimator of the relative quality of statistical models for a given dataset, and is often used for model selection. In simple words, AIC estimates the amount of information lost by the model, and deals with the trade-off between *goodness of fit* and *simplicity* of the model by penalizing log likelihood proportional to the number of model parameters. AIC is reported for the distribution fit obtained via the EM algorithm that maximizes likelihood.

5) Bayesian Information Criterion

BIC [30], the lower the better, is similar to AIC but imposes a larger penalty for the number of parameters. BIC can select the *true model* with probability close to 1 when the number of data points is high. As both AIC and BIC deal with the trade-off between goodness of fit and simplicity of the model, they allow us to to assess both over-fitting and under-fitting of distributions to data, and help select the best model.

B. Inter-Arrival Time (IAT) Trace Analysis

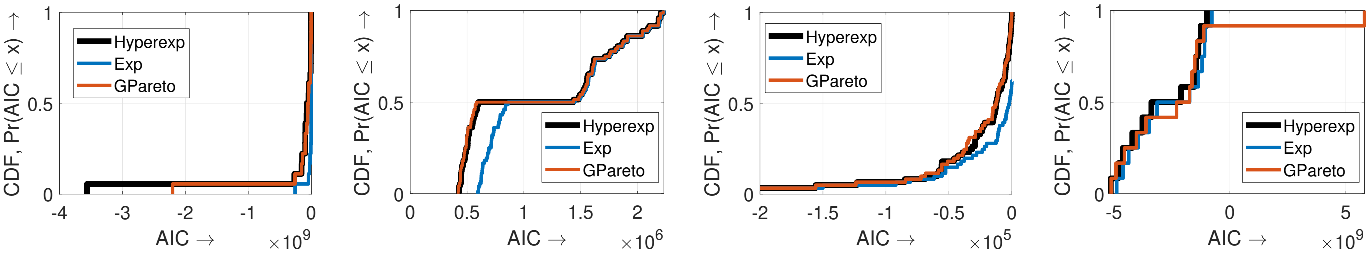
All four trace families that we studied have IAT information. Figures 4, 5 and 6, respectively, show the R^2, JSD and log-likelihood metrics for the different trace families under three distribution fits: (i) Hyper-exponential with two phases (H_2), (ii) Exponential, and (iii) the best alternative distribution (apart from Hyper-exponential and Exponential). We include the Exponential as a baseline as it enables useful performance modeling of systems and has often been used as the default IAT distribution in performance studies [6]–[8].

We see that the Hyper-exponential is always at least as good as the best alternative distribution; the median R^2 for Hyper-exponential ranges from 0.93–0.98 for all trace families. By contrast, the Exponential is typically inaccurate, with the median R^2 for Exponential ranging from 0.4–0.8 for the different trace families. Note the stark contrast around the median in the CDF plots for the Microsoft trace; this is because of the difference in behavior of reads and writes. We separately analyzed reads and writes and found that the fit accuracy for all distributions was better for reads than for writes, indicating bursty IAT behavior of write requests; this is expected as most operating systems batch writes in their page cache and flush them periodically in groups.

Finally, note that the best alternative distribution often changes based on the trace family. Similarly, we observed that the best alternative distribution *for a given trace family* changed with the metric of accuracy. We note that the likelihood value depends on the number of data points, and so likelihood values should not be compared across trace families.

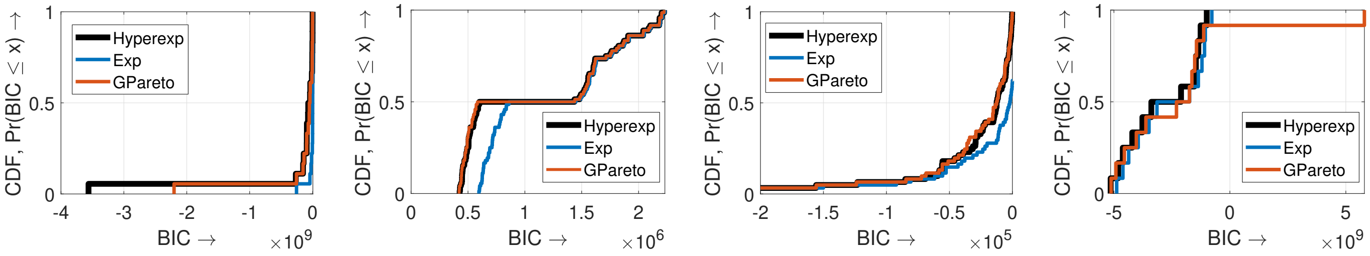
To assess the risk of over-fitting, we now present the AIC and BIC values for the various distribution fits. Figures 7 and 8 show the AIC and BIC metrics, respectively, for the different trace families; we show results for the H_2, Exponential, and the best alternative distribution. We see that H_2 provides a superior fit, and typically has the lowest median AIC and BIC (lower is better). Note that the AIC and BIC results look similar, as they are both based on the log-likelihood metric, with slightly different penalty functions for the number of model parameters.

To better assess the fitting capabilities of the Hyper-exponential, we show the top 3 distribution fits, using the median accuracy, across *each trace family* and for *all* traces in Tables I and III, respectively. We also show the top 3 distribution fits, using the median AIC and BIC values, across *each trace family* and for *all traces* in Tables II and IV, respectively. We see that the Hyper-exponential always ranks in the top 3 for any trace family under all 5 accuracy metrics. For the accuracy metrics in Table I, the Hyper-exponential typically ranks as the top distribution fit for any trace family under at least two metrics, except for Microsoft. Further, the Hyper-exponential resulted in the best fit across all traces we used under JSD and log-likelihood (Table III), with the median being 46.1% and 3.4% more accurate, respectively, than the top alternative distribution. For R^2, the Hyper-



(a) FIU. (b) Microsoft. (c) Mobile. (d) VDI.

Fig. 7: CDF of Akaike information criterion (AIC, lower is better) for Hyper-exponential, Exponential, and the best alternative.



(a) FIU. (b) Microsoft. (c) Mobile. (d) VDI.

Fig. 8: CDF of Bayesian information criterion (BIC, lower is better) for Hyper-exponential, Exponential, and the best alternative.

FIU			**Microsoft**			**Mobile**			**VDI**		
R^2	**JSD**	**LL**	**R^2**	**JSD**	**LL**	**R^2**	**JSD**	**LL**	**R^2**	**JSD**	**LL**
t-location	H_2	H_2	H_2	Burr	GPareto	H_2	H_2	H_2	H_2	H_2	H_2
H_2	GPareto	GPareto	t-location	LogL	GEV	Burr	GEV	GPareto	LogN	Burr	Exp
Burr	Gamma	t-location	LogN	H_2	H_2	LogN	Exp	GEV	Burr	GPareto	GPareto

TABLE I: Ranking of the top 3 fitted distributions (top-to-bottom) for *each trace family*. Here, LogN, LogL, GPareto, and GEV refer to Lognormal, Loglogistic, Generalized Pareto, and Generalized Extreme Value, respectively.

FIU		**Microsoft**		**Mobile**		**VDI**	
AIC	**BIC**	**AIC**	**BIC**	**AIC**	**BIC**	**AIC**	**BIC**
H_2	H_2	GPareto	GPareto	H_2	H_2	H_2	H_2
GPareto	GPareto	GEV	GEV	GPareto	GPareto	Exp	Exp
t-location	t-location	H_2	H_2	GEV	GEV	GPareto	GPareto

TABLE II: Ranking of the top 3 fitted distributions (top-to-bottom), according to AIC and BIC, for *each trace family*. Here, GPareto and GEV refer to Generalized Pareto, and Generalized Extreme Value, respectively.

exponential is a close second, next only to t-location, and only by 0.3%. Finally, the Hyper-exponential also provides the highest quality fit, with the lowest median AIC and BIC values across all traces we used (Table IV). This shows that the Hyper-exponential's superior distribution fit is not a result of over-fitting.

In summary, the Hyper-exponential consistently provides superior distribution fit under diverse accuracy metrics for all trace families we consider.

C. Analyzing the Distribution Fit

Figure 9 shows examples of distribution fits from each family of traces. The x-axis is on a log scale, and the y-axis uses square root scale, a measure that preserves the relative y-axis values per x-axis point and allows us to visually compare tails of empirical data [57]. We show the histogram for the empirical trace data and overlay it with the probability density function (PDF) of the Hyper-exponential (H_2), Exponential, and the top alternative distribution for that trace.

Figure 9(a) represents the case where the H_2 captures the high PDF region (around 10^{-2}) well whereas the other distributions, including the best alternative distribution for this trace, Generalized Pareto, fail to accurately fit around this region. Figure 9(b) represents the case where all distributions perform similarly, but there is a difference at the tail distribution (right of the graph). On close inspection, we see that the Exponential under-fits and the Loglogistic over-fits the tail probability; by contrast, the H_2 accurately fits the tail. Figure 9(c) represents a worst-case fitting example where no distribution performs well. However, we clearly see that the H_2 has two distinct centers of high PDF (around 10^{-3} and 10^2) that provide good coverage of the empirical data. By contrast, the other distributions concentrate around a single IAT range. Finally, Figure 9(d) shows another non-

(a) R^2 (higher is better)

Distribution	$\mathbf{R^2}$
t-location	0.969
Hyper-exponential	0.966
Burr	0.962

(b) JSD (lower is better)

Distribution	JSD
Hyper-exponential	0.062
Lognormal	0.115
Weibull	0.118

(c) Log-likelihood, LL (higher is better)

Distribution	LL
Hyper-exponential	6259
Generalized Pareto	6054
t-location	4846

TABLE III: Median R^2, JSD, and log-likelihood across *all* traces for the top 3 distributions in each case, sorted by accuracy.

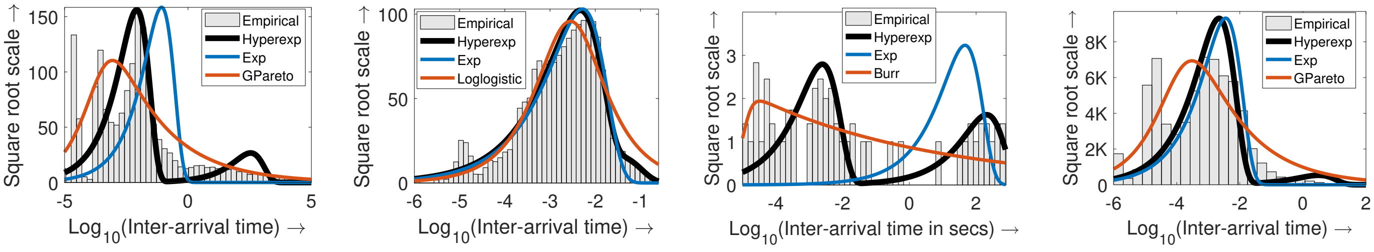
(a) AIC (lower is better)

Distribution	AIC
Hyper-exponential	-12513
Generalized Pareto	-12303
t-location	-10078

(b) BIC (lower is better)

Distribution	BIC
Hyper-exponential	-12496
Generalized Pareto	-12286
t-location	-10057

TABLE IV: Median AIC and BIC values across *all* traces for the top 3 distributions in each case, sorted by accuracy.



(a) FIU.

(b) Microsoft.

(c) Mobile.

(d) VDI.

Fig. 9: Results of distribution fit for a sample trace from each trace family for Hyper-exponential, Exponential, and the best alternative distribution (per log-likelihood).

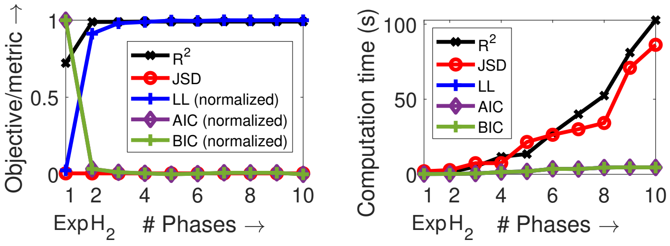
trivial example where the H_2 is able to fit the center of the PDF (around 10^{-3}) as well as the tail (around 10^0), whereas the other distributions only fit the center.

These examples illustrate the flexible and heavy-tailed nature of H_2, which is important for accurately fitting the different types of storage traces, as we alluded to in Section III-A.

D. Sensitivity Analysis for Number of Phases of the Hyper-Exponential

While we only make use of the 2-phase Hyper-exponential in the above analysis, the Hyper-exponential can be extended to include more phases (more Exponentials within the mixture distribution), though at the expense of increased computational complexity. Figure 10(a) shows the accuracy for all five metrics as a function of the number of phases, k, of the k-phase Hyper-exponential. These results are for the *home3* subtrace from the FIU trace family; results are similar for other traces. Note that $k = 1$ refers to the Exponential distribution. We see that accuracy increases as we go from the Exponential to the 2-phase Hyper-exponential, but then largely stabilizes beyond $k = 2$; note that for AIC and BIC, lower values are better. For JSD, whose range of values is small and thus not distinguishable in the figure, the value drops from 0.0061 for $k = 1$ to 0.0058 for $k = 2$ (smaller JSD is better), and then largely remains unchanged.

Figure 10(b) shows the time taken for the distribution fitting for different k values. We see that the computation time scales with the number of phases, as expected. Note that the LL,



(a) Improvement in accuracy.

(b) Increase in computation time.

Fig. 10: Impact of number of phases on (a) improvement in accuracy, and (b) computation time, for different techniques/objectives. Note that $phase = 1$ refers to Exponential and $phase = 2$ refers to 2-phase Hyper-exponential (H_2).

AIC, and BIC data points overlap as the same (EM) algorithm is employed for their fits (see Section V-A). We did not specifically optimize the code for computation time as that is not the focus of this work; however, we used the same code for all phases to enable a fair comparison. In summary, $k = 2$ provides a good trade-off between high accuracy and low computation time, thus representing a good choice for the Hyper-exponential distribution fitting.

E. Service Time (ST) Trace Analysis

We performed a similar distribution fitting for service times (ST). Only one of the trace families we studied, mobile storage traces, had ST information. Our ST analysis results are similar to IAT analysis, so we briefly highlight the results.

We again find that H_2 consistently provides a superior fit for service time under all accuracy metrics. We also find that the top alternative distribution changes with the accuracy metric. For R^2, JSD, and log-likelihood, the best alternative distribution was Lognormal, Birnbaum-Saunders, and Burr, respectively. The Exponential and Generalized Pareto distributions did not provide a good fit for ST.

VI. Performance Modeling Evaluation

We now present our performance modeling study that demonstrates the applicability of the Hyper-exponential distribution fit to predict the mean response time for the modeled storage workload. We first describe the performance models we use, and then present our modeling results.

A. Methodology

As discussed in Section III-B, both the Exponential and Hyper-exponential distributions enable Markov chain models. These, in turn, can be solved analytically or numerically to find the mean response time; for other distributions, only approximate results are available.

1) Hyper-Exponential–Based Model ($H_2/H_2/1$)

When the IAT and ST are distributed as 2-phase Hyper-exponentials, the resulting queueing model is referred to as a $H_2/H_2/1$ queue [5]. For this queue, closed-form analytical expressions for mean response time can be obtained [58]. The analysis involves tracking the queue length in the Markov chain (see Figure 3) given the input (IAT) and output (ST) processes, resulting in a degree 3 polynomial that can be solved to derive the mean queueing time, $E[W]$; here, $E[X]$ denotes the expectation or mean of the random variable X. Adding the mean ST to the mean queueing time gives the mean response time of the system, $[T] = E[W] + E[ST]$. Using the above approach, the mean response time for the $H_2/H_2/1$ model can be obtained in less than one millisecond with negligible CPU and memory overhead.

2) Exponential-Based Model ($M/M/1$)

When the IAT and ST are distributed as Exponentials, we can model the resulting $M/M/1$ system [4] as a simple Markov chain (see Figure 2) that can be easily solved to obtain the mean response time as $E[T] = 1/(E[ST]^{-1} - E[IAT]^{-1})$.

3) Models for Other Distributions ($G/G/1$)

For general IAT and ST distributions, the queueing model is referred to as the $G/G/1$ queue [4]. For $G/G/1$, exact results are not known, and Markov chain modeling is not applicable for distributions other than the Exponential and Hyper-exponential. However, the Kingman's approximation [44] is widely used to estimate the mean waiting time of $G/G/1$ as

$$E[T] \approx E[ST] + \frac{E[ST]^2}{2(E[IAT] - E[ST])} \cdot \left(\frac{Var[IAT]}{E[IAT]^2} + \frac{Var[ST]}{E[ST]^2} \right)$$

where $Var[X]$ denotes the variance of the random variable X. Note that given the parameters of the IAT and ST distribution fit, their mean and variance can be easily computed.

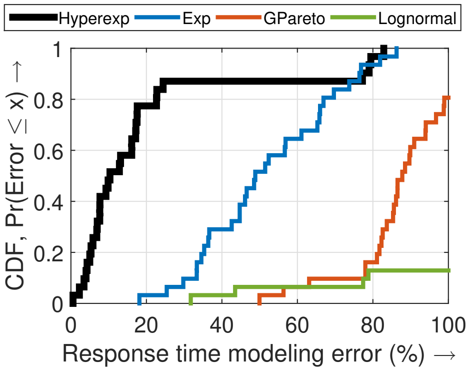


Fig. 11: CDF of mean response time modeling error using the Hyper-exponential and Exponential distribution fits, along with the other top alternative distribution fits for the mobile storage traces.

B. Response Time Modeling Results

We consider the mobile storage subsystem traces (see Section IV-C), which contain IAT, ST, and response time information for 31 traces. Zhou *et al.* [23] reported that the observed response time for their mobile traces is typically 2× the service time, suggesting that there is significant delay in the system. This motivated our response time modeling efforts for these traces. Since the scheduler used for the Flash-based mobile storage subsystem does not maintain different service queues for reads and writes [23], we model the IATs and STs for both request access types together.

Figure 11 shows the CDF of the mean response time modeling error for the 31 mobile storage traces. We show modeling error results for the case of Hyper-exponential and Exponential based distribution fits of IATs and STs, using the $H_2/H_2/1$ and $M/M/1$ queueing models, respectively. Additionally, we show results for the two other top alternative distribution fits (ordered by median accuracy), whose response time is modeled by the $G/G/1$ approximation. We note that while the General Extreme Value, t-location, Burr, and the Loglogistic distribution also resulted in high median accuracy for the IAT and ST distribution fits, their fitted parameters resulted in infinite mean and/or variance. Clearly, this would result in a poor approximation and so we omit these distributions.

We see that the Hyper-exponential-based modeling error for mean response time is significantly lower than the other distributions in Figure 11. The median error for the Hyper-exponential, Exponential, Generalized Pareto, and Lognormal based response time modeling is 17.5%, 48.8%, 87.8%, and 361.2%, respectively. The corresponding mean error numbers are 19.8%, 52.1%, 96.9%, and 672.2%, respectively; the mean error is higher than the median error due to the much higher error values for a few traces. The high error numbers for the Generalized Pareto and Lognormal based modeling should be expected since the response time model is only an approximation for these cases. Note that an error $> 100\%$ indicates that the predicted response time is at least twice the actual response time. Across all traces, the $H_2/H_2/1$ model reduces the relative modeling error by about 64% when

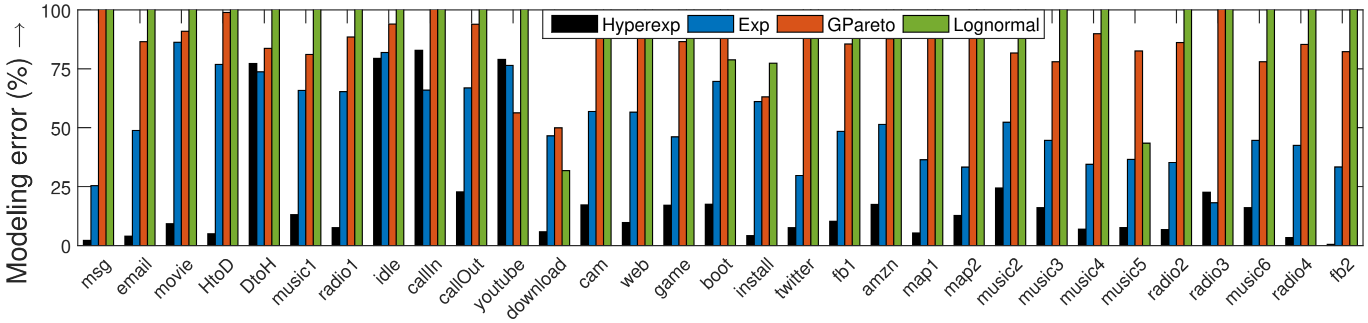


Fig. 12: Response time modeling error for the Hyper-exponential, Exponential, and other top alternative distribution fits for all mobile storage traces.

compared to the $M/M/1$ model.

Figure 12 shows the per-trace response time modeling errors for all 31 mobile storage traces. In most cases, the Hyper-exponential–based $H_2/H_2/1$ model results in low error; the modeling error is less than 20% for 24 of the 31 traces and less than 25% for 27 of the 31 traces. We inspected the remaining 4 traces (DtoH, idle, callIn, and youtube) and found that for most of these, the number of entries in the trace was small. It is likely that the smaller sample size in these traces resulted in poor accuracy for our modeling approach. It should be noted that the modeling error for these 4 traces using the other distributions in Figure 12 is also high. Although our approach does not have a minimum sample size requirement for the trace, in general, the more data points we have, the better is our modeling accuracy. In our evaluation, a minimum trace length of 3,000 provided good modeling accuracy.

Compared to the $M/M/1$ model, our $H_2/H_2/1$ model provides better modeling accuracy for 27 of the 31 traces, lowering the modeling error by about 76% for these traces. For the remaining 4 traces, the $M/M/1$ results in about 7% lower relative error. Note that the $H_2/H_2/1$ is significantly better than the Generalized Pareto and Lognormal based models for almost all traces.

VII. Modeling Use Case: What-If Analysis

An immediate and interesting use case for any system modeling approach is what-if analysis. We now present two such what-if analyses enabled by the $H_2/H_2/1$ performance models we presented in the previous section.

A. *Impact of Request Arrivals on Response Time*

Our first use case analyzes the impact of change in arrival rate of requests (inverse of inter-arrival time, $1/E[IAT]$) on response time. We consider the *msg* subtrace from the mobile storage traces and use the $H_2/H_2/1$ model, whose Markov chain is shown in Figure 3, to obtain the mean response time estimates. As discussed in Section VI-A1, the input to this model is the IAT and ST parameters of the subtrace. We use the IAT and ST Hyper-exponential distribution fit parameters for the *msg* subtrace from Section V as inputs for our modeling and what-if analyses in this section.

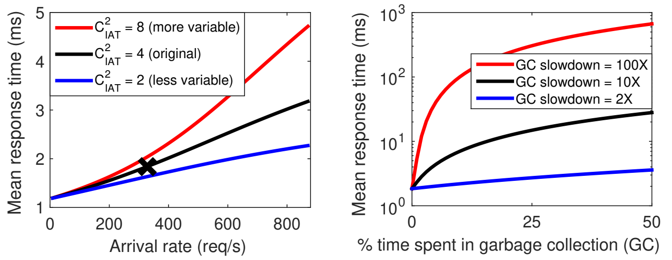
The black line in Figure 13(a) shows the results of our analysis; the cross marker on the line corresponds to the arrival rate observed in the *msg* subtrace (about 330 req/s). As the arrival rate increases, the mean response time increases, as expected. When the request rate doubles to 660 req/s, the mean response time increases from about 1.9ms to 2.7ms.

We also analyzed the impact of variability in inter-arrival time (IAT). A less variable IAT indicates that the request arrivals are more evenly spaced whereas a more variable IAT indicates that the request arrivals are more unevenly spaced (e.g., more temporally batched or bursty requests). We used the squared coefficient of variation of IAT, C^2_{IAT}, to parameterize variability. C^2_{IAT} is the normalized variability of IAT, and is mathematically defined as the variability in IAT divided by the square of the mean IAT. For the *msg* subtrace, $C^2_{IAT} \approx 4$. Note that C^2_{IAT} has no units: larger C^2_{IAT} values imply higher IAT variability. For a fixed IAT or request rate, a doubling of C^2_{IAT} implies a doubling of the IAT variability.

The red, black, and blue lines in Figure 13(a) show the impact on mean response time under $C^2_{IAT} = 8$, $C^2_{IAT} = 4$, and $C^2_{IAT} = 2$, respectively. We modeled the different IAT variabilities by changing the probability parameter (p) of the IAT hyper-exponential distribution (see Eq. (2)). We see that IAT variability significantly impacts response time, especially at high arrival rate. When C^2_{IAT} doubles from 2 to 4, mean response time increases by 29% on average, and by up to 53%, for the arrival rate range considered in Figure 13(a). Likewise, as C^2_{IAT} doubles from 4 to 8, mean response time increases by 33% on average, and by up to 66% at high arrival rates.

B. *Impact of Garbage Collection's Service Rate Slowdown on Response Time*

We analyzed a more complex use case using our performance models: the impact of performance degradation events, such as garbage collection (GC) on response time. To analyze this use case, we extended our $H_2/H_2/1$ Markov chain model for the *msg* subtrace to include an additional regime (or row of states) where the service rate of the storage device (inverse of mean service time, $1/E[ST]$) is reduced to represent the degraded service rate under GC; the service rate slowdown under GC is a model parameter that we vary. We also vary the percentage of time spent in GC, with the frequency of GC set to once every minute (configurable). Essentially, the



(a) Impact of request arrivals.

(b) Impact of garbage collection.

Fig. 13: Results of our what-if analysis. Figure (a) shows the impact of arrival rate on response time under different inter-arrival time (IAT) variabilities. Figure (b) shows the impact of time spent in garbage collection (GC) on response time for different GC service rate slowdowns.

new Markov chain (not shown here) additionally includes a replica of the chain in Figure 3 with lower service rates ($\mu_1 \times slowdown$ and $\mu_2 \times slowdown$), representing the state space under GC. Transitions are then added from every state in the original chain to the corresponding state in the replica chain, with the transition rate determined by the GC frequency. Inverse transitions are likewise added between the replica states and the original states to return the system to the previous, normal service rate.

This extended model is more complex than the $H_2/H_2/1$ model in Figure 3, and has not been analyzed before, to the best of our knowledge. Although this extended model has an infinite state space, the repeating structure of the underlying Markov chain still allows us to obtain the mean response time numerically as a function of the various parameters, using matrix analytic methods [43]—and in less than one second with a small memory footprint (a couple MBs). Note that our model is only an approximation, because real-world SSDs typically have proprietary firmware whose behavior and parameters are largely unknown [45]; nevertheless, to provide useful results, we explored all model parameters over a range of possible values. (More precise models can be constructed using parameters obtained from the GC bounds of a specific storage device under a given workload.)

Figure 13(b) shows the results of our analysis; note the log scale on the y-axis. Here, the arrival rate is the same as that observed in the *msg* subtrace (330 req/s). The range of parameters (percentage time spent in GC and GC slowdown) in the figure was chosen based on numbers reported in prior studies on GC [59], [60]. We see that the service rate slowdown significantly impacts response time, even when the fraction of time spent in GC is quite small. For example, even if we spend only 5% of the time in GC, the mean response time increases to 2ms, 4.4ms, and 40.9ms, under 2×, 10×, and 100× service rate slowdown, respectively. By contrast, without GC, the response time is about 1.8ms. We also tried other GC frequencies, once every 6s and once every 600s, and found the results (and trends) to be qualitatively similar.

The above use cases highlight the benefits of our distribution fitting based performance modeling approach. Our models can also be employed for analyzing other use cases, such as the impact of device aging (by considering multiple service rate slowdown regimes) or the impact of newer hardware (faster service rate) on response time. Without such models, the above what-if analysis would require extensive experimentation and might even be infeasible.

VIII. Related Work

A. Analyzing Disk Access Patterns

Storage workloads have been the focus of analysis since at least as far back as the 1980's when disk access patterns were studied. Ruemmler and Wilkes [35] presented an analysis of disk access patterns on three HP-UX systems collected in 1992. Their analysis focused on the volume of read versus write traffic, and the nature of the traffic, such as sequential, synchronous, swap traffic, etc. A similar study was later conducted by Keeton *et al.* [61] for disk block traces from a mail server and a database server in 2000. While both papers analyze skew in I/O load across devices, distribution fitting of the I/O traffic is not considered. Gomez and Santonja [2] later analyzed and specifically modeled the disk access patterns for the traces provided by Ruemmler and Wilkes. They found that the spatial access pattern is well modeled as a heavy-tailed Pareto distribution; however, the fitted parameters do result in infinite variance.

Verma *et al.* [46] analyzed block-level I/O traces from various servers at FIU's Computer Science department. The analysis focused on the usage of working sets, variability in workload intensity, and read-idle time distribution. The authors observed that data usage was highly skewed, but did not consider distribution fitting.

B. Analyzing Request-Level Storage Traces

Kavalanekar *et al.* [22] analyzed the characteristics of storage workload traces from production servers at Microsoft. The analysis focused on block-level statistics, file access frequencies, and temporal and spatial self-similarity. IAT (inter-arrival time) analysis was also conducted, focusing on the visualization of the IAT histograms and the overall rate of requests, but not on distribution fitting.

Gracia-Tinedo *et al.* [14] analyzed user-level storage workloads for the personal cloud service, UbuntuOne. The analysis shows that some of the inter-operation times, such as those for Upload, are not well approximated by the Exponential distribution and are better approximated by the Pareto distribution.

Zhou *et al.* [23] analyzed block-level I/O traces from common applications, such as email and YouTube, on a Nexus 5 smartphone equipped with a Flash-based eMMC (embedded multimedia card) device. The analysis focused on the size of the requests (or service times, STs) received by the eMMc device and their access type (read versus write). IATs of the traces are also considered, but the analysis is restricted to mean IAT and the extent of batching among requests.

In our analysis here we considered several of the above traces (that are publicly available on SNIA's repository [21]), but focused on the *distribution fitting* of their IAT and ST traces, separated by reads and writes. By finding an accurate distribution fit, we enabled queueing-based performance models that can accurately predict the response time for requests.

C. Analyzing the Aggregate Storage Volume of Workloads

Leung *et al.* [62] analyzed the traffic for two enterprise file servers deployed at NetApp. The paper analyzes various characteristics of file I/O traffic such as volume, lifetimes, access frequency, etc. The authors do find that several of the characteristics are heavy-tailed, but do not identify a specific distribution fit. Birke *et al.* [36] analyzed storage workloads on VMs in an enterprise private cloud and found that the VM-level storage capacity and used storage volume can be well approximated by an Exponential distribution. Mei *et al.* [63] analyzed and modeled a few traces from MSR and found that the spatio-temporal behavior is well modeled as a Gaussian.

Seo *et al.* [64] presented a data-mining and clustering approach to classify and thus characterize I/O workloads using previously published deduplication traces [21], [65]. While classification is a useful characterization of workloads, the classified workloads and clusters are not intuitive and require further analysis. For instance, while the authors do use the mean IAT as a feature, the clustering results do not provide any information about the IAT distribution within the cluster.

D. Analyzing the Network Traffic of Storage Workloads

Lee *et al.* [24] analyzed storage traffic on servers that host commercial virtual desktop infrastructure (VDI) VMs. The analysis is focused on the traffic volume and burstiness of specific applications on these VDI servers. A similar analysis was conducted on a smaller set of traces by Shamma *et al.* [66]. Drago *et al.* [67] analyzed the network traffic to Dropbox from home networks and found that a small number of users are responsible for most of the traffic.

E. Analyzing File System Characteristics

Prior work has also focused on analyzing file system characteristics, such as file size, file count, directory count, etc. For example, the file size distribution was found to be well modeled by a Lognormal [68], Lambda [69] (similar to Normal and Logistic), or Bimodal [70], [71] distribution, depending on the source of the trace. A recent work, Impressions [72], focused on generating realistic file system images with associated metadata to facilitate performance benchmarking of file systems. Our focus in this paper is on *request-level* modeling for IAT and ST (service time). ST is the *time* required to service a request on a device, and is thus different from file size.

F. Other Storage Analysis Works

There are several other storage analysis studies that focus on other metrics such as device failures [73], [74], data corruption [75], data deduplication [65], etc. These are orthogonal to our paper: we focus specifically on request-level distribution fitting and performance modeling.

IX. Conclusions

Storage workload modeling is critical to optimizing storage systems—often the slowest component of any system. Good models depend on an accurate characterization of inter-arrival times and service requirements. Many distributions exist that have been found to accurately model behavior in other domains, but not for storage systems—in part because storage systems exhibit multi-modalities and long tails [45], [76]. And some distributions that do fit storage systems, however, do not provide analytical properties that can be used to, say, build an accurate and efficient performance model for storage systems. This paper makes the following contributions:

1) We undertook a detailed study of distribution fitting for storage workloads, using over 200 traces from four different sources, and evaluated their fitness using 20 different probability distributions under 5 diverse accuracy metrics.
2) We discovered that the seldom used *Hyper-exponential* distribution provided the best fit based on all five metrics of accuracy. Moreover, we found that only two phases were needed to make this distribution fit well: more phases did not improve accuracy by much, and took longer to fit compared to other distributions.
3) This Hyper-exponential distribution with two terms (H_2) is amenable to performance modeling. We built such a model and evaluated it in predicting storage performance. Whereas the few other distributions (e.g., Exponential) that do enable modeling resulted in at least 48% median modeling error, and as high as 361% error, H_2's median error was under 18%.
4) We employed and extended our Hyper-exponential–based performance model to conduct two different what-if analyses. First, we highlighted the severe impact of workload variability on response time. Second, we investigated the performance impact of different parameters of garbage collection; we found that even if garbage collection is only active for a fraction of time, performance can degrade by as much 20×.

While our focus in this work is on modeling request-level characteristics of storage systems, we believe that the applicability of the Hyper-exponential for distribution fitting can extend to other fields as well, such as file system characteristics and other workloads or system traces.

Acknowledgment

This work was made possible in part thanks to Microsoft Azure, Dell-EMC, NetApp, and IBM support; NSF awards 1251137, 1302246, 1305360, 1622832, 1650499, and 1730726; ONR award N00014-16-1-2264; and a Stony Brook OVPR Seed Grant.

References

[1] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny, "Workload Analysis of a Large-scale Key-value Store," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '12, London, England, UK, 2012, pp. 53–64.

[2] M. E. Gomez and V. Santonja, "A new approach in the analysis and modeling of disk access patterns," in *Proceedings of the 2000 IEEE International Symposium on Performance Analysis of Systems and Software*, Austin, TX, USA, 2000, pp. 172–177.

[3] L. Kleinrock, *Queueing Systems, Volume 2*. New York: Wiley-Interscience, 1976.

[4] ——, *Queueing Systems, Volume I: Theory*. Wiley-Interscience, 1975.

[5] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[6] A. Gandhi, Y. Chen, D. Gmach, M. Arlitt, and M. Marwah, "Minimizing Data Center SLA Violations and Power Consumption via Hybrid Resource Provisioning," in *Proceedings of the 2011 International Green Computing Conference*, ser. IGCC '11, Orlando, FL, USA, 2011, pp. 49–56.

[7] X. Chen, L. Rupprecht, R. Osman, P. Pietzuch, F. Franciosi, and W. Knottenbelt, "CloudScope: Diagnosing and Managing Performance Interference in Multi-tenant Clouds," in *Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2015 IEEE 23rd International Symposium on*, Atlanta, GA, USA, 2015, pp. 164–173.

[8] C. Stewart, A. Chakrabarti, and R. Griffith, "Zoolander: Efficiently Meeting Very Strict, Low-Latency SLOs," in *Proceedings of the 10th International Conference on Autonomic Computing*, ser. ICAC '13, San Jose, CA, USA, 2013, pp. 265–277.

[9] Z. L. Li, C.-J. M. Liang, W. He, L. Zhu, W. Dai, J. Jiang, and G. Sun, "Metis: Robustly Tuning Tail Latencies of Cloud Systems," in *Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18)*, Boston, MA, USA, 2018, pp. 981–992.

[10] C. Iorgulescu, R. Azimi, Y. Kwon, S. Elnikety, M. Syamala, V. Narasayya, H. Herodotou, P. Tomita, A. Chen, J. Zhang, and J. Wang, "PerfIso: Performance Isolation for Commercial Latency-Sensitive Services," in *Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18)*, Boston, MA, USA, 2018, pp. 519–532.

[11] S. Yan, H. Li, M. Hao, M. H. Tong, S. Sundararaman, A. A. Chien, and H. S. Gunawi, "Tiny-Tail Flash: Near-Perfect Elimination of Garbage Collection Tail Latencies in NAND SSDs," in *15th USENIX Conference on File and Storage Technologies (FAST 17)*, Santa Clara, CA, USA, 2017, pp. 15–28.

[12] J. He, D. Nguyen, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Reducing File System Tail Latencies with Chopper," in *Proceedings of the 13th USENIX Conference on File and Storage Technologies*, ser. FAST '15, Santa Clara, CA, USA, 2015, pp. 119–133.

[13] M. Hao, G. Soundararajan, D. Kenchammana-Hosekote, A. A. Chien, and H. S. Gunawi, "The Tail at Store: A Revelation from Millions of Hours of Disk and SSD Deployments," in *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST 16)*, Santa Clara, CA, USA, 2016, pp. 263–276.

[14] R. Gracia-Tinedo, Y. Tian, J. Sampé, H. Harkous, J. Lenton, P. García-López, M. Sánchez-Artigas, and M. Vukolic, "Dissecting UbuntuOne: Autopsy of a Global-scale Personal Cloud Back-end," in *Proceedings of the 2015 Internet Measurement Conference*, ser. IMC '15, Tokyo, Japan, 2015, pp. 155–168.

[15] H. Li, D. Groep, and L. Wolters, "Workload Characteristics of a Multi-cluster Supercomputer," in *Proceedings of the 10th International Conference on Job Scheduling Strategies for Parallel Processing*, New York, NY, USA, 2004, pp. 176–193.

[16] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluation," in *Proceedings of the 1998 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, Madison, WI, USA, 1998, pp. 151–160.

[17] A. Iosup and D. Epema, "Grid Computing Workloads," *IEEE Internet Computing*, vol. 15, no. 2, pp. 19–26, 2011.

[18] H. Cramer, *Mathematical methods of statistics*. Princeton University Press, 1946.

[19] D. L. McFadden, "Modelling the Choice of Residential Location," in *Spatial Interaction Theory and Residential Location*. North Holland, 1978, pp. 75–96.

[20] P. R. Tadikamalla, "A Look at the Burr and Related Distributions," *International Statistical Review*, vol. 48, no. 3, pp. 337–344, 1980.

[21] "SNIA IOTTA Repository," http://iotta.snia.org/traces, Storage Networking Industry Association.

[22] S. Kavalanekar, B. Worthington, Q. Zhang, and V. Sharda, "Characterization of storage workload traces from production Windows Servers," in *Proceedings of the 2008 IEEE International Symposium on Workload Characterization*, Seattle, WA, USA, 2008, pp. 119–128.

[23] D. Zhou, W. Pan, W. Wang, and T. Xie, "I/O Characteristics of Smartphone Applications and Their Implications for eMMC Design," in *Proceedings of the 2015 IEEE International Symposium on Workload Characterization*, Atlanta, GA, USA, 2015, pp. 12–21.

[24] C. Lee, T. Kumano, T. Matsuki, H. Endo, N. Fukumoto, and M. Sugawara, "Understanding Storage Traffic Characteristics on Enterprise Virtual Desktop Infrastructure," in *Proceedings of the 10th ACM International Systems and Storage Conference*, ser. SYSTOR '17, Haifa, Israel, 2017, pp. 13:1–13:11.

[25] D. R. Cox, "A use of complex probabilities in the theory of stochastic processes," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 2, pp. 313–319, 1955.

[26] N. Draper and H. Smith, *Applied regression analysis*. Wiley, ch. 1.

[27] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[29] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected papers of hirotugu akaike*. Springer, 1998, pp. 199–213.

[30] G. Schwarz *et al.*, "Estimating the dimension of a model," *The annals of statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[31] D. Freedman, *Statistical Models : Theory and Practice*. Cambridge University Press, 2005.

[32] S. Geisser and W. Johnson, *Modes of Parametric Statistical Inference*, ser. Wiley Series in Probability and Statistics. Wiley, 2006.

[33] D. N. Moriasi, J. G. Arnold, M. W. V. Liew, R. L. Bingner, R. D. Harmel, and T. L. Veith, "Model evaluation guidelines for systematic quantification of accuracy in watershed simulations ," *Transactions of the American Society of Agricultural and Biological Engineers*, vol. 50, no. 3, pp. 885–900, 2007.

[34] A. Jakeman, R. Letcher, and J. Norton, "Ten iterative steps in development and evaluation of environmental models," *Environmental Modelling & Software*, vol. 21, no. 5, pp. 602 – 614, 2006.

[35] C. Ruemmler and J. Wilkes, "UNIX disk access patterns," in *USENIX Winter*. San Diego, CA, USA: USENIX Association, 1993, pp. 405–420.

[36] R. Birke, M. Björkqvist, L. Y. Chen, E. Smirni, and T. Engbersen, "(big)data in a virtualized world: Volume, velocity, and variety in cloud datacenters," in *Proceedings of the 12th USENIX Conference on File and Storage Technologies*, ser. FAST'14, Santa Clara, CA, USA, 2014, pp. 177–189.

[37] L. N. Singh and G. R. Dattatreya, "Estimation of the Hyperexponential Density with Applications in Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 3, no. 3, pp. 311–330, 2007.

[38] H. Papadopoulos and C. Heavey, "Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines," *European journal of operational Research*, vol. 92, no. 1, pp. 1–27, 1996.

[39] M. Ohba, "Software reliability analysis models," *IBM Journal of Research and Development*, vol. 28, no. 4, pp. 428–443, 1984.

[40] A. Feldmann and W. Whitt, "Fitting mixtures of exponentials to long-tail distributions to analyze network performance models," *Performance Evaluation*, vol. 31, no. 3-4, pp. 245–279, 1998.

[41] J. Nair, A. Wierman, and B. Zwart, "The fundamentals of heavy-tails: Properties, emergence, and identification," in *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '13, Pittsburgh, PA, USA, 2013, pp. 387–388.

[42] J. Little, "A Proof of the Queueing Formula $L = \lambda W$," *Operations Research*, vol. 9, pp. 383–387, 1961.

[43] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Philadelphia, PA, USA: ASA-SIAM, 1999.

[44] J. F. C. Kingman, "The single server queue in heavy traffic," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 57, no. 4, pp. 902–904, 1961.

[45] A. Aghayev and P. Desnoyers, "Skylight—a window on shingled disk operation," in *13th USENIX Conference on File and Storage Technologies (FAST 15)*, Santa Clara, CA, USA, 2015, pp. 135–149.

[46] A. Verma, R. Koller, L. Useche, and R. Rangaswami, "SRCMap: Energy proportional storage using dynamic consolidation," in *Proceedings of the 8th USENIX Conference on File and Storage Technologies*, ser. FAST'10, 2010.

[47] R. Koller and R. Rangaswami, "I/O deduplication: Utilizing content similarity to improve I/O performance," *Transactions on Storage*, vol. 6, no. 3, pp. 13:1–13:26, 2010.

[48] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.

[49] F. Glover, "A Template for Scatter Search and Path Relinking," in *Selected Papers from the Third European Conference on Artificial Evolution*, London, UK, 1998, pp. 3–54.

[50] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An Interior Algorithm for Nonlinear Optimization That Combines Line Search and Trust Region Steps," *Mathematical Programming*, vol. 107, no. 3, pp. 391–408, 2006.

[51] R. H. Byrd, M. E. Hribar, and J. Nocedal, "An Interior Point Algorithm for Large-Scale Nonlinear Programming," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[52] D. L. J. Alexander, A. Tropsha, and D. A. Winkler, "Beware of R2: Simple, Unambiguous Assessment of the Prediction Accuracy of QSAR and QSPR Models," *Journal of Chemical Information and Modeling*, vol. 55, no. 7, pp. 1316–1322, 2015.

[53] B. Fuglede and F. Topsoe, "Jensen-Shannon divergence and Hilbert space embedding," in *Proceedings of the 2004 International Symposium on Information Theory*, Chicago, IL, USA, 2004.

[54] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010.

[55] C. P. Robert and G. Casella, *Introducing Monte Carlo Methods with R (Use R)*, 1st ed. Springer-Verlag, 2009.

[56] C. Biernacki, G. Celeux, and G. Govaert, "Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models," *Computational Statistics & Data Analysis*, vol. 41, no. 3, pp. 561 – 575, 2003.

[57] Organisation for Economic Co-operation and Development (OECD), "What are equivalence scales?" http://www.oecd.org/eco/growth/OECD-Note-EquivalenceScales.pdf.

[58] V. N. Tarasov, "Analysis of queues with hyperexponential arrival distributions," *Problems of Information Transmission*, vol. 52, no. 1, pp. 14–23, 2016.

[59] L.-P. Chang, T.-W. Kuo, and S.-W. Lo, "Real-time Garbage Collection for Flash-memory Storage Systems of Real-time Embedded Systems," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 4, pp. 837–863, 2004.

[60] M. Jung, R. Prabhakar, and M. T. Kandemir, "Taking Garbage Collection Overheads off the Critical Path in SSDs," in *Proceedings of the 13th International Middleware Conference*, ser. Middleware '12, Montreal, Quebec, Canada, 2012, pp. 164–186.

[61] K. Keeton, A. Veitch, D. Obal, and J. Wilkes, "I/O Characterization of Commercial Workloads," in *Proceedings of the 3rd Workshop on Computer Architecture Evaluation using Commercial Workloads*, Toulouse, France, 2000.

[62] A. W. Leung, S. Pasupathy, G. Goodson, and E. L. Miller, "Measurement and analysis of large-scale network file system workloads," in *USENIX 2008 Annual Technical Conference*, ser. ATC'08, 2008, pp. 213–226.

[63] L. Mei, G. Xu, W. Yanjun, Z. Chen, and L. Mingshu, "Characterizing the spatio-temporal burstiness of storage workloads," in *Proceedings of the 5th International Workshop on Cloud Data and Platforms*, ser. CloudDP '15, Bordeaux, France, 2015.

[64] B. Seo, S. Kang, J. Choi, J. Cha, Y. Won, and S. Yoon, "IO Workload Characterization Revisited: A Data-Mining Approach," *IEEE Transactions on Computers*, vol. 63, no. 12, pp. 3026–3038, 2014.

[65] R. Koller and R. Rangaswami, "I/O Deduplication: Utilizing Content Similarity to Improve I/O Performance," *Transactions on Storage*, vol. 6, no. 3, pp. 13:1–13:26, 2010.

[66] M. Shamma, D. T. Meyer, J. Wires, M. Ivanova, N. C. Hutchinson, and A. Warfield, "Capo: Recapitulating storage for virtual desktops," in *Proceedings of the 9th USENIX Conference on File and Stroage Technologies*, ser. FAST'11, San Jose, CA, USA, 2011.

[67] I. Drago, M. Mellia, M. M. Munafo, A. Sperotto, R. Sadre, and A. Pras, "Inside dropbox: Understanding personal cloud storage services," in *Proceedings of the 2012 Internet Measurement Conference*, ser. IMC '12, Boston, MS, USA, 2012, pp. 481–494.

[68] J. R. Douceur and W. J. Bolosky, "A large-scale study of file-system contents," in *Proceedings of the 1999 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '99, Atlanta, GA, USA, 1999, pp. 59–70.

[69] K. M. Evans and G. H. Kuenning, "A study of irregularities in file-size distributions," in *In International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, 2002.

[70] S. Liu, X. Huang, H. Fu, and G. Yang, "Understanding data characteristics and access patterns in a cloud storage system," in *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 2013, pp. 327–334.

[71] N. Agrawal, W. J. Bolosky, J. R. Douceur, and J. R. Lorch, "A five-year study of file-system metadata," *Transactions on Storage*, vol. 3, no. 3, 2007.

[72] N. Agrawal, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Generating Realistic Impressions for File-system Benchmarking," in *Proceedings of the 7th Conference on File and Storage Technologies*, ser. FAST '09, San Francisco, CA, USA, 2009, pp. 125–138.

[73] I. Narayanan, D. Wang, M. Jeon, B. Sharma, L. Caulfield, A. Sivasubramaniam, B. Cutler, J. Liu, B. Khessib, and K. Vaid, "SSD Failures in Datacenters: What? When? And Why?" in *Proceedings of the 9th ACM International on Systems and Storage Conference*, ser. SYSTOR '16, Haifa, Israel, 2016.

[74] B. Schroeder and G. A. Gibson, "Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You?" in *Proceedings of the 5th USENIX Conference on File and Storage Technologies*, ser. FAST '07, San Jose, CA, USA, 2007.

[75] L. N. Bairavasundaram, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, G. R. Goodson, and B. Schroeder, "An analysis of data corruption in the storage stack," *Transactions on Storage*, vol. 4, no. 3, pp. 8:1–8:28, 2008.

[76] N. Joukov, A. Traeger, R. Iyer, C. P. Wright, and E. Zadok, "Operating system profiling via latency analysis," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI 2006)*. Seattle, WA: ACM SIGOPS, November 2006, pp. 89–102.