Beyond Poisson: Modeling Inter-Arrival Times of Requests in a Datacenter

Da-Cheng Juan

Electrical and Computer Engineering

Carnegie Mellon University

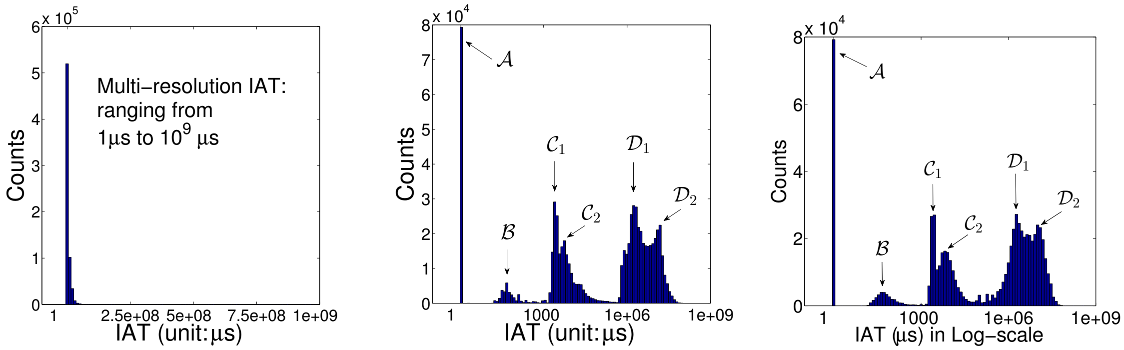
dacheng@cmu.edu

Abstract. How frequently are computer jobs submitted to an industrial-scale datacenter? We investigate the trace that contains job requests and execution collected in one of large-scale industrial datacenters, which spans near half of a Terabyte. In this paper, we discover and explain two surprising patterns with respect to the inter-arrival time (IAT) of job requests: (a) multiple periodicities and (b) multi-level bundling effects. Specifically, we propose a novel generative process, Hierarchical Bundling Model (HIBM), for modeling the data. HIBM is able to mimic multiple components in the distribution of IAT, and to simulate job requests with the same statistical properties as in the real data. We also provide a systematic approach to estimate the parameters of HIBM.

1 Introduction

What are the major characteristics of job inter-arrival process in a datacenter? Could we develop a tool to create synthetic inter-arrivals that match the properties of the empirical data? Understanding the characteristics of job inter-arrivals is the key to design effective scheduling policies to manage massively-integrated and virtually-shared computing resources in a datacenter [11]. Conventionally, during the development of a cloud-based scheduler, job requests are assumed (1) to be submitted independently and (2) to follow a constant rate λ, which results in a simple and elegant model, Poisson process (PP). PP generates independent and identically distributed (i.i.d.) inter-arrival time (IAT) that follows an (negative) exponential distribution [4]. However, in reality, how much does this inter-arrival process deviate from PP?

To demonstrate how the real inter-arrival process deviates from PP, we use Fig. 1 to present the histogram of the IAT for 668,000 jobs submitted and collected in an industrial, large-scale datacenter. The resolution of IAT is 1 microsecond (μs, 10^{-6} sec). As Fig. 1(a) shows, the IATs "seem" to follow an (negative) exponential distribution. However, in logarithmic scale as Fig. 1(b) shows, surprisingly, *four* distinct clusters (denoted as $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$) with either center-or left-skewed shapes can be seen. This distribution (or a mixture of distributions) clearly does not follow an (negative) exponential distribution, which is always right-skewed in logarithmic scale and therefore



(a) Empirical IAT in lin. scale (b) Empirical IAT in log scale (c) Synthetic IAT in log scale

Fig. 1. Deviation from Poisson Process: (a) Histogram of job IAT ($\approx$ 668, 000 jobs) in linear-scale. (b) Same histogram in log-scale. (c) Synthetic IATs from HIBM. In (a), the histogram has limited number of bins to demonstrate IATs of such a fine-resolution, and the marginal distribution may be misidentified as an (negative) exponential distribution. In (b), *four* distinct clusters can be seen: $\mathcal{A}$: 1μs, $\mathcal{B}$: 10-$10^3\mu$s, $\mathcal{C}$: 10^3-$10^5\mu$s, and $\mathcal{D}$: 10^6-$10^9\mu$s. All four clusters are captured by HIBM as shown in (c).

cannot create such shapes. This phenomenon has confirmed that the i.i.d. assumption of PP barely holds since certain job requests may depend on one another. For example, a request of disk-backup may immediately be submitted after a request of Gmail service; this dependency violates the i.i.d assumption and thus invalidates conventional statistical analysis. In this paper we aim at solving the following two problems:

– **P1: Find patterns.** How to characterize this marginal distribution?
– **P2: Pattern-generating mechanism.** What is a possible mechanism that can generate such job inter-arrivals?

This work brings the following two contributions:

– **Pattern discovery.** Two key patterns of job inter-arrivals are provided: (1) multiple periodicities and (2) bundling effects. We show the majority (approximately 78%) of job requests show a regular periodicity with a *log-logistic* noise, a skewed, power-law-like distribution. Furthermore, the submission of a job may depend on the occurrence of its previous job, and we refer to this dependency as the *bundling effect*, since these two associated jobs are considered to belong to the same bundle.
– **Generative model.** We propose HIBM, a "HIerarchical Bundling Model," that is succinct and interpretative. HIBM's mathematical expression is succinct that requires only a handful of parameters to create synthetic job inter-arrivals matching the characteristics of empirical data, as shown in Fig. 1(c). Furthermore, HIBM has the capability to explain the attribution of the four clusters ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$) and the "spikes" ($\mathcal{A}$, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{D}_1$, and $\mathcal{D}_2$) in Fig. 1(b).

The remainder of this paper is organized as follows. Section 2 provides the problem definition, Section 3 details the proposed HIBM, Section 4 provides the discussions and Section 5 surveys the previous work. Finally, Section 6 concludes this paper.

2 Problem Definition

In this work, we use the trace from Google's cluster [12], which is the first publicly available dataset that presents the diversity and dynamic behaviors of real-world service requests, from a large-scale, multi-purpose datacenter. The trace contains the scheduler requests and actions recorded from 29 days (starting at 19:00 EST, on Sunday May 1st, 2011) of activity in a 12,500-machine cluster. Each request submitted by a user forms a *job* and the trace records approximately 668,000 job submissions. For each job, the trace records the timestamps it is submitted, assigned to a machine, or scheduled.

2.1 Terminology and problem formulation

First, we define the terminology used throughout this paper.

Definition 1 (Job type and job instance). *"Job type" represents a certain type of job that can occur once or multiple times, and "job instance" is the actual occurrence of a job request.*

For example, "disk-backup" is a job type that can instantiate several requests; each request (such as "disk-backup at 1:00P.M. on May 2nd") is a job instance.

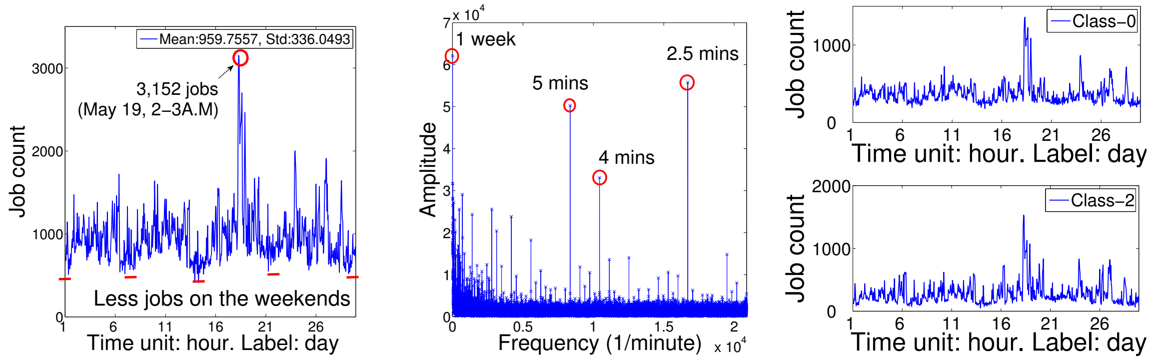
Definition 2 (Job bundle). *"Job bundle" represents the association of two job types – if two job types are in the same job bundle, the IATs of their job instances will be correlated.*

Like the example used in Section 1, two job types "disk-backup" and "Gmail" are functionally-associated, and thus they are considered belonging to the same job bundle. In this case, the inter-arrival of each disk-backup instance will depend on the occurrence of each Gmail instance.

Definition 3 (Job class). *"Job class" represents the priority (or latency sensitiveness) of a job type. In the trace, job class is enumerated as* $\{0, 1, 2, 3\}$ *with a job type of class 3 being the highest priority.*

As mentioned in the Introduction, we have two goals:

- P1: **Find patterns.** Given (1) the job type j, (2) the time stamp of its i^{th} instance (denoted as $t_{j,i}$), and (3) the job class, find the most distinct patterns that are sufficient to characterize the IATs of all job instances in a datacenter.



(a) Job-instance series (b) Single-sided spectrum (c) Class-0 and Class-2

Fig. 2. A burst and periodicities: (a) Job instances per hour. A burst (indicated by the red circle) at May 19th can be observed. (b) Discrete Fourier Transform (DFT) on the job-instance series. The high-amplitude signals correspond to the periods of 1 week, etc. (c) Class-0 (the lowest priority) and class-2 instance series. Notice their similarity (correlation coefficient is 0.94).

– P2: **Pattern-generating mechanism.** Given the patterns found in P1, design a model that can generate IATs that match these characteristics of the empirical data and report the model parameters.

2.2 Dataset exploration

We begin this section by illustrating the number of job instances over time in Fig. 2(a). We collect the time stamp of each job instance when it is first submitted to the datacenter, and then aggregate the total number of job instances within each hour to construct a dataset of one-dimensional time-series. On average, 959.8 job instances are submitted per hour, and in general, less instances are submitted on the weekends whereas more are submitted during weekdays. Interestingly, around 2:00 A.M. on May 19th (Thursday), a burst of 3,152 job instances can be observed, and its amount is approximately three times higher than the amount on typical Thursday midnights.

Discrete Fourier Transform (DFT) is also performed on the job-instance series. Fig. 2(b) provides the amplitude of each discrete frequency, on which we denote four frequencies of high power-spectrum amplitudes: 1-week, 5-min, 4-min and 2.5-min. The reason that the 1-week signal has a high amplitude can be explained by the periodic behavior between weekends and weekdays. Since the other three signals (5-min, 4-min and 2.5-min) also have high amplitudes, they may represent three major periods that exist among the job inter-arrival process. Later in Section 3.1, we characterize the periodicity and show that both 5-min and 4-min periods can be found during the job inter-arrivals.

2.3 Class interdependency

Not all jobs are submitted equal: certain job types have higher priority to be scheduled and executed (class-3, *e.g.*, website services), whereas other jobs do not (class-0, *e.g.* MapReduce workloads) [11].

Observation 1 *The spike $\mathcal{A}$ (1μs) in Fig. 1(b) is attributed to the 1μs IAT between a class-0 and a class-2 instance.*

As shown in Fig. 2(c), the pattern of class-0 job instances (low priority) is highly similar with the pattern of class-2 instances (high priority), in terms of both trend and quantity. As it can be seen that these instances of class-0 and class-2 contribute to the burst on May 19$^{\text{th}}$ observed in Fig. 2(a). Furthermore, the correlation coefficient between class-0 and class-2 instances is 0.94, which makes us think: what is the IAT between a class-0 and a class-2 instance? Surprisingly, this IAT is *exactly 1μs*, which forms the first cluster in Fig. 1(b). This phenomenon immediately piques our interest: how to characterize and attribute the rest of three clusters ($\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$) and the corresponding spikes? The answer lies in the "bundling effect" as we will elaborate in Section 3.

3 HIBM: HIerarchical Bundling Model

In this section, we introduce two major components of HIBM: cross-bundle effects (Section 3.1) and within-bundle effects (Section 3.2). The complete HIBM framework is presented in Section 3.3.

3.1 First component: cross-bundle effect

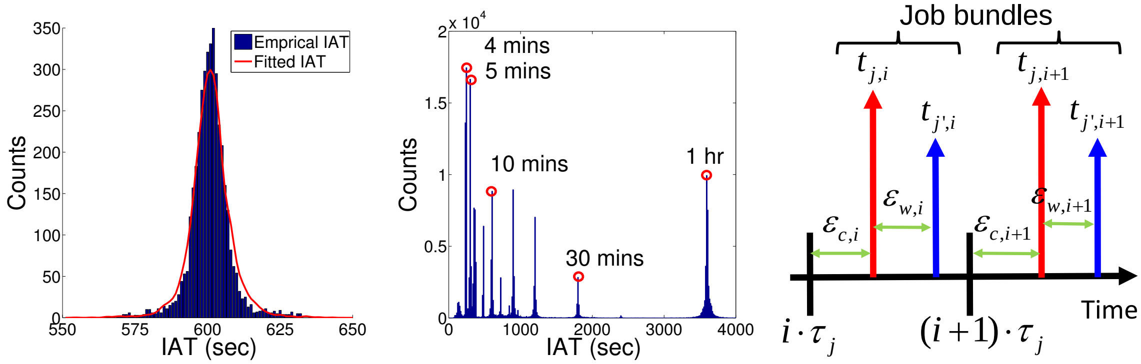
The cross-bundle effects include periodicity and cross-bundle noises. Essentially, each job bundle occurs periodically with some noise (referred as cross-bundle noises in this work). In this section, we first characterize the periodicities and then model the cross-bundle noises.

3.1.1 Multiple periodicities

To characterize the periodicity of each job type, we first calculate the IAT between every two consecutive job instances of that job type as follows:

$$\delta_{j,i} = t_{j,i} - t_{j,i\text{-}1}, \text{ for } i = 1 \dots n_j \tag{1}$$

where $\delta_{j,i}$ is the i^{th} IAT, $t_{j,i}$ represents the occurrence time of the i^{th} instance of job type j, and n_j is the total number of instances of job type j. Fig. 3(a) shows the histogram of such IATs, $\delta_{j,i}$. The histogram is symmetric and has a



(a) IAT of job type j (b) IAT of all job types (c) Cross-and within-bundle

Fig. 3. Multiple periodicities: (a) IAT of job type j and fitted PDF by HIBM. (b) IAT of all job types. (c) Illustration of the cross-bundle noise ($\epsilon_{c,i}$) and the within-bundle noise ($\epsilon_{w,i}$) under the period τ_j.

spike at 600 seconds (10 minutes), which means each instance of job type j arrives approximately every 10 minutes with some noise. Therefore, $t_{j,i}$ can be expressed as:

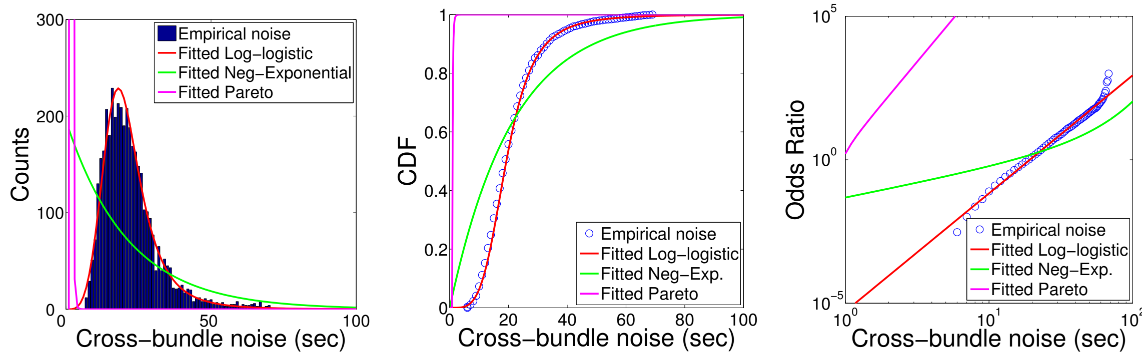
$$t_{j,i} = i \cdot \tau_j + \epsilon_{c,i} \tag{2}$$

where τ_j stands for the period (*e.g.*, 10 minutes in this case) and $\epsilon_{c,i}$ is a random variable representing the "cross-bundle noise." As illustrated in Fig. 3(c), the cross-bundle noise ($\epsilon_{c,i}$) represents the delay of a job bundle from its scheduled time ($i \cdot \tau_j$) and in this example two job types j and j' are in the same bundle. Here, we focus on only the job type j (the red arrows); the within-bundle noise will be elaborated in Section 3.2. In this work, τ_j is estimated by using the median of IATs of job type j; however, what distribution $\epsilon_{c,i}$ follows remains unclear for now.

Observation 2 *Multiple periodicities are observed: 4-min, 5-min, 10-min, 15-min, 20-min, 30-min, and 1-hr.*

One question may arise: is this periodic job type a special case, or do IATs of many job types behave like this? To find the answer, we further collect the IATs from all job types and illustrate them by using Fig. 3(b). For better visualization, only periods smaller than one hour are demonstrated. In Fig. 3(b), multiple periodicities are observed, and the two highest peaks are 4-min and 5-min, which matches the DFT results in Fig. 2(b): the frequencies with high amplitudes are 4-min and 5-min. 4-min is also the smallest period that exists in the trace. We would like to point out that the "10-min peak" in Fig. 3(b) seems sharper than the peak in Fig. 3(a); this is because Fig. 3(b) contains several job types that have the same period (10-min), whereas Fig. 3(a) contains only one such job type.

Now the question is: what random noise $\epsilon_{c,i}$ will create such IAT distribution shown in Fig. 3(a)? Could we use famous "named" distributions, say (negative) exponential or Pareto (power-law), to model this noise?



(a) Cross-bundle noises (b) CDF (c) Odds Ratio

Fig. 4. Modeling cross-bundle noise: (a) PDF, (b) CDF (c) Odds Ratio are demonstrated by using Log-logistic, negative-exponential and Pareto distribution, respectively.

3.1.2 Modeling cross-bundle noise

Among many statistical distributions, we propose to model the cross-bundle noise $\epsilon_{c,i}$ by using Log-logistic distribution (LL), since it is able to model **both the cross-bundle noise and the within-bundle noise** (Section 3.2), leading to the unified expression in HIBM. Also, it provides intuitive explanations for sporadic, large delays. The Log-logistic distribution has a power-law tail and its definition is as follows.

Definition 4 (Log-logistic distribution). *Let T be a non-negative continuous random variable and $T \sim LL(\alpha, \beta)$; the probability density function (PDF) and cumulative density function (CDF) of a Log-logistic distributed variable T are:*

$$PDF(T = t) = f_T(t) = \frac{(\beta/\alpha)(t/\alpha)^{\beta-1}}{(1 + (t/\alpha)^{\beta})^2} \tag{3}$$

$$CDF(T = t) = F_T(t) = \frac{1}{1 + (t/\alpha)^{-\beta}} \tag{4}$$

where $\alpha > 0$ is the scale parameter, and $\beta > 0$ is the shape parameter. The support $t \in [0, \infty)$.

Fig. 4(a) presents the cross-bundle noise $\epsilon_{c,i}$ and three fitted distributions by using Maximum Likelihood Estimate (MLE) [2]. The distribution shows a left-skewed behavior and sporadically, a few job instances suffer from large delays. This phenomenon is difficult to be captured by distributions with tails decaying exponentially fast (*e.g.*, negative-exponential). On the other hand, the Pareto distribution (a power-law probability distribution), which is also a heavy-tail distribution, lacks the flexibility to model a distribution shifted from zero. The goodness-of-fit is tested by using Kolmogorov-Smirnov test [10] with the null hypothesis that the cross-bundle noise is from the fitted

Log-logistic distribution. The resulting P-value is 0.2441, and therefore we retain the null hypothesis under the 95% confidence level and conclude that the cross-bundle noise follows Log-logistic distribution.

To better examine the distribution behavior both in the head and tail, we propose to use the Odds Ratio (OR) function.

Lemma 1 (Odds Ratio). *In logarithmic scale, $OR(t)$ has a linear behavior, with a slope β and an intercept ($-\ln\alpha$), if T follows Log-logistic distribution:*

$$OddsRatio(t) = OR(t) = \frac{F_T(t)}{1 - F_T(t)} = \left(\frac{t}{\alpha}\right)^{\beta} \tag{5}$$

$$\Rightarrow \ln OR(t) = \beta \ln(t) - \ln\alpha \quad \blacksquare$$

As Fig. 4(c) shows, the OR of the cross-bundle noise seems to entirely follow the linear line, which serves as another evidence that its marginal distribution follows a Log-logistic distribution. The Log-logistic distribution presents a modified version of the well known phenomenon – "rich gets richer." We conjecture that this phenomenon can be adapted to explain the cross-bundle noise of periodic job instances – "*those delayed long get delayed longer*." If the submission schedule of a job instance is delayed (or preempted) by other jobs with a higher priority, it is likely that this job instance is going to suffer from being further delayed.

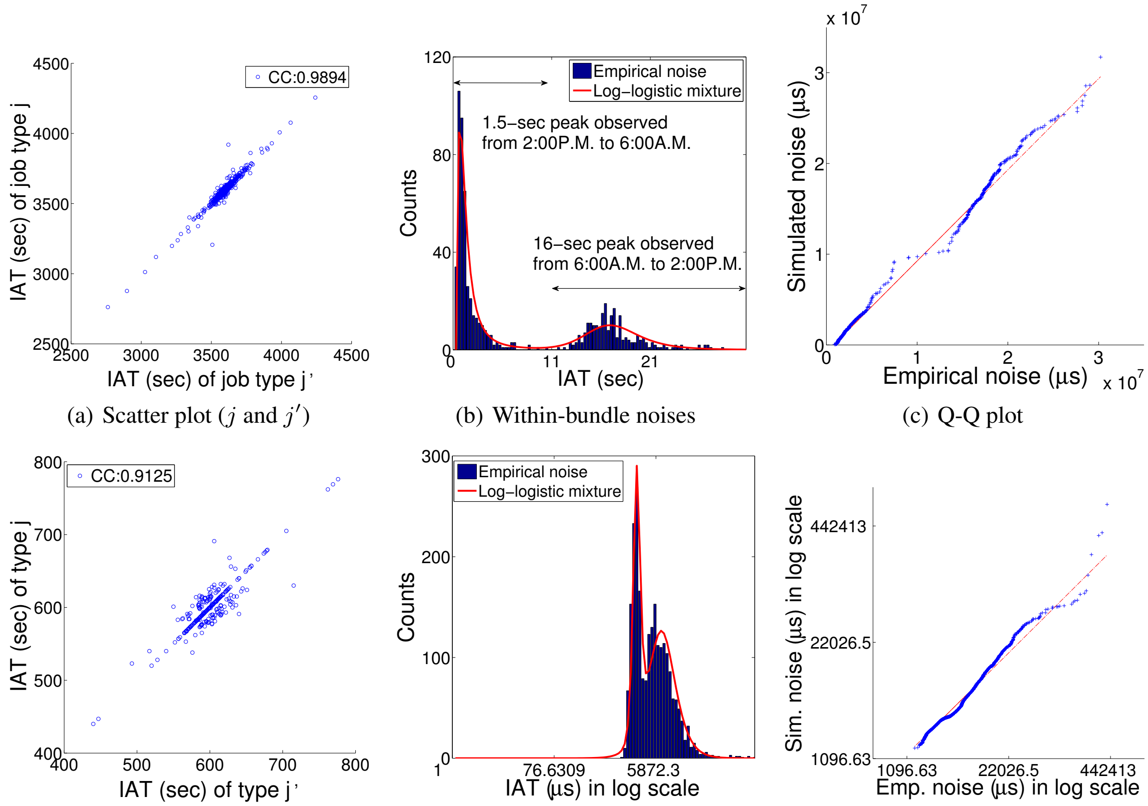
3.2 Second component: within-bundle effect

Here, we elaborate on the within-bundle effects that account for the dependency of two job instances in the same bundle. Next, an algorithm based on "expected occurrence ratio" (EOR) is provided for job bundle detection.

3.2.1 Bundling effect and within-bundle noise

The bundling effect represents the temporal dependency between two job types j and j'. If the instances of two job types (*e.g.*, Gmail and disk-backup, denoted as job type j and j', respectively) are independent from each other, the correlation coefficient of their IATs should be close to zero. However, as Fig. 5(a) shows, IATs of two job types can be highly correlated; the correlation coefficient (CC) is 0.9894. In this context, each $t_{j,i}$ and $t_{j',i}$ must share the same $\epsilon_{c,i}$ due the high correlation. More interestingly, the instances of job type j' always occur after the corresponding instance of j, *i.e.*, $t_{j,i} < t_{j',i}$ as illustrated in Fig. 3(c).

We further examine the IAT between job type j and j', namely, $t_{j',i} - t_{j,i}$, referred as "within-bundle noise" ($\epsilon_{w,i}$). The concept of the within-bundle noise also is illustrated by Fig. 3(c); furthermore, Fig. 5(b) presents a bi-



(a) Scatter plot (j and j') (b) Within-bundle noises (c) Q-Q plot

(d) Scatter plot (j and j') (e) Within-bundle noises (f) Q-Q plot

Fig. 5. HIBM fits real within-bundle noises: (a) IATs of job type j and j' are highly correlated; the correlation coefficient (CC) is 0.9894. Here, both job type j and j' have the period of 1 hour. (b) Within-bundle noise ($\epsilon_{w,i}$) that creates the spikes $\mathcal{D}_1$ and $\mathcal{D}_2$ can be modeled as a mixture of two Log-logistic distributions. (c) Q-Q plot between the empirical $\epsilon_{w,i}$ and the samples drawn from the fitted Log-logistic mixture. (d)(e)(f) demonstrate another $\epsilon_{w,i}$ in millisecond-scale, and have similar explanations. We would like to point out the spikes $\mathcal{C}_1$ and $\mathcal{C}_2$ can be attributed to the within-bundle noise shown in (e).

modal distribution of $\epsilon_{w,i}$: one peak at 1.5-sec observed from 2:00P.M. to 6:00A.M. and the other at 16-sec observed from 6:00A.M. to 2:00P.M.

Observation 3 *The spikes $\mathcal{D}_1$ (1.5sec) and $\mathcal{D}_2$ (16sec) in Fig. 1(b) are attributed to HiBM's within-bundle noise in the scale of seconds.*

A possible explanation is that the submissions of job type j' (class 1, latency-insensitive) are delayed or preempted by other high priority job types during the working hours from 6:00A.M. to 2:00P.M., which creates the second mode (the 16-sec peak). Therefore, we model this bi-modal distribution by using a mixture of two Log-logistic distributions. Fig. 5(c) shows the Q-Q plot between the empirical $\epsilon_{w,i}$ and samples drawn from the fitted Log-logistic mixture. As it can be seen, each quantile of simulated samples matches the empirical $\epsilon_{w,i}$ very well.

A highly similar situation can be observed from another job bundle, shown in Fig. 5(d)(e)(f). Instead of seconds, as Fig. 5(e) shows, $\epsilon_{w,i}$ is bi-modal and in the scale of millisecond.

Observation 4 *The spikes $\mathcal{C}_1$ (3ms) and $\mathcal{C}_2$ (5.5ms) in Fig. 1(b) are attributed to HiBM's within-bundle noise in the scale of milliseconds.*

In this case, $\epsilon_{w,i}$ can also be modeled by a mixture of two Log-logistic distributions as Fig. 5(e)(f) show. For both cases (within-bundle noises in both second-and millisecond-scale), Kolmogorov-Smirnov test is performed; the null hypothesis that $\epsilon_{w,i}$ and the fitted Log-logistic mixture follow the same distribution, is retained under the 95% confidence level. In addition, within-bundle noises are also observed in μs scale, which forms the cluster (and the spike) $\mathcal{B}$ in Fig. 1(b) and can also be modeled by the Log-logistic distribution. This is not shown here due to the space limit. Now we are able to explain and model all the clusters and spikes ($\mathcal{B}$, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{D}_1$ and $\mathcal{D}_2$) with the Log-logistic distribution, leading to the succinctness of HIBM.

Interestingly, even if $\epsilon_{w,i}$ exists, the IATs of job type j and of j' are still highly correlated. The key to create such a phenomenon lies in the hierarchy that cross-bundle noise is always larger than within-bundle noise, $\epsilon_{c,i} > \epsilon_{w,i}$. In the trace, the scale of $\epsilon_{c,i}$ is approximately in the magnitude of minutes, whereas $\epsilon_{w,i}$ is in the magnitude of seconds, milliseconds or even microseconds. Based on this observation, we propose a unified model to describe the IATs of two job types in the same bundle, which serves as the backbone of the proposed HIBM:

$$\begin{cases} t_{j,i} = i \cdot \tau_j + \epsilon_{c,i} \\ t_{j',i} = t_{j,i} + \epsilon_{w,i} = i \cdot \tau_j + \epsilon_{c,i} + \epsilon_{w,i} \end{cases} \tag{6}$$

where $\epsilon_{c,i} \sim LL(\alpha_{c,\kappa}, \beta_{c,\kappa})$, $\epsilon_{w,i} \sim$ a mixture of two LL distributions, expressed as:

$$\epsilon_{w,i} \sim p_{w,\kappa} \cdot LL(\alpha_{w,\kappa}, \beta_{w,\kappa}) + (1 - p_{w,\kappa}) \cdot LL(\alpha_{w',\kappa}, \beta_{w',\kappa}) \tag{7}$$

$p_{w,\kappa} \in [0, 1]$, $\kappa \in \{\mathcal{B}, \mathcal{C}, \mathcal{D}\}$. Given the empirical data, $\alpha_{c,\kappa}$, $\beta_{c,\kappa}$ can be estimated by MLE and $p_{w,\kappa}$, $\alpha_{w,\kappa}$, $\beta_{w,\kappa}$, $\alpha_{w',\kappa}$, $\beta_{w',\kappa}$ can be estimated by Expectation Maximization (EM) [2].

3.2.2 Bundle detection algorithm

After explaining the bundling effect, the next question is how to determine if two certain job types belong to the same job bundle. We ask: given each pair of $t_{j,i}$ and $t_{j',i}$, how do we know these IATs, namely, $|t_{j,i} - t_{j',i}|$, are

Algorithm 1: HIBM Generation

Result: Inter-arrival process of job instances, $t_{j,i}$ for all j and i, given periods τ_j for each job type j, total duration $\mathcal{T}$, $\alpha_{c,\kappa}$, $\beta_{c,\kappa}$, $p_{w,\kappa}$, $\alpha_{w,\kappa}$, $\beta_{w,\kappa}$, $\alpha_{w',\kappa}$, and $\beta_{w',\kappa}$.

initialization: $JS = []$;

for *each* j **do**

  for i = *1 to* $\left\lfloor \frac{\mathcal{T}}{\tau_j} \right\rfloor$ **do**

    if *job type* j *is bundled with job type* j' **then**

      $t_{j,i} = t_{j',i} + \epsilon_{w,i}, \epsilon_{w,i} \sim p_{w,\kappa} \cdot LL(\alpha_{w,\kappa}, \beta_{w,\kappa}) + (1 - p_{w,\kappa}) \cdot LL(\alpha_{w',\kappa}, \beta_{w',\kappa})$;

    else

      $t_{j,i} = i \cdot \tau_j + \epsilon_{c,i}, \epsilon_{c,i} \sim LL(\alpha_{c,\kappa}, \beta_{c,\kappa})$;

    JS = JS appending $t_{j,i}$;

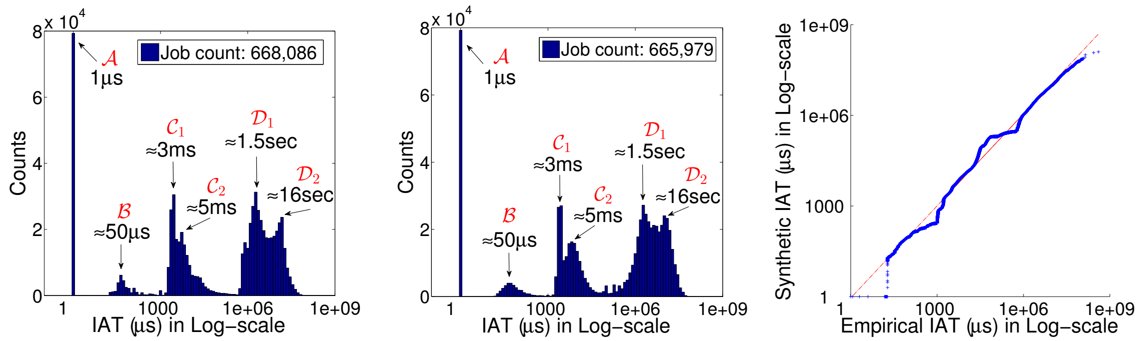
Sort JS in ascending order;

return JS;

caused by within-bundle noises ($\epsilon_{w,i}$), or just coincidentally by a job instance occurring closely to another instance? What if two job types have different periods? To answer these questions, we propose a metric "expected occurrence ratio" (EOR) that compares the empirical counts and the expected counts of within-bundle noises. EOR $\in [0, 1]$ and a high EOR value indicates that job type j and j' are likely to be in the same job bundle. The details of the proposed EOR are in Appendix (Section 7). The intuition is similar to hypothesis testing. We examine the EOR between each pair of job types, and the majority of pairs have EOR less than 0.3, whereas other few pairs have EOR very close to 0.8. In this work, we select an EOR of 0.3 as threshold and therefore two job types are considered unbundled if their EOR is less than 0.3.

3.3 Complete HIBM framework

By assembling the cross-bundle effect (Section 3.1) and the within-bundle effect (Section 3.2) together, we describe here the complete HIBM framework by using Algorithm 1. The inputs to HIBM are user-defined periods, the total duration $\mathcal{T}$, and the parameters of Log-logistic distributions as described in Eq (6). In our case, the periods are set according to the empirical data as shown in Fig. 3(b), the $\mathcal{T}$ is set to one month as mentioned in Section 2.2, and the parameters described in Eq (6) are estimated by MLE and EM. For each job type j, HIBM calculates its total number of instances by $\left\lfloor \frac{\mathcal{T}}{\tau_j} \right\rfloor$. Next, for the i^{th} instance of job type j, there will be two possible cases: (1) $t_{j,i}$ is bundled with $t_{j',i}$ or (2) $t_{j,i}$ is in its own job bundle (not bundled with any other job type). In the first case, $t_{j,i}$ is estimated according to Eq (2), whereas in the second case, $t_{j,i}$ is estimated according to Eq (6). The estimated $t_{j,i}$ is recorded in JS for all j and i. Finally, JS is sorted in ascending order and then HIBM outputs JS as job inter-arrivals.



(a) Empirical IATs (b) Synthetic IATs (c) Q-Q plot

Fig. 6. Comparisons between Synthetic IATs and the empirical IATs: (a) Histogram of empirical IATs in log scale. (b) Histogram of synthetic IATs in log scale. (c) Q-Q plot. The synthetic IATs generated by HIBM match the characteristics of the empirical IATs: the job-instance counts (only 0.3% difference), the four clusters, and all the spikes ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}_1$, $\mathcal{C}_2$ $\mathcal{D}_1$, and $\mathcal{D}_2$). In addition, each quantile of the synthetic IATs matches the corresponding quantile from the empirical data very well.

4 Experimental Results and Discussion

In this section, we first provide the validation of HIBM by using the empirical data, and provide a discussion of HIBM's usage.

4.1 HIBM validation

We validate HIBM by using the empirical data. The comparisons between the synthetic IATs generated by HIBM and empirical IATs are illustrated by Fig. 6. Fig. 6(a)(b) present the histogram of the empirical IATs and the synthetic IATs side by side. As it can be seen, the synthetic IATs match the distinct characteristics of the empirical IATs: the job-instance counts (only 0.3% difference), the four clusters, and all the spikes ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}_1$, $\mathcal{C}_2$ $\mathcal{D}_1$, and $\mathcal{D}_2$). Fig. 6(c) presents the Q-Q plot, from which we can also observe that each quantile of the synthetic IATs matches the corresponding quantile from the empirical data very well.

4.2 HIBM in practice

We begin the discussion with HIBM's succinctness. HIBM requires only a handful of parameters as described in Algorithm 1 to generate job inter-arrivals that match the characteristics from the empirical data, even when the i.i.d. assumption is violated – the submissions of certain instances depend on one another. Therefore, HIBM can be used as a tool to create more realistic job inter-arrivals to design, evaluate, and optimize the cloud-based scheduler of a datacenter.

Also thanks to HIBM's interpretability, we now understand the four distinct clusters observed from the empirical data can be attributed to both class interdependency ($\mathcal{A}$: 1μs) and within-bundle noises ($\mathcal{B}$: 10-$10^3\mu$s, $\mathcal{C}$:10^3-$10^5\mu$s, and $\mathcal{D}$:10^6-$10^9\mu$s). In addition, the 3ms and 5ms spikes ($\mathcal{C}_1$ and $\mathcal{C}_2$) can be attributed to the within-bundle noise shown in Fig. 5(e), and similarly 1.5sec and 16sec spikes ($\mathcal{D}_1$ and $\mathcal{D}_2$) can be attributed to the within-bundle noise shown in Fig. 5(b). Furthermore, the cross-bundle noises in HIBM provides intuitive explanation – "those delayed long get delayed longer" – for the delays occurred on periodic job instances. This explains why sporadically the periodic job instances suffer from large delays.

5 Related Work

Many papers have attempted to model the sequential and streaming data. Leland et al. [9], Wang et al. [14], and Kleinberg et al. [7] have addressed the issues of self-similar and bursty internet traffic. Saveski et al. [13] has adapted active learning to model the web services. Ihler et al. [6] has proposed a time-varying poisson process for adaptive event detection. However, none of these work has addressed the issue of inter-arrivals with both periodicity and bundling effects.

Regarding to the Log-logistic distribution, it has been developed and used for survival analysis [8, 1]. Recently, prior work has demonstrated its use in modeling the duration of telecommunication [3] and software reliability [5]. To the best of our knowledge, this is the first work to use Log-logistic distributions to model the delays of job inter-arrivals in a datacenter.

6 Conclusion

In this work, we investigate and analyze the inter-arrivals of job requests in an industrial, large-scale datacenter. Our paper has two contributions:

- **Pattern discovery.** We discover two key patterns of job inter-arrivals: (a) multiple periodicities and (b) bundling effects. In addition, we propose to use Log-logistic distributions to model both cross-bundle and within-bundle noises.
- **Generative model.** We propose HIBM, a succinct and interpretative model. HIBM requires only a handful of parameters to generate job inter-arrivals mimicking the empirical data. In addition, HIBM also attributes the four distinct clusters and the corresponding spikes to both within-bundle noises and class interdependency, and provides intuitive explanation "those delayed long get delayed longer" to the cross-bundle noises of periodic job types.

References

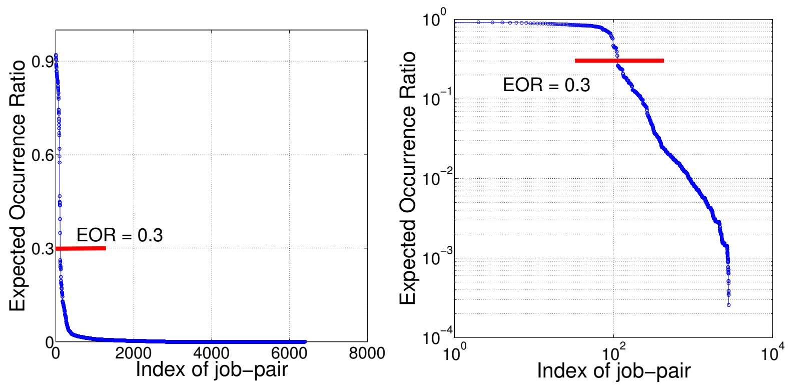
1. Bennett, S.: Log-logistic regression models for survival data. Applied Statistics pp. 165–171 (1983)
2. Casella, G., Berger, R.L.: Statistical inference, vol. 70. Duxbury Press Belmont, CA (1990)
3. De Melo, P.O.V., Akoglu, L., Faloutsos, C., Loureiro, A.A.: Surprising patterns for the call duration distribution of mobile phone users. In: ECML PKDD, pp. 354–369. Springer (2010)
4. Fischer, W., Meier-Hellstern, K.: The markov-modulated poisson process (mmpp) cookbook. Performance Evaluation 18(2), 149–171 (1993)
5. Gokhale, S.S., Trivedi, K.S.: Log-logistic software reliability growth model. In: HASE. pp. 34–41. IEEE (1998)
6. Ihler, A., Hutchins, J., Smyth, P.: Adaptive event detection with time-varying poisson processes. In: KDD. pp. 207–216. ACM (2006)
7. Kleinberg, J.: Bursty and hierarchical structure in streams. Data Mining and Knowledge Discovery 7(4), 373–397 (2003)
8. Lawless, J.F.: Statistical models and methods for lifetime data, vol. 362. John Wiley & Sons (2011)
9. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of ethernet traffic. In: ACM SIGCOMM Computer Communication Review. vol. 23, pp. 183–193. ACM (1993)
10. Massey Jr, F.J.: The kolmogorov-smirnov test for goodness of fit. JASA 46(253), 68–78 (1951)
11. Reiss, C., Tumanov, A., Ganger, G.R., Katz, R.H., Kozuch, M.A.: Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In: SOCC. p. 7. ACM (2012)
12. Reiss, C., Wilkes, J., Hellerstein, J.L.: Google cluster-usage traces. In: https://code.google.com/p/googleclusterdata/wiki/TraceVersion2. Google (2012)
13. Saveski, M., Grčar, M.: Web services for stream mining: A stream-based active learning use case. ECML PKDD 2011 p. 36 (2011)
14. Wang, M., Madhyastha, T., Chan, N.H., Papadimitriou, S., Faloutsos, C.: Data mining meets performance evaluation: Fast algorithms for modeling bursty traffic. In: ICDE. pp. 507–516. IEEE (2002)

7 Appendix

Here, we elaborate on using the expected occurrence ratio (EOR) to determine if two job types are bundled. The expected occurrence ratio (EOR) of job type j and j' can be calculated as follows:

$$EOR(j,j') = \mathcal{N}_\kappa \cdot \left(\frac{\mathcal{T}}{LCM(\tau_j, \tau_{j'})} \cdot \rho_j \cdot \rho_{j'} \right)^{-1} \tag{8}$$

where $\mathcal{N}_\kappa$ represents the number of the IATs occurred in the range of the cluster $\kappa \in \{\mathcal{B}, \mathcal{C}, \mathcal{D}\}$ in Fig. 1(b), $\mathcal{T}$ is the total duration, $LCM(\tau_j, \tau_{j'})$ is the Least Common Multiple (LCM) between two periods τ_j and $\tau_{j'}$, finally ρ_j



(a) **EOR in linear scale** (b) **EOR in log scale**

Fig. 7. Expected occurrence ratio (EOR). (a) EOR in linear scale. The values of EOR for each pair of job types are sorted from high to low. (b) EOR in log scale. Only few job pairs have high EOR values (around 0.8) and the majority of job types are in "their own" job bundle, *i.e.*, they are not bundled with others. 0.3 of EOR is selected the threshold to determine two job types are in the same job bundle or not.

and $\rho_{j'}$ are the missing rates of job type j and j', respectively. The intuition of EOR is similar to hypothesis testing: to compare the empirical count of IATs in the range of a certain cluster with its expected count. The expected count, $\frac{\mathcal{T}}{LCM(\tau_j,\tau_{j'})} \cdot \rho_j \cdot \rho_{j'}$, is calculated under the assumption that job type j and j' are bundled. Therefore, if j and j' are actually bundled, the value of $\mathcal{N}_\kappa$ will be very close to the expected count, resulting in an EOR ≈ 1. On the other hand, if EOR is very close to 0, j and j' are considered unbundled, since the observed $\mathcal{N}_\kappa$ just occurred by coincidence.

We examine the EOR between each pair of job types and illustrate the results by using Fig. 7(a)(b) in linear and log scales, respectively. For better visualization, only the first 6,400 pairs are included, and the values of EOR are sorted from high to low. As it can be seen, the majority of pairs have EOR less than 0.3 , whereas other few pairs have EOR very close to 0.8. In this work, we select an EOR of 0.3 as threshold and two job types are considered unbundled if their EOR is less than 0.3.