An EM-based technique for approximating long-tailed data sets with PH distributions ⋆

Alma Riska

Seagate Research, 1251 Waterfront Place, Pittsburgh PA 15222, USA
e-mail: alma.riska@seagate.com

Vesselin Diev Evgenia Smirni

Department of Computer Science, College of William and Mary, Williamsburg, VA 23187-8795, USA e-mail: {vdiev,esmirni}@cs.wm.edu

Abstract

We propose a new technique for fitting long-tailed data sets into phase-type (PH) distributions. This technique fits data sets with non-monotone densities into a mixture of Erlang and hyperexponential distributions, and data sets with completely monotone densities into hyperexponential distributions. The method partitions the data set in a divide and conquer fashion and uses the Expectation-Maximization (EM) algorithm to fit the data of each partition into a hyperexponential distribution. The fits of all partitions are combined to generate the final fit of the entire data set. The proposed method is accurate and computationally efficient. Furthermore, it allows one to apply existing analytic tools to analyze the behavior of queueing systems with long-tailed arrival and/or service processes via tractable models.

1 Introduction

One of the challenges in the process of designing, capacity planning, and resource allocation of communication systems is the complexity of their workload, as well as the complexity of models that can be used for their analysis. Internet-related workloads are characterized by high variability, long-range dependence, or both [1,12,17]. These characteristics are captured well by stochastic models such as the Markovian Arrival Process (MAP) and the

⋆ This work has been supported by National Science Foundation under grants EIA-9974992, CCR-0098278, and ACI-0090221.

Batch Markovian Process (BMAP) [2,6,17], and an entire analytic methodology exists for their analysis [14,13,10]. A special and widely used class of MAPs is the PH distribution. PH distributions can capture both high and low variabilities in stochastic processes, but in contrast to the wider MAP class, they do not capture long range dependence or self-similar behavior. In this paper, we focus on fitting Internet-related workloads into PH distributions, because such distributions are well-suited for analytical studies.

PH distributions are characterized by the number of exponential phases and the interaction among them [10]. The latter determines special cases of PH distributions such as the Erlang, Coxian, and hyperexponential distributions [9]. The fitting techniques available for PH distributions are either based on moment matching or on maximum likelihood estimators (MLEs). Moment matching techniques are computationally efficient, but apply to somewhat more restrictive models, i.e., models that consists of a few exponential phases only [7]. Among the techniques that are based on MLEs, we distinguish the Expectation-Maximization (EM) algorithm for fitting both data sets and distribution functions into general PH distributions [3], MLAPH [4] for fitting continuous distribution functions into acyclic PH distributions, and a numerical optimization method for fitting heavy-tailed distribution functions into Coxian distributions [6]. The Feldmann-Whitt algorithm [5] is a heuristic-based approach to fit heavy-tailed distribution functions, such as Weibull or Pareto, into hyperexponential distributions.

Accuracy and *efficiency* are equally important in our approach for capturing the long-tailed behavior of data sets, because we are driven by the need for *on-line* performance modeling tools. From this perspective, moment matching methods [7] that apply to simple models fail to accurately capture the long-tailed behavior, while numerical optimization-based techniques, although they provide good fitting accuracy, may be subject to long computation times [6]. Fitting methods for distribution functions instead of data sets [5,6], despite the computational efficiency that some of them provide [5], introduce additional errors in the fitting procedure since one additional step is required to first fit the data set into a distribution function such as Pareto or Weibull. The EM algorithm [3], one of the most popular MLE-based approaches, fits either data sets or distribution functions into general PH distributions making EM superior to other fitting methods. In general, the accuracy and the computational complexity of the EM algorithm increase with the number of exponential phases, with the notable exception of an efficient EM-based algorithm for fitting into hyperexponential distributions [8]. Finally, the EM algorithm may fail to accurately capture the tail of the distribution, because it searches for the "global" optimal solution [3].

In this paper, we propose a new technique for fitting data sets into PH distributions. The new technique strives for both accuracy and efficiency by applying

the popular EM algorithm in a *divide-and-conquer* fashion. The divide-and-conquer EM algorithm benefits from the strengths of the EM algorithm and at the same time reduces the effects of its known weaknesses.

This paper is organized as follows. We motivate our work in Section 2. We describe the PH distributions and existing fitting techniques in Section 3. In Section 4, we describe the proposed fitting method. We present experimental results using the proposed method on five different data sets in Section 5. Section 6 discusses the accuracy of the new approach from the queueing systems perspective. Section 7 reports on the efficiency of the proposed method. A refined version of the fitting method is described in Section 8. Concluding remarks and future directions are outlined in Section 9.

2 Motivation

There is abundance of evidence [1,12] that the workload in Internet-related systems is long-tailed, i.e., highly variable. This characteristic increases the complexity of the systems that operate under them (e.g., Web servers). To motivate our work, we first focus on the performance effects of high variability in the service process of an elementary M/G/1 queue. We analyze the queue behavior by fixing its mean service time but vary the coefficient of variation of service time within a certain range. We solve the system using the well-known matrix analytic methodology [13,14,16].

Figure 1(a) illustrates the effects on the anticipated mean slowdown of the variability in the service process, as a function of the arrival rates. As the arrival rate increases, the average slowdown for the queue with the highest variability in the service process increases much faster than for the queues with lower variabilities. Similar behavior is also observed in Figure 1(b), where we plot the queue length as a function of the variability in the service process. Observe that for high arrival rates the effect of the highly variable service process on performance is more severe comparing to lower arrival rates. This simple yet striking example clearly shows that capturing workload variability is essential for accurate performance analysis.

Long-tailed behavior in a data set is commonly captured by subexponential distributions including Lognormal, Weibull, or Pareto. The analysis of queueing systems that admit these distributions as either the arrival process, or the service process, or both, is complex and most of the times intractable. PH distributions (along with their special cases of Coxian and hyperexponential distributions) offer an attractive alternative that can capture high variability yet in an analytically tractable manner. Their only "drawback" is that they require more parameters than Lognormal or Weibull distributions, making

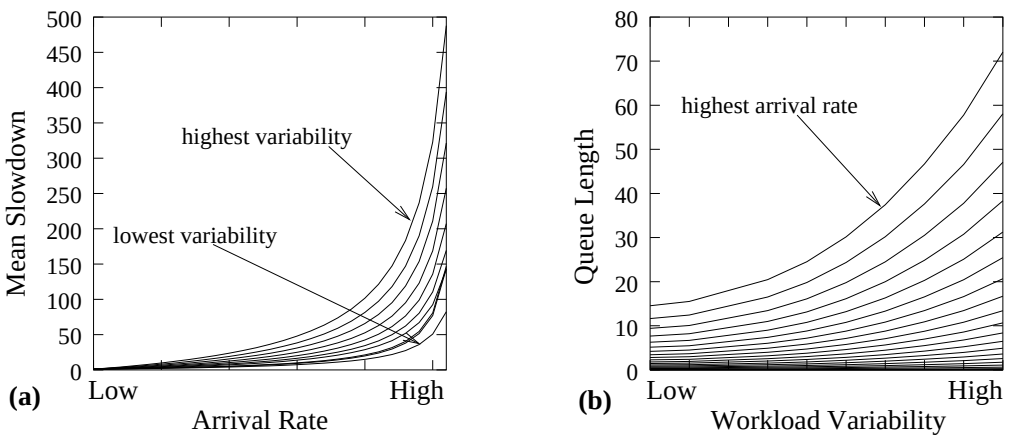

Fig. 1. Performance effects of high variability in the service process of a simple queueing system.

their estimation procedures more complex. To avoid the complicated fitting algorithms and yet benefit from their tractability, Coxian or hyperexponential distributions with two phases only are often used. In such cases, there are only few, i.e., 2 or 3, parameters to estimate, and that is easily done by matching the first and second moments only [9]. Below, we show that moment matching techniques like the one that fits data sets into a two-phase hyperexponential, provide models that are not accurate enough for the complex workloads measured in real systems.

To fit an empirical distribution with a certain mean and CV into a two-phase hyperexponential distribution, which is a mixture of two exponentials only, we need to calculate the parameters p ($0 < p < 1$), λ_1, and λ_2, given that the probability density function of a two-phase hyperexponential is given by:

$$h_2(x) = p\lambda_1 \, \mathrm{e}^{-\lambda_1 x} + (1-p)\lambda_2 \, \mathrm{e}^{-\lambda_2 x}, \qquad x \geq 0, \quad \lambda_1, \lambda_2 > 0.$$

One can match the mean m_d and the coefficient of variation cv_d of the empirical distribution with the mean m_h and the coefficient of variation cv_h of the two-phase hyperexponential distribution using the following formulas [9, page 143]:

$$m_d = m_h = \frac{p}{\lambda_1} + \frac{1-p}{\lambda_2} \quad \text{and} \quad cv_d^2 = cv_h^2 = \frac{\frac{2p}{\lambda_1^2} + \frac{2(1-p)}{\lambda_2^2}}{m_d^2} - 1,$$

resulting in a classic case of a system of two equations and three unknowns. This system may be solved by guessing one of its unknowns (in this case the easiest one to guess is p, since $0 < p < 1$). To illustrate that such a fitting does not capture the complex behavior of the queueing systems that operate under a long-tailed service process, we fit a data trace into a two-phase hyperexponential using the above method. The data set consists of the sizes of the requested files processed by a Web server (one day of server logs from the

1998 World Soccer Cup [1]). Assuming that the requests arrive to the system according to a Poisson process and the service process is determined by the sizes of the requested files, we model the Web server as an $M/H_2/1$ queue. To assess the accuracy of the results obtained from the analytic solution of this queueing model, which is solved using the matrix-geometric method [13], we simulate the same single server queue using an identical arrival process, but now its service process is driven by the original trace data. We present our findings in Figure 2. Figure 2(a) shows the average queue length in the server as function of the arrival rate. Note that for different values of p that result in different fittings, the same average queue length is computed and is in good agreement with simulation results. Consequently, simple models are sufficient for simple measures such as the average queue length. Nonetheless, if we must go a step further and obtain the queue length distribution, simple models fail. Figures 2(b) and 2(c) present the queue length distribution of the body and tail of the queue, respectively. None of the queue length distributions of the various analytic models follows the shape of the queue length distribution obtained by the trace-driven simulation.

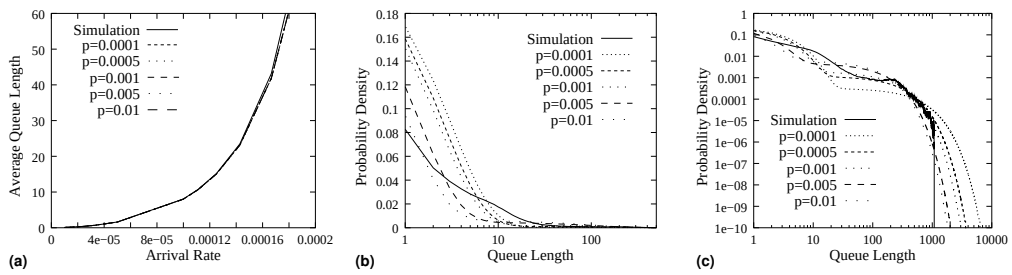

Fig. 2. Average queue, queue length distribution (body and tail) for the $M/H_2/1$ queueing models and the trace-driven simulation. The system is analyzed at a 80% utilization level.

With the examples presented in this section, we emphasize the complexity in the behavior of systems that operate under a highly variable service process and the importance of capturing it correctly for accurate performance analysis results. In the following sections, we present fitting techniques that capture correctly the behavior of the long-tailed trace and consequently can be used to model accurately queueing systems that admit such data.

3 Related Work

In this section, we present a brief overview of the PH distribution and its special cases. We also describe the EM and the Feldman-Whitt algorithms as

[1] Available at http://ita.ee.lbl.gov/

commonly used techniques to fit data sets or distribution functions into PH distributions.

3.1 Phase-type (PH) Distributions

PH distributions are based on the method of stages technique, introduced by Erlang and generalized by Neuts [13]. PH distributions are "general" mixtures of exponentials and represent how basic results on the simple and widely-used exponential distribution are extended to capture complex characteristics while maintaining computational tractability. A PH distribution is represented by a vector $\boldsymbol{\pi}$ and a matrix $\mathbf{T}$ with dimensions $1 \times ph$ and $ph \times ph$ respectively. The value of ph represents the number of phases or stages in the PH distribution. The basic distributional characteristics of a PH distribution are

- the cumulative distribution function: $F(x) = 1 - \boldsymbol{\pi} e^{\mathbf{T}x} \mathbf{e}$
- the density function: $f(x) = \boldsymbol{\pi} e^{\mathbf{T}x} (-\mathbf{T} \cdot \mathbf{e})$
- the n$^{\text{th}}$ moment: $m_n = (-1)^n \cdot n! \boldsymbol{\pi} \mathbf{T}^{-n} \mathbf{e}$,

where $\mathbf{e}$ is a column vector of ones with the right dimension.

Obviously, the number of parameters that define a PH distribution is $ph \times (ph + 1)$, however the cases of PH distributions that are used in practice have the majority of the values in $\boldsymbol{\pi}$ and $\mathbf{T}$ equal to zero. Most commonly-used cases of PH distribution are hypoexponential, Coxian [2], and hyperexponential distributions whose $\boldsymbol{\pi}$ and $\mathbf{T}$ are defined in the following table assuming $ph = 4$:

	Erlang	Coxian	Hyperexponential
$\boldsymbol{\pi}$	$[1,\ 0,\ 0,\ 0]$	$[1,\ 0,\ 0,\ 0]$	$[p_1,\ p_2,\ p_3,\ p_4]$
$\mathbf{T}$	$\begin{bmatrix} -\lambda_1 & \lambda_1 & 0 & 0 \\ 0 & -\lambda_2 & \lambda_2 & 0 \\ 0 & 0 & -\lambda_3 & \lambda_3 \\ 0 & 0 & 0 & -\lambda_4 \end{bmatrix}$	$\begin{bmatrix} -\lambda_1 & \lambda_1^1 & 0 & 0 \\ 0 & -\lambda_2 & \lambda_2^1 & 0 \\ 0 & 0 & -\lambda_3 & \lambda_3^1 \\ 0 & 0 & 0 & -\lambda_4 \end{bmatrix}$	$\begin{bmatrix} -\lambda_1 & 0 & 0 & 0 \\ 0 & -\lambda_2 & 0 & 0 \\ 0 & 0 & -\lambda_3 & 0 \\ 0 & 0 & 0 & -\lambda_4 \end{bmatrix}$

Table 1
Structure of $\boldsymbol{\pi}$ and $\mathbf{T}$ for the Erlang, Coxian, and hyperexponential distributions.

[2] A given exponential phase i, $0 < i < ph$, of a Coxian distribution is associated with two rates: λ_i^1 for reaching the next exponential phase and λ_i^2 for reaching the absorbing state of the underlying Markov chain [10]. The total rate of leaving phase i is $\lambda_i = \lambda_i^1 + \lambda_i^2$.

PH distributions capture a wide range of statistical characteristics including high variability. Note that PH distributions do not capture long-range dependence or self similar behavior. There is another set of processes known as Markovian Arrival Processes (MAPs) [14] that are still based on the method of stages and capture long-range dependence in a data set. PH distributions are a special case of MAPs.

3.2 EM Algorithm

The Expectation-Maximization (EM) algorithm is an iterative method that aims to maximize the log-likelihood function of a data set. In the case of PH distributions, the likelihood function is given by the formula

$$L(\boldsymbol{\pi},\ \mathbf{T};\ \mathbf{y}) = \prod_{i=1}^{n} \boldsymbol{\pi} e^{\mathbf{T} y_i}(-\mathbf{T}\mathbf{e}),$$

where $\mathbf{y}$ represents the data set to be fitted into the PH distribution. The EM algorithm starts from an initial guess for the parameters $(\boldsymbol{\pi}^{(0)},\ \mathbf{T}^{(0)})$. Each iteration j generates a new estimate $(\boldsymbol{\pi}^{(j)},\ \mathbf{T}^{(j)})$, such that the value of the likelihood function with the new estimate is larger than the previous ones

$$L(\boldsymbol{\pi}^{(j)},\ \mathbf{T}^{(j)};\ \mathbf{y}) \leq L(\boldsymbol{\pi}^{(j+1)},\ \mathbf{T}^{(j+1)};\ \mathbf{y}).$$

The sequence of the estimates $(\boldsymbol{\pi}^{(0)},\ \mathbf{T}^{(0)})$, $(\boldsymbol{\pi}^{(1)},\ \mathbf{T}^{(1)}), ...,$ $(\boldsymbol{\pi}^{(N)},\ \mathbf{T}^{(N)})$ should eventually converge after N steps to the stationary point $(\hat{\boldsymbol{\pi}},\ \hat{\mathbf{T}})$ of the likelihood. This stationary point is (hopefully) the one that maximizes the likelihood, but it is not guaranteed that it is the optimal one, as the algorithm can terminate upon reaching an insignificant local maximum or saddle point. The best accuracy in the fittings is achieved when different initial values are tried. More details on the EM algorithm for fitting data sets into PH distributions as well as fitting any non-negative continuous distribution into a PH distribution can be found in [3]. An efficient EM-based algorithm for fitting data sets into hyperexponential distributions is developed in [8].

EM is one of the best algorithms to approximate data sets or continuous distributions by other continuous distributions. In the case of fitting data sets with long-tails into PH distributions, it may be possible not to capture the tail correctly since the procedure searches for a global maximum [6]. As an iterative procedure, its computation complexity depends on the size of the data set and the number of unique entries, as well as the number ph of phases for the fitted PH distribution.

3.3 Feldmann-Whitt (FW) Algorithm

The Feldmann-Whitt (FW) algorithm [5] is a recursive heuristic that fits completely monotone non-negative continuous distributions into hyperexponential distributions. The idea behind the algorithm is to first fit the rightmost portion of the tail into a single exponential distribution determining its parameter and the associated probability (see Table 1 for the definition of hyperexponential distribution). Once the first pair of parameters is known, the weight of the fitted portion is "subtracted" from the distribution and again the rightmost part of the tail of the remaining distribution is fitted into another exponential distribution by computing the same set of parameters. The algorithm continues until the parameters of all phases of the hyperexponential distribution are computed.

This is an elegant and accurate fitting approach for Pareto and Weibull distributions, that are characterized by high variability and completely monotone density functions. Characterization of Internet workloads [1] suggests that hybrid distributions with Lognormal body and power tail give better fits. Since such hybrid distributions do not have completely monotone densities, FW might not yield accurate fits. Another drawback of the FW algorithm is that it works only with distribution functions rather than data sets. Hence an additional step is required in the fitting procedure, i.e., fitting the data set into a Weibull or Pareto distribution. This step introduces additional errors to the final result and may increase the execution time of the algorithm.

4 Fitting Method

In this section, we devise a new technique for fitting highly variable data sets into hyperexponential distributions. The proposed method applies the EM algorithm in a divide and conquer fashion over the initial data set. We call our approach D&C-EM (Divide-and-Conquer-EM).

The high level idea of the proposed method (see Figure 3) is based on the observation that for data sets that exhibit long-tailed behavior, it may be beneficial to partition their entire range of values so as to ensure that each partition exhibits significantly reduced variability in comparison to the variability of the entire data set. The data set of each partition is then fitted into a hyperexponential distribution using the EM algorithm [15] and the final fit for the entire data set is generated by combining together the fitting results for all partitions.

The divide and conquer approach increases the accuracy of the EM algorithm

because the portion of the data set belonging to the tail of its continuous data histogram (CDH) [11] could possibly fit in one or more partitions, reducing the possibility that the EM algorithm does not capture it correctly while searching for the global optimal solution. The approach is efficient because each partition has less data entries and less variability than the entire data set, facilitating an accurate fit in a few phases only.

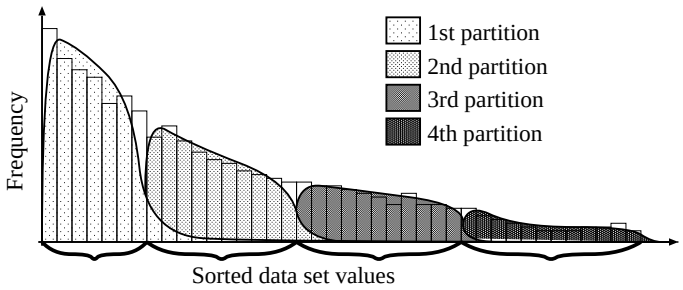

Fig. 3. Splitting of the continuous data histogram (CDH).

We start by building the CDH with only one pass through the data. One additional pass through the CDH is required to define the "splits". Since we are interested to split the CDH such that each partition has reduced variability, we use the coefficient of variation (CV) to determine the partition boundaries. For each partition, we accumulate bins until the accumulated coefficient of variation for that partition, CV_{Acc}, is larger than a threshold, CV_{max}. The value of CV_{max} determines the number of partitions for a given data set. We select CV_{max} to be between 1.2 and 1.5, i.e., slightly higher than the CV of the exponential distribution, in order to fit each partition into a hyperexponential distribution with few phases only using EM. The EM algorithm requires as input only the number of phases, ph, and the actual data. In our experiments, we use $ph = 4$ since it captures well the properties of data sets with CVs as high as CV_{Max} and strikes a good balance between efficiency and accuracy.

We generate the final result by combining together the weight w_i of each partition to the entire CDH with its respective fitted hyperexponential distribution. Since the total number of phases is obtained as the sum of phases over all partitions, the final number of phases from the fitting is not known until the fitting procedure is complete. From the experiments, we observe that data sets with high CV (i.e., higher than 5.0) may contain as many as 6 partitions, while data sets with not-as-high CV (i.e., less than 5.0) may contain as low as 3 partitions. Figure 4 summarizes the D&C-EM algorithm.

Note that by fitting each partition into a hyperexponential distribution, we essentially restrict the applicability of the fitting procedure to completely monotone datasets. We return to this issue in Section 8 where we present an algorithmic refinement that overcomes this deficiency.

1. Build CDH from the data set.
2. while (there are still CDH bins to be considered)
  a. include data of current bin into current partition i
  b. update CV_{Acc}
  c. if $(CV_{Acc} > CV_{max})$
    Use EM to fit partition i into ph phases
    Obtain p_j^i and λ_j^i, $1 \leq j \leq ph$
    Compute weight w_i for partition i
    $CV_{Acc} = 0$.
3. if $(CV_{last\ partition} < CV_{max})$
  a. Merge last two partitions and perform step **2c**
4. Generate final result
  for i from 1 to # of partitions
    for j from 1 to ph
      $p_j^i = p_j^i \cdot w_i$

Fig. 4. The D&C-EM fitting algorithm.

5 Experimental results

In this section, we present results obtained by fitting five different data sets into hyperexponential distributions using the D&C-EM algorithm. We first describe the characteristics of the selected data sets.

5.1 Workload

We have selected five highly variable data sets to test our approach. The first data set (indicated as "trace 1") is a trace from the 1998 World Soccer Cup Web site [3]. It contains the sizes of the requested files received at the Web site during one day. The other four traces are synthetically generated from analytic models that closely approximate Web server traffic [1]. Traces 2 and 3 are generated from Lognormal distributions with shape parameters 1.85 and 1.5, respectively, and the same scale parameter 7.0. Traces 4 and 5 are generated from Weibull distributions with shape parameters 0.25 and 0.35, respectively, and the same scale parameter 9.2. The statistical characteristics of these data sets are shown in Table 2. The number of entries and the number of unique entries for each data set are important for the performance of the D&C-EM as the computation time of the EM algorithm depends on these parameters [15]. Observe that the real trace has less unique entries than the

[3] Available at http://ita.ee.lbl.gov/

Trace	Entries	Unique	Mean	CV
1	16045065	12122	4407.81	7.28
2	25000	25000	6358.23	5.87
3	25000	25000	3459.86	3.13
4	25000	22969	227.27	7.36
5	25000	24298	47.50	3.86

Table 2
Statistical characteristics of the data sets.

synthetically generated ones.

5.2 Statistical Analysis

The size of the data sets precludes using goodness-of-fit tests such as the Kolmogorov-Smirnov and χ^2 tests [11]. Therefore, we evaluate the accuracy of D&C-EM by checking the matching of statistical properties such as the moments and the median, and by plotting PDFs, CDFs, and CCDFs (Complimentary Cumulative Distribution or Survival function).

Table 3 illustrates the means and the CVs of the original data sets, plus various hyperexponential fittings using the EM, FW, and D&C-EM algorithms. In the table, "x ph" means that the EM or the FW algorithms fit the entire data set into a hyperexponential with x phases. Observe that the D&C-EM models match the mean of the traces with maximum error of 4 percent. The coefficient of variation is more difficult to match, since it is obtained using both the first and the second moments. Nevertheless, D&C-EM models match the CV with a maximum error of 20 percent (trace 2). The EM algorithm alone could not generate results for traces 4 and 5 within reasonable amount of computation time (less than a week, running in a Pentium III, 700 MHz, and 1 GB memory machine). Since traces 4 and 5 are synthetically generated from a Weibull distribution, we fit the same distribution functions into hyperexponential distributions using the FW algorithm. The results of Table 3 show that the divide and conquer approach captures better the statistical characteristics of the data sets, when compared to the EM and the FW algorithms.

D&C-EM generates fits that match better even the higher moments of the data sets. We present in Table 4 the relative errors of the fitted third moments from the empirical third moment of traces 1 through 5 (we omit the absolute values because they are too large and not easy to read).

	Data	EM 8 ph	FW 15 ph	D&C-EM
		Trace 1		
Mean	4407.81	4402.35	n/a	4393.56
CV	7.28	3.44	n/a	7.86
Median	938.00	961.51	n/a	950.59
		Trace 2		
Mean	6358.23	6196.27	n/a	6164.50
CV	5.87	4.13	n/a	5.13
Median	1082.91	1063.12	n/a	1061.25
		Trace 3		
Mean	3459.86	3425.72	n/a	3391.06
CV	3.13	2.69	n/a	2.82
Median	1085.32	1084.13	n/a	1086.59
		Trace 4		
Mean	227.27	n/a	221.34	220.21
CV	7.36	n/a	8.12	7.14
Median	2.20	n/a	2.06	2.27
		Trace 5		
Mean	47.50	n/a	46.40	46.61
CV	3.86	n/a	3.95	3.87
Median	3.32	n/a	3.14	3.35

Table 3
Statistical evaluation of the fittings.

	Trace 1	Trace 2	Trace 3	Trace 4	Trace 5
EM 8 ph	75%	91%	70%	n/a	n/a
FW 15 ph	n/a	n/a	n/a	106%	27%
D&C-EM	60%	60%	66%	25%	11%

Table 4
Relative error of the third moment.

Figure 5 plots the PDF, CDF, and CCDF [4] of each trace and the D&C-EM,

[4] Throughout the paper we indicate a data set's empirical distribution, cumulative empirical distribution, and complementary cumulative empirical distribution by PDF, CDF, and CCDF, respectively. This notation, although technically inaccu-

EM, and FW models. The PDF plots for all 5 data sets are shown in the first column of graphs in Figure 5 (note the logscale of the x-axis). D&C-EM offers good fits for all traces with the exception of traces 2 and 3 where we observe that both D&C-EM and EM do not match well the original PDF at the leftmost part of the data-set range. This is a direct outcome of the fact that the algorithm presented in Figure 9 is appropriate for completely monotone PDFs only as the resulting distribution is a hyperexponential one. Trace 1, derived directly from measured data, is even more heavily jagged, making matching of its PDF even more challenging that the other traces.

The CDF plots for all traces (middle column of graphs in Figures 5) illustrate that the D&C-EM provides a good match for the body of the distribution. We investigate the D&C-EM's accuracy on fitting the distribution tail by plotting the CCDFs of the data sets and the respective fitted models (third column of graphs in Figure 5). Observe that D&C-EM models closely match the shape of the data sets distribution tail, which is directly related to their variability. We conclude this section by noting that, consistent with the discussion in [5], the hyperexponential whose PDF is completely monotone is a better fit for traces 4 and 5 with completely monotone CDHs than for traces 1, 2, and 3, whose CDH is not completely monotone.

6 Queueing Systems Analysis

Because we want to provide a methodology that allows for performance analysis of Internet-related systems, we also examine the accuracy of the D&C-EM algorithm from the queueing systems perspective. We consider a Web server and model its operation as an $M/H_r/1$ server queue with exponential interarrival times and hyperexponential service times [5]. The hyperexponential model for the service process is generated from the test data sets (described in Subsection 5.1) using the D&C-EM, the EM, or the FW algorithms. We opt for exponential interarrival times in order to concentrate on the effects of the service process *only* on queueing behavior.

To analyze the $M/H_r/1$ queue that resulted from our fittings, we use the matrix-geometric method proposed by Neuts [13] and implemented in the MAMSolver tool [16]. The solution provides the entire stationary probability distribution for the queueing system under study, the average queue length,

rate, is used for reasons of presentation simplicity for both data sets and distribution functions.

[5] We assume that the service time for each request arriving at the Web server is linearly related to the size of the requested file. Hence by modeling the distribution of the requested file sizes, we also model the Web server's service process.

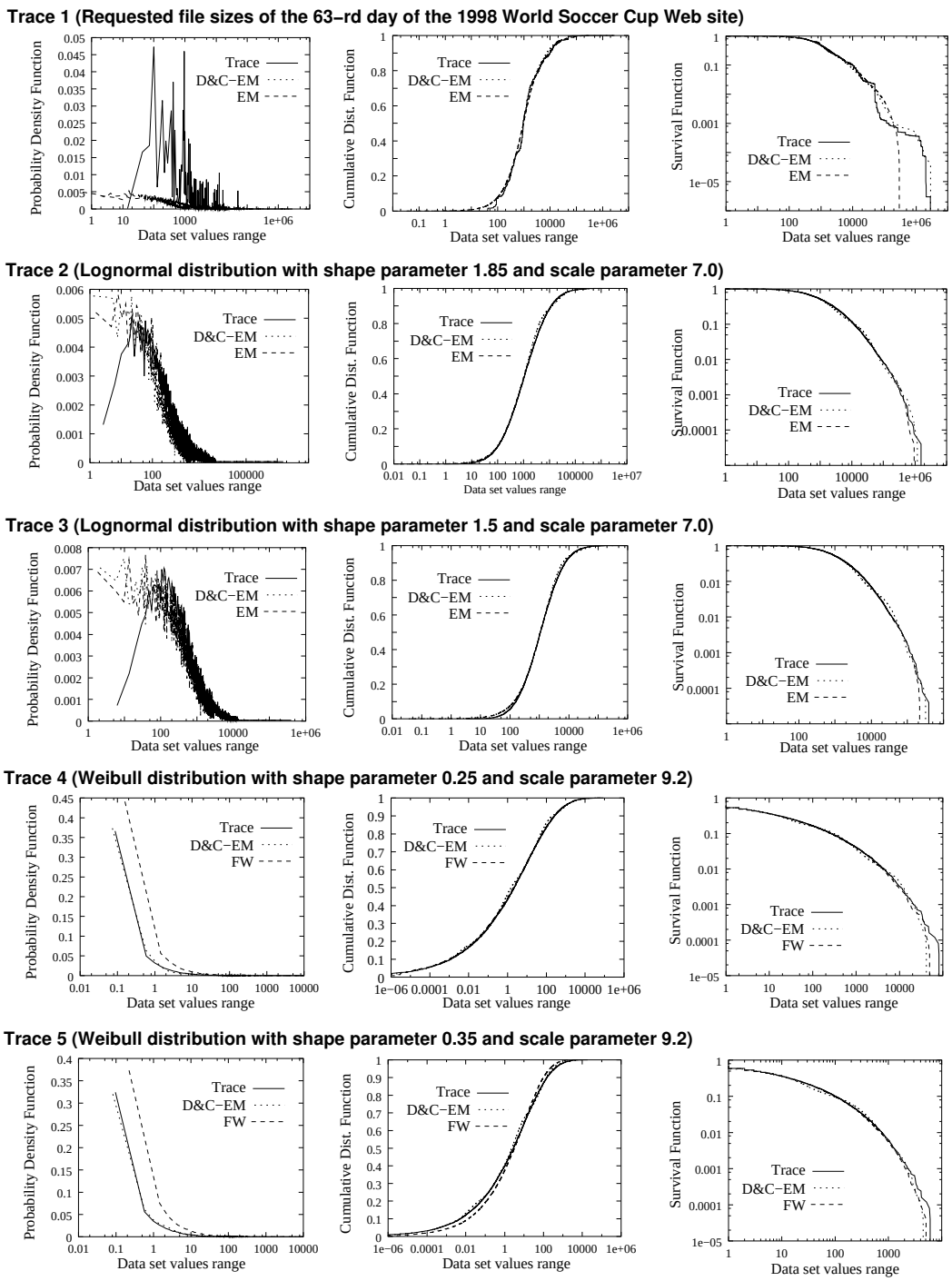

Fig. 5. PDF, CDF, and CCDF of the data-sets and of the fitted models.

and the queue length distribution. In our analysis, we focus on both the average queue length and the queue length distribution. The queue length distribution is an important metric because it can guide system design and at the same time is a strong indicator of model accuracy. Furthermore, because our focus is on long-tailed data sets, we need to demonstrate that the proposed hyperexponential server captures well the tail of the queue length distribution.

To examine the accuracy of the fitting algorithm, we compare the above performance measures that are analytically derived using the $M/H_r/1$ queue with those obtained from trace-driven simulations. The simulation model consists of a single server queue with the same arrival process as in the $M/H_r/1$ model and service process driven from the entries of the test data sets of Table 2. Results are presented in Figure 6. For all data sets, we plot the average queue length as a function of the arrival rate (first column of graphs in Figure 6), the body of the queue length distribution (middle column in Figure 6), and the tail of the queue length distribution (last column in Figure 6). We depict the queue length tail distribution by plotting the queue length distribution in a log-log scale. The queue length distributions in Figure 6 correspond to 80% system utilization levels.

We observe that the models obtained from the D&C-EM fitting generate queueing system results that are close to the ones obtained from the trace-driven simulation. We conclude that the $M/H_r/1$ queueing model captures accurately performance measures of interest and the behavior of systems that operate under highly variable workloads.

7 Computational efficiency of D&C-EM

In this section, we report on the computational efficiency of D&C-EM for fitting of highly variable data sets into hyperexponential distributions. In Figure 7, we illustrate the computation time needed to obtain fittings for the data sets indicated in Table 2 using the D&C-EM and EM algorithms. The experiments were conducted in a Linux machine with Pentium III 800MHz processor and 1GB of memory. D&C-EM is much faster than EM, even for fittings that consist of a few phases only. The efficiency of D&C-EM increases as the number of unique entries in the data sets increases. Note that for traces 4 and 5, we were not able to obtain results even for 4-phase models within a time frame of a week, so the timing results for this trace corresponds to D&C-EM only.

8 Refined D&C-EM fitting algorithm

The hyperexponential distribution is a special case of the PH distribution and has a completely monotone PDF. Therefore, the hyperexponential distribution is suitable for approximating data sets or distribution functions whose PDFs are completely monotone, such as the Weibull distribution with an appropriate selection of the shape parameter as in traces 4 and 5. On the other hand, traces

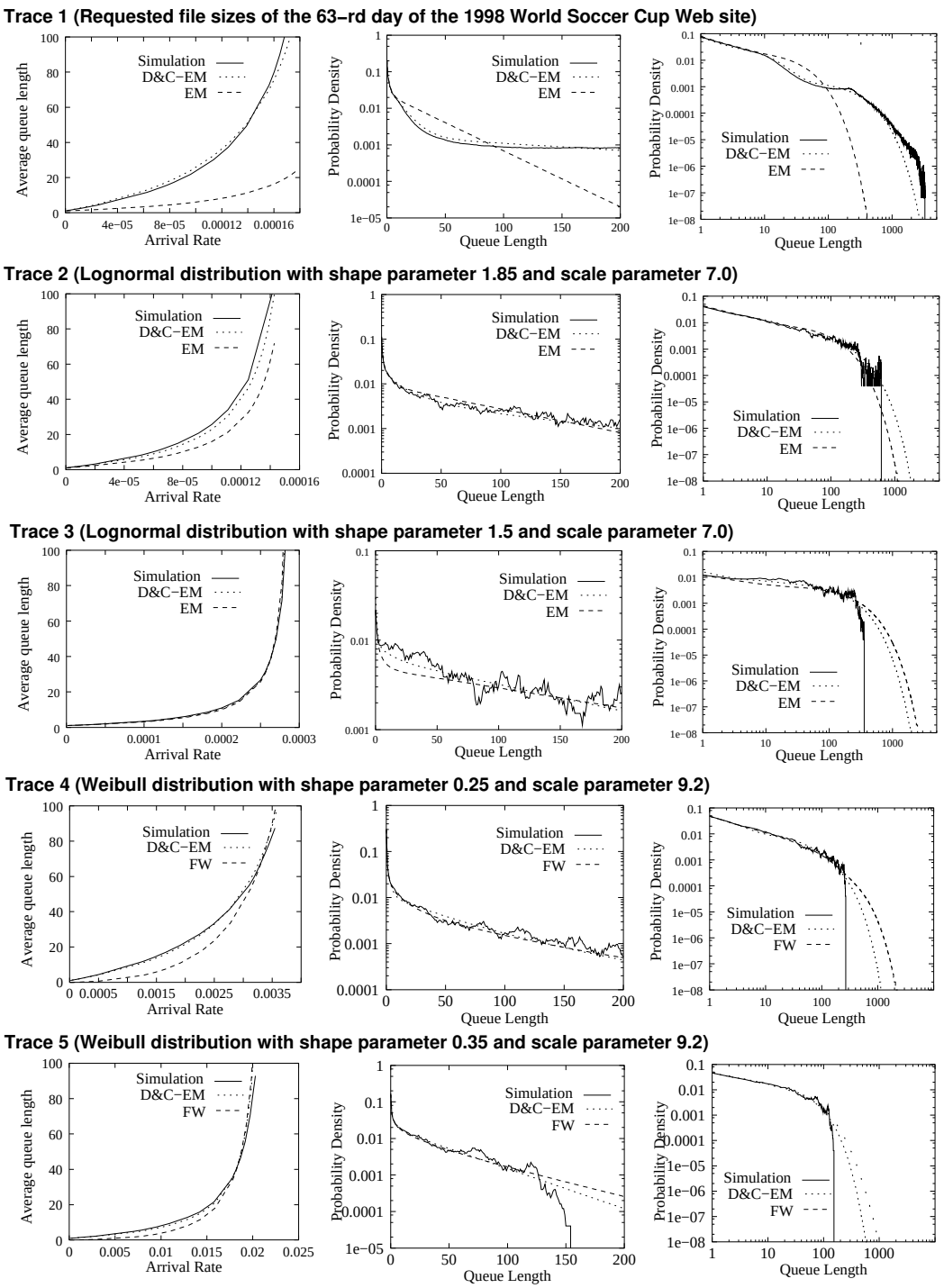

Fig. 6. Average queue length, queue length distribution, and tail queue length distribution (80% utilization levels).

1, 2, and 3 do not have completely monotone PDFs. For the data sets with non-monotone densities, the hyperexponential distribution is not the best choice among the special cases of PH distributions. A PH distribution constructed as a mixture of an Erlang and a hyperexponential distribution results in a better

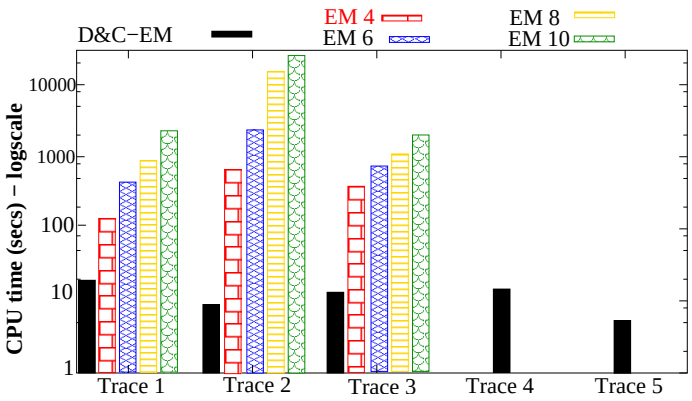

Fig. 7. CPU time for running EM and D&C-EM on all the traces.

fit for data sets whose CDHs shape is similar to traces 1, 2, and 3, i.e., not completely monotone with one peak only [18] (see Figure 8).

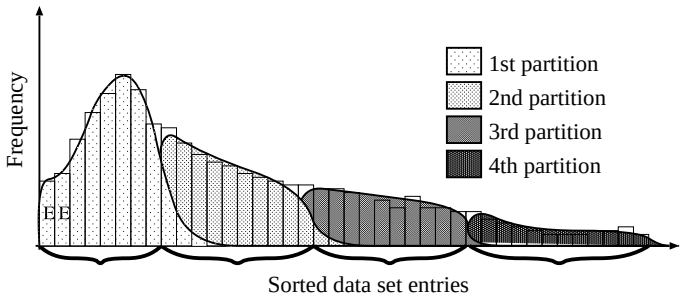

Fig. 8. Splitting of the non-monotone continuous data histogram (CDH).

In order to fit a data set into a mixture of Erlang and hyperexponential distributions, we first determine the portion of the data set's CDH that would be fitted into the Erlang distribution. We start from the first bin of the CDH and accumulate in the same fashion described in Section 4, until we take into consideration approximately 0.5% of the entire data set entries. The value 0.5% is selected as representative to account for just a small portion in the beginning of the CDH [6]. We denote the index of the last bin of this accumulation with Max. In Figure 8 the two leftmost bins, marked with "E", illustrate how we accumulate for the Erlang fitting. In this case, Max is equal to 2. The accumulated data is fitted into an Erlang distribution with 2 phases. Since the Erlang distribution with 2 phases has only one parameter, i.e., the mean λ_E of each of the exponential phases, we use moment matching [9] rather than the EM algorithm to determine this parameter.

Once the parameter of the Erlang distribution is determined, we continue with the D&C-EM algorithm as described in Figure 4 to fit the entire data set into

[6] The fitting algorithm is not sensitive to the amount of collected bins for the Erlang fitting. We experimented with different values in the 0.1% to 1% range and obtained similar results.

1. Build CDH from the data set.
2. Find bin index Max such that:
  a. entries upto bin Max account for 0.5% of all entries,
  b. bin frequencies increase.
3. if $(Max > 0)$
  a. Fit the first Max CDH bins into a 2-phase Erlang
  b. Obtain λ_E
4. while (there are still CDH bins to be considered)
  a. include data of current bin into current partition i
  b. update CV_{Acc}
  c. if $(CV_{Acc} > CV_{max})$
    Use EM to fit partition i into ph phases
    Obtain p_j^i and λ_j^i, $1 \leq j \leq ph$
    Compute weight w_i for partition i
    $CV_{Acc} = 0$.
5. if $(CV_{last\ partition} < CV_{max})$
  a. Merge last two partitions and perform step **4c**
6. Generate the final hyperexponential fit
  for i from 1 to # of partitions
    for j from 1 to ph
      $p_j^i = p_j^i \cdot w_i$
7. if $(Max > 0)$
  a. Obtain the final PH fit by merging
    the Erlang and the hyperexponential fits

Fig. 9. The refined D&C-EM fitting algorithm.

a hyperexponential distribution. Upon completion of the fitting procedure, we merge the Erlang fit with the hyperexponential fit by letting the Erlang fit proceed the hyperexponential one in the final PH distribution. We illustrate the structure of the vector $\boldsymbol{\pi}$ and the matrix $\mathbf{T}$ for the PH distribution in Eq.(1) assuming that the hyperexponential fit has 3 phases and the Erlang only 2.

$$\boldsymbol{\pi} = [1,\ 0,\ 0,\ 0,\ 0] \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} -\lambda_E & \lambda_E & 0 & 0 & 0 \\ 0 & -\lambda_E & p_1\lambda_E & p_2\lambda_E & p_3\lambda_E \\ 0 & 0 & -\lambda_3 & 0 & 0 \\ 0 & 0 & 0 & -\lambda_4 & 0 \\ 0 & 0 & 0 & 0 & -\lambda_5 \end{bmatrix}, \qquad (1)$$

where p_i for $1 \leq i \leq 3$ are the probabilities associated with each phase of the

hyperexponential distribution.

Figure 9 summarizes the refined D&C-EM algorithm for fitting data sets with not completely monotone CDHs into PH distributions. Observe that while we introduce three new steps, i.e., **Step 2, 3**, and **7**, in the improved D&C-EM algorithm, the algorithm does accommodate both cases of monotone and non-monotone densities. Step **2** determines if the data set CDH is non-monotone and there is need for an Erlang fitting. We emphasize that the algorithm described in Figure 9 improves fitting of the body of the CDH while preserving the accuracy in capturing the data set's tail. Most importantly, the refined D&C-EM maintains the same computational efficiency as D&C-EM.

8.1 Results with the refined D&C-EM

In this subsection, we present experimental results obtained by applying the algorithm outlined in Figure 9 to fit traces 1, 2, and 3, i.e., traces with not completely monotone CDHs, into PH distributions with structure as in Eq.(1). In Figure 10, we present the probability densities for traces 1, 2, and 3, and the D&C-EM fits obtained using both versions of the D&C-EM algorithm. The first row presents the hyperexponential-only fits and the second row gives the Erlang-hyperexponential fits. Observe that the refined D&C-EM algorithm captures better the shape of the body of the CDH even for the more challenging case of trace 1.

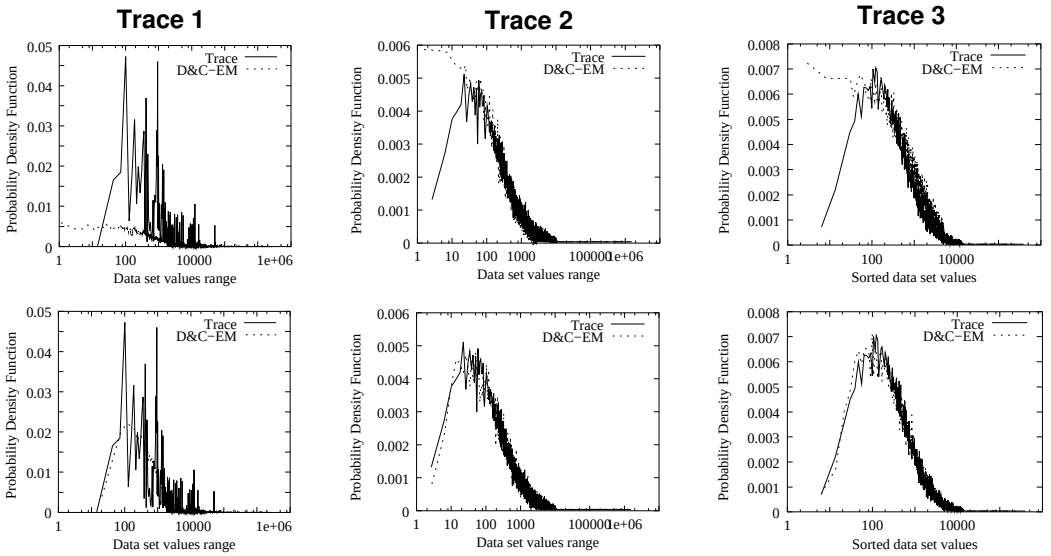

Fig. 10. PDFs of the fittings for Traces 1, 2, and 3 using D&C-EM (first row) and the refined D&C-EM (second row).

We illustrate the moments and median matching for the refined D&C-EM algorithm in Table 5. Observe that the first moments of the fitted models

closely match the first moments of the data sets (a direct outcome of better matching the CDHs). There is no improvement in matching the second or any of the higher moments by applying the refined D&C-EM. Indeed, by including a series of exponential servers *right before* the hyperexponential server, we "reduce" the C.V. of the original distribution and increase the median since the series of exponential servers essentially enforce a decrease in the densities of the lowest values of the data spectrum of the resulting distribution. Note that the new approach still captures closely the tail of the CDH, the most significant characteristic of the data sets of interest.

	Data	D&C-EM	Refined D&C-EM
Trace 1			
Mean	4407.81	4393.56	4407.74
CV	7.28	7.86	7.80
Median	938.00	950.59	967.33
Trace 2			
Mean	6358.23	6164.50	6349.25
CV	5.87	5.13	5.04
Median	1081.14	1061.25	1268.19
Trace 3			
Mean	3459.86	3391.06	3441.82
CV	3.13	2.82	2.79
Median	1085.43	1086.59	1126.60

Table 5
Statistical evaluation of D&C-EM and refined D&C-EM fittings.

The refined D&C-EM algorithm maintains the performance characteristics of the D&C-EM fits when used in queueing systems. Because the first moment of the data sets are better matched, the average queue lengths of the queueing systems described in Section 6 are more accurate than what is shown in Figure 6 (first column). Since both the refined D&C-EM and DCEM algorithms fit the tail of the empirical distribution exactly the same, we stress that the accuracy of capturing the queue length distribution (body and tail) is maintained, emphasizing again that capturing the tail is critical for highly variable data sets.

9 Conclusions and Future Work

In this paper, we proposed a new method to fit data sets exhibiting long-tailed behavior into hyperexponential and more general PH distributions. The proposed method, D&C-EM, uses the EM algorithm in a divide and conquer fashion, which dramatically improves both computational efficiency and fitting accuracy of the original EM. D&C-EM divides the data set into smaller subsets such that each subset has a smaller variability than the entire data set, facilitating an accurate EM fit of the data in each subset using a few phases only. The resulting fits for each subset are combined together to generate the final fit for the original data set. We demonstrated the goodness-of-fit of the new method from both the statistical and queueing systems perspective. Future work includes extending D&C-EM to fit data sets with multi-modal behavior into general PH distributions and incorporating D&C-EM in on-line performance monitoring and prediction tools of Internet-related systems.

Acknowledgments

We would like to thank Qi Zhang for providing us with some of the experimental results presented in Section 2.

References

[1] M. Arlitt and T. Jin. Workload characterization of the 1998 World Cup Web site. Technical report, Hewlett-Packard Laboratories, Sept. 1999.

[2] S. Asmussen and G. Koole. Marked point processes as limits of Markovian arrival streams. *Journal of Applied Probability*, 30:365–372, 1993.

[3] S. Asmussen, O. Nerman, and M. Olson. Fitting phase type distributions via the EM algorithm. *Scandinavian Journal of Statistics*, 23:419–441, 1996.

[4] A. Bobbio and A. Cumani. ML estimation of the parameters of a PH distribution in triangular canonical form. In G. Balbo and G. Serazzi, editors, *Computer Performance Evaluation*, pages 33–46. Elsevier Science Publishers, 1992.

[5] A. Feldmann and W. Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 31(8):963–976, Aug. 1998.

[6] A. Horvath and M. Telek. Approximating heavy tailed behavior with phase type distribution. In G. Latouche and P. Taylor, editors, *Advances in Algorithmic Methods for Stochastic Models*, pages 191–214. Notable Publications, 2000.

[7] M. A. Johnson and M. R. Taffe. Matching moments to phase distributions: Mixtures of Erlang distribution of common order. *Stochastic Models*, 5:711–743, 1989.

[8] R. E. A. Khayari, R. Sadre, and B. R. Haverkort. Fitting world-wide web request traces with the EM-algorithm. *Performance Evaluation Journal*, (forthcoming),2003.

[9] L. Kleinrock. *Queueing Systems, Volume I: Theory.* Wiley, 1975.

[10] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling.* SIAM, Philadelphia PA, 1999. ASA-SIAM Series on Statistics and Applied Probability.

[11] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis, Third Edition.* McGraw-Hill Inc., 2000.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic. *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.

[13] M. F. Neuts. *Matrix-geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, MD, 1981.

[14] M. F. Neuts. *Structured Stochastic Matrices of M/G/1-type and their Applications.* Marcel Dekker, New York, NY, 1989.

[15] M. Olsson. The EMpht-programme. Technical Report http://www.math.lth.se/matstat/staff/asmus/pspapers.htm, Department of Mathematics, Chalmers University of Technology, June 1998.

[16] A. Riska and E. Smirni. MAMSolver: a matrix-analytic methods tool. In T. Field, P. Harrison, J. Bradley, and U. Harder, editors, *Performance TOOLS 2002*, number 2324 in LNCS, pages 205–211, London, UK, April 2002. Springer-Verlag.

[17] A. Riska, M. Squillante, S. Yu, Z. Liu, and L. Zhang. Matrix-analytic analysis of a MAP/PH/1 queue fitted to web server data. In G. Latouche and P. Taylor, editors, *Matrix-Analytic Methods: Theory and Applications*, pages 333–356. World Scientific, 2002.

[18] C. Singh and R. Billinton. *System Reliability Modeling and Evaluation.* Hutchinson, 1977.