A Class of Tractable Models for Run-Time Performance Evaluation

Giuliano Casale Peter Harrison*
Department of Computing
Imperial College London
London SW7 2AZ, U.K.
{g.casale,pgh}@imperial.ac.uk

ABSTRACT

Run-time resource allocation requires the availability of system performance models that are both accurate and inexpensive to solve. We here propose a new methodology for run-time performance evaluation based on a class of closed queueing networks. Compared to exponential product-form models, the proposed queueing networks also support the inclusion of resources having first-come first-served scheduling under non-exponential service times. Motivated by the lack of an exact solution for these networks, we propose a fixed-point algorithm that approximates performance indexes in linear time and linear space with respect to the number of requests considered in the model. Numerical evaluation shows that, compared to simulation, the proposed models solved by fixed-point iteration have errors of about $1\% - 6\%$, while, on the same test cases, exponential product-form models suffer errors even in excess of 100%. Execution times on commodity hardware are of the order of a few seconds or less, making the proposed methodology practical for run-time decision-making.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling Techniques

General Terms

Performance, Algorithms

Keywords

Run-time prediction, analytical modeling, general distributions, closed queueing networks

*The work of G. Casale was supported by an Imperial College Junior Research Fellowship. The work of P. G. Harrison was supported in part by the Engineering and Physical Sciences Research Council of the United Kingdom, research grant number EP/D061717/1.

ICPE'12, April 22–25, 2012, Boston, Massachusetts, USA

1. INTRODUCTION

Run-time management of software systems often requires the prediction, in a short amount of time, of the performance arising from the interaction of multiple requests, resources, and software components. Run-time operation often requires low computational costs, thus leading to the use of simplified performance models which assume exponentially distributed service times. However, this is not a realistic assumption for several software systems. For example, web service workloads exhibit different degrees of variability in their service times [17]. As a result of this, exponential models ignoring this variance tend to suffer high prediction errors, especially when first-come first-served scheduling policies are used at resources. Still, first-come first-serve policies are often used in software performance evaluation, for example to describe admission controls limiting the maximum threading level.

In this paper, we tackle the run-time prediction problem by introducing a fast approximation technique for the analysis of a quite general class of queueing network models that overcomes this issue. Queueing networks are often useful in studying complex resource allocation problems, where either hardware, software or network devices may become a performance bottleneck. The class of queueing networks we consider has the advantage of being better able to represent actual, empirical, distributions of resource service times than models that impose exponential assumptions. The proposed approach leverages on realistic parameterization intervals for the skewness of service time. Motivated by the analysis of a recent public dataset of 15000 web service invocations [28], we find that when the skewness lies in the same range seen in the real world trace, queueing systems parameterized with phase-type (PH) service distributions [20] exhibit regularities in the solution that we exploit to define an efficient, approximate solution method. In particular, the main technical contribution of the paper is a fixed point algorithm that accurately approximates the model in $O(N)$ time as the number of requests N issued to the resources grows. This provides a major advantage over existing methods for generally-distributed workloads which require polynomial time (typically, cubic or quadratic) and hence are often too slow for application in run-time service management. A case study related to connection pooling is provided later in the paper, which illustrates a possible application of the proposed methodology in practice.

The analysis of queueing models with non-exponential workloads has traditionally focused on approximate meth-

ods. Several techniques exist to analyze isolated queues [4], but a relatively small number of results have been obtained for networks of queues. First, in [22], Reiser develops an approximation technique based on the mean value analysis (MVA) equations [4]. The method is enhanced in [11], using a local iterative approximation approach. Diffusion approximation is also an important approximation for networks in heavy-load [15, 12]. Zahorjan *et. al.* develop an approximate technique based on Markov chain decomposition methods [27, 10]. A general methodology based on approximation to $G/G/1$-queues leads to the queueing network analysis proposed by Whitt [26]. More recently, work has been done in approximating networks in which arrival- or service-processes are represented by correlated Markov-modulated processes [13, 25, 7, 9, 8]. Such methods are able to evaluate more expressive models than are studied here, but typically their computational requirements are much larger than the ones we propose[1]. For example, methods based on flow equivalent techniques, such as [8], require computational costs that grow polynomially in the number of requests N. This makes it prohibitive to solve models with more than a few tens of requests. In contrast, using the proposed approach, we were easily able to solve models with several hundreds of requests in just a few seconds.

The rest of the paper is organized as follows. In Section 2 we give background about PH distributions and the workload models used throughout the paper. Section 3 provides motivation for this work by showing unexpected properties of queueing-based solutions when PH distributions are parameterized with a limited range for the skewness. Section 4 develops an approximate scalar expression to characterize $M/PH/1$ and $PH/PH/1$ queues that is accurate under such parameterizations. A fixed-point algorithm for solving the closed PH queueing networks is introduced in Section 5 and validated in Section 6. Section 7 applies our method to a connection pooling problem. Finally, Section 8 concludes the paper.

2. BACKGROUND

2.1 PH Distributions

Phase-type (PH) distributions generalize probability distributions such as exponential, hyper-exponential, Erlang and Coxian [20, 4]. Compared to these models, PH distributions are more flexible in approximating the heavy-tailed distributions that are common in computer workloads [18]. PH models are able in theory to fit any empirical distribution if their order is sufficiently large [1].

Formally, a PH distribution of order K, denoted PH(K), is a continuous-time Markov chain (CTMC) with K transient states and one absorbing state. The transient states are called *phases*. The initial state probability mass function for the CTMC is specified by a row vector $\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}\mathbf{1} = 1$, $\mathbf{1}$ being a column vector of ones of the same length as $\boldsymbol{\alpha}$. The infinitesimal generator matrix for the CTMC is

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{t} \\ \mathbf{0} & 0 \end{bmatrix}, \quad \boldsymbol{t} = -\boldsymbol{T}\mathbf{1},$$

where the $\boldsymbol{T}$ block is called the PH *subgenerator*. A subgenerator $\boldsymbol{T}$ is defined similarly to an ordinary infinitesimal generator except that it satisfies $\boldsymbol{t} \geq 0$, $\boldsymbol{t}^T\mathbf{1} > 0$, where the column vector $\boldsymbol{t} = -\boldsymbol{T}\mathbf{1}$ is called an *exit vector* and represents the rate of jumping to the absorbing state from each of the K transient states of the PH distribution.

Conceptually, PH distributions model inter-arrival times of events as a time to absorption in a CTMC. Let X be the random variable for the time that the CTMC takes to reach the absorbing state after initialization. X can model either job inter-arrival times or service times, making PH distributions a flexible tool to describe the input parameters of a queueing model. In particular, with the above parameterization, it can be shown that the distribution modeled by the PH is

$$\Pr[X \leq x] = 1 - \boldsymbol{\alpha} e^{\boldsymbol{T}x}\mathbf{1}, \quad e^{\boldsymbol{T}x} = \sum_{k=0}^{\infty} \frac{(\boldsymbol{T}x)^k}{k!},$$

Many techniques for fitting PH distributions to empirical datasets have been proposed in the literature, using methods such as the EM algorithm [24] or moment matching [14].

Examples. An exponential distribution with rate μ has

$$\boldsymbol{\alpha} = \begin{bmatrix} 1 \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} -\mu \end{bmatrix},$$

An Erlang-2 process is represented as

$$\boldsymbol{\alpha} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} -\mu & \mu \\ 0 & -\mu \end{bmatrix},$$

A two-phase hyper-exponential distribution with phase-1 selection probability p is defined as

$$\boldsymbol{\alpha} = \begin{bmatrix} p & 1-p \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} -\mu_1 & 0 \\ 0 & -\mu_2 \end{bmatrix},$$

A Coxian distribution with K states has PH representation

$$\boldsymbol{\alpha} = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}, \quad \boldsymbol{T} = \begin{bmatrix} -\mu_1 & p_1\mu_1 & 0 & 0 \\ 0 & -\mu_2 & p_2\mu_2 & 0 \\ \vdots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & -\mu_K \end{bmatrix}.$$

PH renewal process. A sequence of independent and identically distributed (i.i.d.) samples $\{X_1, X_2, \dots\}$ is called a PH renewal process $(\boldsymbol{\alpha}, \boldsymbol{T})$ if the common distribution of the samples is PH with initial state probability mass function $\boldsymbol{\alpha}$ and subgenerator $\boldsymbol{T}$. A queueing system in which arrivals are Poisson and service times constitute a PH renewal process is called a $M/PH/1$ queue. If inter-arrival times also form a PH renewal process, the model is a $PH/PH/1$ queue.

2.2 Closed Queueing Networks

Throughout this paper, we study closed queueing networks as a tool to drive resource allocation decisions. A closed queueing network is populated by a fixed number N of circulating requests that visit a set of M resources. Requests may represent, for instance, a finite pool of N software threads issuing requests to resources. It is established in the literature that, as the population N grows, a closed model accurately approximates an open one [21]. This makes closed networks quite flexible provided that the computational costs of their solution techniques grow slowly with N.

In the models considered here, a request places a demand of X_k processing units when visiting resource k. Such processing units are assumed to include only the actual pro-

[1]It should be noted that the method in [9] is extremely fast; however, it is not currently applicable to models with more than a single non-exponential queue.

cessing and thus disregard the overhead due to contention from other requests. Upon completion, the request is then routed to another resource. We assume that service times X_k belong to a PH renewal process $(\boldsymbol{\alpha}_k, \boldsymbol{T}_k)$ for all resources $1 \leq k \leq M$ and we call the resulting model a closed PH queueing network[2]. In such a network, the average service time at resource k is

$$S_k = \boldsymbol{\alpha}_k(-\boldsymbol{T}_k)^{-1}\mathbf{1},$$

which follows from known expressions for the moments of a PH distribution [20].

Upon completing service at resource j, a request is routed to resource k with probability $p_{j,k}$. The routing matrix

$$\boldsymbol{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & \dots & p_{1,M} \\ p_{2,1} & p_{2,2} & \dots & p_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M,1} & p_{M,2} & \dots & p_{M,M} \end{bmatrix}$$

defines a discrete-time Markov chain, which we assume to be irreducible. The visit ratio to the different resources at steady-state is given by the row vector $\boldsymbol{v}$ satisfying $\boldsymbol{v}\boldsymbol{P} = \boldsymbol{v}$, where $\boldsymbol{v}\mathbf{1} = 1$. Thus, v_k is the fraction of times that a job is routed to resource k at steady-state. Using the convention that a job is completed upon passage through resource M, we define the average number of visits by a job to resource k prior to its completion by $V_k = v_k/v_M$, so that $V_M = 1$.

Finally, let $X(N)$ be the *throughput* of completed requests (as observed at station M) and let $R_k(N)$ be the mean *response time* at resource k, accumulated over V_k visits. The following output performance metrics of the queueing network model are considered throughout the paper for all resources $k = 1, \dots, M$:

- $\rho_k(N) = X(N)V_kS_k$ – the *utilization* of resource k
- $Q_k(N) = X(N)R_k(N)$ – the mean *queue-length* (i.e., backlog) at resource k, including the job in service
- $\pi_k(n)$ – the probability of observing n jobs at resource k

All the above metrics refer to the behavior of the system at steady state. Notice that since, by definition, $Q_k(N) = \sum_{n=1}^{N} n\pi_k(n)$, computing $X(N)$ and $\pi_k(n)$ provides full information also about the response times $R_k(N)$ at each resource. From these, it is easy to compute the system response time as $R(N) = \sum_{k=1}^{K} R_k(N)$, which is the mean time taken to complete a route through the resources before completion. In sections 4 and 5, we discuss the approximate computation of $X(N)$ and $\pi_k(n)$ for all resources $1 \leq k \leq M$.

2.3 Matrix Geometric Method

We briefly discuss the $M/PH/1$ queue where inter-arrival times are Poisson with rate λ and service times form a PH renewal process $(\boldsymbol{\alpha}, \boldsymbol{T})$ of order K. The $M/PH/1$ queue with first-come-first-served scheduling can be modeled as a structured CTMC called a quasi-birth-death (QBD) process. The CTMC state is a tuple (n, k), where n is the population of requests in the queue, including any job in service, and k is the currently active phase, out of K, of the PH service process. The QBD infinitesimal generator matrix then has block-tridiagonal form

$$\boldsymbol{Q} = \begin{bmatrix} -\lambda & \lambda\boldsymbol{\alpha} & 0 & 0 & 0 & \dots \\ \boldsymbol{t} & \boldsymbol{T} & \lambda\boldsymbol{I}_K & 0 & 0 & \dots \\ 0 & \boldsymbol{t}\boldsymbol{\alpha} & \boldsymbol{T} & \lambda\boldsymbol{I}_K & 0 & \dots \\ 0 & 0 & \boldsymbol{t}\boldsymbol{\alpha} & \boldsymbol{T} & \lambda\boldsymbol{I}_K & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where $\boldsymbol{t} = -\boldsymbol{T}\boldsymbol{e}$, $\boldsymbol{t}\boldsymbol{\alpha}$ is a rank-1 matrix and $\boldsymbol{I}_K$ is the identity matrix of order K. The steady-state probability distribution of the CTMC is the set of vectors $\boldsymbol{\pi}(n), n \geq 0$ such that $\pi(n)_k$ is the probability that the state is (n, k) at equilibrium. Thus, $\pi(n) = \sum_{k=1}^{K} \pi(n)_k$ is the marginal queue-length probability mass function.

The steady state distribution $\boldsymbol{\pi}(n)$ is the unique solution of the global balanced equations $\boldsymbol{\pi}\boldsymbol{Q} = 0$, $\boldsymbol{\pi}\mathbf{1} = 1$. An established result of matrix-geometric theory is that such solution may be expressed as a *rate matrix* $\boldsymbol{R}$ such that

$$\boldsymbol{\pi}(n+1) = \boldsymbol{\pi}(n)\boldsymbol{R}, \quad n \geq 1$$

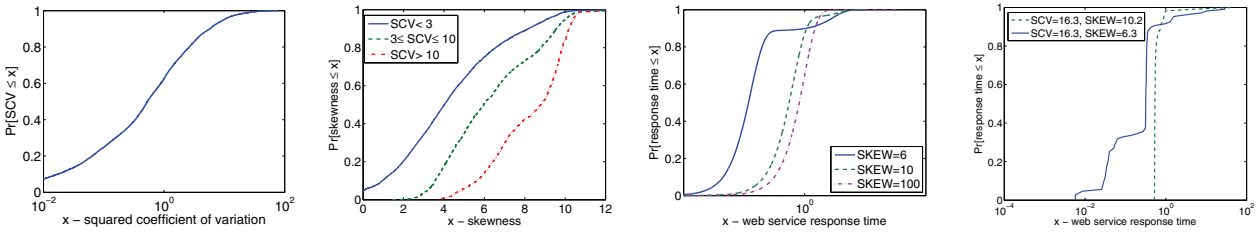
and also $\boldsymbol{\pi}(0)$ can be obtained as a function of $\boldsymbol{R}$ [20]. Since the last expression is similar to one defining a geometric sequence, where the scalar geometric rate is replaced by the matrix $\boldsymbol{R}$, the solution for queueing models parameterized by PH distributions is often referred to as a *matrix-geometric solution*. Finally, we point out that a rich literature is available on the computation of the rate matrix $\boldsymbol{R}$, see [20, 4] and references therein. Publicly available tools for computing $\boldsymbol{R}$ in languages such as C++ and MATLAB have appeared in [23, 3] and are free for download.

3. A MOTIVATING EXAMPLE

We first provide motivation for the approximation developed in the next sections. Let us begin by illustrating the quality of the matrix geometric method when the service times of a $M/PH/1$ queue are parameterized using a real trace from the software domain. To this end, we consider the wsdream dataset recently presented in [28]. This dataset consists of a collection of 15000 response time traces spanning 150 invocation sequences for a set of 100 public web services. Each web service is called several times using different clients deployed on the PlanetLab infrastructure. As a model, we consider a $M/PH/1$ queueing system, where the service time random variable X follows a PH distribution fitted to a trace of the wsdream dataset. This case study may be representive of the performance of a software system that processes requests in first-come first-served order by calling a remote web service every time a request is admitted to service.

Let us first motivate the need for methods that consider non-exponential workloads. The *squared coefficient of variation* (SCV) of a random variable X is defined as the ratio between variance and squared mean: $SCV = Var[X]/E[X]^2$. Further, let $SKEW$ denote the skewness of X, which describes the asymmetry in its probability distribution function. Figure 1(a) illustrates typical values of SCV for the wsdream dataset. The empirical distribution plot shows that both high-variability (i.e., $SCV > 1$) and low-variability ($SCV < 1$) traces are frequent in the dataset, with the median value being $SCV = 0.5294$, close to an Erlang-2 dis-

[2]Notice that a closed PH queueing network might be seen as a specialization of the recently proposed MAP queueing networks [7], which additionally offer the ability to consider service times that are correlated as described by a Markovian arrival process (MAP). However, such models are harder to solve and thus are less appealing for run-time applications.



(a) service time variability (b) service time skewness (c) PH models (d) wsdream trace

Figure 1: Characterization results for the wsdream trace [28]

tribution ($SCV = 0.50$). The fraction of traces with variability greater than an exponential ($SCV > 1$) is significant and represents 37.4% of the total. Such traces correspond to cases where approximating the trace by an exponential distribution results in the largest errors in the prediction of performance metrics by queueing models. Figure 1(b) further illustrates the distribution of $SKEW$ for increasing SCV values, showing that $SKEW$ is typically positive and reaches its maximum value around $SKEW \approx 12$.

Let us now examine the qualitative properties of skewness. For illustration purposes, we first consider PH distributions having same mean and variance but different skewness and we assume these PH models to represent response times for web services. Figure 1(c) illustrates the effects of skewness on the cumulative distribution function for three PH(2) models with mean $E[X] = 1$, $SCV = 16$ and $SKEW \in \{6, 10, 100\}$, where only the first two skewness values are representative of actual values seen in the wsdream trace. As we can see from the figure, for low skewness values, more probability mass is concentrated on small response times. This results in a bimodal (hyper-exponential) distribution, where a significant probability exists of sampling large values. Conversely, as the skewness increases, there is an increasing probability of sampling huge values, but the distribution is unimodal due to the low overall probability mass placed on the tail. As suggested by Figure 1(b), low skewness values are more frequent in web service traces, where it is indeed quite common to observe multi-modal distributions[3]. Figure 1(d) illustrates two distributions taken from the wsdream trace that illustrate such a property in real-world data. Similarly to the PH(2) models shown in Figure 1(c), low skewness values are associated with multi-modal behavior.

3.1 Impact on Queueing Performance

Let us now compare the performance for $M/PH(2)/1$ queueing systems, where service times are parameterized to follow the same distributions used in Figure 1(c). The utilization of the queue is set to 80%. The marginal queue-length probability distributions obtained by the matrix geometric method are plotted in Figure 2(a). Although the high-variability makes these probabilities very different from those of a $M/M/1$ queue, the two cases for low skewness follow a regular geometric decay in the queue-length distribution like in a $M/M/1$, apart for some perturbations around the lowest queue-length sizes. Conversely, the model with $SKEW = 100$ shows a sharp change in the decay rate around $n = 12$ jobs. This behavior can also be observed in other parameterizations, with the differences being more clearly visible at high load and for large SCV. Figure 2(b) shows simulation results for a $M/Trace/1$ queue, where the service time distributions are parameterized empirically using the two traces shown in Figure 1(d). Note that the maximum observed queue-length for the simulation experiment is limited by the simulation length that was set to 10^5 samples. The decay behavior of the two traces is consistent with the one seen in Figure 2(a), as can be seen from the early convergence to a geometric decay in the rate at a queue-length of around $n = 4$.

In order to show that such property is driven by skewness, Figure 3(a) and Figure 3(b) report experiments we performed with the $M/PH(2)/1$ queue to establish at which queue-length n the decay rate of the marginal distribution converges to the asymptotic one η within a 1% tolerance. That is, we seek the value n^* such that

$$n^* = \min_n \left| \frac{\pi_{n+1}}{\pi_n} - \eta \right| < 10^{-2} \qquad (1)$$

where π_n is the marginal equilibrium probability of observing n jobs in the queue (including any in service) and

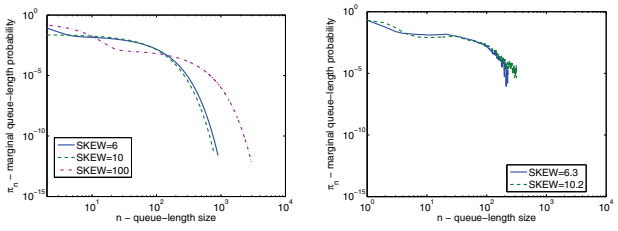
$$\eta = \lim_{n \to +\infty} \frac{\pi_{n+1}}{\pi_n}$$

is called the *caudal characteristic* and represents the asymptotic geometric decay of the queue-length distribution. The results in Figure 3(a) and Figure 3(b) indicate that as the skewness varies within the range observed in the wsdream trace ($SKEW \leq 12$), there is early convergence to the asymptotic decay rate typically for small populations $n^* \leq 10 - 15$. This property does not hold when we consider a larger skewness values outside the range observed for the wsdream trace[4].

Summarizing, real-world software workloads can be characterized by large variability, due to the significant probability of observing large response times, and low skewness arising, in the cases we have found, in connection to multi-modal behavior. Parameterizing the service times of the $M/PH/1$ queue with low skewness values yields marginal queue-length probabilities that decay geometrically with good approximation. Motivated by this important observation, in the

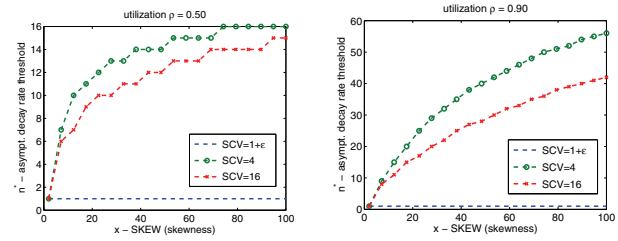
[3]Manual examination of the wsdream traces is sufficient to verify the significant frequency of multi-modal traces.

[4]Notice that the experiment for $SCV = 1$ uses $SCV = 1+\epsilon$, where $\epsilon = 0.01$, so that we have the ability to control $SKEW$ without significantly deviating from the variability of an exponential distribution. In these experiments, it is clear that the geometric decay rate is constant from population $n = 1$ and thus the system always behaves similarly to an $M/M/1$ queue.



(a) (b)

Figure 2: Queueing analysis of the two wsdream traces shown in Figure 1(d).



(a) (b)

Figure 3: Asymptotic decay rate threshold, eq. (1)

next section we derive an approximation for $M/PH/1$ and $PH/PH/1$ queues that is later exploited to define a tractable class of PH queueing network models.

4. PH QUEUE APPROXIMATION

Based on the observations in the previous section, we now seek to obtain new approximate scalar expression for the steady-state distribution of queues with PH service. Let us consider first the asymptotic behavior of the steady-state distribution $\boldsymbol{\pi}(n)$. The *caudal characteristic* η is the spectral radius of the rate matrix $\boldsymbol{R}$, i.e., $\eta = \max_{1\leq k\leq K} |\eta_k|$, where the quantities η_k are the eigenvalues of $\boldsymbol{R}$. Let $\boldsymbol{\ell}$ and $\boldsymbol{r}$ be the left and right unit-eigenvectors corresponding to eigenvalue η such that the rank-1 matrix $\boldsymbol{\Pi} = \boldsymbol{r}\boldsymbol{\ell}$ is called the spectral *projector* for η. Neuts has shown that asymptotically $\boldsymbol{\pi}(n+k) = \boldsymbol{\pi}(n)\eta^k\boldsymbol{\Pi} + o(\eta^k)$ as $n \to \infty$ for $k \geq 1$, which follows from the fact that, for irreducible, non-negative matrices R,

$$\boldsymbol{R}^n = \eta^n\boldsymbol{\Pi} + o(\eta^n) \tag{2}$$

at large n [19]. Therefore the dynamics of a $M/PH/1$ queue when the number of jobs n is large is determined by the eigenvalue η and its projector $\boldsymbol{\Pi}$ only. The result holds also for the $PH/PH/1$ queue, which is considered in the remainder of the paper [20].

The approximation technique that we develop for queueing networks in this paper is based on the following assumptions:

A1. The steady-state solution of a queue with PH service times is entirely determined by the eigenvalue η and its projector $\boldsymbol{\Pi}$. That is, we assume that (2) is valid at all populations $n \geq 1$, not only asymptoticly. Equivalently, this might be seen as studying a model where all non-dominating eigenvalues of $\boldsymbol{R}$ are set to 0, so that $\boldsymbol{R} = \eta\boldsymbol{\Pi}$.

A2. The conditional distribution of phases is identical at each population $n \geq 1$, i.e.

$$\widetilde{\boldsymbol{\theta}} = \frac{\widetilde{\boldsymbol{\pi}}(n)}{\widetilde{\boldsymbol{\pi}}(n)\mathbf{1}}$$

so that $\widetilde{\boldsymbol{\theta}}\mathbf{1} = 1$, where $\widetilde{\boldsymbol{\pi}}(n)$ is the approximate probability distribution as opposed to the exact one $\boldsymbol{\pi}(n)$. We take this conditional distribution to be the exact asymptotic one, i.e., $\widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is a left unit eigenvector of $\boldsymbol{R}$, such $\boldsymbol{\theta}\boldsymbol{\Pi} = \eta\boldsymbol{\theta}$, see Appendix A for details.

From the above approximating assumptions, we have, for all $n \geq 1$, that

$$\widetilde{\pi}(n+1) = \widetilde{\boldsymbol{\pi}}(n+1)\mathbf{1} = \eta\widetilde{\pi}(n)\boldsymbol{\theta}\boldsymbol{\Pi}\mathbf{1} = \eta\widetilde{\pi}(n), \tag{3}$$

Note that the utilization of the queue, ρ, must be $\rho = \sum_{n\geq 1}\widetilde{\pi}(n) = \widetilde{\pi}(1)(1-\eta)^{-1}$, which combined with (3) yields the main result of this section:

$$\widetilde{\pi}(n) = \rho(1-\eta)\eta^{n-1}, \qquad \widetilde{\pi}(0) = 1-\rho, \tag{4}$$

which provides an elegant scalar approximation for the matrix-geometric solution. The phase information is correspondingly approximated as

$$\widetilde{\boldsymbol{\pi}}(n) = \widetilde{\pi}(n)\boldsymbol{\theta}, \qquad \boldsymbol{\theta} = \boldsymbol{\theta}\boldsymbol{\Pi}.$$

Notice that the above expressions require knowledge of $\boldsymbol{\Pi}$ and η, which follow from *exact* calculation of the matrix geometric solution $\boldsymbol{R}$. This may suggest that the above approximation does not provide any computational advantage over an exact solution. However, as we show in the next section, the proposed approximation becomes valuable in the context of closed queueing networks, where the scalar nature of the equations suggest a simple iterative algorithm to compute the probability distribution of the network state at equilibrium. This iterative algorithm is detailed later in Proposition 1.

5. CLOSED PH QUEUEING NETWORKS

This section introduces a technique for the approximate analysis of closed queueing networks with PH service. We begin with a brief review of product-form networks with exponentially-distributed service times.

5.1 Exponential networks

In a closed queueing network with exponential servers, the joint probability mass function of network states may be written as:

$$\Pr[\boldsymbol{n}] = \frac{\prod_{k=1}^{M}(1-\rho_k)\rho_k^{n_k}}{C(M,N)} \tag{5}$$

where n_k is the number of requests at station k, the state is $\boldsymbol{n} = (n_1, n_2, \ldots, n_M)$, including the one in service (if any), $C(M,N)$ is a normalizing constant ensuring that $\sum_{\boldsymbol{n}\in S}\Pr[\boldsymbol{n}] = 1$, the state space of the network is the set

$$S = \left\{(n_1, n_2, \ldots, n_M) : \textstyle\sum_{k=1}^{M} n_k = N, n_k \geq 0, n_k \in \mathbb{N}_0\right\},$$

and $\rho_k = X(N)S_kV_k$ is the utilization of station (or resource) k. Notice also that (5) is often rewritten as

$$\Pr[\boldsymbol{n}] = \frac{\prod_{k=1}^{M}(V_kS_k)^{n_k}}{G(M,N)}, \tag{6}$$

where S_k is the mean service time of station k and $G(M,N) = C(M,N)/\prod_{k=1}^{M}(1-\rho_k)/(X(N))^M$. The two expressions are equivalent, but (6) does not require *a priori* knowledge of the throughput $X(N)$, which becomes an output quantity obtained by solving the model.

Buzen showed that an exponential network can be solved by the convolution algorithm [5], which recursively evaluates

$$G(M,N) = G(M-1,N) + V_M S_M\, G(M,N-1)$$

subject to initial conditions $G(0,N) = 0$ and $G(M,0) = 1$, where $G(M-1,N)$ (resp. $G(M,N-1)$) is the normalizing constant in a model with station M (resp. with a job) removed from the network. Using the normalizing constant, one can immediately find the joint probability mass function (6). This then provides mean performance metrics which are computed as:

$$X(N) = \frac{G(M,N-1)}{G(M,N)},$$

$$\pi_k(n_k) = \frac{(V_k S_k)^{n_k} G(M_k, N-n_k)}{G(M,N)},$$

where $G(M_k, N-n_k)$ is the normalizing constant of a network with station k removed and $N-n_k$ circulating jobs. Utilizations are $U_k(N) = X(N)V_kS_k = \sum_{n=1}^{N}\pi_k(n)$; mean queue-lengths are $Q_k(N) = \sum_{n=1}^{N} n\pi_k(n)$; mean response times follow by Little's law as $R_k(N) = Q_k(N)/X(N)$, for stations $1 \le k \le M$.

5.2 Cyclic PH networks

We first consider the analysis of cyclic PH networks where resources are arranged in series, so that $p_{M,1} = 1$ and $p_{i,i+1} = 1$, for $1 \le i \le M-1$. Assume initially that the network throughput $X(N)$ is known, so that the utilization of each resource i, $\rho_i = X(N)V_iS_i$, is easily determined. Also, let j denote the resource sending jobs to resource k, so here, $k = j+1 \mod M$.

In order to determine the joint probability mass function for the network state, we first study each resource k as a $PH/PH/1$ first-come-first-served queue in isolation. Both the service and arrival processes for such a queue are PH renewal processes $(\boldsymbol{\alpha}_k, \boldsymbol{T}_k)$ and $(\boldsymbol{\alpha}_j, \rho_j\boldsymbol{T}_j)$ respectively. The latter process is defined by assuming approximately that the inter-departure time between jobs at resource j has a PH distribution which is a scaled version of the service process $(\boldsymbol{\alpha}_j, \boldsymbol{T}_j)$; the mean inter-departure rate is adjusted to be $X(N)$ using the scaling factor ρ_j multiplying $\boldsymbol{T}_j$. The set of $PH/PH/1$ queues defined by the above approach can then be solved by the matrix geometric method to obtain the caudal characteristic η_k for each resource $1 \le k \le K$.

Based on our analysis of Section 4, we approximate the joint probability mass function for the state of the whole network by

$$\Pr[\boldsymbol{n}|X(N)] = \frac{\sum_{\boldsymbol{n}\in S}\prod_{k=1}^{M} F_k(n_k)}{G(M,N)} \tag{7}$$

where

$$F_k(n_k) = \begin{cases} 1-\rho_k & \text{if } n_k = 0 \\ \rho_k(1-\eta_k)\eta_k^{n_k-1} & \text{if } n_k > 0 \end{cases}$$

We do not need to re-normalize the marginal probability to sum to unity for $n_k \le N$ since such a normalizing constant is included in $G(M,N)$. From expression (7), it is possible to derive all the usual performance metrics for the queueing network model if the values of $G(M,N)$ and $X(N)$ are known; we discuss below the computation of these terms. Finally, observe that, once the model is solved, phase information for each queue can easily be retrieved using the relation $\boldsymbol{\pi}_k(n) = \pi_k(n)\boldsymbol{\theta}_k$ proposed as an asymptotically correct approximation in Section 4.

5.2.1 Normalizing Constant

We first note that (7) may be written as

$$\Pr[n_1,\ldots,n_M|X(N)] = \frac{\sum_S \prod_{k=1}^{M} q_k(1-\eta_k)\eta_k^{n_k-1}}{H(M,N)}, \tag{8}$$

where $H(M,N) = \prod_{k=1}^{M}(1-\rho_k)G(M,N)$, $q_k = \rho_k(1-\rho_k)^{-1}$. Expression (8) may be interpreted as the joint probability mass function at equilibrium of a Markovian queueing network with load-dependent service rates, where resource k serves jobs with rate

$$\mu_k(n_k) = \begin{cases} q_k^{-1}(1-\eta_k)^{-1}, & n_k = 1 \\ \eta_k^{-1}, & n_k > 1 \end{cases}$$

This enables direct computation of the normalizing constant $H(M,N)$ by the load-dependent convolution algorithm [5]. However, since load-dependent convolution is numerically unstable and existing stabilization techniques become expensive for models with several queues [6], we introduce a specialized method for computing the normalizing constant in (7). This determines efficiently the normalizing constant in $O(MN)$ time and $O(M)$ space.

PROPOSITION 1. *The normalizing constant $G(M,N)$ in (7) can be computed recursively by*

$$G(M,N) = (1-\rho_M)G(M-1,N) + \rho_M G^{aux}(M,N-1)$$
$$G^{aux}(M,N) = (1-\eta_M)G(M-1,N) + \eta_M G^{aux}(M,N-1)$$

with termination conditions

$$G(0,n) = 0, \qquad 1 \le n \le N \tag{9}$$
$$G^{aux}(m,0) = (1-\eta_m)\textstyle\prod_{i=1}^{m-1}(1-\rho_i), \quad 1 \le m \le M, \tag{10}$$

where $G^{aux}(M,N)$ is the normalizing constant of an auxiliary model in which ρ_M is replaced in (7) by η_M.

Proof of the proposition is given in the final appendix. MATLAB code for computing the normalizing constant is reported in Algorithm 1.

5.2.2 Solving the Model

We wish to obtain the throughput $X(N)$ of the PH network by searching for a fixed point of the system of equations defined by (7), with the consistency constraints

$$\rho_k = 1 - \sum_{\boldsymbol{n}:n_k=0} \Pr[n_1,\ldots,n_M|X(N)]$$
$$= 1 - \frac{(1-\rho_k)G(M_k,N)}{G(M,N)}, \tag{11}$$

for $1 \le k \le M$. The key observation that motivates this approach is that (11) is not, in general, satisfied for all guesses of $X(N)$ and thus of $\rho_k = X(N)V_kS_k$. This is because our analysis is approximate and so (7) does not guarantee a consistent description of the steady-state for all values of the

Algorithm 1 *MATLAB code for computing* $G(M, N)$.

Input: M, N, $rho(k) = \rho_k$, $eta(k) = \eta_k$, for $1 \leq k \leq M$
Output: $G(M, N)$
$resource\_indexes = 1 : M$;
$g = zeros(M + 1, N + 1)$;
$gaux = zeros(M + 1, N + 1)$;
for $k = resource\_indexes$ **do**
  $gaux(1+k, 1+n) = prod(1-rho(1 : k-1))*(1-eta(k))$;
end for
for $k = resource\_indexes$ **do**
  for $n = 1 : N$ **do**
    $gaux(1 + k, 1 + n) = max(0, 1 - eta(k)) * g(1 + k - 1, 1 + n) + eta(k) * gaux(1 + k, 1 + n - 1)$;
    $g(1+k, 1+n) = max(0, 1-rho(k)) * g(1+k-1, 1+n) + rho(k) * gaux(1 + k, 1 + n - 1)$;
  end for
end for
return $g(1 + M, 1 + N)$

input parameter $X(N)$. The proposed approximation leverages the heuristic hypothesis that, by guessing a suitable initial value for $X(N)$, a fixed point algorithm that seeks to minimize the violation of (11), while satisfying (7), will terminate with an estimate for $X(N)$ that is close to exact.

Since it is difficult to provide theoretical guarantees on the returned estimate due to the lack of characterizations of the exact solution of a PH queueing network, we first build confidence in the effectiveness of the approach by considering a small case study. The details of the particular fixed point algorithm are deferred to the next subsection. We consider here $M = 2$ resources, each having mean service time $E[X_1] = E[X_2] = 1$, arranged in a cyclic network topology. The population is $N = 10$ requests. Resource 1 has exponentially distributed service times, while resource 2's SCV and $SKEW$ are chosen as follows. We progressively increase SCV from 1 to 64 by powers of 2, while simultaneously setting $SKEW$ to the minimum possible value supported by a $PH(2)$ distribution with the specified SCV[5]. As a result of this parametrization, $SKEW$ grows from 2 to 12, thus remaining representative of the web service traces studied in Section 3. Figure 4 illustrates experimental results. Figure 4(a) shows the exact throughput (exact) computed by numerical solution of the underlying CTMC and the product-form solution of an exponential queueing network (pfexp), solved by the convolution algorithm. As expected, the increase in SCV results in a change of the system throughput that is not reflected in the exponential queueing network solution. Conversely, fp-exact and fp-pfexp are the solutions returned by the fixed point approach described above when the exact and pfexp throughput estimates are used to set up the initial throughput value used by the algorithm. It is found that, regardless of the initial point chosen, the fixed-point method converges to the same solution, which follows the exact trend with a very small approximation error (average 1.85%, maximum 3.56%). Figure 4(b) illustrates the additional time needed to converge to the fixed

[5]Minimum skewness for a PH(2) distribution may be obtained by imposing $E[X^3] = 6E[X]^3(h_2^2 + h_3)$, where $h_2 = (E[X^2]/2 - E[X]^2)/E[X]^2$ and $h_3 = h_2(1 - h_2 - 2\sqrt{-h_2})$ [14].

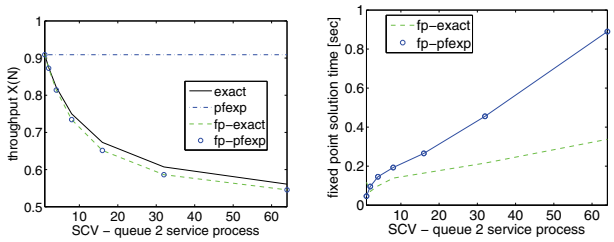


Figure 4: Fixed point algorithm result for a tandem model with $M = 2$ queues. Queue 1 has exponential service times. Population is $N = 10$.

point as a function of the initial throughput value. In all cases a fixed point is found and the execution time in MATLAB is less than 1 second on a laptop PC. For $N = 1000$ jobs, the execution time remains less than 4 seconds.

5.2.3 Fixed-Point Algorithm

The fixed-point algorithm comprises the following steps:
Step 0a. Initialize the iteration number to $i = 0$, set $I \geq 1$ as the maximum number of iterations and $\epsilon_{tol} > 0$ as the tolerance level for acceptable convergence.
Step 0b. Determine an initial throughput guess $X^{(0)}(N)$ by solving a corresponding queueing network model where all stations have exponential service times (with the same mean value) instead of PH. This can be done easily using the convolution algorithm or the mean value analysis (MVA) algorithm [5, 16].
Step 1. Increase the iteration number by setting $i := i + 1$.
Step 2. Compute $\rho_k^{(i-1)} = X^{(i-1)}(N)V_kS_k$, for $1 \leq k \leq M$.
Step 3. Determine the caudal characteristics $\eta_k^{(i)}$ for iteration i by application of the matrix geometric method to the $PH/PH/1$ queues with PH arrival process given by $(\boldsymbol{\alpha}_j, \rho_j^{(i-1)}\boldsymbol{T}_j)$ and PH service process given by $(\boldsymbol{\alpha}_k, \boldsymbol{T}_k)$, for all resources $1 \leq k \leq M$.
Step 4. Solve the PH network parameterized by $X^{(i-1)}(N)$ by computing its normalizing constant using Proposition 1. Then obtain the new utilization estimates

$$\widetilde{\rho}_k^{(i)} = 1 - \frac{(1 - \rho_k^{(i-1)})G(M_k, N)}{G(M, N)}$$

for all resources $1 \leq k \leq M$.
Step 5. Obtain a new throughput estimate $X^{(i)}(N)$ using the average of the throughputs predicted at the resources as follows:

$$X^{(i)}(N) = \sum_{k=1}^{M} \frac{1}{M} \left(\frac{\widetilde{\rho}_k^{(i-1)}}{V_k S_k} \right)$$

Step 6. If $|X^{(i)}(N) - X^{(i-1)}(N)| \leq \epsilon_{tol}$, the algorithm has converged adequately to a fixed point. Terminate, returning the estimate $X(N) = X^{(i)}(N)$. Conversely, if $|X^{(i)}(N) - X^{(i-1)}(N)| > \epsilon_{tol}$, go to Step 1 if $i \leq I$; go to Step 7 otherwise.
Step 7. The algorithm has not converged to a fixed point; terminate returning the sequence average $\sum_{i=1}^{I} X^{(i)}(N)/I$, together with an error message.

5.3 Generalizations

We now develop generalizations of the class of PH queueing networks considered in the previous subsection.

5.3.1 Infinite-server scheduling

Our first generalization involves the integration of $-/G/\infty$ queues, where requests can always be served in parallel thanks to the presence of an ample number of servers. Such resource models are commonly used in queueing networks to describe constant delays on the end-to-end path of a request or to model user think times. In the rest of the paper, we refer to infinite server stations as *delay stations* or simply *delays*.

We integrate delays in (7) as follows. Assume station indices are labeled such that $k = 1, \ldots, D$ are delay stations while $k = D+1, \ldots, D+M$ are PH queues. Then $Z = \sum_{k=1}^{D} V_k S_k$ represents the cumulative mean service time spent by each request in delay stations prior to completing at the reference resource M. We propose to replace the first D delay stations by a single resource with index $k = 0$ having mean service time Z and $V_0 = 1$ visits, and to rename all the station indexes in order to range between 0 and M. Then $\rho_0 = ZX(N)$ is the mean number of requests in the delay stations at steady-state.

We can account for the state of the delays in (7) using the $M/G/\infty$ marginal queue-length distribution. That is, we revise (7) as follows:

$$\Pr[\boldsymbol{n}] = \frac{\sum_{(n_0,n_1,\ldots,n_M)\in S_0} D(n_0) \prod_{k=1}^{M} F_k(n_k)}{G(M,N)},$$

where the state space is now

$$S_0 = \Big\{(n_0, n_1, \ldots, n_M) : \textstyle\sum_{k=0}^{M} n_k = N, n_k \geq 0, n_k \in \mathbb{N}_0\Big\},$$

and the factor for the delay is

$$D(n_0) = \frac{\rho_0^{n_0}}{n_0!},$$

where the local normalizing term $e^{-\rho_0}$ is factored into the network's normalizing constant. The computation of the normalizing constant now follows as before, the only difference being that (9) is replaced by the condition $G(0,n) = D(n)$, for $1 \leq n \leq N$. The fixed-point algorithm is modified according to the assumption that, for the delay station, the inter-arrival time and inter-departure time distributions are identical. This assumption is exact in the limiting case of a deterministic delay. Thus, if station k is fed by a delay station j, which in turn is fed by resource v, we assume that the distribution of arrivals at k is the inter-departure distribution of v. We have observed that in models where the flow was assumed to be Poisson, the throughput approximation errors were about 5-10 times larger than according to the above approximation.

5.3.2 Processor-sharing scheduling

The analysis of processor-sharing resources is simple thanks to the equivalence at steady-state between the queue length metrics of the $M/G/1$ processor-sharing queue and of the corresponding $M/M/1$ first-come-first-served queue with same utilization. We assume that this result remains valid when queues are embedded in PH queueing networks. This is, in general, an approximation since the input process at a resource may not be Poisson anymore.

Stemming from this idea, the product-form factor in (7) for a processor-sharing queue is defined to be

$$F_k(n_k) = (1-\rho_k)\rho_k^{n_k}, \quad n_k \geq 0,$$

which is the marginal queue-length distribution of a $M/M/1$ first-come-first-served queue with utilization ρ_k. Note that it is sufficient to set $\eta_k = \rho_k$ to use the same implementation of both the normalizing constant algorithm and the fixed-point iteration for both first-come-first-served and processor-sharing queues.

5.3.3 General non-cyclic topologies

General topologies introduce complications relating to the splitting and joining of request flows. Consider two queues i and j that feed queue k and assume that they have utilizations ρ_i and ρ_j respectively, such that their departure flows may be approximated as PH renewal processes $(\boldsymbol{\pi}_i, \rho_i \boldsymbol{T}_i)$ and $(\boldsymbol{\pi}_j, \rho_j \boldsymbol{T}_j)$. Further, let $p_{i,k}$ and $p_{j,k}$ be the routing probabilities to queue k for jobs departing from queues i and j, respectively. Then the joined flow seen as input to queue k is *not* in general i.i.d. For example, the joined flow is negatively correlated when the distributions of the PH renewal processes are Erlang.

The problem can be addressed by revising the definition of the block matrices in the QBD for queue k. Let $\otimes$ and $\oplus$ be the Kronecker product and sum operators respectively and let U, V and K be the numbers of phases of the PH service processes of queues i, j and k, respectively. Then the QBD may be written as

$$\boldsymbol{Q}_k = \begin{bmatrix} \boldsymbol{L}_0 & \boldsymbol{F} & 0 & 0 & 0 & \ldots \\ \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & 0 & 0 & \ldots \\ 0 & \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & 0 & \ldots \\ 0 & 0 & \boldsymbol{B} & \boldsymbol{L} & \boldsymbol{F} & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where

$$\begin{aligned} \boldsymbol{L}_0 &= (\rho_i \boldsymbol{T}_i \oplus \rho_j \boldsymbol{T}_j) \otimes \boldsymbol{I}_K \\ \boldsymbol{L} &= \rho_i \boldsymbol{T}_i \oplus \rho_j \boldsymbol{T}_j \oplus \boldsymbol{T}_k \\ \boldsymbol{F} &= (p_{i,k}\rho_i \boldsymbol{t}_i \boldsymbol{\alpha}_i \oplus p_{j,k}\rho_j \boldsymbol{t}_j \boldsymbol{\alpha}_j) \otimes \boldsymbol{I}_K \\ \boldsymbol{B} &= \boldsymbol{I}_{U\cdot V} \otimes \boldsymbol{t}_k \boldsymbol{\alpha}_k \end{aligned}$$

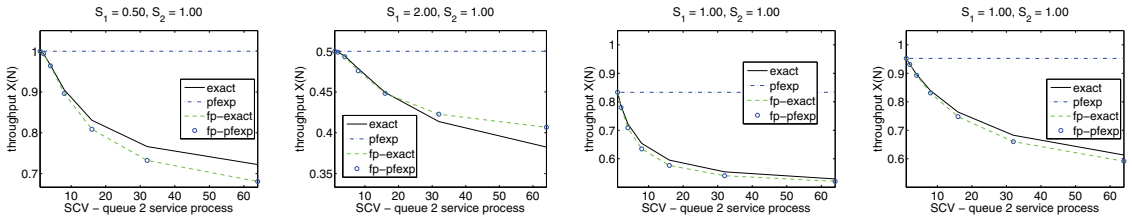
in which $\boldsymbol{I}_n$ is the identity matrix of order n and $\boldsymbol{t}_u = -\boldsymbol{T}_u \boldsymbol{1}$. Solving for the caudal characteristic η provides an immediate generalization of the approach used for a cyclic PH network to a general network topology.

6. VALIDATION

In this section, we study the accuracy of the proposed approximations for networks with increasing size and different service time distributions.

6.1 $M = 2$ resources

We first investigate the accuracy of the proposed approximation method on a network with $M = 2$ resources. We perform experiments similar to the ones depicted in Figure 4, but for different parameterizations of the model. Figure 5(a) and Figure 5(b) illustrate the throughput error when the rates of the two stations are unbalanced. The mean service rate is shown at the top of both figures. The results indicate that for low SCV values the approximation is excellent, whereas at high SCV, there is some deviation between the



(a) exp+ph, $N = 10$ (b) exp+ph, $N = 10$ (c) exp+ph, $N = 5$ (d) exp+ph, $N = 20$

Figure 5: Sensitivity to values of mean service time and to population size; the model has $M = 2$ queues.

exact results and the approximations. The initial throughput value used for the fixed point algorithm is unimportant since fp-exact and fp-pfexp yield identical results. This was also observed in all the other experiments in this section. Quantitatively, the deviation at $SCV = 64$ is just 5.74% in Figure 5(a) and 6.35% in Figure 5(b). Once again, the results for corresponding product-form exponential networks are much worse, with an error of 38.41% in Figure 5(a) and 30.66% in Figure 5(b).

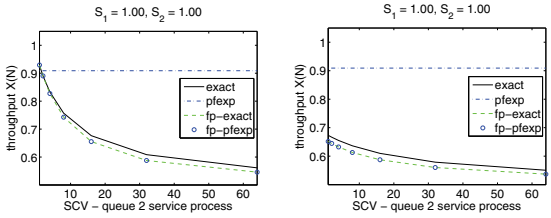
Notice that in all experiments, the maximum resource utilization is $U_{max} = \max_k S_k X(N)$, and Figure 5(a) and Figure 5(b) span a utilization range for the bottleneck queue from 0.70 to 1.00. Figure 5(c) and Figure 5(d) illustrate sensitivity to the network load by altering the number of requests N to $N = 5$ and $N = 20$, respectively. The results indicate that the accuracy of the method is quite insensitive to such changes in the job populations.

Figure 6(a) and Figure 6(b) illustrate the sensitivity of models when resource 1 is set, respectively, to have either an Erlang-2 distribution or a PH distribution identical to the one used in resource 2. Notice that for $SCV = 1$ the exact result no longer matches the exponential case, the first resource no longer being exponential. The former case is indeed very important for web services due to our observation in Section 3 that about 50% of recorded experiments have $SCV < 0.52$. The results in the two figures suggest again that the accuracy of the method is quite insensitive to the change in SCV.

Figure 7(a) shows how the accuracy of the method is affected by the skewness of the distribution. It is clear that, as the skewness raises above the $SKEW \leq 12$ boundary that we considered in Section 3, the scalar approximation we have proposed for the $PH/PH/1$ queue is no longer valid. It is still interesting to note, however, that most of the experiments with large skewness show a good agreement with the product-form exponential solution. This suggests that resources with high-skewness might be approximated as exponential resources without incurring major errors.

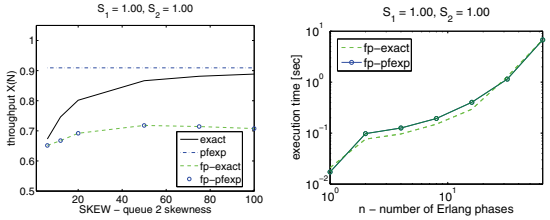
Figure 7(b) shows instead the growth of computational costs as the number of phases increases. For simplicity of parameterization, we consider an Erlang-n process at resource n and explore the values $n = 1, 2, 4, 8, 16, 32$. It is found that a parsimonious PH description with up to 8 states provides execution times that are less than 1 second. For $n = 16$ the time grows to 1.28 seconds and for $n = 32$ it grows to 6.80 seconds. Closer examination of execution traces reveals that the dominating component of the execution time is the matrix-geometric solution used to compute η_k for all resources.

Figure 8 shows sensitivity results for the case where a



(a) erl2+ph, $N = 10$ (b) ph+ph, $N = 10$

Figure 6: Sensitivity to choice of distribution at resource 1; the model has $M = 2$ queues.



(a) exp+ph, $N = 10$ (b) exp+erl, $N = 10$

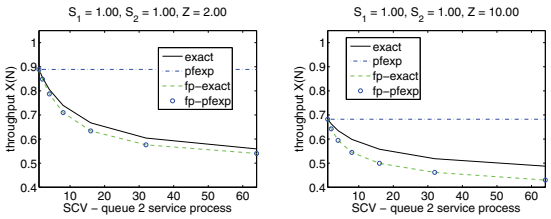
Figure 7: Sensitivity to skewness of distribution and to number of phases; the model has $M = 2$ queues.

delay station is added to the cyclic network, placed after resource 2. The population is $N = 10$. The two cases consider average delays $Z = 2$ and $Z = 10$, respectively resulting in worst-case approximation errors of 3.4% and 11.8%, which are much better than the corresponding errors in product-form models of 59.0% and 40.0%.

The execution times for fp-exact and fp-pfexp in all experiments in this subsection were less than 1 second on MATLAB. Memory requirement was negligible as well – of the order of kilobytes.

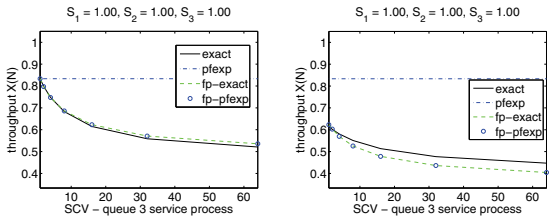
6.2 $M = 3$ resources

We next assessed the sensitivity of the results against the size of the network. Figure 9(a) and Figure 9(b) illustrate two cyclic networks with one or two resources having PH distributed service times. Notice that the range of maximum utilization is larger than in the previous experiments, especially in Figure 9(b), where it is between 0.4 and 1.00. We see that increasing the number of resources results in more accuracy in Figure 9(a), where the maximum error is just 2.84% for the fixed point method against 60.0% for the exponential product-form solution. In Figure 9(b), the errors grow to 6.13% for the fixed point and 109.1% for the



(a) exp+ph, $N = 10$ (b) exp+erl, $N = 20$

Figure 8: Sensitivity to delay stations; the model has $M = 2$ queues and a delay station.



(a) exp+exp+ph, $N = 10$ (b) exp+ph+ph, $N = 10$

Figure 9: Sensitivity to choice of distribution at resource 1; the model has $M = 3$ queues.

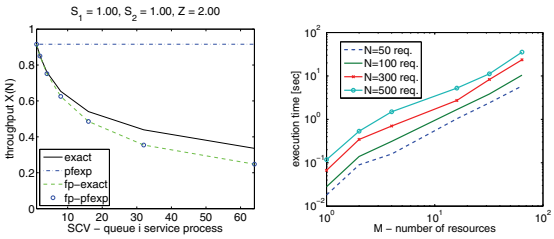
exponential product-form solution. Such a difference from the case in Figure 9(a) may be explained by observing that, when more queues are PH, the request-flows between resources tend to deviate more markedly from a Poisson process. Such a situation cannot be handled by exponential networks[6], whereas our fixed-point method accounts for it effectively, thanks to the scaled input processes $(\boldsymbol{\alpha}_j, \rho_j \boldsymbol{T}_j)$. To test this conjecture, we repeated the experiment in Figure 9(b), replacing the scaled input processes $(\boldsymbol{\alpha}_j, \rho_j \boldsymbol{T}_j)$ by an exponential inter-arrival time with the same mean. Thus, the caudal characteristics η_k are for $M/PH/1$ queues rather than $PH/PH/1$ queues. It is found that the fixed point solution error grows to 37.69% for $SCV = 64$, so it is about 600% larger than with the $PH/PH/1$ approach. For the model in Figure 9(a), the degradation is similar, at 32.9%. These additional experiments provide robust evidence that the proposed approach is effective in describing the distribution of inter-departure times from resources.

Finally, Figure 10 shows sensitivity experiments similar to the ones developed in Figure 5. As before, no major deviations from the exact solution are observed. As with the cases of $M = 2$ resources, for $M = 3$, the execution times for fp-exact and fp-pfexp in all experiments were less than 1 second.

6.3 Large intractable models

We now illustrate the accuracy and scalability of the fixed-point method on models that are intractable by direct solution of the Markov chain. Figure 11(a) shows results for a model with $N = 100$ requests and $M = 10$ resources. The number of states in the underlying Markov process is 4.366×10^{15}, which is clearly intractable. All service times at the queueing resources follow an identical PH(2) distribution with the specified SCV. The exact solution is computed by simulation, using the Java Modelling Tools suite simulator [2], configured with the independent replication method and 95% confidence intervals. Notice that utilization is identical for all resources and equal to the throughput since $S_k = 1$, for $1 \leq k \leq M$. Results indicate that the method is very accurate for $SCV \leq 10$ when the network is more heavily-loaded, whilst it gives slightly larger errors than in the previous examples at large $SCVs$. Overall, the trend is captured fairly well, especially when compared to the pfexp method. We have performed additional experiments and observed that as the bottleneck load grows, performance appears to be captured better at large SCV. Thus, it appears that the proposed approximation tends to perform better at medium/high loads. This is consistent with the discussion in Section 3, since under light load, the different decay rate at queue-length 1 (i.e., $\rho(1-\eta)/(1-\rho)$ as opposed to η) may reasonably become dominant.

[6]Notice that in closed exponential networks the flows are not Poisson either. However, empirical observations suggest that their variability is usually not too far from $SCV = 1$.



(a) 10×ph, $N = 100$ (b)

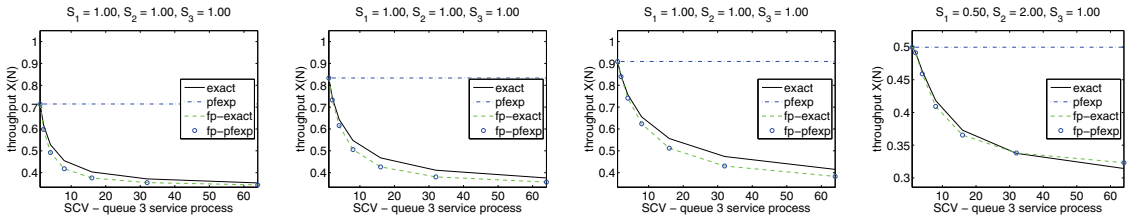
Figure 11: Sensitivity to model size.

Figure 11(b) shows computation times for large models with N up to 500 requests and networks of increasing size, M. In all cases, the fixed-point method can solve the model in a few seconds on a laptop computer. In particular, the execution times scale very efficiently with the population N, thanks to the $O(N)$ complexity of the normalizing constant computation.

7. SERVICE MANAGEMENT EXAMPLE

Finally, we introduce a case study of practical interest for the proposed class of performance models. Consider an application that defines a set of enterprise Java beans (EJBs) to deal with the business logic. Each EJB acquires data from a pool of C connection objects that represent entry-points for web services and database resources, collectively referred to as *data sources*. Assume that there are D data sources and there exist one or more dedicated connection objects for each data source. Thus $C \geq D$ and we denote by C_d the number of connection objects for data source d, $1 \leq d \leq D$. Assume also that queueing delays for outstanding calls at data sources are negligible and that each connection object stores the pending calls it will serve in a first-in-first-out buffer.

We consider the problem of allocating residual bandwidth at run-time by instantiation of new connection objects. Indeed, as the number of connection objects grows, more data can be fetched in parallel per unit of time from data sources and thus increase the application throughput if data acquisition is the performance bottleneck. However, a physical bandwidth limit exists which calls for deciding which data source to prioritize. For simplicity, we focus here on identify-



(a) ph+ph+ph, $N = 5$ (b) ph+ph+ph, $N = 10$ (c) ph+ph+ph, $N = 20$ (d) ph+ph+ph, $N = 10$

Figure 10: Sensitivity to values of mean service time and to the population size; the model has $M = 3$ queues.

ing the data source d^* that would be able to exploit best the greatest portion of the residual bandwidth, and thus acquire more data per time unit.

The decision problem may be tackled by studying a closed queueing network with N circulating requests representing the number of calls issued by EJBs to the connection objects. Define Z_d to be the average time that elapses between completion of a call to data source d and the successive arrival of a new call to that data source. Such a delay may be due to several factors, such as gaps in the arrival stream of requests, time taken by EJBs to process the business logic, level of parallelism for software worker threads used by the application. We model the system as a cyclic network with a first-come-first-served queue, representing the new connection object, a processor-sharing queue, representing available bandwidth, and a delay server with exponential rate Z_d^{-1}. Let R_d be the random variable for the response time of data source d and let $E[B_d]$ be the average data size in bits of a response from data source d over the network. Further, denote by μ the network bandwidth in bits per second and by $\rho_{net,d}$ the current network utilization due to calls to data source d. The average time to transfer data from source d may be estimated as $S_{net,d} = E[B_d]\mu^{-1}$. Since we consider single class models and the network utilization is available at run-time, we consider the scaled quantity $S^*_{net,d} = ((C_d + 1)/C_d)S_{net,d}(1 - \sum_{i \neq d} \rho_{net,i})^{-1}$. Here the utilization scaling factor accounts for the delay due to shared network bandwidth under the assumption that class d will not affect the bandwidth allocated to class $i \neq d$. Instead, the factor $((C_d + 1)/C_d)$ estimates the extra demand placed on the network by a new connection for source d. The example is based on two web service time-traces from the wsdream dataset, having respectively $E[R_1] = 412.43ms$, $SCV[R_1] = 22.35$, $SKEW[R_1] = 9.96$ and $E[R_2] = 661.00ms$, $SCV[R_2] = 0.50$, $SKEW[R_2] = 9.30$, which are fitted by PH(2)s. Thus, data source 1 has high-variability in its response while data source 2 has low variability. The other parameters used in the experiments are given in Table 1.

Notice that source $d = 1$ has the highest demand on network bandwidth and the smallest delay Z_d; thus the choice $d^* = 1$ seems natural. However, we find that low variability makes the choice $d^* = 2$ a better one. Simulation and analytical results are shown in Figure 12. Execution time for the fixed-point algorithm is just $4ms$ per experiment, as opposed to simulation that takes about $10s$ to converge. As we can see, the exponential network model pfexp, which cannot represent high-variability, predicts that an additional connection object would roughly provide the same bandwidth utilization for both data sources, with a slight preference for

$D = 2$	$C_1 = 1$	$C_2 = 1$
d	1	2
N	10	10
Z	$400ms$	$430ms$
$S_{net,d}$	$250ms$	$100ms$
$\rho_{net,d}$	0.5242	0.1518

Table 1: Model parameters.

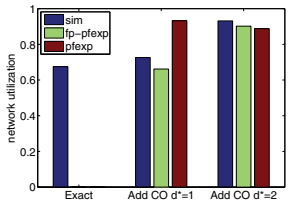


Figure 12: Application example results.

$d = 1$. Conversely fp-pfexp, the fixed-point algorithm initialized with the pfexp solution, correctly predicts that a benefit can be achieved only if the additional connection object is for data source $d = 2$. This is because the high variance of data source 1 would often block the line of requests queueing at the connection object buffer.

Summarizing, this small, but realistic, example shows that the proposed class of models may return surprising – but correct – decisions compared to those suggested by commonly used exponential models. Such predictions are obtained in negligible time and so are compatible with application to run-time decision problems of far greater complexity than that of this exemple.

8. CONCLUSION

We have presented a class of product-form expressions that approximate a diverse range of closed queueing networks with resources having generally distributed processing times. When the skewness of the distribution is not too large (e.g., $SKEW \leq 12$), it was found that the accuracy of the approximation is excellent at all levels of variability, as characterized by the second moment. A fixed-point algorithm that obtains such approximate solutions cheaply in terms of both time and space requirements has been implemented and its application to run-time service management has been illustrated. Future work will focus on the generalization of the proposed method to multi-class workloads as well as load-dependent and multi-server resources.

9. REFERENCES

[1] S. Asmussen and F. Koole. Marked point processes as limits of Markovian arrival streams. *J. Appl. Prob.*, 30:365–372, 1993.

[2] M. Bertoli, G. Casale, and G. Serazzi. User-friendly approach to capacity planning studies with Java Modelling Tools. In *Proc. of SIMUTools 2009*, pages 1–9. ACM, 2009.

[3] D. Bini, B. Meini, S. Steffé, and B. Van Houdt. Structured Markov chains solver: software tools. In *Proc. of SMCTOOLS Workshop*. ACM, 2006, `http://win.ua.ac.be/~vanhoudt/`.

[4] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains*. 2nd ed., Wiley and Sons, 2006.

[5] J. P. Buzen. Computational algorithms for closed queueing networks with exponential servers. *Comm. of the ACM*, 16(9):527–531, 1973.

[6] G. Casale. A note on stable flow-equivalent aggregation in closed networks. *Queueing Systems*, 60(3-4):193–202, 2008.

[7] G. Casale, N. Mi, and E. Smirni. Bound analysis of closed queueing networks with workload burstiness. In *Proc. of ACM SIGMETRICS 2008*, pp. 13–24. ACM Press, 2008.

[8] G. Casale, N. Mi, L. Cherkasova, and E. Smirni. Dealing with burstiness in multi-tier applications: new models and their parameterization. *IEEE Trans. Software Eng.*, to appears.

[9] G. Casale, M. Tribastone. Fluid Analysis of Queueing in Two-Stage Random Environments. in *Proc. of QEST*, Aachen, Germany, Sep 2011.

[10] P. Courtois. Decomposability, instabilities, and saturation in multiprogramming systems. *Comm. of the ACM*, 18(7):371–377, 1975.

[11] D. L. Eager, D. Sorin, and M. K. Vernon. AMVA techniques for high service time variability. In *Proc. of ACM SIGMETRICS*, pages 217–228. ACM Press, 2000.

[12] E. Gelenbe and I. Mitrani. *Analysis and Synthesis of Computer Systems*. Academic Press, London, 1980.

[13] A. Heindl. *Traffic-Based Decomposition of General Queueing Networks with Correlated Input Processes*. Ph.D. Thesis, Shaker Verlag, Aachen, 2001.

[14] A. Heindl, G. Horvath, and K.Gross. Explicit Inverse Characterizations of Acyclic MAPs of Second Order. Proc. of *EPEW*, Springer LNCS 4054, 108–122, 2006.

[15] H. Kobayashi. Application of the diffusion approximation to queueing networks I: equilibrium queue distributions. *Journal of the ACM*, 21(2):316–328, 1974.

[16] E. D. Lazowska, J. Zahorjan, G. S. Graham, and K. C. Sevcik. *Quantitative System Performance*. Prentice-Hall, 1984.

[17] T. Marian, M. Balakrishnan, K. Birman, and R. van Renesse. Tempest: Soft state replication in the service tier. In *DSN*, pages 227–236. IEEE Computer Society, 2008.

[18] N. Mi, Q. Zhang, A. Riska, E. Smirni, and E. Riedel. Performance impacts of autocorrelated flows in multi-tiered systems. *Perform. Eval.*, 64(9-12):1082–1101, 2007.

[19] M. F. Neuts. The caudal characteristic curve of queues. *Adv. Appl. Prob.*, 18:221–254, 1986.

[20] M. F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Marcel Dekker, New York, 1989.

[21] B. Pittel. Closed exponential networks of queues with saturation: the Jackson-type stationary distribution and its asymptotic analysis. *Math. Oper. Res.*, 4:357–378, 1979.

[22] M. Reiser. A queueing network analysis of computer communication networks with window flow control. *IEEE Trans. on Communications*, 27(8):1199–1209, 1979.

[23] A. Riska and E. Smirni. MAMsolver: A matrix analytic methods tool. In *Proc. of TOOLS, pp. 205–211*. Springer-Verlag, 2002, `http://www.cs.wm.edu/MAMSolver/`.

[24] A. Riska, V. Diev, and E. Smirni. An EM-based technique for approximating long-tailed data sets with PH distributions. *Perform. Eval.*, 55(1-2):147–164, Jan. 2004.

[25] R. Sadre and B. R. Haverkort. Fifiqueues: Fixed-point analysis of queueing networks with finite-buffer stations. In *Proc. of MMB*, pages 77–80, 1999.

[26] W. Whitt. The queueing network analyzer. *The bell system tech. journal*, 62(9):2779–2815, Nov. 1983.

[27] J. Zahorjan, E. D. Lazowska, and R. L. Garner. A decomposition approach to modelling high service time variability. *Perform. Eval.*, 3:35–54, 1983.

[28] Z. Zheng and M. R. Lyu. Collaborative reliability prediction of service-oriented systems. In *Proc. of ACM ICSE*, pp. 35–44, May 2010.

APPENDIX

A. ASYMPTOTIC DISTRIBUTIONS

The exact asymptotic distribution of the conditional distribution in a $PH/PH/1$ queue is known to exist from equation (2), i.e.,

$$\begin{aligned}\widetilde{\boldsymbol{\theta}} &= \boldsymbol{\theta} = \lim_{n\to\infty} \frac{\boldsymbol{\pi}(n)}{\boldsymbol{\pi}(n)\mathbf{1}} \\ &= \lim_{n\to\infty} \frac{\boldsymbol{\pi}(1)\boldsymbol{R}^{n-1}}{\boldsymbol{\pi}(1)\boldsymbol{R}^{n-1}\mathbf{1}} \\ &= \lim_{n\to\infty} \frac{\boldsymbol{\pi}(1)\eta^{n-1}\boldsymbol{\Pi}}{\boldsymbol{\pi}(1)\eta^{n-1}\mathbf{1}} \\ &= \frac{\boldsymbol{\pi}(1)\boldsymbol{r}}{\boldsymbol{\pi}(1)\mathbf{1}}\boldsymbol{\ell}\end{aligned}$$

Thus, $\boldsymbol{\theta}$ is a left-Perron-eigenvector of $\boldsymbol{R}$ (parallel to $\boldsymbol{\ell}$) and so, by the previous assumption, $\boldsymbol{\theta}\boldsymbol{R} = \boldsymbol{\theta}\eta\boldsymbol{\Pi} = \eta\boldsymbol{\theta}$. Thus $\boldsymbol{\theta}\boldsymbol{\Pi} = \boldsymbol{\theta}$. Notice that $\boldsymbol{\Pi}$ is the spectral projector for the $\boldsymbol{R}$ matrix associated with the *exact* solution $\boldsymbol{\pi}(n)$.

B. PROOF OF PROPOSITION 1

The normalizing constant may be written as

$$G(M,N) = \sum_{S}\prod_{i=1}^{M}[(1-\rho_i)(1-\delta(n_i)) + \rho_i(1-\eta_i)\eta_i^{n_i-1}\delta(n_i)]$$

where $\delta(n_i) = 1$ if $n_i \geq 1$, 0 otherwise. Observe now that

$$\begin{aligned}G(M,N) = &\sum_{(n_1,\ldots,n_M):n_M=0} (1-\rho_M)\prod_{i=1}^{M-1} x_i(n_i) \\ &+ \sum_{(n_1,\ldots,n_M):n_M\geq 1} \rho_M(1-\eta_M)\eta_M^{n_M-1}\prod_{i=1}^{M-1} x_i(n_i) \quad (12)\end{aligned}$$

where $x_i(n_i) = (1-\rho_i)(1-\delta(n_i)) + \rho_i(1-\eta_i)\eta_i^{n_i-1}$. Factoring out of the two summations $(1-\rho_M)$ and ρ_M, respectively, the right-hand side terms are, by definition, $G(M-1,N)$ and $G^{aux}(M,N-1)$, respectively. The terminations conditions also follow immediately by definition of G and G^{aux}. □