Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

IDENTIFICATION AND APPROXIMATIONS FOR SYSTEMS WITH MULTI-STAGE WORKFLOWS

Parijat Dube
Jian Tan
Li Zhang

IBM T. J. Watson Research Center
19, Skyline Drive
Hawthorne NY 10532, USA

ABSTRACT

Distributed systems with multi-stage workflows are characterized by multiple logical stages which can either execute sequentially or concurrently and a single stage can be executed on one or more physical nodes. Knowing the mapping of logical stages to physical nodes is important to characterize performance and study resource bottlenecks. Often due to the physical magnitude of such systems and complexity of the software, it is difficult to get detailed information about all the system parameters. We show that under light load conditions, the system can be well approximated using first order models and the hence simplifying the system identification problem. For general load, we develop a parameter estimation technique using maximum likelihood and propose a heuristic to solve it efficiently.

1 INTRODUCTION

The increase in scale and complexity of information processing in modern distributed systems has raised many challenging problems. This study investigates a system that is designed to process many job flows on a cluster of hosts. A job flow needs to go through multiple logical processing stages before it is deemed complete. A logical stage can correspond to, for example, a function call in the source code. Each logical processing stage is handled by one or more nodes in the system. These nodes can be physical computers or virtual machines running on shared physical hardware. Each logical stage of a job consumes one or more different physical resources (CPU, memory, I/O, disk) on one or more nodes. Some of the logical stages can be processed in parallel, while some of the logical stages have to be processed in sequential order. The system is required to handle many requests at the same time in a scalable fashion in order to fully utilize all its resources and achieve a high level of throughput. Examples of such distributed processing systems include systems processing complex business process workflows, information/data stream processing systems, management and provisioning systems. The number of processing stages for each request is on the order of several hundreds. Understanding the scalability issues in systems of this magnitude becomes challenging.

Take the streaming computing system for example. The data flow needs to go through a series of operators that possibly run on a set of hosts, including data query, content searching, business analytics, data presentation, etc. From logic perspective, it seems that the operations on different stages of a flow can be isolated and different flows do not interfere each other. However, when the data flow is laid out on a physical cluster, these different stages of a flow can happen to reside on the same host, sharing limited resources (e.g., CPU, memory) and possibly causing performance downgrade. This problem becomes even complicated when running a streaming computing application on virtual machines that reside on fewer physical machines. Due to the abstraction of the virtual machines, it is not clear whether several flows are sharing the same CPU on a physical machine or not.

An important step in identifying bottleneck physical resources involves discovering the mapping between the logical stage and the physical resources involved in the execution of that particular stage. Since most of the software design now-a-days involve use of freely available, pluggable component binaries, there is a lack of clear knowledge of all the logical stages involved in end-to-end flow of a job request let alone knowing the actual physical server(s) and their resources involved in their execution. Matching the logic flow steps onto the physical layout can provide useful information for performance management and prediction. For example, with this knowledge, one may improve the system performance by answering questions like: what the processing time of flow B will be if we increase the input rate of flow A? Should we redeploy the computing flow tenants on the physical hosts to better utilize the resources? Clearly, if flow A and flow B do not share any physical computing resources, they will not impact each other. However, in complex systems, it is very often difficult to tell whether different flows can interfere or not.

Multi-class queueing network is a standard technique to model behavior of systems with different types of flows. Analytical and simulation techniques are then used to study these models. An important step in building these models is identifying the values of system parameters in the model, like the path followed by different flows inside the system and the service time of different flows at different physical nodes in its path. Often these parameters can be inferred by doing a best fit of the parametric model with the measurement data. These problems are typically formulated as optimization problems minimizing some measure of difference between measured value and analytical value from the model. Due to the very nature of queueing behavior and the complexity of interactions between flows and physical servers such problems are non-linear constrained optimization problems with many unknowns.

We provide approximate models for the system under light traffic. These low-order models can be used to simply the problem of inferring model parameters from measurement data. Especially using first order approximations, we show that to study effect on performance due to marginal changes in arrival rate of a flow, the end-to-end delay of the flow can be modeled with an affine function of the flow arrival rate. If we have a detailed discrete event simulation model it may take long time to simulate. The proposed approximate models can also be used for simpler simulations and model analysis. For general load, we develop a Maximum Likelihood Estimation (MLE) formulation for parameter (service time of flows at physical nodes and their path) inference of the queueing network and propose a heuristic approach to solve it efficiently.

The paper is structured as follows. We formulate the system identification problem in Sec. 2. An approach for system identification for a lightly loaded system is developed in Sec. 3 where we also do some numerical comparison of our approximations. For a stationary system under arbitrary load, we formulate the system identification problem using Maximum Likelihood Estimation (MLE) in Sec. 4. We briefly discuss related works in Sec. 5 followed by conclusion in Sec. 6.

2 SYSTEM IDENTIFICATION PROBLEM

Modeling and simulation of an IT system requires information about the physical topology (servers, network and their specifications) and the workload. The workload specifies the different types of flows (or transactions), their arrival rates and their service time at different physical components traversed by them. While its easier to get information about physical topology and to measure end-to-end information related to workload, like arrival rates of different flows, its hard to track path taken by different flows and their service times at different physical components in their end-to-end path. The problem is aggravated in large-scale distributed systems with multi-stage workflows, like stream computing, cloud computing, which typically contain hundreds of stages with sequential and concurrent execution patterns. Due to the complexity of the software and the magnitude of their topology its almost impossible to know with certainty the physical servers where different logical stages of a workflow are executed. This is also because many a times there is dynamic deployment and migration of jobs during run-time. An important step to model system performance is to discover these unknowns for the system. One approach is to use end-to-end measurement data from the system under varying load conditions and then infer these unknowns. The

system can be abstracted using high level queueing network models. The system identification problem (know the path of flows and their service times at different physical servers) then boils down to parameter inference problem.

Let the system be modeled as a multi-class open queueing network with J nodes, indexed by $j = 1, 2, \cdots, J$. Each node is a single server queue of unlimited space with processor sharing discipline. This network is serving F classes of exogenous flows indexed by the subscript $f, 1 \leq f \leq F$, each arriving as an independent Poisson process with rate λ_f. Each flow follows a fixed routing path with finite sequence of nodes. A flow can visit a particular node multiple times bounded by a finite number, implying that the flow finally leaves the system. The amount of work brought by a flow f on node j (also called service time of flow f on node j) has an exponentially distribution with rate μ_{fj}. After completing service at node j, a flow goes to the next node along its routing path until it leaves the network. Let $x_{fj} \geq 0$ be the number of times that flow f visits node j. Let ρ_j be the traffic intensity of node j. We have

$$\rho_j = \sum_{f=1}^{F} x_{fj} \lambda_f s_{fj}. \tag{1}$$

We assume that $\forall j$, $0 \leq \rho_j < 1$ for a stable system. Let T_f denote the average end-to-end response time of a job from flow class f. Under the assumptions, the network is a product form queueing network and hence the expected value T_f can be expressed as:

$$T_f = \sum_{j=1}^{J} \frac{x_{fj} s_{fj}}{1 - \rho_j}. \tag{2}$$

Let $\boldsymbol{\lambda} = \{\lambda_f\}_{1 \leq f \leq F}$ be the vector of arrival rates of different flows. Now, for a sequence of K different input rate vector $\boldsymbol{\lambda}^k$, $k = 1, 2, \cdots, K$, we have the corresponding measurements on the average end-to-end response times for each flow, $\{T_f^k\}_{1 \leq f \leq F}$.

A natural question is: given the end-to-end delays $\{T_f^k\}_{1 \leq f \leq F}$ for $k = 1, 2, \cdots, N$ and the corresponding input rate $\boldsymbol{\lambda}^k$, can we infer the set of nodes a flow f traverse and the corresponding the service rate? Specifically, we want to figure out $\{x_{fj}\}_{1 \leq j \leq J}$ and $\{s_{fj}\}_{1 \leq j \leq J}$ for each f, a total of $2 \times J$ parameters for each flow.

In Section 3, we obtain approximations for the system under light load so as to simplify the parameter estimation problem. Then, in Section 4, we infer x_{fj} and s_{fj} by using a sequence of observations on the number of jobs at each node.

3 APPROXIMATIONS UNDER LIGHT LOAD

For the light traffic approximation, the mean delay of a flow on a Kelly network is approximated by the mean delay of the flow on in reduced dimensionality system. We approximate the end-to-end delay of flow f, by the first two terms of the Taylor expansion of T_f, which works well when the load is small. The approximation error also depends on the actual path followed by the flow.

From (1) and (2) we have:

$$\begin{aligned} T_f &= \sum_{j=1}^{J} \frac{x_{fj} s_{fj}}{1 - \sum_{g=1}^{F} x_{gj} \lambda_g s_{gj}} \\ &= \sum_{j=1}^{J} x_{fj} s_{fj} (1 - \sum_{g=1}^{F} x_{gj} \lambda_g s_{gj})^{-1}. \end{aligned} \tag{3}$$

Consider the light load condition such that $\forall j \in J$, the total load is negligible:

$$\sum_{f=1}^{F} x_{fj} \lambda_f s_{fj} << 1. \tag{4}$$

Proposition 1 Under the light load condition (4), the delay of any flow f can be modeled as $a_{0,f}+a_{1,f}\lambda_f$, where $a_{0,f}$ and $a_{1,f}$ are independent of λ_i and given by

$$a_{0,f}=\sum_{j=1}^{J}x_{fj}s_{fj}(1+\sum_{g=1,g\neq f}^{F}x_{gj}\lambda_g s_{gj}),\quad a_{1,f}=\sum_{j=1}^{J}x_{fj}s_{fj}^2. \tag{5}$$

Proof. Approximating $(1-x)^{-1}$ by $1+x$ when $x<<1$, we have from (3):

$$\begin{aligned}T_f\approx\tilde{T}_f &= \sum_{j=1}^{J}x_{fj}s_{fj}+\sum_{j=1}^{J}x_{fj}s_{fj}\sum_{g=1}^{F}x_{gj}\lambda_g s_{gj}\\ &= \sum_{j=1}^{J}x_{fj}s_{fj}+\lambda_f\sum_{j=1}^{J}x_{fj}s_{fj}^2+\sum_{j=1}^{J}x_{fj}s_{fj}\sum_{g=1,g\neq f}^{F}x_{gj}\lambda_g s_{gj},\end{aligned}$$

which is of the form $a_{0,f}+a_{1,f}\lambda_f$, with $a_{0,f}$ and $a_{1,f}$ given by (5). □

Remark 2 The light traffic can be given a queueing interpretation. From (6) we can express $\tilde{T}_f$ as:

$$\tilde{T}_f=\left(\sum_{j=1}^{J}x_{fj}s_{fj}-\sqrt{\sum_{j=1}^{J}x_{fj}s_{fj}^2}+\sum_{j=1}^{J}x_{fj}s_{fj}\sum_{g=1,g\neq f}^{F}x_{gj}\lambda_g s_{gj}\right)+\left(\sqrt{\sum_{j=1}^{J}x_{fj}s_{fj}^2}+\lambda_f\sum_{j=1}^{J}x_{fj}s_{fj}^2\right). \tag{6}$$

Observe that the term under the first set of parentheses in the last equation is positive, since $(\sum_{j=1}^{J}x_{fj}s_{fj})^2\geq\sum_{j=1}^{J}x_{fj}s_{fj}^2$ and is also independent of λ_f. Thus $\tilde{T}_f$ from (6) is of the form $b_{0,f}+b_{1,f}+b_{1,f}^2\lambda_f$, which has the same form as the end-to-end delay in a two-stage queueing network with arrival rate λ_f, where the first stage is a *lightly loaded* $M/M/1$ queue with service time

$$b_{1,f}=\sqrt{\sum_{j=1}^{J}x_{fj}s_{fj}^2},$$

and the second stage is a constant delay node with delay

$$b_{0,f}=\sum_{j=1}^{J}x_{fj}s_{fj}(1+\sum_{g=1,g\neq f}^{F}x_{gj}\lambda_g s_{gj})-\sqrt{\sum_{j=1}^{J}x_{fj}s_{fj}^2}.$$

For this approximation we need $\lambda_f\sqrt{\sum_{j=1}^{J}x_{fj}s_{fj}^2}<<1$. Since we started with (3) it will be interesting to study how this two-stage queueing network equivalence works for other metrics like queue length distribution of jobs from flow f in the original network and the queue length distribution in the two-stage approximate network.

Remark 3 Note that the constant delay only depends on arrival rate from other flows and is independent of the arrival rate of flow i. Hence for each flow, we can study the increase in its end-to-end delay due to marginal increase in its own load. Further under light load, the system identification problem can be formulated as a simple linear regression problem:

$$\begin{aligned}&\min\sum_{k=1}^{N}\left[T_f^k-(b_{0,f}+b_{1,f}\lambda_f)\right]^2\\ &\text{subject to}: b_{0,f},b_{1,f}\geq 0,\end{aligned} \tag{7}$$

which can be efficiently solved using standard techniques. Observe that, now we only need to infer two parameters for each flow instead of $2\times J$ parameters.

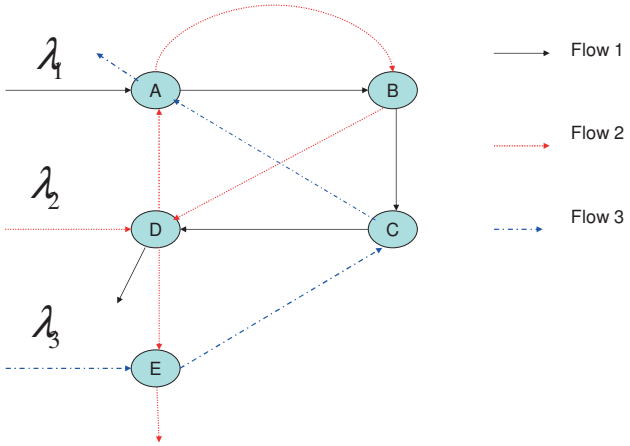

Figure 1: Example system for numerical study.

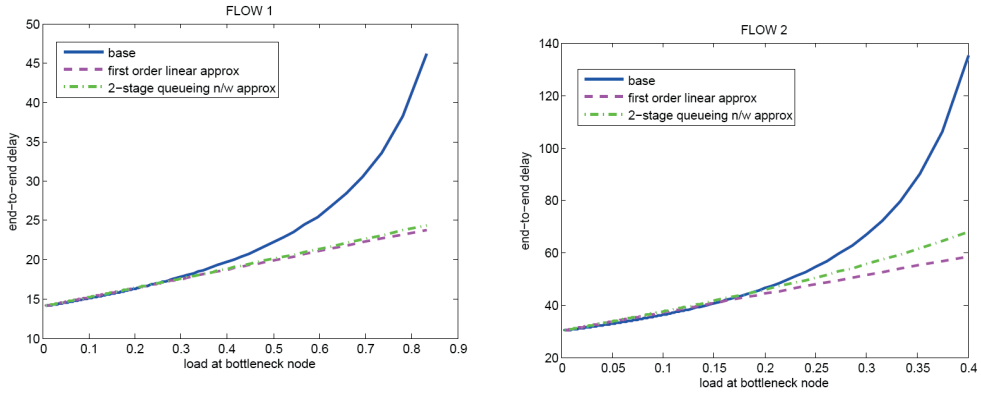

Figure 2: Comparison of actual end-to-end delay for Flow 1 and Flow 2 and their two approximations.

3.1 Numerical Study

We next quantitatively compare the range of load over which our light traffic models approximates the system well. We consider a network with five physical nodes and three flows as shown in Figure 1. For the numerical example we consider a special case where the arrival rate for all flows are identical, i.e., $\forall f, \lambda_f \equiv \lambda$. For this case, a sufficient condition for the light load condition from (4) to be satisfied is:

$$\forall j,\ \lambda \sum_{f=1}^{F} x_{fj} s_j << 1 \Rightarrow \lambda << \min_j \frac{1}{\sum_{f=1}^{F} x_{fj} s_j}. \tag{8}$$

The service time matrix $S = (s_{fj})$ for the example is:

$$S = \begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 4 & 6 & 8 & 10 & 12 \\ 6 & 9 & 12 & 15 & 24 \end{pmatrix}.$$

Note that the smallest service time in S is 2. We increased λ from $2/1800$ to $2/15$. Since $\forall f$, $\lambda_f \equiv \lambda$, changing λ changes arrival rate of all flows by the same amount but the load at different nodes in the network changes by different amounts as the paths followed by flows are different. Figures 2 and 3 compare the

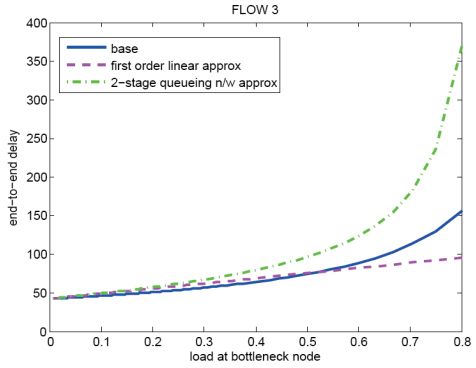

Figure 3: Comparison of actual end-to-end delay for Flow 3 and its two approximations.

actual end-to-end delay (T_f, labeled as *base*) and its two approximations: ($\tilde{T}_f$, labeled as *first order linear approximation*) and the *2-stage queueing network approximation* for the example network. The $x-axis$ is the load at the *bottleneck node* for the particular flow. The bottleneck node for a flow is the physical node in its path that has the maximum total load. Observe that while both the first order linear approximation and the 2-stage queueing network very closely match the base end-to-end delay at low loads (upto 0.2), the delay estimate from the 2-stage network may be optimistic (for Flow 1 and Flow 2) or conservative (for Flow 3) at higher loads. So appropriate ranges of load should be identified for these approximations to work.

4 IDENTIFICATION UNDER GENERAL LOAD

Similar to the previous section, we investigate the scenario when the whole system is modeled as a multiclass product form queueing network. Now, we can conduct K different experiments to measure the queue length information. For the kth experiment, we fix the input rate $\lambda_f^k, 1 \le f \le F$ and observe N_k samples of the number of jobs on each node $\{q_j^k(t)\}_{1\le t\le N_k}$ where t is the index of the sample and can also be viewed as the time interval when the sample is taken. Therefore, (1) can be also represented in matrix form

$$\boldsymbol{\rho} = \boldsymbol{R} \times \boldsymbol{\lambda}, \tag{9}$$

with $\boldsymbol{\rho} = \begin{bmatrix} \rho_1^1 & \cdots & \rho_1^K \\ \vdots & \vdots & \vdots \\ \rho_J^1 & \cdots & \rho_J^K \end{bmatrix}$, $\boldsymbol{R} = \begin{bmatrix} x_{11}s_{11} & \cdots & x_{1F}s_{1F} \\ \vdots & \vdots & \vdots \\ x_{K1}s_{11} & \cdots & x_{KF}s_{1F} \end{bmatrix}$ and $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1^1 & \cdots & \lambda_1^K \\ \vdots & \vdots & \vdots \\ \lambda_F^1 & \cdots & \lambda_F^K \end{bmatrix}$.

We use $\mathbf{Q^k(t)}$ for the kth experiment to denote the vector where the jth coordinate $Q_j^k(t)$ represents the number of jobs on node $j, 1 \le j \le J$ at time t. Furthermore, we assume that these measurements $\{\mathbf{Q}(t)\}$ are independent and are sampled when the system reaches stationarity. Theoretically, when fixing λ_f^k and routing paths, $\{\mathbf{Q(t)}\}$ is a Markov chain. For a high speed computing system, when the sample interval is reasonably large, these samples can be viewed to be nearly independent and drawn from the stationary distribution.

Under the above conditions, the joint queue length distribution at any time t has a product form.

Lemma 4 For the multiclass network that satisfies the conditions described in this section, the stationary distribution of the number of jobs $\{\mathbf{Q(t)}\}$ has the following form

$$\mathbb{P}[Q_1^k(t) = q_1, Q_2^k(t) = q_2, \cdots, Q_J^k(t) = q_J] = \prod_{j=1}^{J} (1-\rho_j^k)\left(\rho_j^k\right)^{q_j}, \tag{10}$$

where $q_j = 0, 1, 3, \cdots$ and $1 \le j \le J$.

Remark 5 The preceding theorem shows that the total number of flows follows a geometric distribution with parameter ρ_j^i at node j. It can be shown that when stable the number of jobs from different flows at each node follow the same distribution as the node operates in isolation and is fed with a set of independent Poisson streams. Additionally, the arrival process for every flow of jobs at each node has the nice property that Poisson arrivals see time average.

Remark 6 From Lemma 4 and the definition of ρ_j^k in (1), we can assume that x_{fj} only takes values in $\{0,1\}$, since if $x_{fj} = y > 1$ we can compute a solution $x'_{fj} = 1$ and $s'_{fj} = s_{fj}x_{fj}$. Therefore, from this point, we make this assumption in the rest of the paper.

Lemma 7 When $N_k \to \infty$ for $1 \le k \le K$ and $\text{rank}(\boldsymbol{\lambda}) = \text{F}$, the matrix $\boldsymbol{R}$ is unique with probability one.

Proof. At time t, we observe $Q_j^k(t) = q_j^k$. Using law of large numbers we know, almost surely,

$$\lim_{N_k \to \infty} \frac{\sum_{t=1}^{N_k} Q_j^k(t)}{N_k} = \frac{\rho_j^k}{1-\rho_j^k}, \tag{11}$$

for $1 \le k \le K$ and $1 \le j \le J$. Since $\text{rank}(\boldsymbol{\lambda}) = \text{F}$, we can always pick F linearly independent columns in $\boldsymbol{\lambda}$. Using these F vectors we form a $F \times F$ square matrix $\boldsymbol{\lambda}'$. The indices used to find these columns in matrix $\boldsymbol{\lambda}$ can be used to form submatrices $\boldsymbol{\rho}'$ and $\boldsymbol{R}'$ from the given matrices $\boldsymbol{\rho}$ and $\boldsymbol{R}$. Therefore, $\boldsymbol{R}' = \boldsymbol{\rho}' \times \boldsymbol{\lambda}^{-1}$ has a unique solution. Finally, the definition of ρ_j^k in (1) and (11) ensure the unique solution of $\boldsymbol{R} = (x_{fj}s_{fj})$ almost surely. □

From Lemma 7, once the unique solution $(x_{fj}s_{fj})$ is obtained, we can set $x_{fj} = 0$ if $x_{fj}s_{fj} = 0$ to denote that flow f is not attended at node j, and $x_{fj} = 1$ if $x_{fj}s_{fj} > 0$ to denote the other cases. However, for a finite sequence of observations, fitting the curve admits the widely known problem of overfitting the model. Specifically, the product $(x_{fj}s_{fj})$ is likely to be always greater than zero, and thus we can compute a result which shows that each flow will be attended at all the nodes. This problem is related to the subset selection or shrinkage methods in linear regression (Hastie, Tibshirani, and Friedman 2001). In order to overcome this problem, we take a Bayesian treatment by assuming that parameters x_{fj} are random variables drawn from a distribution function. This assumption assumes that we have some prior knowledge about the routing paths. For example, if we know that roughly on average each flow only traverses on n nodes from the total J number of nodes, then, in view of Remark 6, we can assume that x_{fj} are realizations of i.i.d. Bernoulli random variables X_{fj} with $\mathbb{E}[X_{fj}] = n/J \triangleq p$.

Using Lemma 4 which shows the product form of the joint queue length distribution at any time and the independence assumption on the observed queue lengths at different times, we can write down the logarithm of the probability to observe the whole sequence of samples.

$$\begin{aligned}
\text{ML} &= \log\left[\left(\prod_{k=1}^{K}\prod_{t=1}^{N_i}\prod_{j=1}^{J}\mathbb{P}[\boldsymbol{Q}_j^k(t) = q_j^k(t)|X_{fj} = x_{fj}]\right)\mathbb{P}[X_{fj} = x_{fj}]\right] \\
&= \log\left[\left(\prod_{k=1}^{K}\prod_{t=1}^{N_i}\prod_{j=1}^{J}(1-\rho_j^k)\left(\rho_j^k\right)^{q_j^k(t)}\right)\left(\prod_{f=1}^{F}p^{x_{fj}}(1-p)^{1-x_{fj}}\right)\right],
\end{aligned}$$

which is same as,

$$\text{ML} = \sum_{k=1}^{K}\sum_{t=1}^{N_k}\sum_{j=1}^{J}\left[\log\left(1-\rho_j^k\right) + q_j^k(t)\log\rho_j^k\right] + \left(\sum_{f=1}^{F}\sum_{j=1}^{J}x_{fj}\right)\log\frac{p}{1-p} + F \times J \times \log(1-p).$$

Using (1) the maximum likelihood estimation is to solve the following optimization problem:

$$\begin{aligned} &\text{Maximize} \\ &\sum_{k=1}^{K}\sum_{j=1}^{J}\left[N_k \log\left(1-\sum_{f=1}^{F} x_{fj}\lambda_f^k s_{fj}\right)+\left(\sum_{t=1}^{N_k} q_j^k(t)\right)\log\left(\sum_{f=1}^{F} x_{fj}\lambda_f^k s_{fj}\right)\right]+\left(\sum_{f=1}^{F}\sum_{j=1}^{J} x_{fj}\right)\log\frac{p}{1-p} \\ &\text{Subject to:} \quad x_{fj}\in\{0,1\} \\ &\qquad\qquad\quad s_{fj}\geq 0. \end{aligned} \tag{12}$$

Remark 8 It is easy to observe that if $p \geq 1/2$, then optimal solution will be obtained at $x_{fj} = 1$ for $1 \leq f \leq F, 1 \leq j \leq J$. Therefore, in the following discussion, we assume that $0 < p < 1/2$.

This is a mixed nonlinear programming. In order to solve it, we introduce an auxiliary optimization problem. For a given sequence $y_{fj}, 1 \leq f \leq F, 1 \leq j \leq J$ that only takes values in $\{0,1\}$, we define the following problem that solves the optimal values for $\{s_{fj}\}$.

$$\begin{aligned} &\text{Maximize } \Phi(\boldsymbol{s}) \stackrel{\Delta}{=} \sum_{k=1}^{K}\sum_{j=1}^{J}\left[N_k \log\left(1-\sum_{f=1}^{F} y_{fj}\lambda_f^k s_{fj}\right)+\left(\sum_{t=1}^{N_i} q_j^k(t)\right)\log\left(\sum_{f=1}^{F} y_{fj}\lambda_f^k s_{fj}\right)\right] \\ &\text{Subject to:} \quad \mathrm{s_{fj}} \geq 0. \end{aligned} \tag{13}$$

Lemma 9 The optimization problem defined in (13) is a convex programming.

Proof. Observe that in (13), $\sum_{f=1}^{F} y_{fj}\lambda_f^k s_{fj}$ is linear with respect to s_{fj}. Next note that $\log(x)$ is a convex function on $(0,\infty)$. Since composition with an affine mapping preserve convexity, we know that (13) is convex as well. □

Remark 10 Convex programming is easy to solve numerically, for example, using the standard steepest descent search. To this end, the partial derivatives are computed as below.

$$\frac{\partial\Phi(\boldsymbol{s})}{\partial s_{fj}} = \sum_{k=1}^{K}\sum_{j=1}^{J}\left[\frac{-N_k y_{fj}\lambda_f^k}{1-\sum_{f=1}^{F} y_{fj}\lambda_f^k s_{fj}}+\left(\sum_{t=1}^{N_i} q_j^k(t)\right)\frac{y_{fj}\lambda_f^k}{\sum_{f=1}^{F} y_{fj}\lambda_f^k s_{fj}}\right].$$

Since it is a convex function, the local maxima is the global maximum.

Based on the solution to the convex optimization (13), we may enumerate all the possibilities of $\{y_{fj}\}$, and compute the optimal value for the problem defined in (12). However, we need to solve $2^{F\times J}$ number of convex optimization problems. In some scenarios when the number of physical nodes and flows is small, solving (12) by searching all the realizations of $\{X_{fj}\}$ is possible. Since the time complexity grows exponentially fast with respect to $F \times J$, it will take a lot of time to solve when $F \times J$ is large. To this end, we propose a more efficient method based on heuristic approximation.

In view of Lemma 7, we now assume that $\boldsymbol{\lambda}$ is full rank with $\mathrm{rank}(\boldsymbol{\lambda}) = \mathrm{F}$. Under this assumption, we exactly conduct $K = F$ different sets of experiments. Now, we proceed with solving the following problem by setting $y_{fj} = 1$ in (13).

$$\begin{aligned} &\text{Maximize } \Phi(\boldsymbol{s}) = \sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{j}=1}^{\mathrm{J}}\left[\mathrm{N_k}\log\left(1-\sum_{\mathrm{f}=1}^{\mathrm{F}}\lambda_{\mathrm{f}}^{\mathrm{k}}\mathrm{s_{fj}}\right)+\left(\sum_{\mathrm{t}=1}^{\mathrm{N_i}}\mathrm{q_j^k(t)}\right)\log\left(\sum_{\mathrm{f}=1}^{\mathrm{F}}\lambda_{\mathrm{f}}^{\mathrm{k}}\mathrm{s_{fj}}\right)\right] \\ &\text{Subject to:} \quad \mathrm{s_{fj}} \geq 0. \end{aligned} \tag{14}$$

Defining $\xi_j^k = \sum_{f=1}^{F}\lambda_f^k s_{fj}$, we obtain

$$\Phi(\boldsymbol{s}) \leq \sum_{k=1}^{K}\sum_{j=1}^{J}\left(\max\left[N_k\log\left(1-\xi_j^k\right)+\left(\sum_{t=1}^{N_k} q_j^k(t)\right)\log\xi_j^k\right]\right). \tag{15}$$

Each summand on the right-hand side of the preceding inequality reaches maximum when $\xi_j^k = \sum_{t=1}^{N_k} q_j^k(t)/N_k$. Therefore, we obtain $K \times J$ number of simple optimization problem

$$\max \left[N_k \log\left(1-\xi_j^k\right) + \left(\sum_{t=1}^{N_k} q_j^k(t)\right) \log \xi_j^k \right]. \tag{16}$$

Using Lemma 7 and the assumption that $\text{rank}(\boldsymbol{\lambda}) = \text{F}$, we can uniquely determine s_{fj} after solving ξ_j^k and in the meanwhile the solution is feasible to (14), thus optimal as well.

Next we propose a heuristic method to solve (12) using the optimal solution obtained from solving (15). We first pick the smallest $s_{f^*j^*}$ that is also positive in $\{s_{fj}\}$, and determine whether we should make $x_{f^*j^*} = 0$ or not by computing the changes of the value in (12). Using Taylor expansion, we obtain the first order approximation to the changes in the following form

$$\sum_{k=1}^{K} \left(\frac{-N_k \lambda_{f^*}^k s_{f^*j^*}}{1-\sum_{f=1}^{F} \lambda_f^k s_{fj}} + \frac{\left(\sum_{t=1}^{N_k} q_j^k(t)\right) \lambda_{f^*}^k s_{f^*j^*}}{\sum_{f=1}^{F} \lambda_f^k s_{fj}} \right) - \log \frac{p}{1-p}, \tag{17}$$

where $-\log p/(1-p) > 0$ by the assumption $0 < p < 1/2$ and

$$\sum_{k=1}^{K} \left(\frac{-N_k \lambda_{f^*}^k s_{f^*j^*}}{1-\sum_{f=1}^{F} \lambda_f^k s_{fj}} + \frac{\left(\sum_{t=1}^{N_k} q_j^k(t)\right) \lambda_{f^*}^k s_{f^*j^*}}{\sum_{f=1}^{F} \lambda_f^k s_{fj}} \right) \leq 0$$

, since the initial values maximize (15). If (17) is positive, then after setting x_{uv} to zero, we expect the total value of $\Phi(s)$ will increase. Next, we again choose the smallest positive value in $\{s_{fj}\}$, and repeat this procedure until the value in (17) is negative.

5 RELATED WORK

There are many queueing network simulation tools available both in open source and as commercial products. OMNeT++ (http://www.omnetpp.org) is a popular C++ based simulation library and framework for building network simulators. Java Modeling Tools (JMT) (Bertoli, Casale, and Serazzi 2006) is another open source tool for modeling and simulating different type of queueing networks (open, closed, hybrid) with different scheduling and routing policies. However, simulation models require knowledge of service time of a transaction at different system components in its end-to-end path which is difficult to measure.

There is an extensive literature on parameter estimation of queues. Many studies focus on inferring parameters for a single-node queue (Basawa, Bhat, and Lund 1996, Basawa, Bhat, and Zhou 2008, Chen, Walrand, and Messerschmitt 1994, Ross, Taimre, and Pollett 2007) using only partial information . (Basawa, Bhat, and Lund 1996) develops maximum likelihood estimators for parameters of a GI/G1/I queue using only waiting time data. (Chen, Walrand, and Messerschmitt 1994) applies maximum likelihood to estimate the arrival rate of "unobserved" flow using arrival rate and waiting times of observed flow. The work that is most close to our study is (Liu, Wynter, Xia, and Zhang 2006), where the investigation is to fit the model for a given queueing network using the end-to-end delay measurements. However, the routing paths therein are known, and only the service rates need to be estimated from measurements. In our study we study a related problem but the measurements are the total number of jobs at each node and the routing path of each flow is also unknown along with its service time at each node.

6 CONCLUSION AND FUTURE WORK

System identification using measurements is important for understanding performance, finding bottleneck resources, and placement of workload. The problem gains prominence in systems with multi-stage workflows

due to the complexity of the software components, and techniques like virtualization, and run-time migration of workloads across physical nodes. Under light load, we can approximate the system using simple queueing models with much smaller set of parameters, which can be easily estimated from measured data. The approximate models can also be used to speed-up system simulation. The system identification methodology that we propose using MLE goes beyond existing techniques by incorporating the path identification problem along with service time estimation. Further work needs to be done targeted at quantitative evaluation of our proposed methodology and the heuristic solution approach over different systems and its comparison with existing techniques. We are also working to study the feasibility of applying the approximation and identification methodology for specific problems in cloud environments, in particular, the VM provisioning delay analysis and bottleneck identification.

REFERENCES

Basawa, I., U. Bhat, and J. Zhou. 2008, September. "Parameter estimation using partial information with applications to queueing and related models". *Statistics & Probability Letters* 78:1375–1383.

Basawa, I. V., U. N. Bhat, and R. Lund. 1996. "Maximum likelihood estimation for single server queues from waiting time data". *Queueing Systems - Theory and Applications* 24:155–167.

Bertoli, M., G. Casale, and G. Serazzi. 2006. "Java Modeling Tools: an Open Source Suite for Queueing Network Modeling and Workload Analysis". *Quantitative Evaluation of Systems, International Conference on*:119–120.

Chen, T. M., J. Walrand, and D. G. Messerschmitt. 1994. "Parameter estimation for partially observed queues". *IEEE Transactions on Communications* 42:2730–2739.

Hastie, T., R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Liu, Z., L. Wynter, C. H. Xia, and F. Zhang. 2006, January. "Parameter inference of queueing models for IT systems using end-to-end measurements". *Performance Evaluation* 63:36–60.

Ross, J. V., T. Taimre, and P. K. Pollett. 2007, February. "Estimation for queues from queue length data". *Queueing Syst. Theory Appl.* 55:131–138.

AUTHOR BIOGRAPHIES

PARIJAT DUBE. received his M.S. in Electrical Communication Eng. from Indian Institute of Science, Bangalore in 2001 and his Ph.D. in Computer Science from University of Nice-Sophia Antipolis in 2002 where he was affiliated to INRIA. He joined IBM T. J. Watson Research Center, Hawthorne, New York in 2002. His research interests include performance analysis and control of computer systems, distributed computing, stochastic modeling and game theory. His email address is pdube@us.ibm.com.

JIAN TAN is a Research Staff Member in the System Analysis and Optimization group at IBM T. J. Watson Research Center. His research activity focuses on performance modeling and reliability evaluation for information networks and complex service systems. He received his Masters and PhD (2008) from the Electrical Engineering Department at Columbia University and undergraduate (2002) from University of Science and Technology of China. His email address is tanji@us.ibm.com.

LI ZHANG graduated from IEOR Dept. Columbia University after receiving degrees from Purdue University and Beijing University. He is the manager of the System Analysis and Optimization group at IBM T.J. Watson Research Center. His research interests include control and performance analysis of computer systems, statistical techniques for traffic modeling and prediction, scheduling and resource allocation in parallel and distributed systems, as well as measurement based clock synchronization mechanisms. His email address is zhangli@us.ibm.com.