RC23664 (W0507-080) July 8, 2005
Computer Science

IBM Research Report

Optimal Capacity Allocation for Web Systems with End-to-End Delay Guarantees

Wuqin Lin
School of Industrial and System Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

Zhen Liu, Cathy H. Xia, Li Zhang
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Optimal Capacity Allocation for Web Systems with End-to-End Delay Guarantees

Wuqin Lin [a], Zhen Liu [b], Cathy H. Xia [b,*], Li Zhang [b],

[a] *School of Industrial and System Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA*

[b] *IBM Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA*

Abstract

Providing quality of service guarantees have become a critical issue during the rapid expansion of the e-Commerce area. We consider the problem of finding the optimal capacity allocation in a clustered Web system environment so as to minimize the cost while providing the end-to-end performance guarantees. In particular, we consider constraints on both the average and the tail distribution of the end-to-end response times. We formulate the problem as a nonlinear program to minimize a convex separable function of the capacity assignment vector. We show that under the mean response time guarantees alone, the solution has a nice geometric interpretation. Various methods to solve the problem are presented in detail. For the problem with tail distribution guarantees, we develop an approximation method to solve the problem. We also derive bounds and show that the solution is asymptotically optimal when the service requirement becomes stringent. Numerical results are presented to further demonstrate the robustness of our solutions under data uncertainty.

Key words: resource allocation, queueing network, end-to-end delay, QoS

1 INTRODUCTION

With the increasing bandwidth and connectivity associated with the Internet, e-Commerce has become more and more popular. Many traditional services

* Corresponding author.

Email addresses: linwq@isye.gatech.edu (Wuqin Lin), zhenl@us.ibm.com (Zhen Liu), cathyx@us.ibm.com (Cathy H. Xia), zhangli@us.ibm.com (Li Zhang).

have been transformed or converted to Web-based services. In addition to becoming a cost effective solution for many traditional businesses, e-Commerce is also creating new business opportunities. A variety of e-Commerce models now exist, ranging from, for example, on-line shopping, on-line auction, on-line reservation, on-line banking and on-line trading to customer relation management, personnel management, etc. E-Commerce has become such a critical component of many companies that guaranteeing performance and availability has become essential. Thus, the design and development of e-Commerce infrastructure (or more specifically, Web systems) should meet a two-fold challenge. On one hand, it must meet customer expectations in terms of quality of service (QoS). On the other hand, companies have to control information technology (IT) costs to stay competitive. It is therefore crucial to understand the tradeoffs between costs and service levels so as to determine the most cost-effective architecture and system configuration.

One of the most common architectures of Web service infrastructures is the clustered system architecture, where requests are served by the system through different clusters of servers, from the Web server cluster to the application server cluster, and possibly to the database server cluster. In general, it is possible for requests to be served by a subset of the clusters. Although architecturally simple, the system is quite complex and large in general, with multiple clusters of servers, each of which can have many components. A typical Web system is comprised of hundreds of nodes with tens of applications running on them. Given the great complexity of the overall system, IT planners are constantly puzzled with questions regarding: how many servers to place at each cluster in the current infrastructure; what layout can deliver the best QoS; is there enough capacity available to support the expected business opportunities and future growth.

Another important characteristic of today's Web service environment is the diversity of services that one system can support. Multiple classes of services are commonly provided to multiple clients, all of which are time-sharing and competing for the same set of resources. The service provider has contracts to each individual client and agrees to guarantee a certain level of QoS for each class of service. In addition, performance guarantees can be on the mean end-to-end response times and/or on the percentage of time that the end-to-end response times are above a given threshold. The intense competition in e-Commerce has driven many companies to sign contracts that promise both, that is, performance guarantees for both the mean and the tail distribution of the end-to-end response times. Thus, we need to develop techniques and algorithms to determine the most cost-effective capacity allocation in a clustered system while providing guarantees on one or more end-to-end performance measures.

In this paper, we study the cost-effective capacity planning problem in a clus-

tered Web system with multiple classes of services under service level guarantees. We consider a general class of service guarantees that include both the mean and the tail distribution of per-class end-to-end delay. We assume that the cost of a given capacity allocation is a convex separable increasing function of the capacity assignment vector. Such a cost function can be used to handle many e-Commerce cost structures. For example, cost functions linear in the capacity assignment can be used to represent the case if the cost structure is purely on the IT expenses. One can also choose cost functions to be functions of the mean delay or the tail probability so as to represent the profits or penalty related to the QoS contracts. We shall focus exclusively on Web systems although the techniques developed here can be applied to more general settings including networks.

Studies on resource allocation to deliver end-to-end performance are quite limited. Most literature has focused on a single cluster and addressed either scheduling or load balancing issues [9,10,7]. Recently, Menascé et al. [13] considered the problem of resource scheduling in e-commerce sites with the aim of maximizing revenue; Liu et al. [12] considered the problem of both routing and scheduling in the large-scale server farm environment in order to guarantee the tail distribution of a single-tiered architecture.

We propose a nonlinear programming problem formulation and investigate its structural properties. We show that under the mean response time guarantees alone, the solution has a nice geometric interpretation and can be solved in polynomial time. The problem with guarantees on the tail distribution of response times is more complicated. We develop approximation methods which can provide solution that is only away from the optimal solution by a constant factor, independent of the service demands of the job classes at all tiers. We also show that our algorithm can achieve asymptotic optimality when the service requirement becomes stringent. The problem with both types of end-to-end response time guarantees can then be solved easily. Numerical results further show that the proposed methods are robust under data uncertainty.

The paper is organized as follows. In section 2, we present the problem and the corresponding queueing network model. Sections 3, 4 and 5 focus respectively on the problems with only mean performance guarantees, with only tail performance guarantees, and the problem with both types of guarantees. Robustness of the solution under data uncertainty is then discussed in section 6.

2 The Problem

We consider the problem of capacity planning in multiple clustered Web architectures in the presence of multi-class performance constraints. The objective

is to find the most cost effective capacity allocation while satisfying the *dual* type of QoS requirements on both mean and tail distribution of the response times for each class of service. Our approach is based on the use of queueing network model and optimization methods to capture the e-Commerce service process.

2.1 e-Commerce Service Environment

Today's e-commerce service environment is quite complex. Typically it involves multiple clusters of machines. Figure 1 illustrates such a 3-clustered architecture for example. Each cluster handles a particular set of functions. Front-end Web servers handle the serving of requests for static pages. Requests for dynamic pages may require processing by the application server, some of which further involve obtaining, or updating information from the database server. The service of different requests may involve multiple visits to multiple clusters in different order.

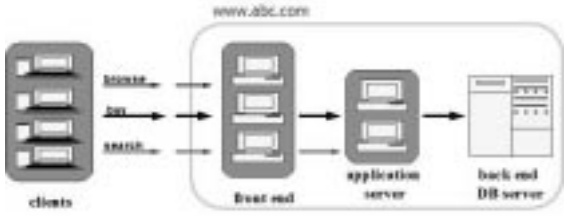

Fig. 1. A Three-Cluster Architecture

Several classes of requests can be served on each server. The service discipline is either First-Come-First-Serve (FCFS) or Processor-Sharing (PS) depending on the server type. FCFS service discipline may be required, for example, on a database server when processing write transactions. The assumption of PS service discipline is reasonable for a wide range of Web servers in practice. We assume that the resource requirement of each job class at each server is well-understood. This can be measured, for example, with certain instrumentation on the various servers in a control environment.

The workload and the use of the e-commerce service resources depends upon the navigational behavior of the Web clients which is characterized by Web sessions consisting of a sequence of alternating transactions. For example, a typical client scenario might consist of several browse, search requests, possibly followed by a buy transaction, in an iterative manner. In between the transactions, there might be network delays or client-based delays (or think times), which may be different for different sessions.

2.2 Queueing Network Model

We use a general queueing network model to capture the e-Commerce service environment described above. There are $t = 1, \cdots, T$ multiclass single-server stations (or servers), one infinite-server queue ($t = 0$), and $k = 1, \cdots, K$ (external) classes of requests. The server stations $t = 1, \cdots, T$ represent the collection of Web servers, the infinite-server queue $t = 0$ represents the client think times, and the job class represents the various types of user transactions within a Web session.

We assume there are J different types of Web sessions, each representing a different client navigational behavior (e.g. when there are multiple sites co-hosted on the same system, then navigation for each different web site corresponds to a different session type). User sessions of type j have exogenous Poisson arrival rate α_j, and begin with a class k_j request, $j = 1, \ldots, J$. Upon the completion of a class k request, clients of session type j incur a random think time d_j at an infinite-server queue (labeled as $t = 0$), and either return to the system as a class k' request with probability $p^{(j)}_{k,k'}$ or exit the system (thus complete the session) with probability $1 - \sum_{k'=1}^{K} p^{(j)}_{k,k'}$. Let $\Lambda^{(j)}_k$ denote the aggregate rate of class k arrivals from session j. Then $\Lambda^{(j)}_k = \sum_{k'} \Lambda^{(j)}_{k'} p^{(j)}_{k'k} + \alpha_j$.

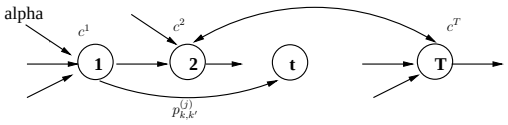

Fig. 2. Queueing Network Model

Requests of class k visit the stations along route k which is a deterministic sequence: $t(k,1), t(k,2), \cdots, t(k,T_k)$, where a class k job visits server station $t(k,l)$ at hop l, and T_k is the total number of hops on route k. Denote ν^t_k the total (random) number of visits to station t by a class k request ($\sum_{t=1}^{T} \nu^t_k = T_k$). Let $\Lambda_k := \sum_{j=1}^{J} \Lambda^{(j)}_k$. Then $\lambda^t_k = \nu^t_k \Lambda_k$ is the effective arrival rate of class k requests to station t.

At each server station, there could be multiple servers. We assume each individual server has a unit capacity and the total number of servers c^t at station t is yet to be determined. Let $\vec{c} = (c^1, c^2, \cdots, c^T)$ be a *capacity assignment* so that station t has capacity c^t. For simplicity, we assume that all servers that belong to station t work together and execute the incoming requests as a single server. Assume the (nominal) service requirements of class k jobs at station t are i.i.d. random variables S^t_k with mean $m^t_k = \mathbf{E}[S^t_k]$. We use the term 'nominal' because S^t_k has not taken into account the capacity c^t at server t which is a decision variable. If capacity c^t is assigned to server t, then the

service time for a class k job at station t is S_k^t/c^t (providing that station t spends its total capacity c_t on this job). The service discipline at each station is either FCFS or PS. If station t serves under FCFS, we then assume that the service requirements S_k^t are exponential with the same mean for all class k jobs at station t. If station t serves under PS, we assume that the service requiremens S_k^t of class k jobs at station t follow a general distribution including heavy-tailed distributions.

It should also be emphasized that we choose to use the above Kelly-Type network [8,4] setting in order to maintain a product-form solution. For more general systems, experiments show that these solutions also work well, indicating that product-form queueing network models can serve as good approximations to capture the high-level queueing dynamics of real systems.

2.3 End-to-end Delay and Performance Guarantees

We define the end-to-end delay (or sojourn time) of a class k job, denoted by R_k, as the total elapse time from the time it enters the system to the time it exits the system. Let $R_k^t(i)$ be the delay experienced by a class k request at station t in its i-th visit, $i = 1, \ldots, \nu_k^t$. Then, $R_k = \sum_{t=1}^{T} \sum_{i=1}^{\nu_k^t} R_k^t(i)$. We shall consider the following two types of service guarantees:

$$\mathbf{E}\left[R_k\right] \leq U_k, \text{ for } k = 1, \ldots, K, \tag{1}$$

and

$$\mathbf{P}\left[R_k > V_k\right] \leq \varepsilon_k, \text{ for } k = 1, \ldots, K. \tag{2}$$

Constraint (1) guarantees the average end-to-end delay of class k jobs to be no more than U_k, and constraint (2) guarantees that the probability that the end-to-end delay of class k is greater than V_k is no more than ϵ_k

Let q^t be the *nominal* server utilization at station t such that $q^t = \sum_k \lambda_k^t m_k^t$. Then under capacity c^t, the server utilization at station t will be $\rho^t = q^t/c^t$. In order for the system to be stable, the minimum capacity assignment $\vec{c}$ must satisfy:

$$c^t > q^t, \text{ for all } t = 1, \cdots, T. \qquad \text{(Stability Condition)} \tag{3}$$

We assume that the cost structure is a separable increasing function of the capacity assignment $\vec{c} = (c^1, c^2, \cdots, c^T)$. That is, for all $t = 1, \cdots, T$, $f_t(\cdot)$ is a convex increasing function. If capacity c^t is assigned to server t, then a cost of $f_t(c^t)$ will be incurred. The total cost of this assignment is simply $\sum_{t=1}^{T} f_t(c^t)$. The problem is to minimize this total cost while satisfying one or both of end-to-end delay guarantees.

3 MEAN DELAY GUARANTEE

We first consider the problem with only mean delay guarantees (1).

Let N_k^t(resp. R_k^t) denote the steady-state number (resp. response time) of class k jobs at station t. Denote $N^t = \sum_{k=1}^K N_k^t$ the steady-state number of all class jobs at station t. Similarly, denote N_k the steady-state total number of class k jobs in system and R_k the steady-state end-to-end response time (or sojourn time) of class k jobs.

Based on the product form (see also [17] Section 6-8 of Chapter 11), we know that the number of jobs at station t has the same distribution as that of the corresponding M/M/1 queues. Moreover, a job belongs to class k with probability ρ_k^t/ρ^t, where $\rho_k^t = \lambda_k^t m_k^t/c^t$ and $\rho^t = \sum_{k=1}^K \rho_k^t$. Hence,

$$\mathbf{E}\left[N^t\right] = \frac{\rho^t}{1-\rho^t}, \qquad \text{and} \qquad \mathbf{E}\left[N_k^t\right] = \frac{\rho_k^t}{1-\rho^t}.$$

Applying Little's law, we can obtain the mean response time of class k jobs at station t as follows:

$$\mathbf{E}\left[R_k^t\right] = \frac{\mathbf{E}\left[N_k^t\right]}{\lambda_k^t} = \frac{m_k^t/c^t}{1-\rho^t} = \frac{m_k^t}{c^t - q^t}. \tag{4}$$

Based on Little's law, the mean end-to-end delay for class k jobs is therefore

$$\mathbf{E}\left[R_k\right] = \frac{\mathbf{E}\left[N_k\right]}{\Lambda_k} = \frac{\sum_t \mathbf{E}\left[N_k^t\right]}{\Lambda_k} = \sum_t \frac{\lambda_k^t \mathbf{E}\left[R_k^t\right]}{\Lambda_k} = \sum_t \frac{\nu_k^t m_k^t}{c^t - q^t}. \tag{5}$$

The problem of finding the minimum capacity such that the mean delay requirement is guaranteed can be formulated as:

$$\min \qquad \sum_t f_t(c^t) \tag{6}$$

$$\text{s.t.} \qquad \sum_t \frac{\nu_k^t m_k^t}{c^t - q^t} \le U_k, \qquad k = 1, \ldots, K; \tag{7}$$

$$c^t > q^t, \qquad t = 1, \ldots, T. \tag{8}$$

Condition (8) is exactly the stability condition. The capacity c^t required by each server t must be strictly greater than q^t to have finite response time. Assign a new variable x^t, such that $c^t = q^t + 1/x^t$. Then $1/x^t$ is the extra capacity (above the minimum requirement q^t) that is allocated to server t. Denote further $w_k^t = \frac{\nu_k^t m_k^t}{U_k}$. Hence w_k^t can be interpreted as the weight (or

relative ratio) of the total required mean nominal service time for a class-k job at server t (in all visits) to its required mean end-to-end delay upper bound U_k.

Now the problem is simplified to the following:

$$\begin{aligned}
\textbf{(M)} \qquad \min \quad & \sum_t g_t(x^t) \\
\text{s.t.} \quad & \sum_t w_k^t x^t \ \leq \ 1, \qquad k = 1, \ldots, K, \qquad (9) \\
& x^t \ \geq \ 0, \qquad t = 1, \ldots, T, \qquad (10)
\end{aligned}$$

where $g_t(x^t) \equiv f_t(q^t + 1/x^t) = f_t(c^t)$. Note that we have relaxed the strict $>$ to $\geq$ in (8) so that the feasible region is compact. It is obvious $x^t = 0$ will never be the optimal solution.

The problem becomes to minimize the total cost which occurs due to the extra system capacity $1/x^t$. It is naturally to assume function f_t is non-decreasing for each t. Therefore, g_t is a non-increasing function. Moreover, in the rest of this section, we assume g_t is convex function. It is easy to see that this assumption is satisfied if f_t is convex. However, this is not necessary. This assumption is also satisfied for some well-known concave cost function: for example, $f_t(c^t) = \ln(c^t)$.

For notation simplicity, denote $\vec{x} = (x^1, \ldots, x^T)$ and and $\vec{w_k} = (w_k^1, \ldots, w_k^T)$ for $k = 1, ..., K$. Let Polyhedron$_+(\vec{w_1}, ..., \vec{w_K})$ be the polyhedral set [3] defined by $\vec{w_1}, ..., \vec{w_K}$ on the positive quadrant $\mathcal{R}_+^T$, that is,

$$\text{Polyhedron}_+(\vec{w_1}, ..., \vec{w_K}) := \{\vec{x} \in \mathcal{R}_+^T \mid \vec{w_k} \cdot \vec{x} \ \leq \ 1, \ k = 1, \ldots, K \ \}.$$

Clearly Polyhedron$_+(\vec{w_1}, ..., \vec{w_K})$ defines the feasible region of (M), which is a closed convex set.

It's important to observe that because the objective function in (M) is monotone non-increasing in each variable, the optimal solution must lie on the 'north-eastern' frontier of the convex feasible region. We call such frontier as the *efficient frontier*. We then have

Proposition 1 *The optimal solution to (M) must be on the efficient frontier of Polyhedron$_+(\{\vec{w_1}, \ldots, \vec{w_K}\})$.*

Notice that this is slightly counter-intuitive comparing with the general convex optimization problems where the optimal solutions are very often interior points.

For a given vector set $\Omega = \{\vec{w}_k, k = 1, ..., K\}$, we say that a subset $\Omega_1 = \{\vec{w}_\ell, \ell \in \mathcal{S}\}$ with $\mathcal{S} \subset \{1, ..., K\}$ is the *critical set* of Ω, if Ω_1 is the minimum cardinality set such that, for any $\vec{w}_k \in \Omega$, there exist scalars $\{\alpha_\ell\}_{\ell \in \mathcal{S}}$ such that

$$\vec{w}_k \leq \sum_{\ell \in \mathcal{S}} \alpha_\ell \vec{w}_\ell, \quad \sum \alpha_\ell = 1, \ \alpha_\ell \geq 0, \ \forall \ \ell \in \mathcal{S}.$$

Clearly Polyhedron$_+(\Omega)$ is a subset of Polyhedron$_+(\Omega_1)$. On the other hand, for any $\vec{x} \in$ Polyhedron$_+(\Omega_1)$, we have

$$\vec{w}_k \cdot \vec{x} \ \leq \ \sum_{\ell \in \mathcal{S}} \alpha_\ell \vec{w}_k \cdot \vec{x} \ \leq \ \sum_{\ell \in \mathcal{S}} \alpha_\ell \ = \ 1.$$

Hence,

$$\text{Polyhedron}_+(\Omega) = \text{Polyhedron}_+(\Omega_1).$$

This means that the constraint $\vec{w}_k \cdot \vec{x} \leq 1$ corresponding to $\vec{w}_k$ can be derived from the constraints corresponding to the critical set $\{\vec{w}_\ell, \ell \in \mathcal{S}\}$. Therefore, we can ignore the constraints corresponding to the vectors $\vec{w}_k$, if it is dominated by a convex combination of vectors in the critical set. Therefore,

Proposition 2 *The optimal solution to (M) is determined by the critical set* $\{\vec{w}_\ell, \ell \in \mathcal{S}\}$.

Property 2 can also be illustrated by Figure 3. Consider the T-dimensional space, where point w_k corresponds to vector $\vec{w}_k = (w_k^1, \ldots, w_k^T)$, , $k = 1, ..., K$. If a point w_k is not on the 'north-eastern' frontier of Convex Hull$(w_1, ..., w_K)$, then it will be dominated by a convex combination of other w's who are on the frontier, thus it will not play a role in determining the optimal solution to (M). In the context of the original problem, when allocating server capacities, one or more classes of jobs satisfying the mean end-to-end response time constraint could imply another class of jobs' mean response time constraint be automatically satisfied. In this case, the mean response time constraint for the latter job class is less stringent, and can be derived from the constraint for the former job classes. Here, the vertices of the convex hull correspond to the class of jobs whose response time constraints are more stringent, and can not be derived from the constraints for other job classes.

After simplifying the formulation, let's turn to how to find the optimal solutions to the problem. The optimization problem (M) is a separable convex programming problem with linear constraints. Note that it has been shown in [5] that such a separable convex programming problem with linear constraints can be converted into a linear program and solved in polynomial time.

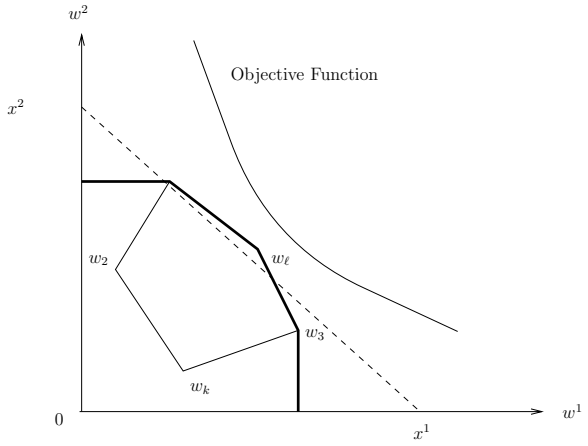

Fig. 3. Convex Hull $(w_1, ..., w_K)$.

4 TAIL DISTRIBUTION GUARANTEE

In this section, we consider the capacity planning problem with the tail distribution guarantee constraint (2).

The problem can be formulated as:

$$\begin{array}{lll} \textbf{(T)} \quad \min & \sum_t f_t(c^t) & \\ \text{s.t.} & \mathbf{P}\left[R_k > V_k\right] \le \varepsilon_k, \quad k = 1, \ldots, K; & (11) \\ & c^t \ge q^t, \qquad t = 1, \ldots, T. & \end{array}$$

The tail distribution of the end-to-end delay $\mathbf{P}\left[R_k > x\right]$ can be quite complicated. Bounds on the tail asymptotics of maximum daters (time to empty the network when stopping future arrivals) for FCFS Jackson networks were considered in [1,2]. For the general Kelly-type network under service disciplines such as Processor Sharing, the tail asymptotics of the end-to-end delay is still an open question. Furthermore, in practice, the threshold V_k's in the tail performance guarantees in (11) are often limited to some prespecified finite levels, thus such asymptotics may not apply. For simplicity, we shall consider as approximation that $R_k \stackrel{d}{=} \sum_{t=1}^{T} \sum_{i=1}^{\nu_k^t} R_k^t(i)$, where $R_k^t(i)$'s, $i = 1, \ldots, \nu_k^t$ are i.i.d. replicas of the steady-state response times R_k^t of a single server queue (corresponding to server t) with capacity c^t and arrival rates λ_k^t, $k = 1, \ldots, K$, independent of other server queues. Denote

$$G_k^t(c, y) = \mathbf{P}\left[R_k^t \le y | c^t = c\right].$$

For a capacity allocation vector $\vec{c} = (c^1, c^2, \cdots, c^T)$, $\mathbf{P}\left[R_k > V_k\right]$ is the convolution of $G_k^t(c^t, y)$ with respective to y for all $t = t(k, i)$, $i = 1, \ldots, T_k$, along the route visited by class k jobs.

Under the above simplifications, problem **(T)** then becomes a non-linear program. However, the convolution is a very complicated function which makes the non-linear problem hard to solve. We therefore search for good approximating solutions. In this paper, by approximating the constraint (11), we derive upper and lower bounds and give a near optimal solution to problem **(T)**. Indeed, under some assumptions on the distribution function G_k^t, the solution is proven to be asymptotically optimal.

4.1 A Lower Bound

Denote $\mathcal{K}(t)$ the set of job classes that visit server t. Let $\mathcal{T}(k)$ denote the set of servers that a class k job visits. Recall that T_k denotes the total number of hops on the route for class k jobs, where $T_k = \sum_{t=1}^{T} \nu_k^t$, with ν_k^t denoting the total (random) number of visits to server t in the route.

The following lemma is straightforward based on the fact that

$$R_k = \sum_{\tau} \sum_{i=1}^{\nu_k^\tau} R_k^\tau(i) \geq R_k^t(i) \stackrel{d}{=} R_k^t.$$

Lemma 3 *For an arbitrary $k = 1, ..., K$, if the end-to-end delay R_k satisfies* (11)*, then*

$$\boldsymbol{P}\left[R_k^t > V_k\right] \leq \varepsilon_k, \ \ \forall 1 \leq k \leq K, 1 \leq t \leq T. \tag{12}$$

Lemma 3 basically says that if the tail distribution guarantees on the end-to-end delays are satisfied then these guarantees will be satisfied for the delay at each server. Therefore, replacing constraint (11) by (12), we can obtain a lower bound for problem (T) which is stated as follows.

Theorem 4 *$\sum_t f_t(c_*^t)$ is a lower bound on problem (T), where $\vec{c}_* = (c^t, t = 1, ..., T)$, and*

$$c_*^t = \max_{k \in \mathcal{K}(t)} \min\{c : G_k^t(c, V_k) \geq 1 - \varepsilon_k\}. \tag{13}$$

4.2 A Feasible Solution and Upper Bound

The solution $\vec{c}_*$ obtained by (13) may not be feasible to (T) as it does not guarantee the tail distribution requirement on the end-to-end delay. To derive a feasible solution for (T), we first define the random variable $R_k^{t,n}$ to be the summation of n independent copies of R_k^t and assume the following.

Assumption 5 *For each class k, if*

$$\boldsymbol{P}\left[R^{t,T_k} > V_k\right] \leq \varepsilon_k, \quad \textit{for each } t \in \mathcal{T}_k, \tag{14}$$

then

$$\boldsymbol{P}\left[R_k > V_k\right] \leq \varepsilon_k. \tag{15}$$

Assumption 5 is satisfied if for any fixed class k, all R_k^t's are comparable in the stochastic ordering sense [15], $1 \leq t \leq T$. We say that X and Y are comparable in the stochastic ordering sense $\leq_{st}$ if either $X \leq_{st} Y$ or $Y \leq_{st} X$. Indeed, in this case, for each fixed class k, there is a bottleneck server τ such that the response time R_k^τ stochastically dominates the others. Then R_k^{τ,T_k} stochastically dominates $R_k = \sum_t \sum_{i=1}^{\nu_k^t} R_k^t(i)$. Hence $\mathbf{P}\left[R_k^{\tau,T_k} > V_k\right] \leq \varepsilon_k$ implies immediately inequality (15). For example, if for any fixed class k, R_k^t's are all exponential or are all of Weibull distribution with the same shape parameter, then they are stochastically comparable. Also, if for any fixed class k, the nominal service times S_k^t's for $t = 1, ..., T$, are identical in distribution, then one can show using coupling arguments [16] that R_k^t's are stochastically comparable.

Define $G_k^{t,n}(c, y)$ to be the n-th convolution of $G_k^t(c, y)$ with respective to y, and then

$$G_k^{t,n}(c, y) = \mathbf{P}\left[R_k^{t,n} \leq y | c^t = c\right]. \tag{16}$$

The following theorem gives a feasible solution to problem (T).

Theorem 6 *Suppose Assumption 5 holds. Let $\vec{c^*} = (c^{*1}, \cdots, c^{*T})$ where*

$$c^{*t} = \max_{k \in \mathcal{K}(t)} \min\{c \geq q^t : G_k^{t,T_k}(c, V_k) \geq 1 - \varepsilon_k\}. \tag{17}$$

Then $\vec{c^*}$ is feasible to problem (T), and $\sum_t f_t(c^{*t})$ is an upper bound for problem (T).

PROOF. It is easy to see that for each t,

$$\mathbf{P}\left[R^{t,T_k} > V_k\right] \leq \varepsilon_k, \quad \text{for all } k \in \mathcal{K}_t, \tag{18}$$

This is equivalent to (14). Then (11) follows under assumption 5.

4.3 Asymptotic Optimality

In this section, we discuss the effectiveness of the solutions we proposed in Section 4.1 and 4.2. Recall that S_k^t denotes the nominal service requirement

random variable for class k at server t. First we make the following *simplifying assumption*:

Assumption 7 *For each class k and each server t,*

$$\boldsymbol{P}\left[R_k^t > y\right] \sim \boldsymbol{P}\left[\frac{S_k^t}{c^t - q^t} > y\right]. \tag{19}$$

Note that (19) is only an approximation. In the case of M/G/1 PS queue with subexponential service times, the above assumption has been shown [6] to be asymptotically true for large y when F_k^t has a heavier tail than $e^{-\sqrt{y}}$. Here $F_k^t(y) = \mathbf{P}\left[S_k^t \leq y\right]$. Denote further the complementary of F_k^t to be $\overline{F_k^t} = 1 - F_k^t$, and the inverse function of $\overline{F_k^t}$ to be

$$\varphi_k^t(\epsilon) = \arg\min_y \{\overline{F_k^t}(y) \leq \epsilon\}.$$

For the asymptotic analysis to make sense, we assume that the support of F_k^t is $(0, \infty)$. Therefore,

$$\varphi_k^t(\epsilon) \to \infty \text{ as } \epsilon \to 0.$$

Under assumption 7, we then have $G_k^t(c, V_k) = F_k^t\left((c - q^t)V_k\right)$, and the lower bound in (13) can be written as

$$c_*^t = q^t + \max_{k \in \mathcal{K}(t)} \varphi_k^t(\varepsilon_k)/V_k. \tag{20}$$

Similarly, denote $F_k^{t,n}$ to be the n-th convolution of F_k^t, the complementary $\overline{F_k^{t,n}} = 1 - F_k^{t,n}$, and the inverse function of $\overline{F_k^{t,n}}(y)$:

$$\varphi_k^{t,n}(\epsilon) = \arg\min_y \{\overline{F_k^{t,n}}(y) \leq \epsilon\}.$$

Then we can simplify the upper bound in (17) as

$$c^{*t} = q^t + \max_{k \in \mathcal{K}(t)} \varphi_k^{t,T_k}(\varepsilon_k)/V_k, \tag{21}$$

Assumption 8 *For each k, t,*

$$\limsup_{y \to \infty} \frac{\overline{F_k^{t,T_k}}(y(1+\delta))}{\overline{F_k^t}(y)} < 1, \quad \textit{for any } \delta > 0. \tag{22}$$

Lemma 9 *For each k, t, if there exists $0 \leq \kappa_0 < \infty$, and $0 < \kappa_1 \leq \infty$, such that*

$$\limsup_{y \to \infty} \frac{\overline{F_k^{t,T_k}}(y)}{\overline{F_k^t}(y)} = \kappa_0, \tag{23}$$

and for any $\delta > 0$,

$$\limsup_{y\to\infty} \frac{\overline{F_k^t}(y(1+\delta))}{\overline{F_k^t}(y)} \leq \frac{1}{\kappa_0 + \kappa_1}. \tag{24}$$

then, Assumption 8 holds.

PROOF. Please refer to [11].

Remark 10 *Suppose* F_k^t*'s are subexponential. Then* $\kappa_0 = T_k$*. If* (24) *holds for* $\kappa_1 = \infty$*, then assumption 8 is always true. If* κ_1 *is finite, then* (24) *will be eventually violated as the number of servers* T_k *gets larger. For a large class of subexponential distributions with moderate heavy tails such as lognormal and Weibull distributions,* (24) *holds for* $\kappa_1 = \infty$ *and therefore assumption 8 is true for any* T_k*.*

Both exponential and Weibull distributions satisfy Assumption 8. For exponential distribution, one can directly verify (22), although Lemma 9 does not apply. The subexponential Weibull family with shape parameter $0 < \beta < 1$ satisfies Assumption 8, with $\kappa_0 = T_k$, and $\kappa_1 = \infty$. However, Assumption 8 does not hold for Pareto family.

Proposition 11 *If Assumption 8 holds, then for each* $1 \leq k \leq K, 1 \leq t \leq T$*,*

$$\varphi_k^{t,T_k}(\epsilon)/\varphi_k^t(\epsilon) \to 1 \text{ as } \epsilon \to 0, \tag{25}$$

PROOF. Please refer to [11].

Combine Proposition 11 with (20) and (21), it follows immediately that

Theorem 12 *If Assumptions (5-8) hold, then the solutions obtained by* (13) *and* (17) *are asymptotically optimal in the sense that for each* t*,*

$$c^{*t}/c_*^t \to 1 \text{ as } \varepsilon_k \to 0 \text{ for all } k.$$

In fact, for the above theorem to hold, it only requires the minimum ε_k to go to 0, which is stated as follows.

Corollary 13 *Suppose Assumptions (7-8) hold. The solutions obtained by* (13) *and* (17) *are asymptotically optimal in the sense that for each* t*,*

$$c^{*t}/c_*^t \to 1 \text{ as } \quad \min_k \varepsilon_k =: \varepsilon_0 \to 0.$$

PROOF. Please refer to [11].

Note that Pareto distribution does not satisfy the Assumption 8, and the asymptotic results do not hold. However, the ratio of the bounds obtained in the previous sections is bounded by a constant factor as ε_0 goes to zero. That is,

Corollary 14 *For Pareto service times that* $\overline{F_k^t}(y) = y^{-a_k^t}, 0 < a_k^t \leq 2$, *we have*

$$\limsup_{\varepsilon_0 \to 0} c^{*t}/c_*^t \leq 2^{1/a},$$

where $a = \min_{k,t} a_k^t$.

4.4 Example: Exponential Case

We consider the case for which the response time at each server R_k^t is exponential distributed, that is, $\overline{F_k^t}(y) = e^{-y/m_k^t}$. Denote $\gamma_n(\epsilon)$ to be the inverse of the Gamma tail distribution function with parameter $(n, 1)$. Then $\varphi_k^t(\epsilon) = m_k^t \gamma_1(\epsilon)$ and $\varphi_k^{t,n}(\varepsilon) = m_k^t \gamma_n(\epsilon)$. We then have

$$c^{*t} = q^t + \max_{k \in \mathcal{K}_t} v_k^t \gamma_{T_k}(\varepsilon_k), \qquad {c_*}^t = q^t + \max_{k \in \mathcal{K}_t} v_k^t \gamma_1(\varepsilon_k),$$

and

$$c^{*t}/c_*^t \leq \max_{k \in \mathcal{K}_t} \frac{q^t + v_k^t \gamma_{T_k}(\varepsilon_k)}{q^t + v_k^t \gamma_1(\varepsilon_k)},$$

where $v_k^t = m_k^t / V_k$ denotes the weight (or relative ratio) of the mean nominal service time to its required $(1-\epsilon)$-percentile upper bound V_k on the end-to-end delays for class k jobs at server t. That is, the ratio of the upper bound over the lower bound is bounded by a constant depending only on T and ε.

In addition, because Assumptions 5 - 8 are satisfied for the exponential case, the asymptotic results we obtained in previous section holds.

Corollary 15 *For each class* k, *if the response times* $\{R_k^t, 1 \leq t \leq T\}$ *are i.i.d. exponentially distributed, then the solution obtained by* (17) *is asymptotically optimal.*

5 MEAN AND TAIL GUARANTEES

To satisfy both constraint (1) and (2), the minimum required capacity is obtained by solving the following problem.

$$\textbf{(C)} \quad \min \ \sum_t g_t(x^t)$$
$$\text{s.t.} \ \ \mathbf{E}\,[R_k] \le U_k, \ k = 1, \ldots, K$$
$$\mathbf{P}\,[R_k > V_k] \le \varepsilon_k, \ k = 1, \ldots, K$$
$$x^t \ \ge \ 0, \ t = 1, \ldots, T$$

where $1/x^t = c^t - q^t$ is the extra capacity (above the minimum requirement q^t) that is allocated to server t, and R_k is a function of $x^t, t = 1, \ldots, T$.

Again the tail distribution constraints make the problem difficult. Using similar bounding techniques on the tail distribution constraints as we did in section 4, we can obtain upper bound on the optimal solution by solving

$$\textbf{(C1)} \quad \min \ \sum_t g_t(x^t)$$
$$\text{s.t.} \qquad \sum_t w_k^t x^t \ \le \ 1, \quad k = 1, \ldots, K$$
$$0 \le x^t \le 1/B_F^t, \quad t = 1, \ldots, T,$$
$$\text{where } B_F^t = \max_{k \in \mathcal{K}(t)} \min\{c : G_k^{t,T_k}(c, V_k) \ge 1 - \varepsilon_k\} - q^t;$$

or we can obtain a lower bound on the optimal solution by solving

$$\textbf{(C2)} \quad \min \ \sum_t g_t(x^t)$$
$$\text{s.t.} \qquad \sum_t w_k^t x^t \ \le \ 1, \quad k = 1, \ldots, K$$
$$0 \le x^t \le 1/B_L^t, \quad t = 1, \ldots, T,$$
$$\text{where } B_L^t = \max_{k \in \mathcal{K}(t)} \min\{c : G_k^t(c, V_k) \ge 1 - \varepsilon_k\} - q^t.$$

Based on Theorem 6, 4 and 12, the following theorem is immediate.

Theorem 16 *Under Assumption 5, the following hold:*

i). *Any feasible solution to (C1) is feasible to (C);*

ii). *The optimal solution to (C) is feasible to (C2), and the optimal solution to (C2) provides a lower bound for the solution to (C);*

iii). *Under Assumption 7-8, the optimal solutions to (C1) and (C2) x^* and x_* are asymptotically optimal in the sense that $x^{*t}/x_*^t \to 1$ for each t as $\varepsilon_k \to 0$ for all k.*

6 Robustness

One practical concern is that the mean service time requirements m_k^t are usually obtained through some measurement and prediction mechanism. Errors are very common in the measurement and predictions. Therefore one would like to allocate the capacity in such a way that not only the total capacity is minimized but most of the end-to-end response time constraints are still satisfied when the input parameters vary slightly from the predicted ones. That is, the solution is required to be robust under data uncertainty. For simplicity, we only show the robustness results for the case where only mean end-to-end response time constraints are considered.

In Section 3, we notice that the constraints on those non-bottleneck classes are less likely to be violated than the bottleneck classes when the service time requirements have some uncertainty, and the number of bottleneck classes are less than T. Therefore, when the number of servers is small, we expect the optimal solution to problem (M) to be quite robust. To demonstrate this fact through numerical example, we consider the estimate $\hat{w}_k^t$ is uniformly distributed in $[w_k^t(1-\Delta), w_k^t(1+\Delta)]$, where w_k^t is the true value. Here Δ measures the degree of uncertainty of the prediction on w_k^t. Table 6 gives the average (over 100 samples) number of violated constraints under different K, T, and Δ when the true input parameters are $(\hat{w}_k^t)$ but the capacity allocated based on the optimal solution of the problem (M) with the predicted parameters (w_k^t). The robustness of the solution is quite satisfactory because all values in Table 6 are small. This implies that if we use the solution of (M) to allocate the capacity, then only very few classes will exceed their mean end-to-end delay thresholds even if the parameter could be 25% away from what we predict.

Table 1
100 Clusters ($K = 100$)

T \ Δ	5%	10%	15%	20%	25%
4	0.79	0.89	1.15	1.45	1.7
8	1.05	1.2	1.43	1.66	1.94
12	1.49	1.64	1.86	2.1	2.39
16	1.48	1.71	1.99	2.22	2.51
20	1.48	1.76	1.98	2.25	2.68

Table 2
4 Classes ($T = 4$)

K \ Δ	5%	10%	15%	20%	25%
40	0.78	0.91	1.09	1.21	1.36
80	0.83	1.03	1.26	1.45	1.74
120	0.9	1.08	1.29	1.61	1.94
160	0.75	0.93	1.13	1.52	1.9
200	0.82	1.04	1.3	1.75	2.22

Tables 1&2: Average Number of Violated Constraints under Data Uncertainty

By looking into the value of the tables along each row and/or column, we can observe how the average number of violated constraints are affected by Δ, T

and K. When the uncertainty level Δ or the number of servers T increases, slightly more classes will experience unexpected long end-to-end delay. However, the total number of classes in the system K does not have much impact on the number of classes whose mean response time constraint is violated.

Table 3 gives the relative difference of the optimal objective values of the problems with input parameters (w_k^t) and $(\hat{w}_k^t)$. It shows that the total cost does not change much when planning under parameter (w_k^t) or $(\hat{w}_k^t)$.

T \ Δ	5%	10%	15%	20%	25%
4	0.23%	−0.07%	−0.71%	−1.5%	−2.65%
8	0.12%	−0.01%	−0.34%	−0.88%	−1.58%
12	0.06%	−0.01%	−0.68%	−0.97%	−1.3%
16	−0.4%	−0.49%	−1.19%	−0.66%	−1.78%

Table 3
Relative changes of the optimal cost under perturbation $K = 100$

7 Conclusion

We have investigated a resource allocation problem in a clustered system environment delivering end-to-end performance guarantees. Specifically, we considered service guarantees on both the mean and on the tail distribution of the end-to-end response times for each class of requests. For the problem with mean delay guarantees alone, we present a nonlinear program formulation and provide a nice geometric interpretation of optimal solution structure. For the problem with tail distribution guarantees, we develop an approximation method to solve the problem. Under suitable assumptions, we gave a constant factor bound for the solution and also showed that it is asymptotically optimal. These assumptions are quite general, and are easily satisfied by a collection of common problems. Numerical results further demonstrated the solutions are robust under data uncertainty.

Although these results were established with the restriction to Poisson arrivals and feed-forward Kelly-type networks, they can be applied as good approximations to capture the high-level queueing dynamics of more general systems. Future research can focus on more general arrival processes and general network types.

References

[1] F. Baccelli, S. Foss (2004). Moments and Tails in Monotone-Separable Stochastic Networks, *Ann. Appl. Probab.*, 14, 612-650.

[2] F. Baccelli, S. Foss and M.Lelarge. (2003). Tails in Generalized Jackson Networks with Subexponential Service Distributions. To appear in *Ann. Appl. Prob.*.

[3] M. S. Bazaraa, H. D. Sherali and C. M. Shetty (1993). *Nonlinear Programming*, Second Edition, John Wiley & Sons, New York. 25

[4] P.J. Burke (1964). The dependence of delays in tandem queues. *Ann. Math. Statist.* 35 874-875.

[5] S. Hochbaum and J. G. Shanthikumar (1990). Convex separable optimization is not much harder than linear optimization, *J. Assoc. Comput. Mach.*, 37 pp. 843–862.

[6] P. Jelenkovic and P. Momcilovic (2002). Resource Sharing with Subexponential Distributions. *Proceedings of IEEE INFOCOM*, June 2002, Vol. 3, 1316-1325.

[7] N. Katoh and T. Ibaraki. *Resource Allocation Problems: In Handbook of Combinatorial Optimization, D.-Z. Du and P. Pardalos, editors.* Kluwer Academic Publishers, Boston, Massachusetts, 1998.

[8] F.P. Kelly, *Reversibility and Stochastic Networks*, Second Ed., John Wiley & Sons; 1979.

[9] L. Kleinrock, *Queueing Systems Volume II: Computer Applications.* John Wiley & Sons, 1976.

[10] W.E. Leland and T.J. Ott (1986). Load-balancing heuristics and process behavior. *Proceedings of Performance and ACM Sigmetrics*, pp. 54-69.

[11] W. Lin, Z. Liu, C.H. Xia and L. Zhang. (2005) Optimal Capacity Allocation for Web Systems with End-to-End Delay Guarantees. *IBM Research Report*, RC-XXXX, 2005.

[12] Z. Liu, M.S. Squillante and J.L. Wolf. (2002) Optimal Control of Resource Allocation in e-Business Environments with Strict Quality-of-Service Performance Guarantees, *CDC 2002.*

[13] D. A. Menascé, V. A. F. Almeida, R. Fonseca, and M. A. Mendes (2000). Business-oriented resource management policies for e-commerce servers. *Performance Evaluation*, 42:223–239.

[14] R. Nunez-Queija. *Processor-sharing Models for Integrated-Services Networks.* Ph.D. thesis, Eindhoven University of Technology, Jan. 2000.

[15] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models.* John Wiley and Sons, Berlin, 1983.

[16] J. Walrand. *An introduction to queuing networks.* Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[17] R.W. Wolff, *Stochastic Modeling and the Theory of Queues*, Prentice Hall, 1989.

[18] A.P. Zwart and O.J. Boxma (2000). Sojourn time asymptotics in the M/G/1 processor sharing queue. *Queueing Syst. Theory and Appl.*, 35(1/4):141-166.