APPROXIMATE MODEL FOR A MERGE CONFIGURATION OF MULTISERVERS WITH FINITE CAPACITY INTERMEDIATE BUFFERS

Angel Vassilev Nikolov

Department of Mathematics and Computer Science
National University of Lesotho
Roma 180, LESOTHO

Abstract: We consider a queueing network consisting of multiservers in parallel and connected to a merger queue. All servers have an infinite capacity buffers. The configuration is decomposed into two subsystems: merging multiservers and merger server and then analyzed in isolation. All times are assumed to be exponentially distributed. First we set and solve the equations for the aggregated merger queue, then express the occupancy probabilities of the original queue through the probabilities of the aggregated one, which reduces significantly the number of equations describing the behavior of the network. Using the fore-mentioned result, we derive expressions for the occupancy probabilities and other parameters of the merging queue.

AMS Subject Classification: 90B22

Key Words: multiserver, merging, flow conservation, aggregation

1. Introduction

Queueing network models are extensively used to describe the behavior of complex real-life physical systems in manufacturing and production, health care, telecommunication, computer and software engineering, see e.g. [2, 7, 8, 9, 11]. The methods applied fall into two major categories – simulation and analytical. The simulation most sophisticated and detailed representation of of the system and most accurate results related to performance [8]. It is most expensive, how-

ever, in terms of development and computing time, and prove to design errors because of its complexity.

The analytical models are simpler and less expensive and in many cases, especially at early design stages, provide good estimates of the performance metrics. The general exact model represents the network by a continuous time discrete-state Markov chain [1]. Blockings of different types are taken into account by the number of blocked servers in the states of the network. These methods are the most accurate analytical methods but applicable mostly to small networks because the number of states grows enormously for a large number of servers. Mean Value Analysis (MVA) is used for closed configurations, infinite intermediate buffer, and some types of blocking [9]. Under some assumptions MVA can be applied to finite capacity buffers.

Approximate approaches significantly overcome the limitations of the exact methods. Diffusion approximation is based on the replacement of the discrete queueing process by a continuous one, without losing the characteristics of the original process [4]. It gives relatively accurate results for large networks although accuracy is difficult to be checked. Decomposition principle, one of most commonly used approach is implemented in three steps:

- 1) network decomposition into subsystems,
- 2) analysis of each subsystem in isolation,
- 3) analysis of the new aggregated system.

The methods differ usually in Steps 2 and 3. Phase-type approximation (PH) assumes PH distributed service times, and is restricted to small networks due to the complexity of the PH mechanism [6]. In [7] the network is broken down to nodes (single servers or multiservers) and a system of equations for the underlying Markov process is developed. The equations for all nodes are to be solved simultaneously. The states of each node are the numbers of active, waiting and blocked units, respectively, and this gives rise to the number of the unknowns. Another heuristic model is developed in [10]. Introduction of pseudo-arrival rates and effective service rates makes possible analysis of broad range of network types where blocked arrivals are rejected. The occupancy probabilities for the aggregated merger server of merge configuration with single servers and finite buffers can be computed from a simple birth-death process, and then performance metrics can be expressed as a function of these probabilities, thus significantly reducing the number of the unknowns [5, 6]. In this paper we develop model of merged multiservers also taking the advantage of the birth-death process.

2. Model Description and Analysis

The system studied consists of M parallel multiservers, or merging queues, with outputs connected to a single server, or merged queue, referred below as a server θ or queue θ as well. The arrivals to multiserver i are independent and follow exponential distribution with a rate λ_i ($i = 1, \dots, M$). Each multiserver has N_i single identical units with exponentially distributed service times with rate μ_i and a common buffer with B_i positions ($i = 1, \dots, M$). The service time of the server θ is also exponentially distributed with rate μ and its buffer size is N . The index i can be viewed as the sequential number of the multiservers and runs from 1 to M unless otherwise stated. The arrivals to any multiserver are dispatched to some of its units if there is one free, and if all single units are occupied the request waits in the corresponding buffer. When an arrival finds no free position in the buffer (buffer is full), however, it is lost, or rejected. After completion of the service at multiserver i the request proceeds to server θ , and is taken immediately into service if the server is free (no queue at server θ), otherwise is placed into the buffer θ and waits. Requests at queue θ are served on First Come First Served (FCFS) basis until there is at least one free position in buffer θ . When the buffer θ becomes full all incoming arrivals from multiservers i are blocked, and the multiservers are forced to wait, so that there is no loss of requests at server θ , but some delay is inserted due to the fact the this blocking after service (BAS) leads to a closing of a single unit from the multiserver from which the request originates. The number of the operating units of that multiserver is reduced by one. As the time progresses the number of units L_i forced to shut down can vary from θ to N_i , i.e. at some point all units of multiserver i are blocked and it produces no requests to server θ . Blocking contributes to the degradation of the total system throughput, i.e. to the increase of the number of the rejected requests at the inputs of the multiservers by inserting additional time required to wait to join the merged queue. If $j = \sum_{i=1}^M L_i$ units are blocked by the merged queue then any newly generated request from the merging queues must wait for the server θ to complete the service of $j+1$ requests, or in other words to make free $j+1$ places in its buffer. The approach used is to decompose the system into individual queues, and analyze each individual queue separately. We assume that there are Poisson arrivals with rate $\tilde{\lambda}_i^{N_i-L_i}$ from multiserver i to server θ as long as $N_i - L_i$ units are not blocked. Obviously arrival rates decrease as the number of the blocked unit L_i increases, and $\tilde{\lambda}_i^0 = 0$, that is at some point multiserver i becomes fully detached. Since blocked units are forced to wait at queue θ the size of the

buffer of queue θ can be regarded as extended by $K = \sum_{i=1}^M N_i$ places so its total size is $N + K$.

We introduce the following notations:

$T_i^{N_i-L_i}$ – clearance time of multiserver i given that $N_i - L_i$ units are not blocked. This is the time between entrance of service at queue i and arrival at queue θ . If there are no blocked units at queue θ $T_i^{N_i-L_i}$ is equal to the service time of the unit $\frac{1}{\mu_i}$. If, however, j units in the system are blocked the time needed to complete the service of $j + 1$ requests by server θ must be added to $\frac{1}{\mu_i}$ as explained above. Its distribution represents the sum of $j + 1$ independent exponentially distributed random variables, each of which has a parameter of μ .

$f_i^{N_i-L_i}$ P[i^{th} queue is full given $N_i - L_i$ units are operational (not blocked)]
 $\bar{\lambda}_i^{N_i-L_i}$ arrival rate to the multiserver i as long as $N_i - L_i$ units are operational (not blocked). Apparently rejected (lost) requests are excluded from this rate.

$b_i^{N_i-L_i}[n]$ P[more than L_i units of multiserver i are blocked and the number of requests (waiting + blocked units) at queue θ is $N + n$]

$\bar{\lambda} = \sum_{i=1}^M \bar{\lambda}_i$ total system throughput

$\delta_{m,n}$ – the Kronecker delta symbol

$\sum_n^m = 0$, if $n > m$

$\prod_n^m = 1$, if $n > m$.

The system is considered to be balanced and all probabilities in this paper are steady-state ones.

2.1. Analysis of the Merged Queue

The goal of this analysis is to express the occupancy probabilities of queue θ as a function of the arrivals rates $\tilde{\lambda}_i^{N_i-L_i}$, N_i , and N . The arrival rates $\tilde{\lambda}_i^{N_i-L_i}$ are measured on the output of the multiserver i . The states of the queue are represented by the number of the requests in the queue, but if there is a blocking the order of the blocked units counts, since they are being unblocked on a FCFS basis, and consequently the number of the states increases enormously. The problem is dealt with disregarding the order of the blocked units, and aggregating the states, which have same number of units into one state. The states of such aggregated system are denoted by k ($k = 0, \dots, N + K$) and represent the number of units waiting for service at queue θ including the

blocked units, if any. We define $\lambda_a(k)$ as a transition rate from state k to state $k + 1$ for the aggregated system. After solving the aggregated queue the occupancy probabilities inclusive of the order of arrivals of the blocked units are expressed in terms of the aggregated probabilities. The aggregated states of queue θ follow simple birth and death process [3] and its steady-state probabilities are determined by the following expressions

$$P[k] = P[k-1] \frac{\lambda_a(k-1)}{\mu} \quad (k = 1, \dots, N+K), \quad (1)$$

$$P[k] = P[0] \prod_{j=1}^k \frac{\lambda_a(j)}{\mu} \quad (k = 1, \dots, N+K), \quad (2)$$

$$\sum_{k=0}^{N+K} P[k] = 1. \quad (3)$$

Let us denote by $\mathbf{L}_n^p = [L_1, \dots, L_i, \dots, L_M]$ an integer $1 \times M$ vector for which both $0 \leq L_1 \leq N_1, \dots, 0 \leq L_i \leq N_i, \dots, 0 \leq L_M \leq N_M$ and $\sum_{i=1}^M L_i = n$ hold. Superscript p points to a particular vector from the set of such vectors. Let us also define an $1 \times n$ vector $\mathbf{z}_n^{p,q} = [z_1, \dots, z_l, \dots, z_n]$ ($z_l \in \{1, \dots, i, \dots, M\}$) for which $\sum_{l=1}^n \delta_{i,z_l} = L_i$ holds for all L_i elements of the vector $\mathbf{L}_n^p$. Index q runs from 1 to $\prod_{i=1}^M \binom{n - \sum_{l=1}^{i-1} L_l}{L_i}$, where the term $\prod_{i=1}^M \binom{n - \sum_{l=1}^{i-1} L_l}{L_i}$ is the number of ways $L_1, \dots, L_i, \dots, L_M$ distinct elements can be placed in n distinct places. In the context of nonaggregated queue θ each vector $\mathbf{z}_n^{p,q}$ represents a unique order of arrivals from multiservers i ($i = 1, \dots, M$), where the numbers of units from each multiservers are uniquely determined by the vector $\mathbf{L}_n^p$. For example if $M = 3$, $n = 4$ and $\mathbf{L}_4^p = [2, 0, 2]$ the corresponding vectors $\mathbf{z}_4^{p,q}$ are $\mathbf{z}_4^{p,1} = [1, 1, 3, 3]$, $\mathbf{z}_4^{p,2} = [1, 3, 1, 3]$, $\mathbf{z}_4^{p,3} = [1, 3, 3, 1]$, $\mathbf{z}_4^{p,4} = [3, 3, 1, 1]$, $\mathbf{z}_4^{p,5} = [3, 1, 3, 1]$, and $\mathbf{z}_4^{p,6} = [3, 1, 1, 3]$. The occupancy probabilities of the original queue θ are denoted as $P[k]$ for $k = 1, \dots, N$ and $P[N+n; \mathbf{z}_n^{p,q}]$ for $n = 1, \dots, K$. The relationship between the steady-state probabilities of the original queue θ and those of the aggregated queue θ is:

$$P[N+n; z_1, \dots, z_l, \dots, z_n] = P[N+n] \frac{\prod_{i=1}^M \prod_{k=N_i-L_i+1}^{N_i} \tilde{\lambda}_i^k}{\Omega_n}, \quad (4)$$

where

$$\Omega_n = \sum_{\forall \mathbf{L}_n^P} \prod_{i=1}^M \binom{n - \sum_{l=1}^{i-1} L_l}{L_i} \prod_{k=N_i-L_i+1}^{N_i} \tilde{\lambda}_i^k. \quad (5)$$

Proof. We define the aggregated rates $\lambda_a(k)$ as follows:

$$\lambda_a(k) = \tilde{\lambda} = \sum_{i=1}^M \tilde{\lambda}_i^{N_i}; \quad k = 1, \dots, N \quad (6)$$

and

$$\lambda_a(N+n) = \frac{\Omega_{n+1}}{\Omega_n}; \quad n = 1, \dots, K-1. \quad (7)$$

The balance equations of the original nonaggregated system are

$$P[0] \tilde{\lambda} = P[1] \mu, \quad (8)$$

$$P[k] (\tilde{\lambda} + \mu) = P[k-1] \tilde{\lambda} + P[k+1] \mu, \quad k = 1, \dots, N-1, \quad (9)$$

$$P[N] (\tilde{\lambda} + \mu) = P[N-1] \tilde{\lambda} + \sum_{i=1}^M P[N+1; i] \mu, \quad (10)$$

$$\begin{aligned} P[N+n; z_1, \dots, z_l, \dots, z_n] & \left(\sum_{i=1}^{N_i} \left(1 - \delta_{\mathbf{L}_n^P(i), N_i} \right) \tilde{\lambda}_i^{N_i - \mathbf{L}_n^P(i)} + \mu \right) \\ & = P[N+n-1; z_1, \dots, z_l, \dots, z_{n-1}] \tilde{\lambda}_{z_n}^{N_{z_n} - \mathbf{L}_n^P(z_n)} \\ & + \sum_{i=1}^M P[N+n+1; i, z_1, \dots, z_l, \dots, z_n] \mu; \quad n = 1, \dots, K-1, \end{aligned} \quad (11)$$

$$P[N+K; z_1, \dots, z_l, \dots, z_K] \mu = P[N+k; z_1, \dots, z_l, \dots, z_{K-1}] \tilde{\lambda}_{z_K}^1. \quad (12)$$

If we insert (1) and (4) into the above equations after some simplifications one can show that they are satisfied, and this concludes the proof.

2.2. Analysis of the Merging Queues

In this subsection we first express the rates $\tilde{\lambda}_i^{N_i-L_i}$ in terms of the parameters of multiservers. Applying the conservation flow rule to queue i yields

$$\begin{aligned} \bar{\lambda}_i^{N_i-L_i} &= \tilde{\lambda}_i^{N_i-L_i} P[\text{there are } N_i-L_i \text{ nonblocked units at multiserver } i] \\ &= \tilde{\lambda}_i^{N_i-L_i} \left(\delta_{L_i,0} \sum_{k=0}^N P[k] + \sum_{\forall(\mathbf{z}_n^{\mathbf{P},\mathbf{Q}} \ \& \ \mathbf{L}_n^{\mathbf{P}}(i)=L_i)} P[N+n; z_1, \dots, z_l, \dots, z_n] \right); \\ L_i &= 0, \dots, N_i - 1, \end{aligned} \quad (13)$$

and

$$\bar{\lambda}_i = \sum_{L_i=0}^{N_i} \bar{\lambda}_i^{N_i-L_i}. \quad (14)$$

By summing the above equations over i , we obtain

$$\sum_{k=0}^{N+K-1} P[k] \lambda_a(k) = \bar{\lambda}, \quad (15)$$

which is in fact the conservation flow for queue θ .

The rate $\bar{\lambda}_i^{N_i-L_i}$ is the rate measured at the input of queue θ and it is smaller than $\tilde{\lambda}_i^{N_i-L_i}$ due to the blocking.

From the definition of $b_i^{N_i-L_i}$ it follows

$$\begin{aligned} b_i^{N_i-L_i}[0] &= 0, \\ b_i^{N_i-L_i}[n] &= \sum_{\forall(\mathbf{z}_n^{\mathbf{P},\mathbf{Q}} \ \& \ \mathbf{L}_n^{\mathbf{P}}(i)>L_i)} P[N+n; z_1, \dots, z_l, \dots, z_n]; \\ L_i &= 0, \dots, N_i - 1. \end{aligned} \quad (16)$$

After inserting (4) into the above, one can show that

$$b_i^{N_i-L_i}[n] = P[N+n] \sum_{\forall(\mathbf{z}_n^{\mathbf{P},\mathbf{Q}} \ \& \ \mathbf{L}_n^{\mathbf{P}}(i)>L_i)} \frac{\prod_{i=1}^n \prod_{k=N_i-L_i+1}^{N_i} \tilde{\lambda}_i^k}{\Omega_n}. \quad (17)$$

We now define $\alpha_i^{N_i-L_i}[N+j]$ as the conditional probability that, upon completion of service at server i when N_i-L_i units are operational and there are $N+j$ units at queue θ , a request is blocked. It can be viewed as well as a probability that at the point of completion of service at multiserver i , the new

arrival sees only L_i units blocked, there are $N + j$ units in the extended buffer of server 0 , given that $N_i - L_i$ units are not blocked,

$$\alpha_i^{N_i-L_i} [N + j] = \frac{P [N + j] - b_i^{N_i-L_i} [j]}{1 - \sum_{n=1}^K b_i^{N_i-L_i} [n]};$$

$$i = 1, \dots, M, \quad j = 0, \dots, K. \quad (18)$$

The expected value of the clearance time $E[T_i^{N_i-L_i}]$ is

$$E [T_i^{N_i-L_i}] = \frac{1}{\mu_i} + \sum_{j=0}^{K-1} \alpha_i^{N_i-L_i} [N + j] \frac{j+1}{\mu}; \quad i = 1, \dots, M. \quad (19)$$

We set $\rho_i^{N_i-L_i} = \lambda E[T_i^{N_i-L_i}]$ and derive the probabilities of rejection $f_i^{N_i-L_i}$, [3],

$$f_i^{N_i-L_i} [B_i] = P_i^{N_i-L_i} [0] \frac{\left(\rho_i^{N_i-L_i}\right)^{N_i-L_i}}{(N_i - L_i)!}; \quad B_i \leq N_i - L_i,$$

$$f_i^{N_i-L_i} [B_i] = P_i^{N_i-L_i} [0] \frac{\left(\rho_i^{N_i-L_i}\right)^{B_i}}{(N_i - L_i)^{B_i-N_i+L_i} (N_i - L_i)!};$$

$$B_i > N_i - L_i, \quad (20)$$

where

$$P_i^{N_i-L_i} [0] = \left[\sum_{k=0}^{N_i-L_i} \frac{\rho_i^k}{k!} + \frac{\rho_i^{N_i-L_i}}{(N_i - L_i)!} \sum_{k=N_i-L_i+1}^{B_i} \frac{\rho_i^k}{(N_i - L_i)^k} \right]^{-1}.$$

By definition $\bar{\lambda}_i^{N_i-L_i}$ is determined from the formulae

$$\bar{\lambda}_i^{N_i-L_i} = \lambda_i \left(1 - f_i^{N_i-L_i} [B_i] \right), \quad [3]. \quad (21)$$

The majority of the analyses of queueing networks having splitting assume the routing probabilities from/to queues are constants and known in advance. In some cases, however, for example, asynchronous web servers, this concept is not applicable [8]. After completion of service at queue 0 the job is routed to a server for specific processing tied to the upstream. The jobs processed at the multiserver i after server 0 proceed to a server that processes only requests originating from this particular multiserver. The routing probability equals obviously $\bar{\lambda}_i/\bar{\lambda}$.

3. Example

The state-transition diagram for a configuration of two multiservers each having two single servers is shown in Figure 1.

Figure 1

The occupancy probabilities are

$$\begin{aligned}
 P[N+1; 1] &= P[N+1] \frac{\tilde{\lambda}_1^2}{\Omega_1}; \quad P[N+1; 2] = P[N+1] \frac{\tilde{\lambda}_2^2}{\Omega_1}; \\
 P[N+2; 1, 1] &= P[N+2] \frac{\tilde{\lambda}_1^2 \tilde{\lambda}_1^1}{\Omega_2}; \quad P[N+2; 2, 2] = P[N+2] \frac{\tilde{\lambda}_2^2 \tilde{\lambda}_2^1}{\Omega_2}; \\
 P[N+3; 1, 1, 2] &= P[N+3; 1, 2, 1] = P[N+3; 2, 1, 1] = \\
 &= P[N+3] \frac{\tilde{\lambda}_1^2 \tilde{\lambda}_1^1 \tilde{\lambda}_2^2}{\Omega_3}; \\
 P[N+3; 1, 2, 2] &= P[N+3; 2, 1, 2] = P[N+3; 2, 2, 1] = \\
 &= P[N+3] \frac{\tilde{\lambda}_1^2 \tilde{\lambda}_2^2 \tilde{\lambda}_2^1}{\Omega_3};
 \end{aligned}$$

$$\begin{aligned}
P[N+4; 1, 1, 2, 2] &= P[N+4; 1, 2, 1, 2] = P[N+4; 1, 2, 2, 1] = \\
&P[N+4; 2, 1, 1, 2] = \\
P[N+4; 2, 1, 2, 1] &= P[N+4; 2, 2, 1, 1] = P[N+4] \frac{\tilde{\lambda}_1^2 \tilde{\lambda}_1^1 \tilde{\lambda}_2^2 \tilde{\lambda}_2^1}{\Omega_4},
\end{aligned}$$

where

$$P[N+n] = P[N+n-1] \frac{\lambda_a(n)}{\mu}, \quad n = 1, 2, 3;$$

$$\begin{aligned}
\lambda_a(N+n) &= \frac{\Omega_{n+1}}{\Omega_n} \text{ for } n = 1, 2, 3 \text{ and } \lambda_a(N+4) = 0; \quad \Omega_1 = \tilde{\lambda}_1^2 + \tilde{\lambda}_2^2; \quad \Omega_2 = \\
&\tilde{\lambda}_1^2 \tilde{\lambda}_1^1 + 2\tilde{\lambda}_1^2 \tilde{\lambda}_2^2 + \tilde{\lambda}_2^2 \tilde{\lambda}_2^1; \quad \Omega_3 = 3\tilde{\lambda}_1^2 \tilde{\lambda}_1^1 \tilde{\lambda}_2^2 + 3\tilde{\lambda}_1^2 \tilde{\lambda}_2^2 \tilde{\lambda}_2^1; \quad \Omega_4 = 6\tilde{\lambda}_1^2 \tilde{\lambda}_1^1 \tilde{\lambda}_2^2 \tilde{\lambda}_2^1.
\end{aligned}$$

4. Concluding Remarks

The merits of this model could be summarized as follows:

- 1) Relatively small number of equations.
- 2) Deep insight into the performance metrics since all probabilities, including those of all permutations of the blocked units can be calculated with insignificant computational effort.
- 3) Applicable to different splitting configurations since the output rates of the downstream of queue 0 are explicitly computed.

For $N_i = 1$, $i = 1, \dots, M$, the equations derived here fully coincide with those reported in [5], [6].

References

- [1] I.A. Akyyildiz, H.V. Brand, Exact solutions for networks of queues with blocking-after-service, *Theoretical Comp. Sci.*, **125** (1994), 111-130.
- [2] S. Balsamo, V. De N. Persone, P. Inverardi, A review on queueing network models with finite capacity queues for software architectures performance prediction, *Perf. Eval.*, **974** (2002), 1-20.
- [3] U.N. Bhat, *Introduction to Queueing Theory: Modelling and Analysis in Applications*, Springer (2008).

- [4] H. Chen, H.Q. Ye, Methods of diffusion approximation for multi-server systems: Sandwich, uniform attraction and state-space collapse, In: *Queueing Networks: a Fundamental Approach*, Springer (2011), 488-530.
- [5] H.S. Lee, S.M. Pollock, Approximate analysis for the merge configuration of an open queueing network with blocking, *Technical Report, University of Michigan*, **87-11** (1987), 1-28.
- [6] H.S. Lee, S.M. Pollock, Approximate analysis of open exponential queueing networks with blocking: General configuration, *Technical Report, University of Michigan*, **48-109** (1987), 1-31.
- [7] C. Osorio, M. Bierlaire, A finite capacity queueing network model capturing blocking, congestion and spillbacks, *European J. Operational Res.*, **196** (2009), 996-1007.
- [8] U. Praphamontripong, S. Gokhale, A. Gokhale, J. Gray, Performance analysis of an asynchronous web server, In: *Comp. Software and Appl. Conf., 2006 COMPSAC'06 30th Ann. Intl.*, **2** (2006).
- [9] R. Suri, S. Sahu, M. Vernon, Approximate mean value analysis for closed queueing networks with multiple-server stations, In: *Proc. of the 2007 Ind. Eng. Res. Conf.*, Nashville (2007).
- [10] Y. Takahachi, H. Miyahara, T. Hasegawa, An approximation method for open restricted queueing networks, *Operational Res.*, **28** (1980), 594-602.
- [11] Y. L. Zhu, S. Y. Zhu, H. Xiong, Performance analysis and testing of the storage area network, In: *Proc. of the 10th NASA Goddard Conf. on Mass Storage Systems and Technologies 2002*, Singapore (2002).