A Multiserver Approximation for Cloud Scaling Analysis

Siyu Zhou
Dept of Systems and Computer Engineering
Carleton University
Ottawa, Canada
zhousiyut@gmail.com

Murray Woodside
Dept of Systems and Computer Engineering
Carleton University
Ottawa, Canada
cmw@sce.carleton.ca

ABSTRACT

Queueing models of web service systems run at increasingly large scales, with large customer populations and with multiservers introduced by scaling up the services. "Scalable" multiserver approximations, in the sense that they that are insensitive to customer population size, are essential for solution in a reasonable time. A thorough analysis of the potential errors, which is needed before the approximations can be used with confidence, is the goal of this work. Three scalable approximations are evaluated: an equivalent single server SS, an approximation RF introduced by Rolia, and one based on a binomial distribution for queue state AB. AB and SS are suggested by previous work but have not been evaluated before. For AB and SS, multiple classes are merged into one to calculate the waiting. The analysis employs a novel traffic intensity measure for closed multiserver workloads. The vast majority of errors are less than 1%, with the worst cases being up to about 30%. The largest errors occur near the knee of the throughput/response time curves. Of the approximations, AB is consistently the most accurate and SS the least accurate.

CCS CONCEPTS

• Software and its engineering → Software performance
• Computer systems organization → Cloud computing
• Mathematics of computing → Queueing theory

KEYWORDS

Web scaling, software performance, multiserver queueing, approximations.

ACM Reference format:

Siyu Zhou, Murray Woodside. 2018. A Multiserver Approximation for Cloud Scaling Analysis. In *In Companion of the 2022 ACM/SPEC International Conference on Performance Engineering (ICPE '22 Companion), April 9–13, 2022, Bejing, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3491204.3527472

ICPE '22 Companion, April 9–13, 2022, Beijing, China.

ACM ISBN 978-1-4503-9159-7/22/04...$15.00.
https://doi.org/10.1145/3491204.3527472

1 Introduction

Cloud performance management can only exploit contention (queueing) models if they can be solved quickly and robustly; accurate prediction is important but secondary. This work seeks improvement in analyzing FIFO multiservers, which are the usual model for contention at a replicated software server.

FIFO multiservers are the rule in web service systems deployed in clouds. In a performance model there is no exact solution for these servers, so approximations must be used to predict the queueing delays. Most known solutions require calculation effort which increases at least linearly with the user population (e.g. [10]), which is often huge (related work is described in Section 6). One well-established scalable approximation by Rolia [11] has convergence problems, as will be described further below.

This work contributes a new and intensive evaluation of three approximations for the waiting time of *closed FIFO multiservers* which are insensitive to the user population size:

- RF, the approximation by Rolia [11]
- SS, an equivalent single server, which has been used in cloud models (without much scrutiny, e.g. [14][15])
- AB, which uses a binomial distribution for the queue.

Their accuracy is evaluated over a wide range of parameters, which indicates that AB is the best by a modest margin.

The scalability problem is illustrated by the solution times for the simple model shown in Figure 1. It represents a set of 50 service instances attached to a load-balancer, which is subsumed in the queue. There are three classes of customers with sizes [2000, 400, 100].

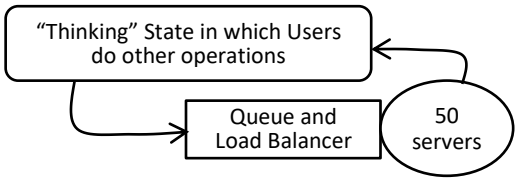


Figure 1 A Simple Multiserver Model for a Web Service

Table 1 compares the times by the LQNS solver [6] using three multiserver approximations. RF [11], and Conway [5] are the most-used built-in algorithms in the solver, while AB is new. The times shown are the average of 1000 LQNS solutions. The impact of even a linear term in N or C is evident.

Table 1 Comparison of LQNS Solution Times

Approximation	RF	Conway	AB
Complexity per iteration (N customers, m servers, C classes)	O(C)	O(NC^3)	O($m + C$)
Average of 1000 solution times (ms)	0.786	401.4	0.705

AB and RF are about the same, and Conway is much slower. For the same model one simulation, giving confidence intervals of 0.2%, took 1001 s., 2500 times as long as Conway.

2. The Model

The abstract model is a closed FIFO M/M/m//N queue serving a single class of customers:

- N = the number of customers,
- Z= the mean "think time" (between leaving the server and arriving for the next service) ,
- S= the mean service time (normalized to 1),
- T = the traffic intensity, the ratio of the maximum arrival rate N/Z to the maximum departure rate m/S:

$$T = NS/(mZ) \quad (1)$$

- W = mean waiting time,
- $R = W + S$ = response time ,
- $\lambda = N/(R + Z)$ = throughput in responses/s.,
- L^* = an estimate of the mean expected number in queue (not service) when there are N-1 customers, used in approximate Mean Value analysis (AMVA),
- λ^* = the corresponding throughput ,
- P = the probability that a customer is in the queue or at the server:

$$P = (W + S)/(W + S + Z) \quad (1)$$

- $p(i)$ = probability that there are i customers in the queue+server,
- PB = the probability that all m servers are busy.

The service time S is normalized to 1 so that in general, Z represents Z/S. Approximate results are subscripted by the algorithm name, represented here by a subscript APP.

- W_{APP} and R_{APP} = approximate mean waiting time and response time found by method APP (APP is one of Exact, AB, RF, SS, Sim) .
- It is useful to recall the asymptotic bounds of R for a single class with think time Z and service time S [9]:
- $R > \max(S, NS/m - Z)$
- which are shown in Figure 2. We see that T = 1 at the point where the diagonal bound crosses the axis, near the intersection of the bounds. The actual response time curve has a knee near this point, where it turns upwards, and it is near this value that the largest approximation errors were found.

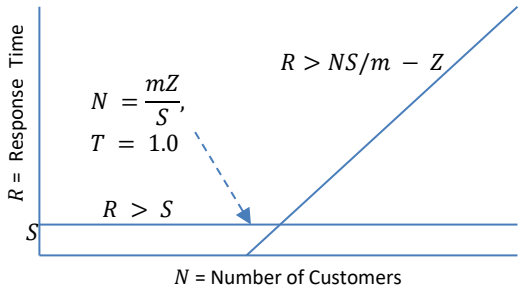


Figure 2 The Asymptotic Bounds on R for One Class

The exact solution for the state probabilities is found from a birth-death Markov model (see, e.g. [2]). Then the waiting and response times are given (using Little's formula) by:

$$\lambda_{\text{Exact}} = \sum_i \min(i, m)p(i)/S \;;$$
$$R_{\text{Exact}} = N/\lambda_{\text{Exact}} - Z;$$
$$W_{\text{Exact}} = R_{\text{Exact}} - S \quad (2)$$

Errors will be reported as their relative error RE, normalized to the response time. For approximation APP, the error is

$$RE(\text{APP}) = (W_{APP} - W_{Exact})/R_{Exact} \quad (3)$$

The algorithms APP to be reported include Equivalent Single Server (SS), Rolia-Franks (RF) and the Arrival-theorem Binomial (AB).

The notation ARE is used for the absolute RE, and $MARE$ for the mean ARE.

3. The Approximations for One Class

3.1 Rolia-Franks (RF, giving W_{RF})

Rolia's approximation [11] modified by Franks [7] assumes the servers are independent, to estimate the probability PB that all servers are busy. Conditioned on all servers busy, it estimates waiting by a conventional iterative AMVA approach. The update to W at each iteration is:

$$W = S + [(U(1)^*)^m/m]\; S\, L^* \quad (4)$$

where $U(1)^*$ is the utilization of each server separately with one less customer, and L^* is the expected customers to wait for. The mean queue length L^* with one less customer is found from the previous value of W by any AMVA approach. For example the Bard-Schweitzer Proportional Estimation (PE) algorithm [12] gives the iterative relationship:

$$L^* \cong [(W + S)/(W + S + Z)][(N - 1)/N] \quad (5)$$

In applying RF with a fixed-point iteration, convergence requires under-relaxation of the form

$$\text{updated } W = \alpha\ (\text{new } W) + (1 - \alpha)(\text{previous } W) \quad (6)$$

with a relaxation parameter α that was set here to 0.2. The complexity of RF with PE for one multiserver queue is O(1) per iteration. This makes RF highly scalable, and it has been the algorithm of choice in the LQNS solver for many years. However there have been problems with convergence.

3.2 The Equivalent Single Server (SS, giving W_{SS})

Some authors (e.g. [14][15]) have approximated the waiting time for a set of servers, used to scale up a single service in a cloud deployment, by a single faster server. A set of m servers is represented by a single server of m times the speed (giving a service time S/m). To provide the correct total delay at very light loads, an additional delay $S(1 - 1/m)$ is added to represent the remaining service delay (this is not found in the references). The additional delay is added to the response time of the server, but does not contribute to its utilization. For a closed model the exact solution may be used, making its complexity O(N) for a single class, but using AMVA it has lower complexity. For example with PE the time complexity of each iteration is O(1); with Linearizer [4] it is larger but also O(1).

3.3 The "Arrival-theorem Binomial" Approximation (AB)

AB assumes that the movement of customers between the thinking and server states is independent, with the probability for each customer being at the server (waiting or in service) of $P = R/(Z + R)$. Independence gives a binomial distribution for the customers at the server, from which the probability $p^{(B)}(i)$ of i customers is found for $i = 0$ to $m - 1$. This is similar to the single-class version of the approach by deSouza e Silva and Muntz described in [13].

$$p_B(i) = (N!\,(N-i)!/i!)P^i(1-P)^{N-i}$$

The probability that all servers are busy is PB_B:

$$PB_B = 1 - \sum_1^{m-1} p^{(B)}(i)$$

The throughput is then

$$\lambda = (1/S)\sum_1^{m-1} ip^{(B)}(i) + (m/S)PB_B$$

W can be found from this throughput using Little's formula but it was found to give inferior estimates. A better approach called AB was found by applying the Arrival Theorem [10] and considering a queue with $N - 1$ customers, indicated by a superscript (N-1). The probability P is assumed to be the same, and L^* estimates the mean number in in the queue:

$$p_{AB}^{(N-1)}(i) = ((N-1)!\,(N-1-i)!/i!)P^i(1-P)^{N-1-i}$$
$$L^* = \sum_{i=m+1}^{N-1}(i-m)p_{AB}^{(N-1)}(i) \qquad (7)$$

Eq (7) can be re-arranged so that it uses only the first m probabilities:

$$L^* = (N-1)P - (m-1)(1-\sum_{i=1}^{m-1} p^{(N-1)}(i)) - \sum_{i=1}^{m-1} ip^{(N-1)}(i) \qquad (8)$$

By the Arrival Theorem a customer waits on average for L^* departures before entering service, giving

$$W_{AB} = L^* S/m$$

Using Eq. (8) the complexity of AB is O(m).

4. Approximation Accuracy

Figure 3 shows a quick look at the approximation errors for RF, and AB, for a model with N =100. The traffic level T takes values from 0 to 5, m takes a range of values as shown, S = 1 and. Z was computed from T as:

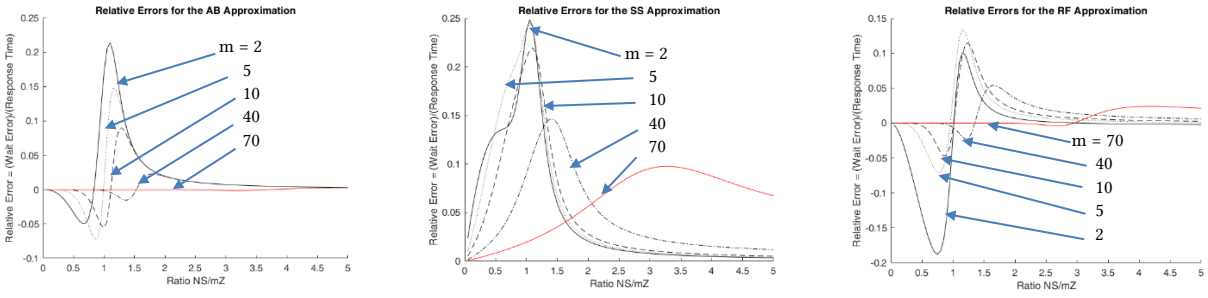
$$Z = N/(mT)$$

The errors have magnitudes less than 25%, and are mostly concentrated around T = 1, although for large m, in both SS and RF the peak error tends towards higher values of T. SS is considerably the worst, and this was found to be generally true. In AB and SS, the errors converge towards zero as T increases (rather slowly in the case of SS with large m). For RF with m = 70 the error increases above T=3 and takes a value of about 2%. Large m and large T model very large sets of servers under substantial load, which is an important case. For each algorithm there appears to be a defined "worst" value of T, mostly close to 1.0, but varying with m and N.

It is notable that AB and SS tend to overestimate W (positive error), while RF tends to underestimate it. For different m and N the curves are different, and AB sometimes underestimates as well.

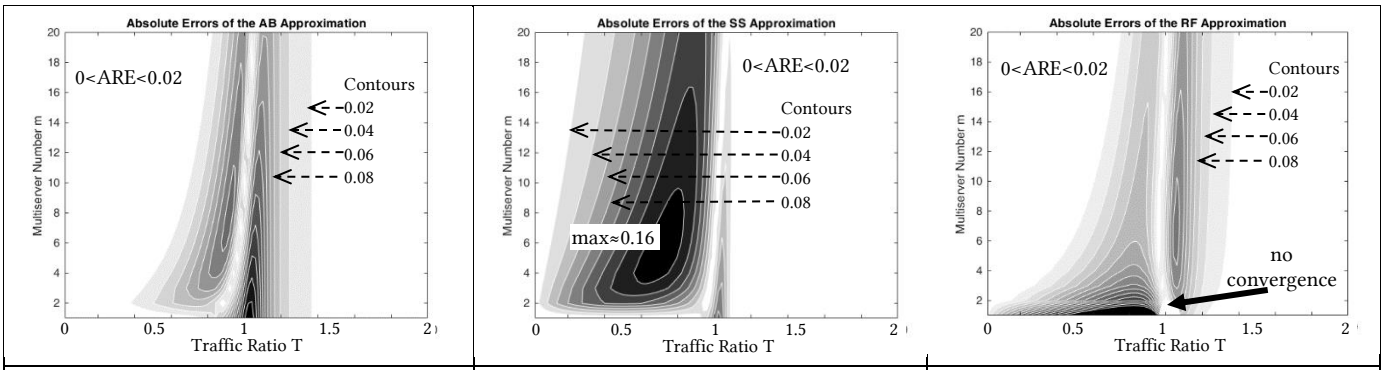
4.1 Absolute Relative Errors (ARE)

To go beyond the error graphs in Figure 3, a contour map of ARE against T and m, for N = 500, is shown in Figure 4 above, for the three approximations. The range of T was restricted below 2 because the errors are quite small for all the approximations for larger T and this customer population.



(a) RE(AB) (b) RE(SS) (c) RE(RF)

Figure 3 Relative Error of Three Approximations against the Traffic Level T, for N = 100



(a) AB (c) SS (d) RF

Figure 4 Contours of the Absolute Relative Errors of the Three Approximations for 1<m<20, 0<T<2, and N = 500

The contours are spaced 0.02 apart. Server saturation is approached for T>1.2. All the approximations show ridges of error near T = 1, corresponding to the positive and negative peaks below and above T = 1 in the curves in Figure 3 above. AB and RF appear to be comparable, although the errors are concentrated differently. SS has a very large black area of large errors, but with a smaller peak value (0.16 vs 0.24 for AB and 0.28 for RF). The largest errors for AB and RF are for single servers, but these are unimportant, since there are better solvers for single servers.

For a quantitative comparison the average and maximum ARE was calculated for 200000 cases covering the parameter ranges m ∈ [2, 50], N ∈ [5, 1000], four ranges of traffic intensity T with 10 values in each range, with $Z = N/(mT)$:

Moderate loads below the saturation point, T ∈ [0, 1],
Mid-range loads near the saturation point, T ∈ [1,3],
Heavy loads which saturate the server, T ∈ [3,10],
Very heavy loads, T ∈ [10,36],

The cases included all combinations of 25 values of m, 200 values of N and 40 values of T. Table 2 shows the average and maximum values for each load range. The iteration in the RF approximation failed to converge in many cases with heavy traffic and the percentage of non-converged cases is shown in the last column. Convergence depends on a relaxation parameter which was set to 0.2; smaller values improve convergence but increase the average run-time.

In Table 1, AB has smaller mean errors than RF everywhere, and smaller than SS in three out of four ranges. In the important range of moderate traffic (row 1) AB is much the best. Near saturation (row 2), the differences between the approximations are smaller and AB is between SS and RF. For maximum error, the smallest occurs for AB in the two heaviest ranges, for SS in the "Moderate" cases, and for RF in the Mid-Range" cases. From Table 1, either AB or SS would be preferred over RF, both for reasons of error and of convergence, with AB being preferred over SS because of the "Moderate" cases.

Table 2. Absolute Relative Errors Over 200000 Cases

		AB	SS	RF	RF non-converged
Moderate 0<T<1	Mean ARE	0.01496	0.05791	0.01866	0
	Max ARE	*0.2020*	*0.1697*	*0.2487*	
Mid-range 1<T<3	Mean ARE	0.01441	0.008969	0.02556	20542 41%
	Max ARE	*0.2169*	*0.1919*	*0.1532*	
Heavy 3<T<10	Mean ARE	0.001019	0.002056	0.003450	36682 73%
	Max ARE	*0.06617*	*0.1926*	*0.08451*	
Very Heavy 10<T<36	Mean ARE	0.0002134	0.001604	0.002931	39531 79%
	Max ARE	*0.02289*	*0.1932*	*0.09561*	

For additional insight into the error distributions, Figure 5 shows histograms of the errors for the first three ranges of T. In the histograms the first cell (for relative errors less than 1%) is omitted because its count is so large, but its count is reported.

Many results from all the approximations have less than 1% error. SS and RF are relatively poor in moderate traffic. RF shows some large errors in light traffic. Overall, AB and SS both appear to have acceptable errors.

4.2 Worst-Case Analysis

It is useful to know how the maximum errors are related to the parameter values. The maximum ARE, denoted as $E^*(m,N)$ were mapped over values of (m,N) with m in (1, 20) and N in (5, 500). For each configuration of m, N, there is a "worst" traffic level $T^*(m,N)$ (giving the largest value of $E^*(m,N)$). T^* was found by a numerical search over T and is usually near to 1.0. Figure 6 displays $E^*(m,N)$ as a contour map, and Figure 7 displays $T^*(m,N)$.

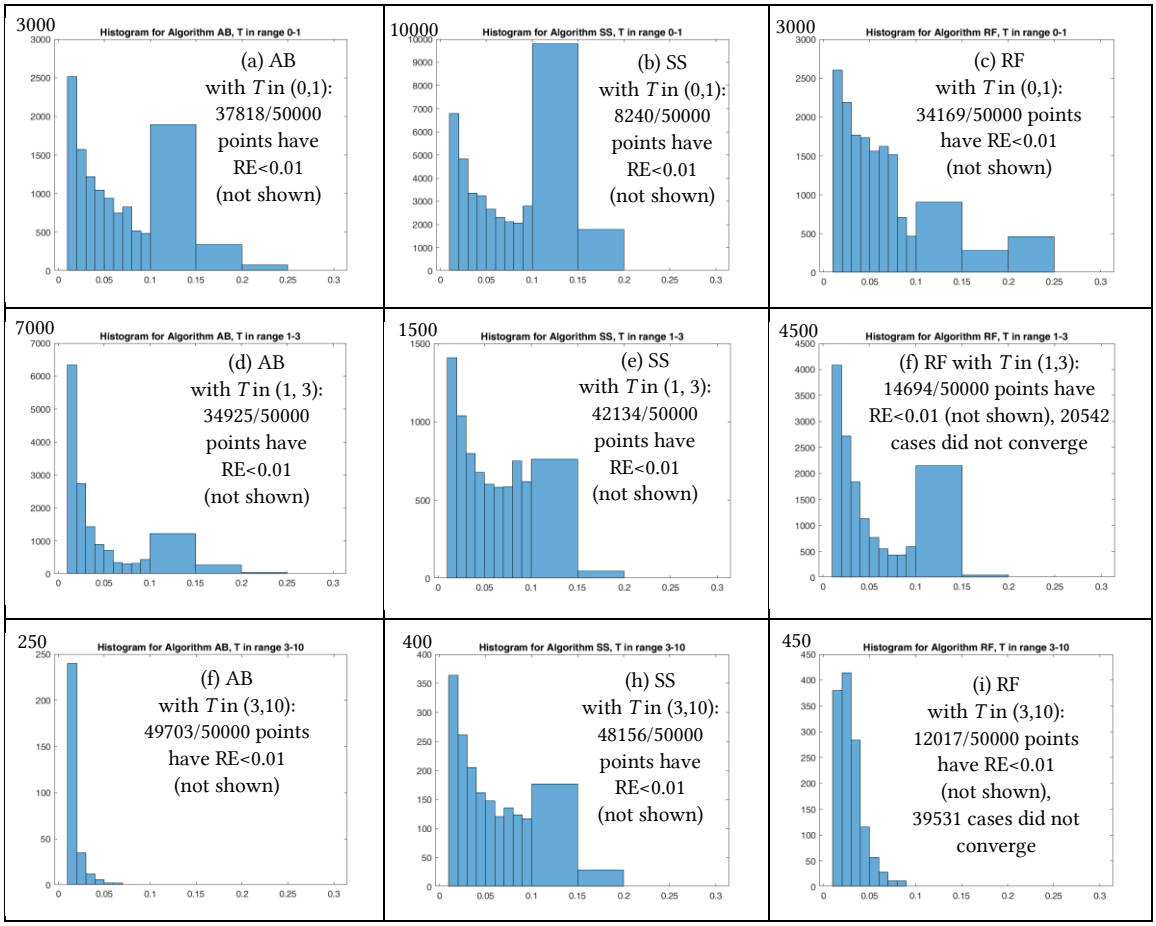
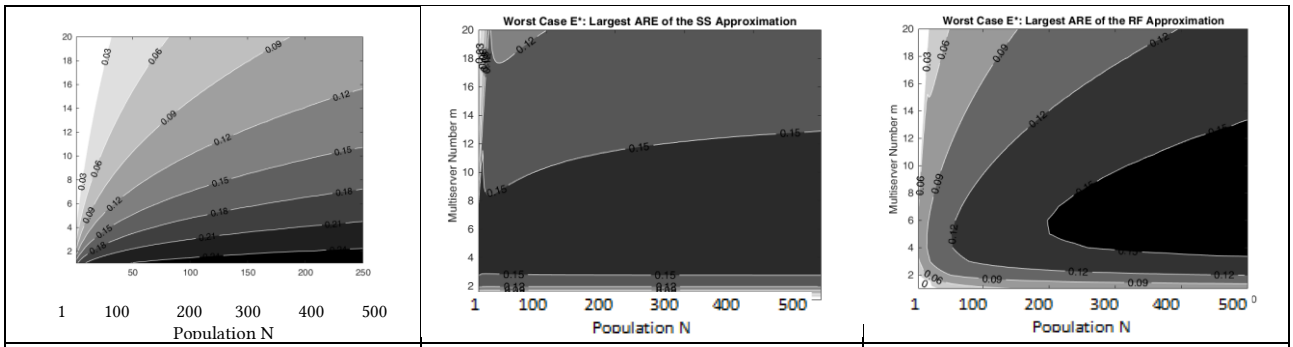


Figure 5 Histograms of the Absolute Relative Errors (AREs) at Moderate, Mid-Range and Heavy Loads

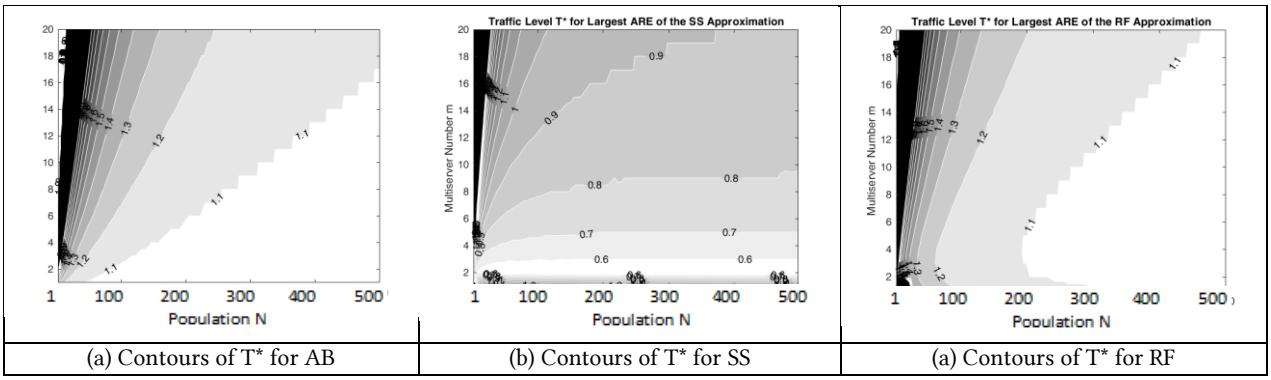


(a) Max Error E* for AB	(a) Max Error E* for SS	(a) Max Error E* for RF

Figure 6. Worst Case Analysis: Contour Plots of the Maximum Errors $E^*(m, N)$

The maximum ARE follows different patterns for the three approximations:

- For AB the worst-case ARE is less than 0.1 over almost half the space (the upper left half of the chart) The largest values (around 0.24) occur in the bottom right corner. These are cases with many customers, just one or two servers. Figure 7(a) shows values of $T^* \approx 1$ in that corner, so think times are long relative to service times, to give T near to 1.
- For SS the worst-case ARE is greater than 0.1 for all m, N but its highest value is smaller, less than 0.16. There is a band across the chart for cases with 3-12 servers, giving worst-case errors between 0.15 and 0.16. In Figure 7(b) the worst-case traffic levels in this area are between 0.5 and 0.9, indicating moderately heavy traffic.



(a) Contours of T* for AB | (b) Contours of T* for SS | (a) Contours of T* for RF

Figure 7. Worst Case Analysis: Contour Plots of the Worst-Case Traffic Metric $T^*(m, N)$

- For RF the worst-case ARE is above 0.15 in a large region on the right, with worst-case traffic levels (in figure 7(c) just above 1.0, thus right at the knee of the response time curve. The error is relatively small along the X and Y axes (that is, for small m or small N)

5. Example Software Performance Application

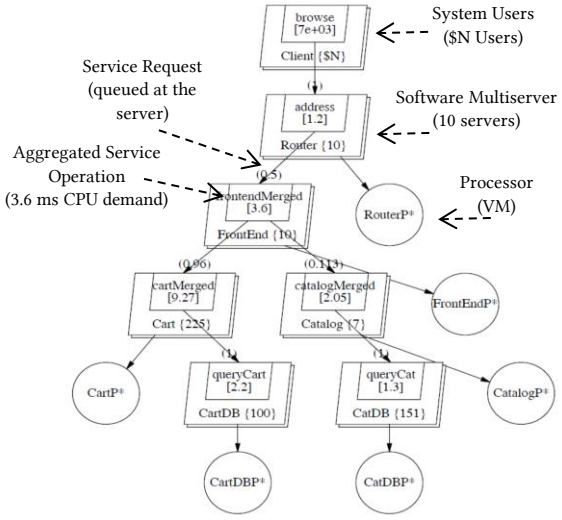


Figure 8 Layered Queueing Model of a Version of the SockShop Microservices Demonstration Software

Figure 8 shows a layered queueing model of the Sockshop microservices demonstration software taken from [8], with the multiclass servers aggregated to a single class. The solution was found using different calculation options for the multiserver waiting. "Sim" applied the LQSIM tool described in [6], so as to obtain 95% confidence intervals of ±1%. Analytic approximations used in LQNS [6] were "Reiser" which applied exact MVA to the layer submodels, and AB and RF as described above; SS was not included in these experiments.

Table 3 compares the solution times obtained. The simulation time varied with population. AB is the fastest approximation and also the most robust (along with exact MVA), since with AB the LQNS solver never failed to converge. These solution times are all quite short, however the difference is still significant when a model must be solved many times in a search process. Also, the multiplicity of 10 at the Router component limits the populations which contend for the lower layer servers to 10 and this limits the impact of the greater complexity.

Table 3 Time for LQNS to solve the SockShop model

Option	Simulation (1500 clients)	Reiser (MVA)	Conway	RF	AB
Time (ms)	1168000	47	39	15	10
Non-convergence in 25 cases		0	1	6	0

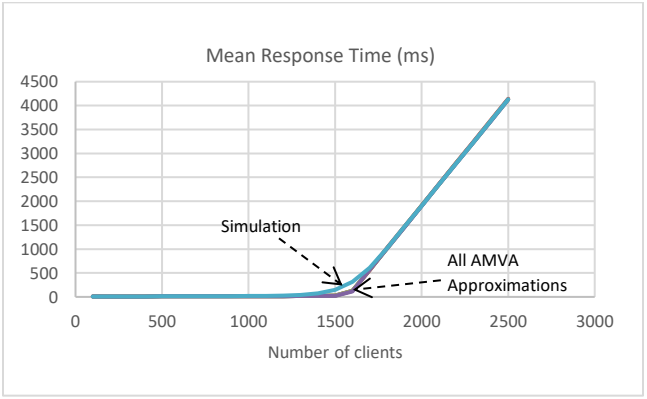


Figure 9 Response Times for Varying Populations

Figure 9 shows the response time solution obtained as the population is varied up to 2500. All the approximations gave the same result, with differences much less than 1% for all populations. However they all gave low estimates (compared to simulation) at the corner in the neighborhood of the onset of

saturation; this difference is a result of the LQNS solution strategy and may be due to LQNS ignoring service time variance in all its multiserver approximations (the coefficients of variation of the service time of the Router and the FrontEnd were estimated at 2.97 and 1.78 respectively).

6. Related Work

Related work on FIFO multiservers with closed workloads begins with separable models with exact solutions by Mean Value Analysis (MVA) [10]. The high complexity of MVA, due to the need for marginal distributions of queue length, has led to AMVA approximations to multiservers, such as Linearizer [4], improved in SCAT [2] and in [1] by using an ad hoc distribution shape based on the mean. QD-MVA [3] takes a different approach and reformulates the solution as an optimization problem.

For multiclass queues de Souza e Silva and Muntz approximated the queue state by a multinomial distribution [13] based on the mean values, to get the class waiting times, and combined these with AMVA methods. The single class version of the multinomial is the binomial distribution used in AB. Conway [5] adapted their method to AMVA, as it is used in LQNS. Rolia and Sevcik [11] described a much simpler AMVA calculation (RF in this work) that approximates the probability that all servers are busy, by the product of the individual probabilities (effectively assuming the servers are independently busy). Franks made a small improvement described in [7].

There are many other works on multiserver approximations that address different models, such as open arrivals, general service times and multiple classes.

7. Conclusions

The AB approximation has been shown to be a useful addition to numerical methods for multiserver queue waiting. It is a little more accurate, and a little faster than the best alternative which is RF, and is more robust in terms of convergence. In defense of RF it should be mentioned that using a smaller relaxation coefficient a in Eq (6) improves its convergence, at the expense of longer computation times. In other experiments not reported here we have been forced to make a as small as 0.02 to obtain convergence.

The example in Section 4 shows that the faster queue solution by AB translates into faster overall solutions.

The in-depth investigation of AB, SS and RF show that SS is surprisingly good given its simplicity. It has substantial errors for many cases but its maximum errors are the least of the three, to it has a safety factor.

The results are reported against the traffic ratio T, and the relationship of T to the server utilization is of interest. Utilization for a given m is close to T or less than T; it increases with T and reaches saturation between T=1 and T=2, in most cases, depending on the balance of m, N and Z. For larger T the server is saturated but the error behaviour still depends on T, indicating that it varies according to the balance of N and Z. Larger T gives behaviour more like an open queue.

The extension of AB and SS to multiple classes will be reported elsewhere.

ACKNOWLEDGMENTS

This research was funded by the Natural Sciences and Engineering Research Council of Canada, grant 06274-2016.

REFERENCES

[1] I.F. Akyildiz, G. Bolch, "Mean Value Analysis Approximation for Multiple Server Queueing Networks", Performance Evaluation, v 8, pp 77-91, 1988.

[2] G. Bolch, S. Greiner, H. Meer, K. Trivedi, Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications, Second Edition, April 2006.

[3] G. Casale, J. Perez, W. Wang, "QD-AMVA: evaluating systems with queue-dependent service requirements", Performance Evaluation, v 91, pp 80-98, 2015.

[4] K.M. Chandy, Doug Neuse, "Linearizer: A heuristic algorithm for queueing network models of computing systems", Comm. of the ACM, v 25 n 2, pp 126-134, 1982.

[5] A.E. Conway, "Fast approximate solution of queueing networks with multi-server chain-dependent FCFS queues", in *Modeling Techniques and Tools for Computer Performance Evaluation*, pp 385-396, Plenum, New York, 1989.

[6] G. Franks et al, "Layered Queueing Network Solver and Simulator User Manual", http://www.sce.carleton.ca/rads/lqns/LQNSUserMan-jan13.pdf.

[7] G. Franks, Performance analysis of distributed server systems, PhD thesis, Carleton University, 1999.

[8] A.I. Gias, G. Casale, M. Woodside, "ATOM: Model-Driven Autoscaling for Microservices", ICDCS 2019.

[9] E. Lazowska, J. Zahorjan, G.S. Graham, K. Sevcik, Quantitative System Performance, Wiley, 1984.

[10] M. Reiser, S.S Lavenberg, "Mean-value analysis of closed multichain queueing networks", J. of ACM, v27 n 2, pp 312–322, 1980.

[11] J.A. Rolia, K.C. Sevcik, The Method of Layers, IEEE Trans Software Engineering, v 21 pp 689 – 700, 2015.

[12] P.J. Schweitzer, "Approximate analysis of multiclass closed networks of queues", Proc. Int. Conf. on Stochastic Control and Optimization, pp 25-29, Amsterdam, 1979.

[13] E. de Souza e Silva, R.R. Muntz, "Approximate solutions for a class of non-product form queueing network models", Performance Evaluation, v 7, pp 221-242, 1987.

[14] Q. Zhang, Q. Zhu, M.F. Zhani, R. Boutaba, J.L. Hellerstein, "Dynamic Service Placement in Geographically Distributed Clouds", IEEE J. on Selected Areas in Communications, v 31, n 10, Oct. 2013.

[15] Q. Zhang, Y. Xiao, F. Liu, J.C.S. Lui, J. Guo, T. Wang, "Joint Optimization of Chain Placement and Request Scheduling for Network Function Virtualization", Int. Conf. Distrib. Computing Systems (ICDCS), pp 731-741, 2017.