Mean Value Technique for Closed Fork-Join Networks

Elizabeth Varki
Department of Computer Science
University of New Hampshire
email: varki@cs.unh.edu

Abstract

A simple technique for computing mean performance measures of closed single-class fork-join networks with exponential service time distribution is given here. This technique is similar to the mean value analysis technique for closed product-form networks and iterates on the number of customers in the network. Mean performance measures like the mean response times, queue lengths, and throughput of closed fork-join networks can be computed recursively without calculating the steady-state distribution of the network. The technique is based on the mean value equation for fork-join networks which relates the response time of a network to the mean service times at the service centers and the mean queue length of the system with one customer less. Unlike product-form networks, the mean value equation for fork-join networks is an approximation and the technique computes lower performance bound values for the fork-join network. However, it is a good approximation since the mean value equation is derived from an equation that exactly relates the response time of parallel systems to the degree of parallelism and the mean arrival queue length. Using simulation, it is shown that the relative error in the approximation is less than 5% in most cases. The error does not increase with each iteration.

1 Introduction

A popular method of improving performance of computer and storage systems is by distributing the work across multiple systems in parallel. The parallel job has a simple fork-join structure. A parallel job is divided (*forked*) into a number of tasks which are processed concurrently on the different units of the parallel system. On completing execution, a task waits at the *join* point for its sibling tasks to complete execution. A job leaves the system once all its tasks complete execution (i.e., tasks of a job are joined before departing the system). While the concept is simple, the synchronization constraints introduced by forking and joining and the resulting complexity of the system design makes it difficult to analyze these systems and answer questions about their current and future capability to perform their functions.

Several papers study fork-join queueing networks and propose tools for analyzing their performance. Due to the difficulty of analyzing fork-join models exactly, many studies on fork-join queues concentrate on approximation techniques [1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 16, 17, 18] The approximation techniques are largely complex and do not scale well. Some of these techniques are proposed without formal proofs. This paper gives a simple approximation technique for computing mean performance measures like the mean response times, mean queue lengths, mean throughput, and utilization of closed fork-join networks. The approximation technique is similar to the Mean Value Analysis (MVA) for closed product-form networks developed by Reiser and Lavenberg [14]. MVA is an iterative technique which iterates on the number of customers. The analysis is based on the arrival equation for product-form networks which maps the relationship between the mean number of waiting customers and the mean queue length of a system with one customer less. This arrival equation and Little's Law are used to compute performance measures of the network without having to compute its steady state distribution. While the mean-value equation for product-form networks is exact, it is shown to be an approximation in the case of fork-join networks. However, the simulation results indicate that the error margins are small and do not grow with each iteration. Another key contribution of this paper is the insight it provides to the behavior of parallel systems. While the mean-value equation is shown to be an approximation, it is derived from an equation which relates the exact mean response time of parallel systems to the degree of parallelism and the mean arrival queue length.

The remainder of this paper is organized as follows. Section 2 introduces the fork-join model. The mean value technique for fork-join networks is given in sections 3, 4, 5, and 6. Section 7 states the conclusions and suggests possible future work.

2 The Fork-Join Queueing Network Model

Closed fork-join queueing networks (FJQNs) of the type shown in Figure 1 are studied in this paper. A fixed number of identical *jobs* circulate in the network. The network consists of one or more interconnected *subsystems*, where each subsystem consists of one or more independent queueing systems. Each queueing system consists of a single service center and an infinite capacity queue. All servers have a first-come-first-served queueing discipline and all demands at the service centers have a negative exponential distribution. The subsystems are broadly classified into two types, namely, *parallel* and *serial* subsystems. A parallel subsystem consists of $K > 1$ identical queueing systems. Upon arrival at a parallel subsystem, a job instantaneously forks into K independent and identical *tasks*, where task k, $k = 1, 2, \cdots, K$, is assigned to the kth queueing system. On completing execution at the kth service center, the task waits at the join point for its sibling

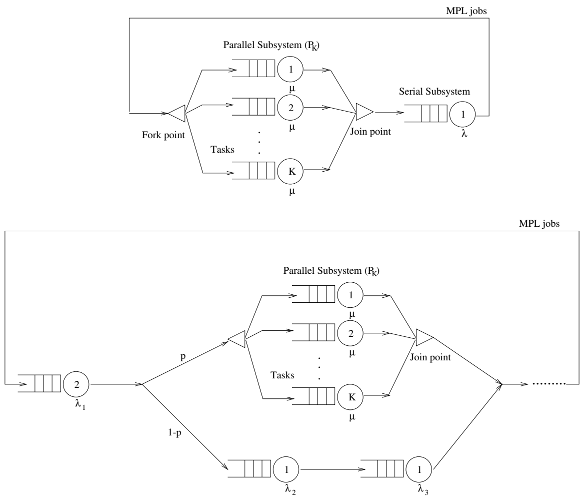

Figure 1: Fork-Join Queueing Network Model

tasks to complete execution. A job leaves the subsystem as soon as all its tasks complete execution and arrive at the join point. Such a subsystem is also referred to as a *K-sibling* fork-join subsystem (P_K). The response time of a job in a parallel subsystem is the time taken from arrival instant until all tasks of the job complete execution and the job departs the subsystem. A serial subsystem consists of a single queueing station and jobs are not split into tasks. A serial subsystem can be considered as a special case of a parallel subsystem with $K = 1$ (P_1).

The subsystems in a FJQN are arbitrarily numbered from 1 to N. The notation used in this paper is:

- N: Number of subsystems in the FJQN.
- m, MPL: Multiprogramming level of the network (i.e., number of jobs in the network).
- μ_i: Service rate of a service station in $subsystem_i$.
- K_i: Number of tasks a job is split into upon arrival at $subsystem_i$. ($K_i = 1$ for a serial subsystem.)
- V_i: Average number of visits per job to $subsystem_i$.
- $R_i(m)$: Mean response time at $subsystem_i$ when the multiprogramming level is m.

- $X_i(m)$: Mean throughput at $subsystem_i$ when the multiprogramming level is m.
- $A_i(m)$: Mean number of jobs seen by a job arriving at $subsystem_i$ when the multiprogramming level is m.
- $Q_i(m)$: Mean number of jobs in $subsystem_i$ when the multiprogramming level is m.
- $CT_{NET_i}(m)$: Mean cycle time spent in a closed network. This equals the time it takes to cycle through the network and return to $subsystem_i$.
- H_k: The kth harmonic number defined as $\sum_{i=1}^{k} \frac{1}{i}$.

3 Overview of the Mean Value Technique

The mean value (MVA) technique for FJQNs is based on the mean value technique for product-form networks developed by Reiser and Lavenberg [14]. A brief explanation of the mean value technique for product-form networks is given here. The mean value technique is based on the *arrival* theorem which states that the number of jobs "seen" by a job upon arrival at a subsystem is equal to the mean queue length at the subsystem computed when the network has one less job [7, 15]. The response time of the arriving job is equal to the sum of the arriving job's service time plus the time required to service the jobs seen ahead. From the arrival theorem, it follows that

$$R_i(m) = \frac{1}{\mu_i}[1 + Q_i(m-1)] \qquad (1)$$

From Little's Law, the subsystem throughput and queue length are given by

$$\begin{aligned} X_i(m) &= \frac{m}{\sum_{n=1}^{N}[R_n(m)V_n/V_i]} = \frac{m}{CT_{NET_i}(m)} \quad \text{and} \\ Q_i(m) &= X_i(m) \times R_i(m) \end{aligned}$$

The MVA technique involves iteratively solving these three equations for $m = 1, 2, \cdots, MPL$ where the initialization of the iteration is $Q_i(0) = 0$ for all subsystems.

The MVA technique is based on Equation 1 which relates the mean response time of a subsystem to its mean service time and its mean queue length computed when the network has one less customer. The other two equations follow directly from Little's Law. Due to its significance to the MVA technique, Equation 1 is referred to as the *mean-value* equation. The key to developing the MVA technique for FJQNs is to find an equivalent mean-value equation that holds for these networks. However, finding such an equation is complicated since fork-join networks do not have product-form properties and the arrival theorem need not hold. In addition, unlike serial subsystems, the mean service time of a job arriving at a parallel subsystem depends on the state seen at arrival instant. So, the mean service time of the parallel subsystem varies as the multiprogramming level of the network changes. (This can be confirmed by solving small systems and using Little's Law to compute mean service times.) Thus, the computation of both the mean service time of a parallel subsystem and the mean number of waiting jobs seen in the parallel subsystem by an arriving job is non-trivial. The next section addresses this issue.

4 Mean Response Time Analysis of Parallel Subsystems

This section summarizes the work in [19]. The response time of a job in a parallel subsystem is the time taken from arrival instant at the fork point until all tasks of the job complete execution and the job departs the join point. In order to compute the mean response time of a closed network (without using Little's Law), it is necessary to know the mean service time of a job arriving to a parallel subsystem and the mean number of jobs seen ahead of this arriving job. When there is just one job in the closed network, the mean service time of this job at a K-sibling parallel subsystem is equal to $\frac{H_K}{\mu_i}$, the mean value of the K^{th} order statistic of an exponential random variable. When there are $m > 1$ jobs in the network, the service time of a job arriving at the parallel subsystem is less than $\frac{H_K}{\mu_i}$ since the execution time of some of the K tasks of this job will overlap with the execution time of tasks of jobs ahead of it.

The response time of a K-sibling parallel subsystem can also be analyzed by viewing the subsystem from the perspective of the K phases of a job's response time in the parallel subsystem as explained here. During $phase_k$ $(k = K, \cdots, 1)$ of a job's response time, k tasks of the job are in the queues of the service centers and $K - k$ tasks are waiting at the join point. A phase ends with the movement of one of the executing tasks to the join point. The time spent completing each phase of a job's response time in the parallel subsystem can be viewed as the time spent getting service at K serial subsystems $S_K, S_{K-1}, \cdots, S_1$. The mean service rates at these K service stations can be directly computed from the underlying Markov diagram of the K-sibling parallel subsystem. The mean service rate at service center S_k $(k = K, \cdots, 2)$ varies according to the number of customers at service centers S_i, $i = k-1, k-2, \cdots, 1$ as follows: if there is at least one customer in service center S_{k-1}, the mean service rate at S_k equals μ, else if there is at least one customer in service center S_{k-2}, the mean service rate at S_k equals 2μ, else if there is at least one customer in service center S_{k-3}, the mean service rate at S_k equals 3μ, $\cdots$, else if there is at least one customer in service center S_1, the mean service rate at S_k equals $(k-1)\mu$, else (if there are no customers in $S_{k-1}, S_{k-2}, \cdots, S_1$) the mean service rate at S_k equals $k\mu$. The service rate at server S_1 is equal to μ. Thus, service rates at all servers S_k, $k = K, \cdots, 2$ are dependent on the states at the service stations S_i $(i = k-1, \cdots, 1)$. Only server S_1 is state independent. The response time of a job in the state dependent model is equal to the sum of the response times at the K service centers. By construction, this state-dependent model is equivalent to a K-sibling parallel subsystem and have identical Markov diagrams. Thus, the response time of a job in a parallel subsystem is equal to the response time of the job in its equivalent state-dependent model.

The equivalence of the state-dependent model to a parallel subsystem is used to prove that for a K-sibling parallel subsystem, the response time is given by:

$$R_{P_K}(m) = \frac{1}{\mu_i}\left[H_K + AQ_K(m)\right]$$

where,
$AQ_K = \frac{1}{K}(A_{S_K S_K} + A_{S_K S_{K-1}} + \cdots + A_{S_K S_1}) +$

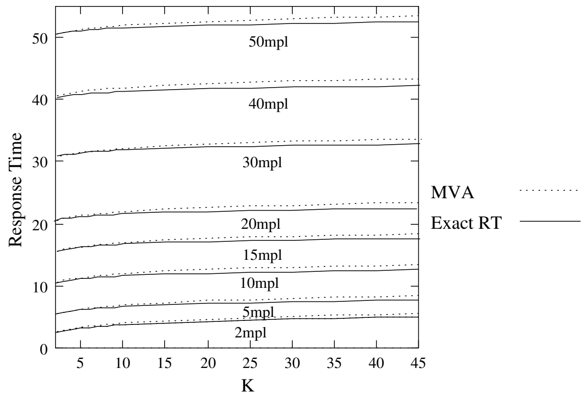

Figure 2: Response Times (MVA and Exact) for fixed MPL as K varies

$\frac{1}{K}(A_{S_K S_K} + A_{S_{K-1} S_{K-1}} + A_{S_{K-1} S_{K-2}} + \cdots + A_{S_{K-1} S_1}) + \cdots +$
$\frac{1}{K}(A_{S_K S_K} + A_{S_{K-1} S_{K-1}} + \cdots A_{S_3 S_3} + A_{S_2 S_2} + A_{S_2 S_1}) +$
$\frac{1}{K}(A_{S_K S_K} + A_{S_{K-1} S_{K-1}} + \cdots + A_{S_2 S_2} + A_{S_1 S_1})$

The notation $A_{S_i S_k}$ represents the mean arrival queue length at subsystem S_k seen by a job just prior to arrival at subsystem S_i. It is further shown that $AQ_K(m) \leq A_{P_K}(m)$, where A_{P_K} is the mean number of jobs seen at the parallel subsystem by a job arriving to the parallel subsystem. This gives:

$$R_{P_K}(m) \leq \frac{1}{\mu_i}\left[H_K + A_{P_K}(m)\right] \qquad (2)$$

5 Mean Value Equation for FJQNs with a Single Parallel Subsystem

In case of fork-join networks that just consists of a single subsystem, namely, a K-sibling parallel subsystem, Equation 2 can be simplified. For such systems, $A_{P_K}(m)$, the mean number of jobs seen in the parallel subsystem by a job arriving to the parallel subsystem, is just equal to the multiprogramming level of the network when there is one less job in the network. Hence, the response time of the parallel subsystem P_K is given by:

$$R_{P_K}(m) \leq \frac{1}{\mu}\left[H_K + (m-1)\right]$$

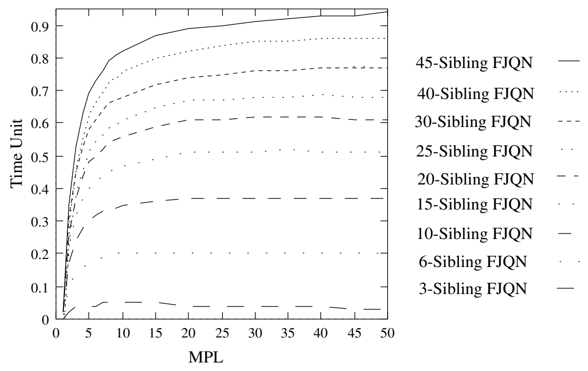

Figure 3: Relative Error in MVA Bound

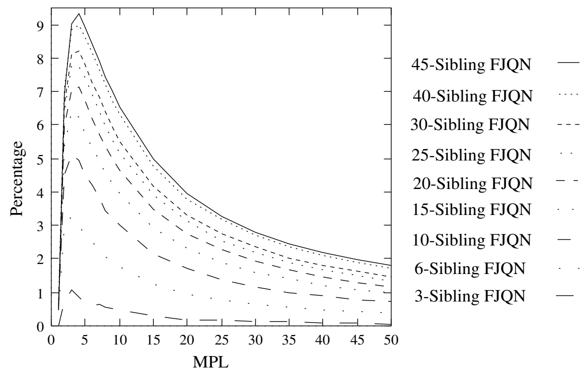

Figure 4: Percentage Relative Error in MVA Bound

In [19], a stronger result is shown. It is proved that

$$
\begin{aligned}
R_{P_K}(m) &= \frac{1}{\mu}\left[H_2 + (m-1)\right] when\ K = 2 \\
&< \frac{1}{\mu}\left[H_K + (m-1)\right] when\ K > 2.
\end{aligned}
$$

While the formal proof is not given here, some of the simulation runs that validate the result are given. The response times are plotted for values of K ranging from $2, \cdots, 45$ and m ranging from $1, \cdots, 50$. The mean service time is set at 1.00 time unit. Figure 2 plots the upper and exact response times for fixed values of m as K varies from 2 to 45. The upper bound and exact response time values are very close and grow at the same rate. The response time curves are quite flat showing that for a given m, R_{P_K} grows at a slower rate than H_K. The relative and percentage error in the response time bounds are plotted in Figures 3 and 4, respectively. The relative error in the bound increases till $m \approx 5$ and then becomes constant. The maximum percentage error in the upper bound is around 9 percent and occurs when $K = 45$ and $m = 5$. However, for the majority of values, the percentage error is less than 3 percent. For fixed K, the percentage relative error increases sharply till $m \approx 5$ and then decreases as m increases. Figure 3 shows that the relative error is a non-decreasing function of m. This implies that for values of m at which percentage relative error is at its greatest ($m \approx 5$), the distance between the bound and the exact response time is lesser than for values of $m > 5$ (when percentage error is less). Thus, the bounds are tight even at points where the percentage relative error is the greatest.

6 Mean Value Technique for FJQNs

The MVA technique for fork-join networks is based on Equation 2, namely, $R_{P_K}(m) \leq \frac{1}{\mu}\left[H_K + A_{P_K}(m)\right]$, and $R_{P_1}(m) = \frac{1}{\mu}\left[H_1 + A_{P_1}(m)\right]$. This gives

$$
CT_{FJQN_k}(m) \leq \sum_{i=1}^{N} \frac{1}{\mu_i}\left[H_{K_i} + A_i(m)\right]\frac{V_i}{V_k} \qquad (3)
$$

where $1 \leq k \leq N$. In product-form networks, the mean number of jobs seen upon arrival to a subsystem is equal to the mean queue length when the number of jobs in the network is one less. For fork-join networks, the relationship between the arrival instant distribution and the steady state distribution of the network with one less customer is not known. However, equations 2 and 3 imply that

$$R_i(m) \approx \frac{1}{\mu_i}\left[H_{K_i} + Q_i(m-1)\right] \qquad (4)$$

Equation 4 is referred to as the mean value equation for FJQNs. Therefore, the complete MVA technique involves iteratively solving the equations

$$\begin{aligned} CT_{FJQN_i}(m) &= \sum_{n=1}^{N}[R_n(m)V_n/V_i] \\ X_i(m) &= \frac{m}{CT_{FJQN_i}} \\ Q_i(m) &= X_i(m) \times R_i(m) \end{aligned}$$

for $m = 1, 2, \cdots, MPL$ where the initialization of the iteration is $Q_i(0) = 0$ for all subsystems. Note that the performance measures computed by the MVA technique are approximations since they are based on Equation 4 which is an approximation.

The closeness of the MVA generated response time values to the exact response time values is investigated by running simulations. The cycle times are plotted for values of MPL ranging from 1 to 50. Figures 5 and 6 plot the cycle times obtained via simulation and the MVA technique for four different FJQNs. (In all these cases, the visit counts to all subsystems within a network are set to be equal.) Graph (a) of Figure 5 compares the cycle times for a 2-subsystem FJQN where $subsystem_1$ is serial and $subsystem_2$ is a 2-sibling parallel subsystem. Graph (b) of Figure 5 compares the cycle times for a 2-subsystem FJQN where $subsystem_1$ is serial and $subsystem_2$ is a 20-sibling parallel subsystem. Graph (c) of Figure 6 compares the cycle times for a 3-subsystem FJQN where $subsystem_1$ is serial, $subsystem_2$ is a 2-sibling parallel subsystem, and $subsystem_3$ is a 3-sibling parallel subsystem. Graph (d) of Figure 6 compares the cycle times for a 2-subsystem FJQN where $subsystem_1$ is a 2-sibling

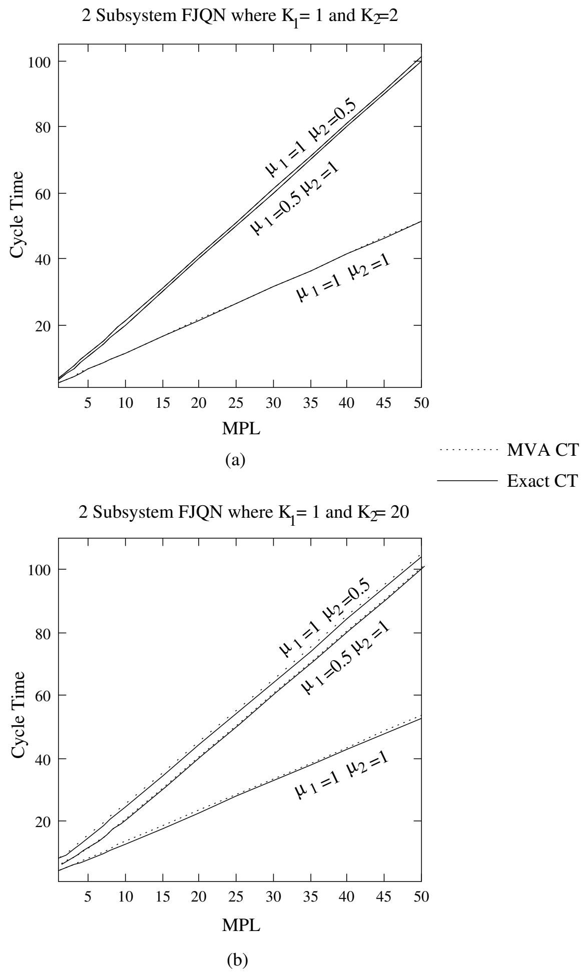

Figure 5: Cycle Times (Exact and MVA)

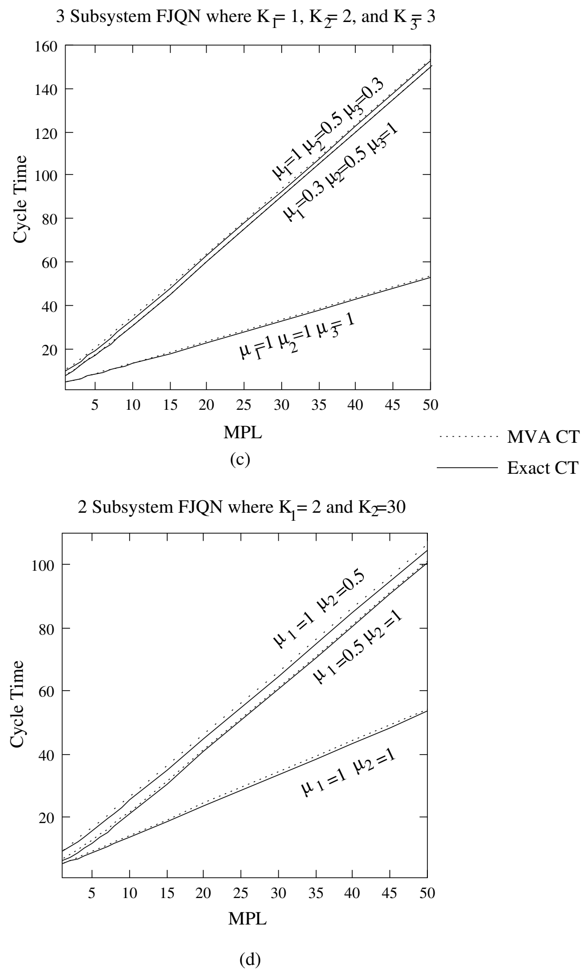

Figure 6: Cycle Times (Exact and MVA)

	$\frac{1}{\mu_1}=3,\ \frac{1}{\mu_2}=2,\ \frac{1}{\mu_3}=1$		
MPL	Exact	MVA	%Error
1	7.83	7.83	0.00
2	9.95	9.98	0.30
3	12.19	12.25	0.49
4	14.54	14.63	0.62
5	17.01	17.11	0.59
6	19.57	19.67	0.51
7	22.22	22.32	0.45
8	24.93	25.03	0.40
9	27.71	27.79	0.29
10	30.54	30.61	0.23
15	45.12	45.15	0.07
20	60.03	60.03	0.00
25	75.01	75.01	0.00
30	90.01	90.01	0.00
35	105.02	105.02	0.00
40	120.02	120.02	0.00
45	135.02	135.02	0.00
50	150.02	150.02	0.00

Table 1: Cycle Times for a 3 Subsystem FJQN where $K_1 = 1$, $K_2 = 2$, and $K_3 = 3$

	$\frac{1}{\mu_1} = 1$, $\frac{1}{\mu_2} = 1$, $\frac{1}{\mu_3} = 1$		
MPL	Exact	MVA	%Error
1	4.33	4.33	0.00
2	5.29	5.33	0.76
3	6.23	6.33	1.61
4	7.18	7.33	2.09
5	8.12	8.33	2.59
6	9.07	9.33	2.87
7	10.03	10.33	2.99
8	10.99	11.33	3.09
9	11.96	12.33	3.09
10	12.94	13.33	3.01
15	17.86	18.33	2.63
20	22.84	23.33	2.15
25	27.82	28.33	1.83
30	32.82	33.33	1.55
35	37.81	38.33	1.38
40	42.81	43.33	1.21
45	47.81	48.33	1.09
50	52.81	53.33	0.98

Table 2: Cycle Times for a 3 Subsystem FJQN where $K_1 = 1$, $K_2 = 2$, and $K_3 = 3$

	$\frac{1}{\mu_1} = 1$, $\frac{1}{\mu_2} = 2$, $\frac{1}{\mu_3} = 3$		
MPL	Exact	MVA	%Error
1	9.50	9.50	0.00
2	11.84	11.97	1.10
3	14.22	14.53	2.18
4	16.71	17.14	2.57
5	19.30	19.81	2.64
6	21.98	22.53	2.50
7	24.73	25.30	2.30
8	27.54	28.09	2.00
9	30.40	30.92	1.71
10	33.28	33.77	1.47
15	47.96	48.29	0.69
20	62.83	63.06	0.37
25	77.74	77.93	0.24
30	92.68	92.85	0.18
35	107.66	107.79	0.12
40	122.66	122.76	0.08
45	137.64	137.73	0.07
50	152.62	152.71	0.06

Table 3: Cycle Times for a 3 Subsystem FJQN where $K_1 = 1$, $K_2 = 2$, and $K_3 = 3$

parallel subsystem and $subsystem_2$ is a 30-sibling parallel subsystem. Each graph plots three sets of cycle time values for the same network by varying the service time values for the subsystems. Thus, in each graph, there are a total of six cycle time curves, two (exact and MVA) for each set. However, it appears that there are only 3 cycle time curves in each graph since the actual and MVA generated cycle time values are so close that they overlap closely and it is difficult to distinguish them.

Tables 1, 2, and 3 compare the cycle times obtained via simulation and the MVA technique for the network in graph (c) of Figure 6. The tables plot three sets of cycle time values for the same network by varying the service time values for the subsystems. The tables also show that the MVA bounds are very close to the cycle time values generated by simulation. The percentage error increases from $MPL = 2$ till $MPL \approx 5$ and then decreases. The key point to note is that the percentage error does not increase with each iteration. The error increases with increasing values of K and networks where the service rate at the parallel subsystem is less than the serial subsystem. The maximum percentage error is 7% and occurs at $MPL = 5$ for the 2 subsystem FJQN where $subsystem_1$ is a 2-sibling parallel subsystem with service rate $\mu_i = 1$ and $subsystem_2$ is a 30-sibling parallel subsystem with service rate $\mu_2 = 0.5$ (Graph (d)). For the networks analyzed with MPL ranging from 1 to 50, 95% of the relative error in the approximation is less than 5%. In general, the percentage error increases as the value of K_i increases and decreases as MPL increases.

The simulation results indicate that the MVA technique gives a lower performance bound for fork-join networks. Thus,

$$CT_{FJQN_k}(m) \leq \sum_{i=1}^{N} \frac{1}{\mu_i} \left[H_{K_i} + Q_i(m-1)\right] \frac{V_i}{V_k} \quad (5)$$

and

$$X_i(MPL) \geq \frac{MPL}{CT_{FJQN_k}(MPL)}$$

The validity of Equation 5 is formally proved for fork-join networks with just a single parallel subsystem and for balanced fork-join networks [19]. For more general fork-join networks a formal proof is unavailable.

7 Conclusions and Future Work

The MVA technique for FJQNs is an iterative technique that iterates on the number of customers in the network. It is based on the mean-value equation which relates the mean response time of a subsystem in a FJQN to the mean service time at a server in the subsystem and the queue length of the subsystem with one less job in the network. The mean value equation is an approximation in the case of fork-join networks. The simulation results indicate that the performance measures computed using the MVA technique are a lower performance bound for the network. A formal proof showing that this technique computes a lower performance bound is only available for specific classes of the FJQN [19]. The MVA technique is a good approximation and the relative error in the approximation for a given network is less than 5% for the majority of the cases considered. For multiprogramming levels approximately greater than 5, the percentage error decreases with each iteration.

Another contribution of this paper is the insight it provides to the behavior of parallel systems. The mean-value equation shows how the response time of a parallel system varies as the number of customers and the degree of parallelism varies. For a given multiprogramming level, the response time of a parallel subsystem P_K grows at a slower rate than H_K, as K varies.

The work here only considers single-class FJQNs, the exponential service time distribution, and the first-come-first-served scheduling discipline. The work can be extended to multi-class networks with other service time distributions and scheduling disciplines. Also, networks with more general parallel subsystems can be considered.

Acknowledgments

I thank Larry Dowdy, Mark Squillante, and anonymous referees for their comments which have greatly improved this paper.

References

[1] Almeida, V.A.F., Dowdy, L.W., "A reduction technique for solving queueing network models of programs with internal concurrency", Proceedings of the 3^{rd} International Conference on Supercomputing, Boston, May 1988.

[2] Baccelli, F., Massey, W. A., Towsley, D. "Acyclic fork-join queueing networks", Journal of the ACM, 36, 3, July 1989, pp. 615 – 642.

[3] Baccelli, F., Liu, Z. "On the execution of parallel programs on multiprocessor systems – A queueing theory approach", Journal of the ACM, 37, 2, April 1990, pp. 373 – 414.

[4] Balsamo, S., Donatiello, L., Van Dijk, N.M., "Bound performance models of heterogeneous parallel processing systems", IEEE Trans. on Parallel and Distributed Systems, Vol. 9, No. 10, October 1998.

[5] Heidelberger, P., Trivedi, S. K. "Analytic queueing models for programs with internal concurrency", IEEE Trans. on Computers, 32, Jan 1983, pp. 73 – 82.

[6] Kumar, A., Shorey, R. "Performance analysis and scheduling of stochastic jobs in a multicomputer system", IEEE Trans. on Parallel and Distributed Systems, 4, 10, Oct. 1993, pp. 1147 – 1164.

[7] Lavenberg, S.S., Reiser, M. "Stationary state probabilities at arrival instants for closed queueing networks with multiple types of customers", Journal Appl. Prob., 17, 1980, pp. 1048 – 1061.

[8] Lee, E.K., Katz, R.H., "An analytic performance model of disk arrays", Proceedings of ACM SIGMETRICS, 1993.

[9] Liu, Y. C., Perros, H. G. "A decomposition procedure for the analysis of a closed fork/join queueing system", IEEE Transactions on Computers, 40, 3, Mar. 1991, pp. 365 – 370.

[10] Lui, J.C.S., Muntz, R.R., Towsley, D, "Bounding the response time of a minimum expected delay routing system: An algorithmic approach", IEEE Trans. on Computers, vol 44, no. 5, pp. 1371 – 1382, May 1995.

[11] Lui, J.C.S., Muntz, R.R., Towsley, D, "Computing performance bounds of fork-join parallel programs under a multiprocessing environment", IEEE Trans. on Parallel and Distributed Systems, vol. 9, no. 3, pp. 295-311, March 1998.

[12] Makowski, A., Varma, S., "Interpolation approximations for symmetric fork-join queues", Performance Evaluation 20, pp. 145 – 165, 1994.

[13] Nelson, R., Tantawi, N. "Approximate analysis of fork/join synchronization in parallel queues", IEEE Transaction on Computers, 37, 6, June 1988, pp. 739 – 743.

[14] Reiser, M., Lavenberg, S. S. "Mean-value analysis of closed multichain queueing networks", Journal of the ACM, 27, 2, April 1980, pp. 313 – 322.

[15] Sevcik, K.C., Mitrani, I. "The distribution of queueing network states at input and output instants", Journal of the ACM, 28, 2, April 1981, pp. 358 – 371.

[16] Thomasian, A., Tantawi, A.N., "Approximate solutions for M/G/1 fork-join synchronization", Proc. 1994 Winter Simulation conf., pp. 361 – 368, Orlando, Fla., Dec 1994.

[17] Thomasian, A., Menon, J., "Approximate analysis for fork-join synchronization in RAID5", Computer Systems: Science and Eng., 1997.

[18] Varki, E., Dowdy, L.W. "Analysis of closed balanced fork-join queueing networks", Proceedings of ACM/SIGMETRICS 1996.

[19] Varki, E., "Response time analysis of closed fork-join networks", Technical Report, University of New Hampshire, 1998.