Approximate Analysis of a Closed Fork/Join Model

Y.C. Liu
Center for Communications and Signal Processing
Electrical and Computer Engineering

and

H.G. Perros
Center for Communications and Signal Processing
Department of Computer Scjience
North Carolina State University
Raleigh, NC 27695-8206

CCSP-TR-87/8

Abstract

An approximation algorithm for analyzing a closed queueing systems with a K-dimensional fork/join queue is presented. The procedure is based on Norton's theorem. Comparisons with exact numerical data and simulation data show that the approximation procedure gives results which are an upper bound of the mean response time of the fork/join operation and a lower bound of the system throughput, for both homogeneous and non-homogeneous cases. A modification of this procedure, applicable only to the homogeneous case, is also presented. This procedure was found to give very good results for the system throughput and the mean response time of the fork/join operation (the relative error is less than 3%).

1. Introduction

In recent years, there has been a growing interest in the development of tools for analyzing the performance of distributed and parallel processing systems. In such systems, quite often, a job is split into two or more sub-jobs. These sub-jobs execute independently of one another, and at the end of their execution, they recompose to the original job. These type of operations are known as fork/join or disassembly/assembly in production system. They occur in multiprocessor computing systems, distributed database systems, telecommunication systems and flexible manufacturing systems.,

In this paper, we consider a closed queueing system with a K-dimensional fork/join queue and M identical jobs. When a primary job finishes its service at the fork node it is split into a fixed number K ($K \geq 2$) of sub-jobs called siblings. Each sibling executes independently of one another. A sibling upon completion of its service enters the synchronization queue where it waits for the other siblings. As soon as all the siblings have been served, they merge into the original primary job. We call the time between the fork and the join operation of a job as the response time. Also, the time a sibling spends waiting for the other siblings is referred to as the synchronization delay.

Queueing networks with fork/join operations are in general difficult to analyze. An exact numerical approach to modelling a fork/join queue results in an explosion of the state space and makes computational intractable. Also, in general, fork/join models do not have analytically closed form solutions. In view of this, most of these systems reported in the literature have been analyzed using various approximation methods.

Flatto [6,7] considered a fork/join queueing system, assuming that jobs arrive in a poisson fashion. Upon arrival, a job is split into two siblings, each sibling is served by a different server. The service time at each server is exponentially distributed with a different mean. As soon as the two siblings have been served, they recompose into a primary job which leaves the system immediately. Flatto obtained the stationary distribution of the response time, and for each queue he obtained the queue length distribution and its expectation conditioned upon the other queue. Heidelberger and Trivedi [8] considered a queueing network model of a computing system in which jobs divide into two or more asynchronous tasks, i.e., synchronization between tasks is not required. They developed an algorithm which use an iterative technique for solving a sequence of product form type queueing networks. In [9], they extended the model to include a join node. Two approximation methods were developed. The first one was based on a decomposition approximation, consisting of an inner model, a product form queueing network, and an outer model, a finite state markov chain. The other approximation was based on the complementary delays method, which iteratively solves a sequence of product form queueing networks. Nelson and Tantawi [10] proposed an approximation technique, called the scaling approximation. They assumed that the mean response time increases at the same rate as the number of sibling increase. They derived a closed-form approximate expression of the mean response time for a homogeneous fork/join queueing systems consisting of K (K≥2) identical servers. The fork/join queue was analyzed numerically as a closed network by Duda and Czachorski [4]. The numerical approach is inherently limited to small problems. Also, Duda [5] developed an approximation algorithm for analyzing the same system, by constructing an approximately equivalent queueing network with a product-form solution. This approximation method was validated for only 3 siblings. The general model was considered by Baccelli and Makowski [2]. Their analysis was based on renewal type of arguments. They concluded that an upper bound of the response time can be obtained using a GI/G/1 mutually independent parallel queueing system and a lower bound can be obtained using a D/G/1 parallel queueing system. The tightness of these bound was not analyzed. Ammar and Gershwin [1] presented an equivalence relation for disassembly/assembly operations in production systems. In particular, by studying the movement of holes, they converted a 2-dimension disassembly/assembly system into an equivalent tandem queueing network.

In this paper, we consider a closed queueing system with a K-dimensional fork/join queue. The system consists of homogeneous or non-homogeneous exponential servers. The buffer size of each queue is infinite. Using Norton's theorem (see Chandy, Herzog, and Woo [3]), we iteratively reduce the K-dimensional fork/join queue into a 2-dimensional fork/join queue. This approximation method gives a lower bound of the system throughput and an upper bound of the response time. For the homogeneous case, we empirically observed that the difference of the system throughput between the approximate solution and the exact solution proportionally increased as the number of siblings increased. Hence, we developed a modification procedure in order to improve the accuracy of the approximation algorithm. The relative error of the system throughput and the response time of the modified approximation algorithm was found to be less then 3%.

In the following section, we describe the fork/join model. In section 3, we present an approximation algorithm for analyzing a closed queueing system with a fork/join queue. In section 4, we describe the modification procedure. Finally, the conclusions are given in section 5.

2. Model Description

The model studied in this paper is a closed queueing system consisting of a primary node and a K-dimensional fork/join queue, as shown in figure 1. The K-dimensional fork/join queue consists of K servers arranged in parallel. Each server has its own queue, hereafter called the sibling queue. For each sibling queue there is a synchronization queue. The capacity of each queue is assume to be infinite, the service time of each server is exponentially distributed, and the service discipline is FIFO. Let M be the number of primary jobs in the system. Each primary job is split up into K siblings (fork operation) upon completion of its service at the primary node. Each sibling joins a different sibling queue. Upon completion of its service, a sibling enters its synchronization queue where it waits for the other siblings associated with the same job. When all siblings have completed their service, i.e., each sibling is in its synchronization queue, they recompose immediately to the original primary job which joins the primary node. At time t, let P(t) be number of primary jobs, $S_i(t)$ be number of siblings in sibling queue i, $W_i(t)$ be number of siblings that have been serviced and are waiting in the ith synchronization queue, i = 1,2...K. Then, at any given time t, $M = P(t) + S_i(t) + W_i(t)$, i = 1,2...K.

The motivation behind studying this fork/join system as a closed queueing network, is that it allows us to incorporate approximately fork/join operations in a BCMP type of a queueing networks. Let us consider, for example, a queueing network of the BCMP type, and let us assume that jobs after completing service at node i, are split up into K siblings. Each sibling enters a different queue. When all siblings complete their service, they merge into a single job which joins some other queue in the system. Queue i along with the fork/join queue can be replaced approximately by an equivalent server, so that the resulting queueing network is in the BCMP type. This equivalent server can be obtained by analyzing queue i and the fork/join queue as a closed queueing network (as shown in figure 1), with M jobs in it, where M varies from 1 to the total number in the system. This paper concentrates on the analysis of such closed queueing networks with a fork/join queue. The approximate substitution of this system by an equivalent server in a BCMP queueing network is not investigated, seeing that such an approximation is common place and quite well understood (cf. Perros, Nilsson, and Liu [11]).

In order to analyze the above model, we developed both a numerical procedure and an approximation algorithm. The numerical procedure involved the following steps: (1) generation of all the states of the system; (2) generation of the rate matrix Q; and (3) solution of the linear system $Qx^T = 0$ in order to obtain the steady state probability distribution. We note that the state space increases very fast seeing that the number of states is equal to $(M+1)^K$. In view of this, the numerical procedure is only suitable for analyzing small systems. On the other hand, the approximation procedure described below can be used for any number of siblings.

3. The Approximation Procedure

In this section, we discuss an approximation procedure for analyzing the closed queueing network with a K-dimensional fork/join queue as described above and shown in figure 1. The approximation procedure can be briefly summarized as follows. We first analyze approximately the K-dimensional fork/join queue shown in figure 2. In particular, using Norton's theorem, this system is approximately reduced to a 2-dimensional system, which is then used to replace the K-dimensional system in the original queueing network as shown in figure 3. This queueing system is then analyzed numerically, as described above.

Let us first consider the fork/join queue shown in figure 2, assuming that K = 2. Let $\mu_1(i)$ and $\mu_2(i)$ be the state-dependent service rate of sibling queue 1 and 2, i = 1,2,...M. Let $P_{i,j}$ be the steady state probability that there are i and j jobs in sibling queue 1 and 2 respectively. It can be easily shown that this system has a product-form solution (see also Duda and Czachorski [4]). The stationary equations are as follows :

$$P_{M,M}\ (\mu_1(M)+\mu_2(M)) = \mu_1(M)\ P_{M,M-1} + \mu_2(M)\ P_{M-1,M} \tag{1}$$

$$P_{M,N}\ (\mu_1(M)+\mu_2(N)) = \mu_1(M)\ P_{M,N-1} + \mu_2(N+1)\ P_{M,N+1}, \tag{2}$$

$$1 \leq N \leq M-1$$

$$P_{M,0}\ \mu_1(M) = P_{M,1}\ \mu_2(1) \tag{3}$$

$$P_{N,M}\ (\mu_1(N)+\mu_2(M)) = \mu_1(N+1)\ P_{N+1,M} + \mu_2(M)\ P_{N-1,M}, \tag{4}$$

$$1 \leq N \leq M-1$$

$$P_{0,M}\ \mu_2(M) = P_{1,M}\ \mu_1(1) \tag{5}$$

$$P_{M,M} + \sum_{i=0}^{M-1} P_{M,i} + \sum_{j=0}^{M-1} P_{j,M} = 1 \tag{6}$$

We can easily obtain

$$P_{N,M} = \prod_{j=N+1}^{M} \rho_2(j)\ P_{M,M}, \quad 0 \leq N \leq M-1 \tag{7}$$

$$P_{M,N} = \prod_{j=N+1}^{M} \rho_1(j)\ P_{M,M}, \quad 0 \leq N \leq M-1 \tag{8}$$

and

$$P_{M,M} = \frac{1}{1 + \sum\limits_{i=0}^{M-1} \prod\limits_{j=i+1}^{M} \rho_2(j) + \sum\limits_{j=0}^{M-1} \prod\limits_{N=j+1}^{M} \rho_1(N)} \tag{9}$$

where $\rho_2(j) = \dfrac{\mu_1(j)}{\mu_2(M)}$ and $\rho_1(j) = \dfrac{\mu_2(j)}{\mu_1(M)}$

The system throughput, T(M), is :

$$T(M) = \sum_{i=1}^{M} \mu_1(i)\ P_{i,M} + \sum_{j=0}^{M-1} \mu_1(M)\ P_{M,j} \tag{10}$$

Using expression (10) in conjunction with Norton's theorem. we can now reduce approximately the K-dimensional fork/join system to a 2-dimensional fork/join system as follows.

Let us first number the sibling queues in figure 2 from 1 to K starting from the top. The approximation algorithm can now be summarized as follows:

Step 1. Consider the closed queueing system consisting of sibling queues 1 and 2 (and their corresponding synchronization queues) obtained from figure 2 by shorting out the remaining sibling queues. Let T (i), i = 1,2,..M, be the throughput of this system, obtained using expression (10). Replace this system by an equivalent composite node, call it C_1, with a state-dependent service rate equal to T (i). Now, use composite node C_1 and sibling queue 3 to form a new 2-dimensional fork/join queue. As above, construct an equivalent composite node, call it C_2.

Proceed as above until the K-1 sibling queues are replaced by an equivalent composite node C_{K-2}. The K-dimensional fork/join queue has now been reduced to a 2-dimensional queue consisting of the composite node C_{K-2} and sibling queue K and their associated synchronization queues.

Step 2. In the original queueing network system substitute the K-dimensional fork/join queue by the 2-dimensional fork/join queue obtained from step 1 (see figure 3). Solve this model numerically to obtain the system throughput, the mean queue length of the primary node, the Kth sibling queue, and its synchronization queue. Also, by applying Little's relation one can obtain the mean waiting time in each of these three queues, and the mean response time of the fork/join operation, i.e., the mean time elapsing from the moment a job is split up to the moment it joins again the primary node.

The above procedure yields performance measures for the primary node and the Kth sibling queue (and its synchronization queue). We can obtain performance measures of any other sibling queue i, i = 1,2,..,K-1 and its associated synchronization queue, by simply exchanging sibling queue i with sibling queue K and then apply the above algorithm.

The above algorithm was implemented on a VAX 11/785. The approximate results were compared against exact numerical values for 3 and 4-dimensional fork/join queues, and with simulation results for 8-dimensional fork/join queues. The results for both homogeneous cases and non-homogeneous cases are presented in tables 1 to 6. Each table gives the approximate and exact (or simulation) results for the system throughput, mean queue lengths, and the mean response time of the fork/join operation for various values of μ_0. The relative error (expressed as a percentage) is also given. We note that the relative error increases as K increases and

$$\frac{\mu_0}{\mu_1} \leq 1 .$$

In general, the results obtained from the above procedure give a lower bound of the system throughput. This is due to the additional synchronization delay introduced each time a composite node is used to substitute a 2-dimensional fork/join queue. Obviously, this solution gives an upper bound of the mean response time of the fork/join operation.

4. The Modified Approximation Procedure for Homogeneous Fork/Join Queues

In this section, we present a modification of the procedure described above that can be used to analyze the model shown in figure 1, under the assumption that all the sibling queues are served at the same rate (homogeneous system). The modified procedure aims at improving the accuracy of step 1. It is based on the empirical observation that the difference between the exact throughput $T_n(i)$ of the n-dimensional fork/join queue and the approximate throughput $\hat{T}_n(i)$ obtained from step 1, proportionally increases as n increases. In particular,

$$T_n(i) \approx \hat{T}_n(i)[1+(n-2)\alpha(i)], \quad i=1,2,...M.$$

where

$$\alpha(i) = \frac{T_3(i)}{\hat{T}_3(i)} - 1,$$

and n = 3, 4,...,K. Therefore, we can numerically analyze the 3-dimensional fork/join queue, in order to obtain $T_3(i)$, i = 1, 2,...,M. We can also obtain $\hat{T}_3(i)$, i = 1, 2,..,M, approximately from step 1 of the above approximation algorithm. Using these two quantities, we can obtain the correction factor $\alpha(i)$, i = 1,2,..,M, which can then be used to construct an improved estimate of the system throughput, $T_{e,K}(i)$. i = 1,2,..,M. of the final 2-dimensional fork/join queue obtained from step 1. We have

$$T_{e,K}(i) = \hat{T}_K(i)[1+(K-2)\,\alpha(i)], \quad i=1,2,..,M.$$

Before, we proceed with step 2, we need to adjust the state-dependent service rate $\mu_c(i)$, i = 1, 2,..,M, of the composite queue of the final 2-dimensional fork/join queue so that the system throughput is equal to $T_{e,K}(i)$, i = 1, 2,..,M. This is calculated as follows :

1. set $m = 1$;

2. set $i = 0$; $\mu_c^{(i)}(m) = \mu_c(m)$; $T_K^{(i)}(m) = \hat{T}_K(m)$.

3. $\mu_c^{(i+1)}(m) = \dfrac{T_{e,K}(m)}{T_K^{(i)}(m)} * \mu_c^{(i)}(m)$. Use $\mu_c^{(i+1)}(m)$ to obtain the adjustment throughput $T_K^{(i+1)}(m)$ from equation (10).

4. set $i := i + 1$; if $| T_{e,K}(m) - T_K^{(i)}(m) | < \epsilon$, then set $\mu_c(m) = \mu_c^{(i)}(m)$ and go to step 5, else go to step 3.

5. if $m < M$, then $m := m + 1$, go to step 2, else stop.

We can apply the same numerical procedure as in step 2 to obtain the approximate solution.

This modified approximation algorithm was implemented on a VAX-11/785. The main results obtained is the system throughput, and the mean queue length of the primary node, and of each sibling and synchronization queue. From this, other performance measures, such as the mean response time of the fork/join operation and the mean synchronization time for each sibling, can also be obtained. The approximate results were compared against exact numerical values for 4-dimensional fork/join .queues, and with simulation results for 8-dimensional fork/join queues. The results obtained are summarized in tables 7 to 8. Each table gives the approximate and exact results or simulation results for the throughput, mean queue lengths and mean response time of the fork/join operation for various values of μ_0. The relative error (expressed as a percentage) is also given.

The modified procedure gives very good results for the system throughput and the mean response time of the fork/join operation (the relative error is less than 3%). But, we get a big error for the mean queue lengths of a sibling queue and its synchronization queue when $\frac{\mu_0}{\mu_1} > 1$. We note, however, that the sum of these two mean queue lengths is practically equal to the exact solution.

5. Conclusion

We developed an approximation procedure for analyzing a closed queueing network with a K-dimensional fork/join queue. The approximation procedure is based on Norton's theorem, and it was shown through a number of examples that it gives results which are a lower bound of the system throughput and an upper bound of the mean response time of the fork/join operation. The modified procedure is only applicable to the homogeneous case. It gives very good results for the system throughput and the mean response time of the fork/join operation.

We note that the original queueing network shown in figure 1 can be also approximately reduced·a two-node closed queueing networks consisting of the original primary node and a composite node with a state-dependent service rate equal to $T_{e,N}(i)$. This closed queueing network can be easily analyzed. The system throughput, the mean queue length of the primary node and the mean response time of the fork/join operation obtained from this model are very close to the exact results. The main drawback of this approach is that we loose information regarding the synchronization delay.

REFERENCES

[1] **Ammar, M.H. and Gershwin, S.B.**, "Equivalence Relations in Queueing Models of Manufacturing Networks", Proceeding of the nineteenth IEEE Conference on Decision and Control. 1981.

[2] **Baccelli, F. and Makowski, A.M.**, "Simple Computable Bounds for the Fork-Join Queue", The John Hopkins Conf. on Inf. Sc., John Hopkins U., 1985.

[3] **Chandy, K.M. and Herzog, U. and Woo, L.**, "Parametric Analysis of Queueing Networks", IBM J. Res. Develop. 19, 1975, 36-42.

[4] **Duda, A. and Czachorski, T.**. "Performance Evaluation of Fork and Join Synchronization Primitives". to appear in ACTA Information.

[5] **Duda, A.**. "Approximate Performance Analysis of parallel Systems". Proc. of 2nd Inf. Workshop on Appl. Mathematics and Performance/Reliability Models of Computer/Communication Systems. Rome. Italy. May 25-27. 1987.

[6] **Flatto,L. and Hahn, S.**, "Two Parallel Queues Created by Arrivals with Two Demands I", SIAM J. Appl. Math. vol. 44, Oct. 1984, 1041-1053.

[7] **Flatto,L.**, "Two Parallel Queues Created by Arrivals with Two Demands II", SIAM J. Appl. Math. vol. 45, Oct. 1985, 861-878.

[8] **Heidelberger, P. and Trivedi, S.K.**, "Queueing Network Models for Parallel Processing with Asynchronous Tasks", IEEE Trans. on Comp. vol. 31, Nov. 1982, 1099-1109.

[9] **Heidelberger, P. and Trivedi, S.K.**, "Analytic Queueing Models for Programs with Internal Concurrency", IEEE Trans. on Comp vol. 32, Jan. 1983, 73-82.

[10] **Nelson, R. and Tantawi, A.N.**, "Approximate Analysis of Fork/Join Synchronization in Parallel Queues", IBM Research Report RC 11481.

[11] **Perros, H.G., Nilsson, A.A. and Liu, Y.C.**, "Approximate Analysis of Product Form Type Queueing Networks with Blocking and Deadlock", To be published in Performance Evaluation.

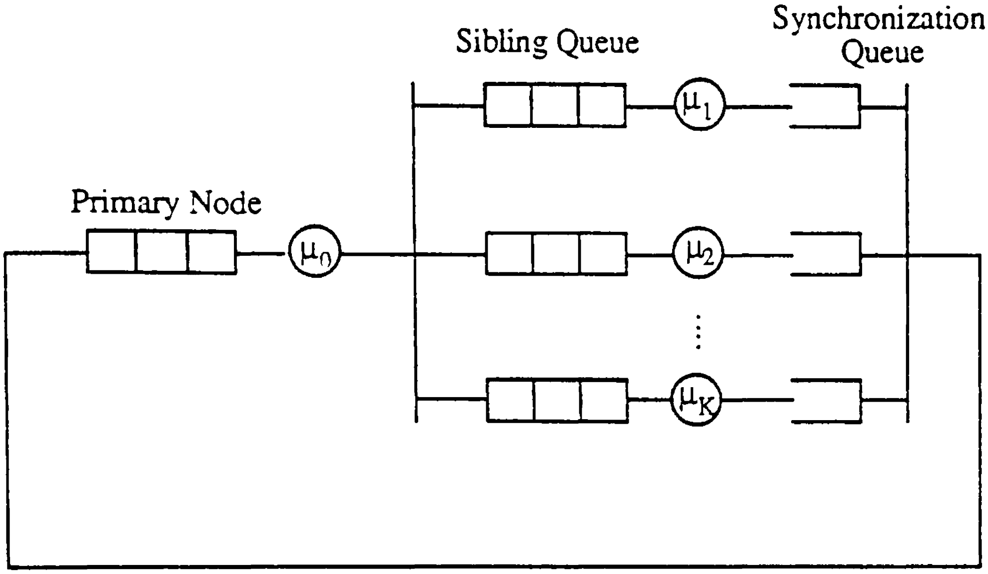

Fig. 1 The System Under Study

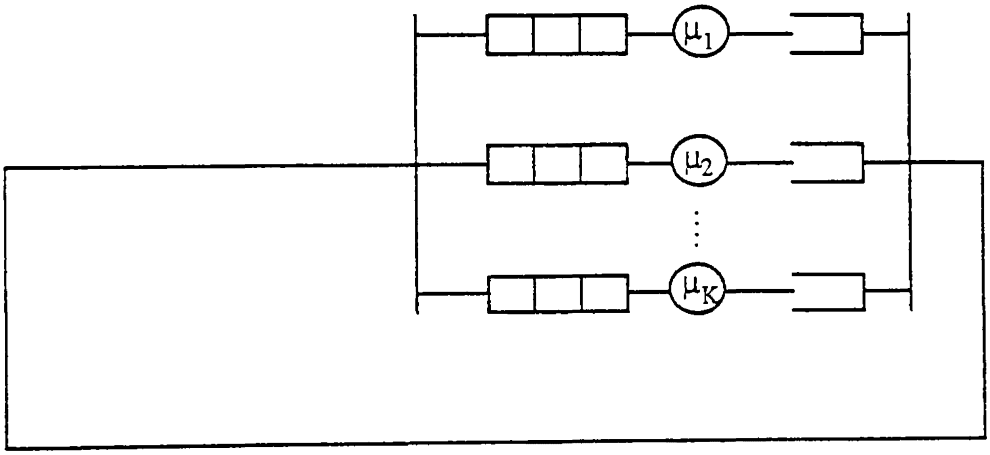

Fig. 2 The K-dimensional Fork/Join System

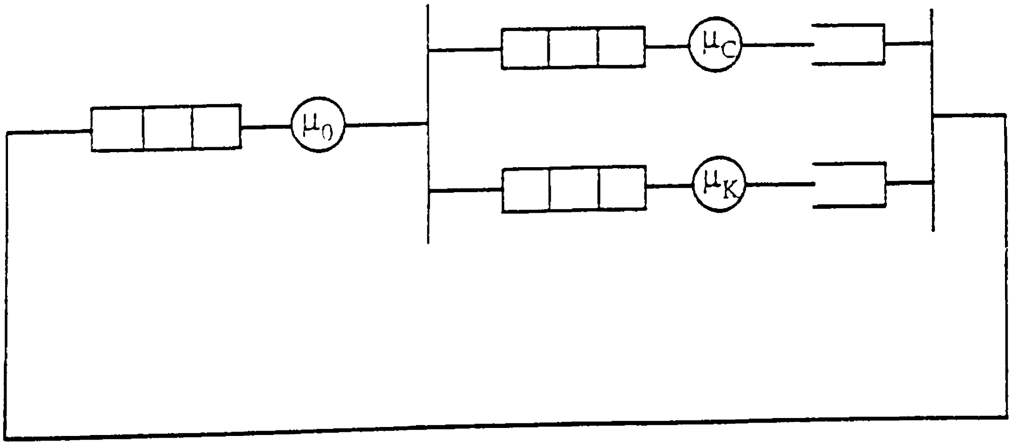

Fig. 3 The Reduced Network

Table 1 : 3-dimensional homogeneous fork/join model; M =10; $\mu_1=\mu_2=\mu_3=2$.

System Throughput

μ_0/μ_1	Exact	Appro.	error
10.0	1.853	1.843	0.54%
5.0	1.852	1.840	0.65%
1.0	1.712	1.699	0.76%
0.5	0.998	0.998	0.00%
0.1	0.200	0.200	0.00%

Mean Response Time of Fork/join Operation

μ_0/μ_1	Exact	Appro.	error
10.0	5.342	5.371	0.55%
5.0	5.279	5.314	0.66%
1.0	3.751	3.823	1.92%
0.5	1.705	1.792	5.10%
0.1	1.005	1.043	3.79%

Mean Queue Length at Primary Node

μ_0/μ_1	Exact	Appro.	error
10.0	0.1014	0.1009	0.49%
5.0	0.2238	0.7230	0.35%
1.0	3.578	3.5050	2.00%
0.5	8.298	8.2190	0.95%
0.1	9.799	9.7910	0.08%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	6.203	6.215	0.19%
5.0	6.123	6.130	0.11%
1.0	3.934	3.860	1.88%
0.5	0.991	0.998	0.70%
0.1	0.111	0.111	0.00%

Mean Queue Length at Synchronization Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	3.69600	3.68400	0.32%
5.0	3.65300	3.64700	0.16%
1.0	2.48800	2.63500	5.90%
0.5	0.71020	0.79030	11.20%
0.1	0.08994	0.09762	8.30%

Table 2 : 3-dimensional non-homogeneous fork/join model; $M = 10$; $\mu_1=1, \mu_2=2, \mu_3=3$.

System Throughput

μ_0/μ_1	Exact	Appro.	error
10.0	0.9998	1.0000	0.02%
5.0	0.9998	1.0000	0.02%
1.0	0.9090	0.9091	0.01%
0.5	0.4997	0.4997	0.00%
0.1	0.0999	0.1000	0.10%

Mean Response Time of Fork/join Operation

μ_0/μ_1	Exact	Appro.	error
10.0	9.8910	9.8890	0.02%
5.0	9.7520	9.7500	0.02%
1.0	5.5620	5.5720	0.18%
0.5	2.1610	2.1860	1.16%
0.1	1.3240	1.3510	2.06%

Mean Queue Length at Primary Node

μ_0/μ_1	Exact	Appro.	error
10.0	0.1111	0.1111	0.00%
5.0	0.2499	0.2499	0.00%
1.0	4.9440	4.9340	0.20%
0.5	8.9200	8.9080	0.13%
0.1	9.8680	9.8650	0.03%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	9.8880	9.888	0.00%
5.0	9.7490	9.748	0.01%
1.0	4.9980	4.999	0.02%
0.5	0.9945	0.9947	0.02%
0.1	0.1111	0.1111	0.00%

Mean Queue Length at Synchronization Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	0.001106	0.00136	22.6%
5.0	0.001311	0.00161	22.4%
1.0	0.057850	0.06693	15.7%
0.5	0.085330	0.09750	14.3%
0.1	0.021140	0.02399	13.5%

Mean Queue Length at Sibling Queue 2

μ_0/μ_1	Exact	Appro.	error
10.0	0.99880	0.9990	0.22%
5.0	0.99870	0.9989	0.02%
1.0	0.81910	0.8176	0.18%
0.5	0.33300	0.3300	0.90%
0.1	0.05263	0.0526	0.00%

Mean Queue Length at Synchronization Queue 2

μ_0/μ_1	Exact	Appro.	error
10.0	8.89000	8.8900	0.00%
5.0	8.75100	8.7510	0.00%
1.0	4.23700	4.2700	0.78%
0.5	0.74690	0.7700	3.10%
0.1	0.07962	0.0824	3.50%

Mean Queue Length at Sibling Queue 3

μ_0/μ_1	Exact	Appro.	error
10.0	0.49970	0.5000	0.06%
5.0	0.49970	0.5000	0.06%
1.0	0.43180	0.4304	0.32%
0.5	0.19980	0.1998	0.00%
0.1	0.03448	0.0345	0.00%

Mean Queue Length at Synchronization Queue 3

μ_0/μ_1	Exact	Appro.	error
10.0	9.38900	9.3890	0.00%
5.0	9.25000	9.2500	0.00%
1.0	4.62400	4.6830	1.28%
0.5	0.88000	0.9156	4.00%
0.1	0.09777	0.1008	3.10%

Table 3 : 4-dimensional homogeneous fork/join model; $M = 5$; $\mu_1=\mu_2=\mu_3=\mu_4=1$.

System Throughput

μ_0/μ_1	Exact	Appro.	error
10.0	0.8315	0.8122	2.32%
5.0	0.8288	0.8082	2.48%
1.0	0.7244	0.7060	2.54%
0.5	0.4786	0.4741	0.94%
0.1	0.1000	0.1000	0.00%

Mean Response Time of Fork/join Operation

μ_0/μ_1	Exact	Appro.	error
10.0	5.9050	6.0490	2.44%
5.0	5.8020	5.9540	2.62%
1.0	4.6400	4.8510	4.55%
0.5	3.4300	3.6820	7.33%
0.1	2.2760	2.4560	7.91%

Mean Queue Length at Primary Node

μ_0/μ_1	Exact	Appro.	error
10.0	0.08913	0.08735	1.99%
5.0	0.19150	0.18790	1.88%
1.0	1.63900	1.72600	3.90%
0.5	3.35900	3.25500	3.10%
0.1	4.77200	4.75400	0.37%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	2.69100	2.65600	1.30%
5.0	2.63000	2.58800	1.60%
1.0	1.79200	1.72600	3.68%
0.5	0.84750	0.83400	1.60%
0.1	0.11100	0.11100	0.00%

Mean Queue Length at Synchronization Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	2.21900	2.25700	1.70%
5.0	2.17900	2.22400	2.10%
1.0	1.56900	1.69900	8.30%
0.5	0.79390	0.91140	14.80%
0.1	0.11660	0.13460	15.00%

Table 4 : 4-dimensional non-homogeneous fork/join model; M =5; $\mu_1=1, \mu_2=2, \mu_3=3, \mu_4=4$.

System Throughput

μ_0/μ_1	Exact	Appro.	error
10.0	0.9997	0.9989	0.08%
5.0	0.9992	0.9984	0.08%
1.0	0.8317	0.8310	0.08%
0.5	0.4919	0.4918	0.02%
0.1	0.1000	0.1000	0.00%

Mean Response Time of Fork/join Operation

μ_0/μ_1	Exact	Appro.	error
10.0	4.8905	4.8945	0.08%
5.0	4.7572	4.7612	0.08%
1.0	3.1121	3.1375	0.82%
0.5	2.0248	2.0683	2.15%
0.1	1.3400	1.3940	4.03%

Mean Queue Length at Primary Node

μ_0/μ_1	Exact	Appro.	error
10.0	0.1106	0.1105	0.09%
5.0	0.2470	0.2466	0.16%
1.0	2.4120	2.3930	0.79%
0.5	4.0040	3.9830	0.52%
0.1	4.8660	4.8610	0.10%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	4.8560	4.8470	0.18%
5.0	4.7150	4.7050	0.21%
1.0	2.4840	2.4780	0.24%
0.5	0.9037	0.9033	0.04%
0.1	0.1111	0.1111	0.00%

Mean Queue Length at Synchronization Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	0.0330	0.0421	27.53%
5.0	0.0384	0.0486	26.60%
1.0	0.1043	0.1293	23.97%
0.5	0.0923	0.1139	23.40%
0.1	0.0229	0.0283	23.22%

Mean Queue Length at Sibling Queue 2

μ_0/μ_1	Exact	Appro.	error
10.0	0.9620	0.9624	0.04%
5.0	0.9561	0.9560	0.01%
1.0	0.6714	0.6661	0.79%
0.5	0.3229	0.3323	0.19%
0.1	0.0526	0.0526	0.00%

Mean Queue Length at Synchronization Queue 2

μ_0/μ_1	Exact	Appro.	error
10.0	3.9270	3.9270	0.00%
5.0	3.7970	3.7970	0.00%
1.0	1.9170	1.9530	1.88%
0.5	0.6731	0.7058	4.86%
0.1	0.0814	0.0866	6.32%

Mean Queue Length at Sibling Queue 3

μ_0/μ_1	Exact	Appro.	error
10.0	0.4925	0.4967	0.85%
5.0	0.4913	0.4949	0.73%
1.0	0.3737	0.3701	0.96%
0.5	0.1951	0.1946	0.26%
0.1	0.0345	0.0345	0.00%

Mean Queue Length at Synchronization Queue 3

μ_0/μ_1	Exact	Appro.	error
10.0	4.3970	4.3920	0.11%
5.0	4.2620	4.2570	0.11%
1.0	2.2140	2.2540	1.80%
0.5	0.8009	0.8404	4.93%
0.1	0.0996	0.1049	5.34%

Mean Queue Length at Sibling Queue 4

μ_0/μ_1	Exact	Appro.	error
10.0	0.3298	0.3324	0.79%
5.0	0.3294	0.3315	0.64%
1.0	0.2583	0.2556	1.04%
0.5	0.1398	0.1394	0.29%
0.1	0.0256	0.0256	0.00%

Mean Queue Length at Synchronization Queue 4

μ_0/μ_1	Exact	Appro.	error
10.0	4.5600	4.5570	0.06%
5.0	4.4240	4.4200	0.09%
1.0	2.3300	2.3720	1.80%
0.5	0.8563	0.8991	5.00%
0.1	0.1084	0.1139	5.07%

Table 5 : 8-dimensional homogeneous fork/join model; $M = 15$; $\mu_1 = \mu_2 = ... = \mu_8 = 1$.

System Throughput

μ_0/μ_1	simulation	Appro.	error
10.0	0.9187	0.8906	3.06%
5.0	0.9132	0.8894	2.61%
1.0	0.8659	0.8421	2.75%
0.5	0.4931	0.4998	1.36%
0.1	0.0976	0.1000	2.46%

Mean Response Time of Fork/join Operation

μ_0/μ_1	simulation	Appro.	error
10.0	16.3130	16.7400	2.50%
5.0	16.2493	16.6236	2.25%
1.0	12.2200	13.2395	7.70%
0.5	4.8097	5.9049	18.55%
0.1	2.9668	3.6230	18.11%

Mean Queue Length at Primary Node

μ_0/μ_1	simulation	Appro.	error
10.0	0.10210	0.09743	4.79%
5.0	0.22056	0.21500	2.58%
1.0	4.40822	3.85100	14.47%
0.5	12.63000	12.05000	4.81%
0.1	14.70000	14.64000	0.52%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	simulation	Appro.	error
10.0	6.8389	6.8150	0.35%
5.0	6.6968	6.7360	0.58%
1.0	4.7367	4.4280	6.97%
0.5	0.9742	0.9983	2.41%
0.1	0.1129	0.1111	1.62%

mean Queue Length at Synchronization Queue 1

μ_0/μ_1	simulation	Appro.	error
10.0	8.0567	8.0870	0.37%
5.0	8.0825	8.0490	0.41%
1.0	5.8550	6.7210	12.89%
0.5	1.3956	1.9530	28.50%
0.1	0.1875	0.2512	25.36%

Table 6 : 8-dimensional non-homogeneous fork/join model; M =15; $\mu_1=\mu_2=1, \mu_3=\mu_4=2, \mu_5=\mu_6=3, \mu_7=\mu_8=4$.

System Throughput

μ_0/μ_1	simulation	Appro.	error
10.0	0.9770	0.9666	1.06%
5.0	0.9668	0.9663	0.05%
1.0	0.9171	0.9126	0.49%
0.5	0.5034	0.4899	2.68%
0.1	0.0986	0.1000	1.40%

Mean Response Time of Fork/join Operation

μ_0/μ_1	simulation	Appro.	error
10.0	15.40820	15.4044	0.03%
5.0	15.10850	15.2709	1.06%
1.0	9.95020	9.4570	4.96%
0.5	3.10450	3.3192	6.47%
0.1	1.81640	2.0778	12.58%

Mean Queue Length at Primary Node

μ_0/μ_1	simulation	Appro.	error
10.0	0.10544	0.1070	1.46%
5.0	0.24202	0.2393	1.15%
1.0	6.32950	5.9273	6.79%
0.5	13.43600	13.3400	0.72%
0.1	14.82000	14.7900	0.20%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	simulation	Appro.	error
10.0	11.0072	10.9100	0.89%
5.0	10.9437	10.8200	1.14%
1.0	6.3467	6.4430	1.49%
0.5	1.0543	0.9997	5.46%
0.1	0.1083	0.1111	2.52%

mean Queue Length at Synchronization Queue 1

μ_0/μ_1	simulation	Appro.	error
10.0	3.8873	3.98000	2.33%
5.0	3.8142	3.94600	3.34%
1.0	2.3237	2.63800	11.91%
0.5	0.5092	0.62650	18.71%
0.1	0.0708	0.09654	26.68%

Mean Queue Length at Sibling Queue 3

μ_0/μ_1	simulation	Appro.	error
10.0	0.90400	0.93690	3.51%
5.0	0.92257	0.93610	1.45%
1.0	0.85246	0.82390	3.47%
0.5	0.33888	0.33333	1.67%
0.1	0.05314	0.05263	0.97%

mean Queue Length at Synchronization Queue 3

μ_0/μ_1	simulation	Appro.	error
10.0	13.9905	13.9600	0.22%
5.0	13.8354	13.8200	0.11%
1.0	7.8179	8.2400	5.12%
0.5	1.2245	1.3290	7.86%
0.1	0.1259	0.1549	18.72%

Mean Queue Length at Sibling Queue 5

μ_0/μ_1	simulation	Appro.	error
10.0	0.47130	0.47600	0.99%
5.0	0.47450	0.47560	0.23%
1.0	0.43900	0.43240	1.53%
0.5	0.20200	0.20000	1.00%
0.1	0.03369	0.03448	2.29%

mean Queue Length at Synchronization Queue 5

μ_0/μ_1	simulation	Appro.	error
10.0	14.4233	14.4200	0.02%
5.0	14.2835	14.2800	0.02%
1.0	8.2314	8.6430	4.88%
0.5	1.3614	1.4730	7.58%
0.1	0.1454	0.1734	16.14%

Mean Queue Length at Sibling Queue 7

μ_0/μ_1	simulation	Appro.	error
10.0	0.319318	0.31900	0.01%
5.0	0.322000	0.31880	1.00%
1.0	0.297300	0.29300	1.47%
0.5	0.143550	0.14280	0.53%
0.1	0.025460	0.02564	0.70%

mean Queue Length at Synchronization Queue 7

μ_0/μ_1	simulation	Appro.	error
10.0	14.5752	14.5700	0.04%
5.0	14.4368	14.4409	0.03%
1.0	8.3730	8.7850	4.68%
0.5	1.4199	1.5340	7.44%
0.1	0.1536	0.1824	15.79%

Table 7 : Modified approximation for a 4-dimensional homogeneous fork/join model; $M = 5$; $\mu_1=\mu_2=\mu_3=\mu_4=1$.

System Throughput

μ_0/μ_1	Exact	Appro.	error
10.0	0.8315	0.8394	0.95%
5.0	0.8288	0.8358	0.84%
1.0	0.7244	0.7268	0.33%
0.5	0.4786	0.4783	0.06%
0.1	0.1000	0.1000	0.00%

Mean Response Time of Fork/join Operation

μ_0/μ_1	Exact	Appro.	error
10.0	5.9050	5.8490	0.09%
5.0	5.8020	5.7480	0.09%
1.0	4.6400	4.5620	1.70%
0.5	3.4300	3.3640	1.90%
0.1	2.2760	2.2180	2.60%

Mean Queue Length at Primary Node

μ_0/μ_1	Exact	Appro.	error
10.0	0.08913	0.09055	1.59%
5.0	0.19150	0.19560	2.14%
1.0	1.63900	1.68500	2.80%
0.5	3.35900	3.39100	0.95%
0.1	4.77200	4.77800	0.13%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	2.69100	2.88700	7.28%
5.0	2.63000	2.81300	6.96%
1.0	1.79200	1.83500	2.40%
0.5	0.84750	0.84910	0.19%
0.1	0.11100	0.11100	0.00%

Mean Queue Length at Synchronization Queue 1

μ_0/μ_1	Exact	Appro.	error
10.0	2.21900	2.02300	8.83%
5.0	2.17900	1.99100	8.63%
1.0	1.56900	1.48100	5.60%
0.5	0.79390	0.76010	4.26%
0.1	0.11660	0.11030	5.06%

Table 8 : Modified approximation for an 8-dimensional homogeneous fork/join model; M =15; $\mu_1=\mu_2=....=\mu_8=1$.

System Throughput

μ_0/μ_1	simulation	Appro.	error
10.0	0.9187	0.9150	0.41%
5.0	0.9132	0.9142	0.11%
1.0	0.8659	0.8643	0.18%
0.5	0.4931	0.4999	1.40%
0.1	0.0976	0.1000	2.46%

Mean Response Time of Fork/join Operation

μ_0/μ_1	simulation	Appro.	error
10.0	16.3130	16.2800	0.20%
5.0	16.2493	16.1600	0.55%
1.0	12.2200	12.3200	0.85%
0.5	4.8097	4.9610	3.10%
0.1	2.9668	2.9790	0.41%

Mean Queue Length at Primary Node

μ_0/μ_1	simulation	Appro.	error
10.0	0.10210	0.10040	1.67%
5.0	0.22056	0.22200	0.65%
1.0	4.40822	4.34800	1.37%
0.5	12.63000	12.52000	0.87%
0.1	14.70000	14.70000	0.00%

Mean Queue Length at Sibling Queue 1

μ_0/μ_1	simulation	Appro.	error
10.0	6.8389	7.9180	15.78%
5.0	6.6968	7.8280	16.89%
1.0	4.7367	4.9770	5.07%
0.5	0.9742	0.9991	2.56%
0.1	0.1129	0.1111	1.62%

mean Queue Length at Synchronization Queue 1

μ_0/μ_1	simulation	Appro.	error
10.0	8.0567	6.9810	13.35%
5.0	8.0825	6.9500	14.01%
1.0	5.8550	5.6740	3.09%
0.5	1.3956	1.4810	6.11%
0.1	0.1875	0.1868	0.37%