Response Time Approximations in Fork-Join Queues

Abigail S. Lebrecht and William J. Knottenbelt
Department of Computing
Imperial College London
London, SW7 2AZ, United Kingdom
{asl102,wjk}@doc.ic.ac.uk

Abstract

Fork-join queueing networks model a network of parallel servers in which an arriving job splits into a number of sub-tasks that are serviced in parallel. Fork-join queues can be used to model disk arrays. A response time approximation of the fork-join queue is presented that attempts to comply with the additional constraints of modelling a disk array. This approximation is compared with existing analytical approximations of the fork-join queueing network.

1 Introduction

Engineers of modern computer and communication systems need good analytical models to help predict performance behaviour for a wide range of workloads. In many cases of practical interest, system workload can be abstracted as a job stream, in which each job is split into many synchronised tasks, that are processed in parallel at various, possibly heterogeneous, servers. Examples of such systems include disk arrays (where each logical I/O request becomes several physical I/O requests spread across disk devices), multiprogramming and manufacturing systems.

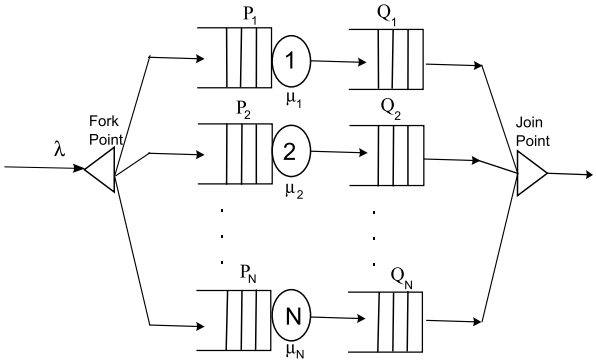

Figure 1: Fork-join queueing model

Conceptually, such systems can be modelled as a fork-join queue within a closed queueing network (see Figure 1). In a fork-join queueing system, each incoming job is split into N tasks at the fork point. Each of these tasks queues for service at a parallel service node before joining a queue for the join point. When all N tasks in the job are at the front of their respective queues, they rejoin (synchronise) at the join point.

It is difficult to model moments of job response time in a fork-join synchronisation analytically. Indeed, to date, exact analytical results exist only for the mean response time of a two server system [4]. For more than two parallel servers, approximations exist for the mean response time of homogeneous servers. Ideally a universal solution or accurate approximation is needed to solve for moments of job response time in generic fork-join networks. The closest to this is Varki's modification of mean value analysis applied to closed fork-join networks [7], which approximates mean values only.

In this case, fork-join queues are being studied specifically for the modelling of disk arrays. Therefore certain constraints on the fork-join model are preferable, for an accurate disk array model. Each disk in the array is modelled as one of the parallel servers of the fork-join queue. The service time of a disk drive is dependent on the disk cylinder seek time and rotational latency and is unlikely to be distributed exponentially. Hence, the disk array requires a fork-join model with M/G/1 parallel queues. The service time distributions and mean service times on each disk are unlikely to be identical, hence any analytical approximation must allow for heterogeneous parallel servers. Finally, a disk array could consist of up to fifty disk drives so the analytical approximation needs to be capable of generating results quickly for a large number of disks.

This paper makes some progress towards a good approximation of the response time distribution for a fork-join queue from which the mean and further moments can be calculated. This is obtained by calculating the maximum cumulative distribution function of a collection of random variables exactly. This result is applicable to a variety of different queues with both homogeneous and heterogeneous servers. Section 2 outlines some of the other approximations for fork-join queues. In Section 3 the analytical approximation for the response time distribution of a fork-join network is presented. Section 4, compares this result with the other fork-join mean response time approximations discussed in Section 2, for fork-join queues with either M/M/1 or M/G/1 parallel queues and homogeneous or heterogeneous parallel servers.

2 Background

In order to solve fork-join queueing networks analytically, most results assume that the parallel queues are independent and identically distributed (iid). The arrival rate to the fork is λ and mean service rate for each queue is μ. Initially Nelson and Tantawi [4] define bounds for the mean response time of an N-branch fork-join system of M/M/1 queues, R_N. The mean response time can then be approximated, based on the observation that both the lower and upper bounds of the mean response time grow at the same rate as a function of the number of servers. By running simulations of the fork-join network with different values of N, the mean response time approximation is calibrated to the following result:

$$R_N \approx \left[\frac{H_N}{H_2} + \frac{4}{11}\left(1 - \frac{H_N}{H_2}\right)\rho\right]\left(\frac{12-\rho}{8}\right)\frac{1}{\mu(1-\rho)} \qquad N \geq 2 \tag{1}$$

where H_N is the harmonic series, $\sum_{i=1}^{N} \frac{1}{i}$. Varma and Makowski [9] use interpolation between light and heavy traffic to approximate the mean response time for M/M/1 queues:

$$R_N \approx \left[H_N + \left(\left(\sum_{i=1}^{N}\binom{N}{i}(-1)^{i-1}\sum_{m=1}^{i}\binom{i}{m}\frac{(m-1)!}{i^{m+1}}\right) - H_N\right)\frac{\lambda}{\mu}\right]\frac{1}{\mu-\lambda} \quad 0 \leq \lambda < \mu \quad N \geq 2 \tag{2}$$

This result can be extended to non-exponential service and arrival times, but in all cases, is only applicable for homogeneous servers.

Varki et al [8] present another approximation for the same conditions as equations (1) and (2),

$$R_N \approx \frac{1}{\mu}\left(H_N + \frac{\rho}{2(1-\rho)}\left(\sum_{i=1}^{N}\frac{1}{i-\rho} + (1-2\rho)\sum_{i=1}^{N}\frac{1}{i(i-\rho)}\right)\right) \tag{3}$$

David [1] describes an upper bound for the mean of the maximum of a set of n iid random variables, X_i.

$$E[X_{(n)}] \leq \mu + \frac{\sigma(n-1)}{\sqrt{2n-1}} \tag{4}$$

where μ is the mean of X and σ is the standard deviation. Thomasian and Tantawi [6] adapt equation (4). Using Nelson and Tantawi's method of observing simulation results they present an analytical result for an approximation to the mean response time of a fork-join queue with M/G/1 queues in parallel service. Their approximation proposes:

$$R_N(\rho) \approx R_1(\rho) + \sigma_1(\rho)F_N\alpha_N(\rho) \tag{5}$$

$R_1(\rho)$ and $\sigma_1(\rho)$ are the mean response time and standard deviation respectively for one M/G/1 queue with no fork-join properties. F_N is a constant dependent on the service time distribution of the parallel servers and $\alpha_N(\rho)$ scales according to observations from simulation results. $\alpha_N(\rho)$ will have to be recalculated and hence resimulated for any change of service time distribution.

Harrison and Zertal derive a method for finding the maximum of multiple random variables [3]. This gives an approximation to a fork-join synchronisation, by modelling a similar network called the split-merge model exactly (figure 2). In the split-merge model, a job splits into N tasks which are serviced in parallel. Only when all the tasks finish servicing and rejoin can the next job split into tasks and start servicing. This will lead to a slower mean response times than its fork-join equivalent.

Let $f_n(\boldsymbol{\alpha}, t)$ be a probability density function that describes the maximum of n independent, negative exponential random variables, with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots \alpha_n)$. The following recurrence relation can be obtained for the Laplace transform of $f_n(\boldsymbol{\alpha}, t)$, $L_n(\boldsymbol{\alpha}, s)$.

$$(s + \sum_{j=1}^{m}\alpha_j)L_m(\boldsymbol{\alpha}, s) = \sum_{j=1}^{m}\alpha_j L_{m-1}(\boldsymbol{\alpha}_{\setminus j}, s) \quad s \geq 0 \tag{6}$$

for $1 \leq m \leq n$, where $\boldsymbol{\alpha}_{\setminus j} = (\alpha_1, \ldots \alpha_{j-1}, \alpha_{j+1}, \ldots \alpha_m)$, $L_0(\boldsymbol{\epsilon}, s) = 1$ and $\boldsymbol{\epsilon}$ is the zero vector.

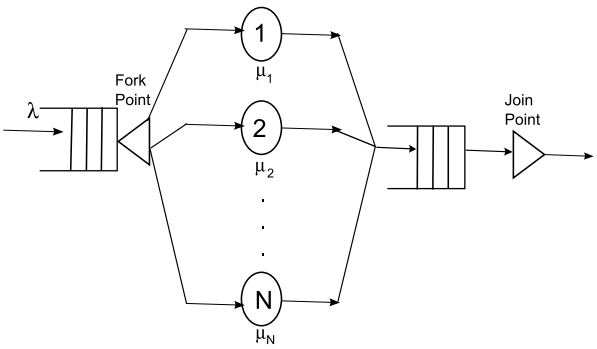

Figure 2: Split-merge queueing model

The kth moments, $M_n(\boldsymbol{\alpha}, k)$ for $f_n(\boldsymbol{\alpha}, t)$, can be derived by differentiating equation (6) k times using Leibnitz's theorem and setting s to 0. A recurrence relation for approximating the mean value of the maximum of n independent, non-negative random variables with means $\mathbf{m} = (m_1, \ldots, m_n)$ follows. $I(n, \boldsymbol{\alpha}, \mathbf{M})$ is the approximation function, where, $\boldsymbol{\alpha} = (m_1^{-1}, \ldots, m_n^{-1})$, second moments $\mathbf{M} = (M_1, \ldots, M_n)$ and recurrence relation for $k = 2, \ldots, n$,

$$I(k, \boldsymbol{\alpha}, \mathbf{M}) = \frac{1}{k} \sum_{i=1}^{k} I(k-1, \boldsymbol{\alpha}_{\setminus i}, \mathbf{M}_{\setminus i}) + \alpha_i M_i L_{k-1}(\boldsymbol{\alpha}_{\setminus i}, \alpha_i)/2 \tag{7}$$

$$I(1, \alpha_1, M_1) = 1/\alpha_1$$

The result is exact if the n random variables are exponentially distributed.

3 The Maximum Order Statistic

An alternative to Harrison and Zertal's method [3], is to find the mean of the maximum of a set of random variables by utilising the properties of Order Statistics [5, 1].

Definition Any random variables, $X_1, X_2, \ldots, X_n$ can be reordered as $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$, where $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$. Then $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$ are the order statistics of $X_1, X_2, \ldots, X_n$.

The maximum of n random variables, using order statistics is $X_{(n)}$, the maximum order statistic. The mean value of this maximum and further moments can be found if the cumulative distribution function (cdf) of $X_{(n)}$ is calculated.

$$F_{X_{(n)}}(x) = P(X_{(n)} \leq x) = \forall i P(X_{(i)} \leq x)$$

Thus, if $X_1, X_2, \ldots, X_n$ are iid with cdf $F(x)$,

$$F_{X_{(n)}}(x) = (F(x))^n$$

If the random variables are independent but not identically distributed, and X_i has cdf $F_i(x)$,

$$F_{X_{(n)}}(x) = \prod_{i=1}^{n} F_i(x)$$

The mean of the maximum of n independent random variables is then

$$E[X_{(n)}] = \int_{-\infty}^{\infty} x \left(\sum_{i=1}^{n} \frac{f_i(x)}{F_i(x)} \right) \prod_{i=1}^{n} F_i(x) dx \tag{8}$$

If the random variables are iid, equation (8) simplifies to

$$E[X_{(n)}] = n \int_{-\infty}^{\infty} x f(x) (F(x))^{(n-1)} dx \tag{9}$$

Further moments, M_k, can be calculated,

$$M_k = E[X^k_{(n)}] = n \int_{-\infty}^{\infty} x^k f(x)(F(x))^{(n-1)} dx$$

These results always give exact solutions to the mean of the maximum random variable, immaterial to the distribution of the random variables.

4 Results

To validate this analytical result, it is compared to simulation and analytical results from Harrison and Zertal [3]. The simulations were run 100,000 times, giving 98% confidence bands of the order 0.01.

The two analytical results are the same for exponential random variables, since equation (7) is exact for exponential random variables. Table 1 compares the two models and simulation results for an Erlang-k distribution, with parameter k. The column HZ contains the results from the approximation in [3] and OS contains the results using equation (9). The approximation suffers with low variance as $N \to \infty$, with a constantly increasing percentage error for larger N. Equation (9) consistently delivers better percentage errors with no clear performance deficits.

N	Exp-1	Erlang-2				
		Sim	HZ	% err	**OS**	% err
1	1.000	1.003	1.000	-0.334	**1.000**	-0.334
2	1.500	1.373	1.375	0.135	**1.375**	0.135
4	2.083	1.772	1.813	2.265	**1.774**	0.089
8	2.718	2.182	2.288	4.881	**2.180**	-0.078
16	3.381	2.588	2.786	7.648	**2.587**	-0.035

N	Erlang-3					Erlang-4				
	Sim	HZ	% err	**OS**	% err	Sim	HZ	% err	**OS**	% err
1	0.999	1.000	0.062	**1.000**	0.062	0.999	1.000	0.060	**1.000**	0.060
2	1.271	1.313	3.281	**1.313**	3.281	1.195	1.281	7.207	**1.273**	6.127
4	1.546	1.677	8.448	**1.630**	5.153	1.380	1.609	16.64	**1.544**	10.6230
8	1.806	2.074	14.84	**1.945**	7.147	1.555	1.966	26.43	**1.808**	13.993
16	2.061	2.488	20.74	**2.254**	8.562	1.716	2.339	36.30	**2.063**	16.820

Table 1: Comparison with Erlang (low-variance)

For a high variance situation, a Pareto distribution is used. Table 2 compares the models and simulation results for a heavy tailed Pareto-β distribution. This has cdf $F_P(x) = 1 - \alpha(x + \gamma)^{-\beta}$, with $\alpha = \gamma^\beta$ and $\gamma = \beta - 1$. The results are consistently better for the maximum order statistic result than the approximation and are also significantly more accurate than the low variance case.

N	Exp-1	Pareto-4					Pareto-5				
		Sim	HZ	% err	**OS**	% err	Sim	HZ	% err	**OS**	% err
1	1.000	1.004	1.000	-0.381	**1.000**	-0.381	0.994	1.000	0.614	**1.000**	0.614
2	1.500	1.579	1.750	10.82	**1.571**	-0.509	1.567	1.667	6.350	**1.556**	-0.707
4	2.083	2.327	2.625	12.81	**2.319**	-0.345	2.269	2.444	7.744	**2.266**	-0.132
8	2.718	3.261	3.577	9.698	**3.255**	-0.184	3.129	3.290	5.173	**3.129**	-0.001
16	3.381	4.394	4.571	4.027	**4.395**	0.023	4.153	4.174	0.512	**4.149**	-0.096

Table 2: Comparison with Pareto (high-variance)

4.1 M/M/1 Queues

Describing the mean response time as the mean of the maximum of a set of random variables used in equations (7) and (9) are compared to the other mean response time approximations described in Section 2. All these approximations only apply to M/M/1 queues. Since M/M/1 queues have both exponential arrival and service time distributions, the approximation in equation (7) is exact and yields the same results as equation (9). Figure 3 compares the mean response time for a fork-join network of M/M/1 queues using the method of finding the mean of the slowest queue (OS), with three of the response time approximations described in Section 2. The results were calculated with a mean arrival rate, $\lambda = 1$ and a mean service rate, $\mu = 1.1$ for each server and the number of servers varying between 1 and 25. Equation (1) is the line NT,

equation (2) is VM and equation (3) is VMC. Equation (2) has significantly worse performance results than all the other approximations, due to the presence of a double sum. All these results are compared to a simulation of the network, with the line labelled SIM. The simulation is run 100,000 times and then each run is replicated 30 times, with either the mean or median result chosen, depending on the range and frequency of results. All further simulations in this paper are carried out with these standards applied, using the JINQS queueing network simulator [2].

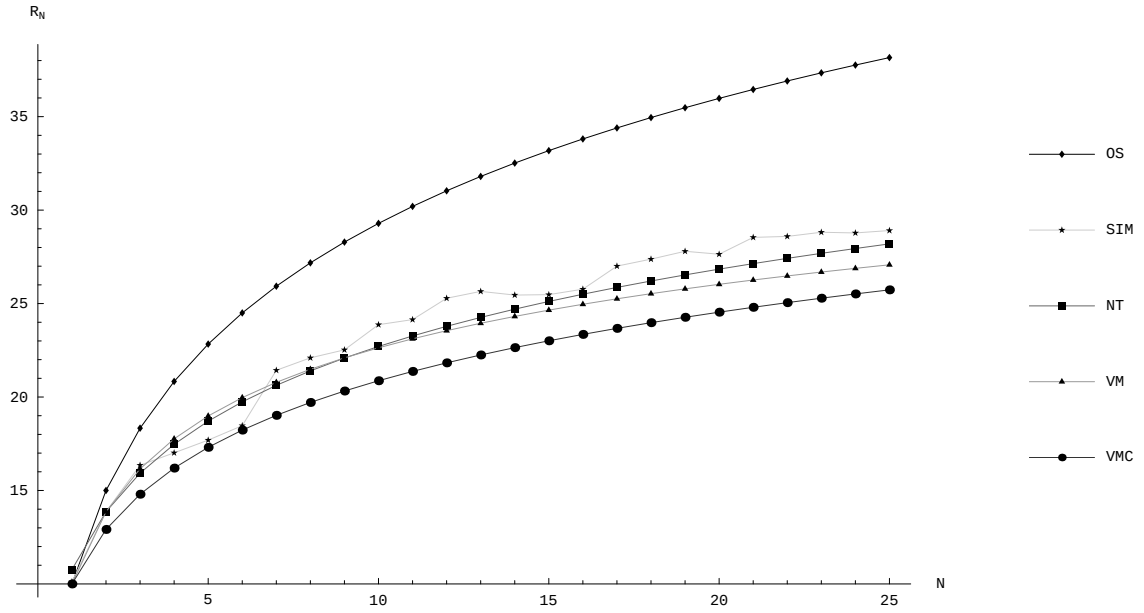

Figure 3: Mean response time R_N for M/M/1 fork join-queue with N queues, $\lambda = 1$, $\mu = 1.1$

The mean of the maximum order statistic, which gives exact results for a split-merge queue but only approximates the fork-join model, performs worst out of all the approximations for the M/M/1 queue. This could be expected as the split-merge model waits for all parallel servers to finish servicing before a new job begins service and will hence be significantly slower than the fork-join model. However, although the accuracy of this result is worse than other models, its strengths are in its potential. The three approximations it is compared with can only approximate fork-join queues that consist of homogeneous M/M/1 queues. This result will therefore perform better with M/G/1 queues or fork-join queues with heterogeneous servers. Additionally, the maximum order statistic provides a useful and easily computable response time upper bound.

4.2 M/G/1 queues

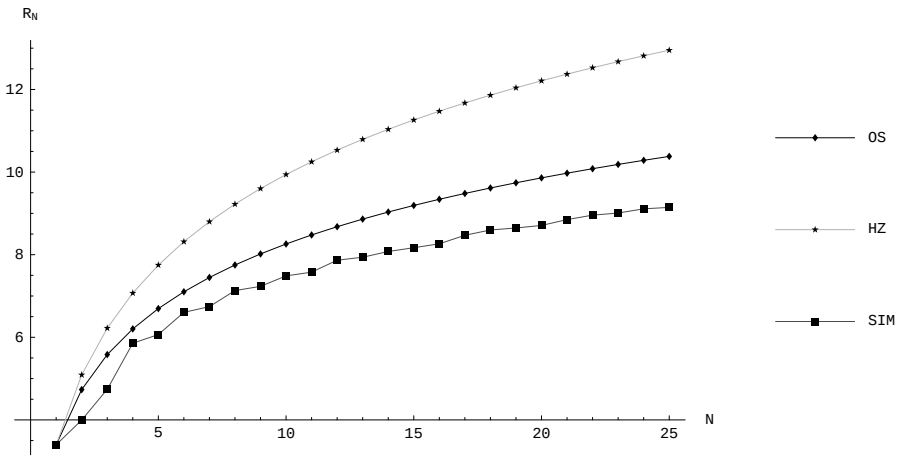

Figure 4: Mean response time R_N for Erlang-2 M/G/1 fork-join queue with N queues, $\lambda = 0.1$, $\mu = 0.375$

The benefits of this result become more apparent with an M/G/1 queue. The maximum order statistic is calculated by inverting the Laplace transform of the response time for each individual queue using mathematical software. The approx-

imations defined for M/M/1 queues (equations (1), (2) and (3)) only apply for M/M/1 queues and the only approximations that exist for M/G/1 queues are computationally intensive [9], or reliant on simulation results (equation (5)). Harrison and Zertal's method is an approximation to the split-merge queue, whereas this result is exact for the split-merge queue.

Figure 4 plots mean response time for an N server fork-join queue. The service time distribution has an Erlang-2 distribution with mean 0.375 and arrival rate 0.1. The graph compares Harrison and Zertal (HZ), the mean of the maximum order statistic (OS) and a simulation (SIM).

4.2.1 A Large Number of Parallel Queues

Disk arrays often consist of up to fifty individual disk drives. Any analytical approximation of a disk array needs to quickly and accurately calculate the mean response time as the number of parallel queues increases. Finding the mean of the maximum order statistic is computationally fast for a large number of parallel queues. However, simulating large fork-join queues is very slow. Therefore, figures 3 and 4 only show results up to 25 disks. To show how these results compare as the number of queues gets very large, tables are presented for the cases when there are 40 and 50 parallel queues. Table 3 shows mean response times for the M/M/1 fork-join queue described above with arrival rate 1 and service rate 1.1. Table 4 displays mean response times for the M/G/1 fork-join queue with Erlang-2 distributed service times, arrival rate 0.1 and service rate 0.375. Both tables are labelled with the same key for the results as figures 3 and 4.

N	Simulation	Confidence Interval half width	OS	NT	VM	VMC
40	32.195	1.201	42.785	31.055	29.196	28.263
50	32.450	0.684	44.992	32.42	30.171	29.469

Table 3: M/M/1 parallel queues with many servers

N	Simulation	Confidence Interval half width	OS	HZ
40	10.0126	0.0160	11.481	14.521
50	10.406	0.0178	12.0054	15.27

Table 4: Erlang-2 M/G/1 parallel queues with many servers

4.3 Heterogeneous Servers

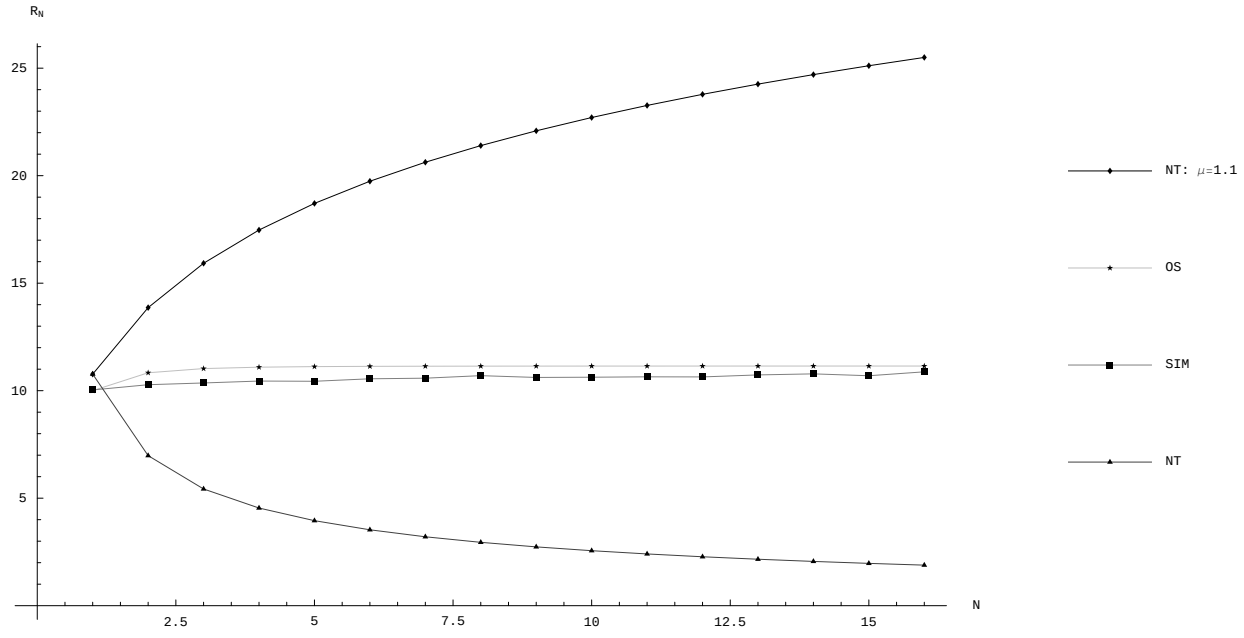

Figure 5: Mean response time R_N for an heterogeneous M/M/1 fork-join queue with N queues

Heterogeneous parallel servers in a fork-join queue create a situation in which approximating the response time with the maximum order statistic is an improvement upon other analytical approximations for fork-join synchronisation. The

other fork-join approximations discussed in this paper are only applicable for homogeneous servers. Figure 5 measures the mean response time for an M/M/1 fork-join queue with heterogeneous servers. It plots the mean response time for an N branch fork-join queue in which each server has a mean service rate of $1.1 + 0.2i$, where $i = 0, 1, \ldots, N-1$. The line SIM represents a simulation for $N = 1, \ldots, 16$ and the line OS is an approximation using the mean of the maximum order statistic. To show that the approximations for fork-join queues with homogeneous servers cannot approximate the heterogeneous result, two lines are plotted assuming homogeneous servers, using Nelson and Tantawi's approximation (see Equation (1)), which was shown in figure 3 to be the most accurate analytical approximation of M/M/1 fork-join synchronisation. Firstly, we approximate the heterogeneous servers by assuming homogenous servers with the minimum, and hence the slowest service rate, 1.1. This is displayed in the line NT $\mu = 1.1$. Secondly in line NT, we define the service rate of the homogenous servers as the mean of all the service rates on the heterogeneous servers.

The results using the order statistics method stay consistently closer to the fork-join simulation than in the homogenous case. Furthermore, both attempts at approximating parallel systems with heterogeneous servers by modelling them as homogenous servers have increasingly large percentage errors as N increases.

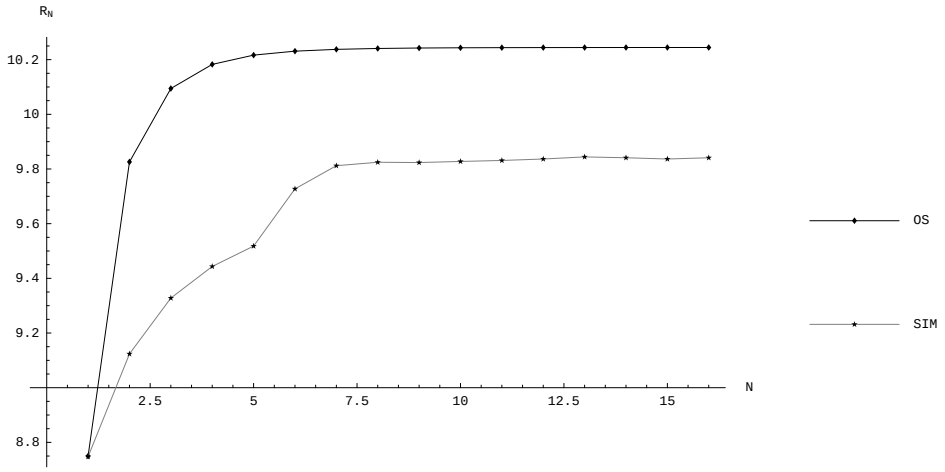

Figure 6: Mean response time R_N for an heterogeneous M/G/1 fork-join queue with N queues and an Erlang-2 service time distribution, with mean $0.2 + 0.1N$

Figures 6 and 7 compare simulation results with the mean of the maximum order statistic for M/G/1 heterogeneous fork-join queues. Figure 6 charts the mean response time for N M/G/1 queues with an Erlang-2 distribution, but with a mean service rate that varies according to N ($\mu = 0.2 + 0.1N$). Figure 7 keeps the mean service rate constant at 0.375, but varies the service time distribution according to N. The service time distribution is Erlang-$N + 1$.

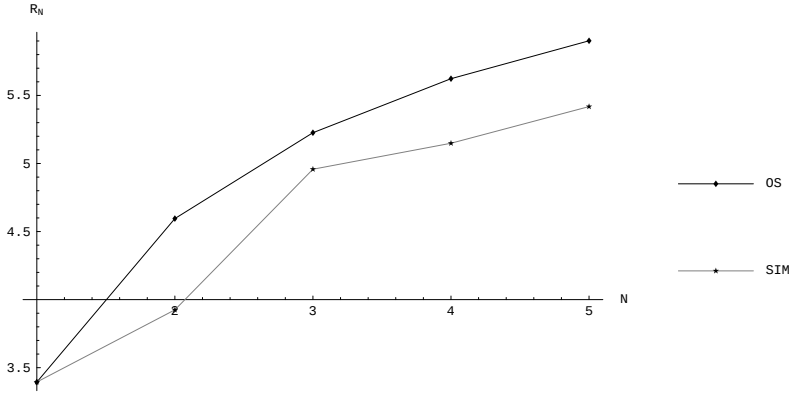

Figure 7: Mean response time R_N for an heterogeneous M/G/1 fork-join queue with N queues and a service time distribution of Erlang-$N + 1$,with mean 0.375

In figures 5 and 6, the mean response time tends to a constant value as N increases. This is because queues are added to the network with increasingly fast mean response times. The slow response times of the queues initially added to the network have a larger effect on the overall mean response time of the fork-join network.

5 Conclusion and Future Work

This paper discusses existing analytical solutions to the fork-join network. We present an approximation using order statistics and compare it to the existing methods. The use of the maximum order statistic enables a number of useful features to the result, not available with the other approximations. The other approximations only calculate a value for the mean response time, whereas using the result presented here, the response time density is derived, enabling not just the mean, but all further moments and other statistical measures to be calculated. This approximation enables the features needed for a performance model of a disk array.

The work presented above for solving fork-join synchronisation analytically can be improved. The approximation discussed above is a pessimistic model of fork-join synchronisation. This raises the possibility of scaling the result down to approximate fork-join synchronisation better. Thomasian and Tantawi [6] use the properties of order statistics to scale their result; however this is done by first simulating the fork-join queue with the specified service time distribution. Our aim is to scale the result without using simulation results. This would create a simple, fast and accurate approximation for M/G/1 fork-join queues which would create a good performance model for disk arrays.

Other assumptions made in this fork-join approximation need to be considered. At present, the fork-join model splits each job into exactly as many sub-tasks as there are servers. In disk arrays, however, an arriving job, or I/O request, will split into more or less tasks than the number of servers or disks. Additionally, throughout this paper we assume a constant Markovian arrival rate to the fork-join queue. The I/O request stream that sends requests to a disk array is unlikely to have a constant arrival rate and would be better modelled with MMPP (Markov Modulated Poisson Process) arrivals. The approximation discussed in this paper needs to be extended and tested to comply with these requirements.

Acknowledgements

The authors would like to thank Peter Harrison for his advice and contribution to this work.

References

[1] H. A. David. *Order Statistics*. John Wiley and Sons, Inc, 1981.

[2] A. J. Field. *JINQS: An Extensible Library for Simulating Multiclass Queueing Networks*. Imperial College London, August 2006.

[3] P. G. Harrison and S. Zertal. Queueing models with maxima of service times. In *Proc. TOOLS Conference*, pages 152–168, 2003.

[4] R. Nelson and A. N. Tantawi. Approximate analysis of fork/join synchronization in parallel queues. *IEEE Transactions on Computers*, 37(6):739–743, June 1988.

[5] R. A. Sahner and K. S. Trivedi. Performance and reliability analysis using directed acyclic graphs. *IEEE Transactions on Software Engineering*, 13(10):1105–1114, 1987.

[6] A. Thomasian and A. N. Tantawi. Approximate solutions for M/G/1 fork/join synchronization. In *Proc. of the 26th conference on Winter simulation (WSC '94)*, pages 361–368, San Diego, CA, USA, 1994. Society for Computer Simulation International.

[7] E. Varki. Mean value technique for closed fork-join networks. In *Proc. ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 103–112, 1999.

[8] E. Varki, A. Merchant, and H. Chen. The M/M/1 fork-join queue with variable sub-tasks. Unpublished.

[9] S. Varma and A. M. Makowski. Interpolation approximations for symmetric fork-join queues. In *Proc. of the 16th IFIP Working Group 7.3 international symposium on Computer performance modeling measurement and evaluation (Performance '93)*, pages 245–265, 1994.