Validation of Large Zoned RAID Systems

Abigail S. Lebrecht Nicholas J. Dingle William J. Knottenbelt *

Abstract

Building on our prior work in [8], we present an improved model for for large partial stripe following full stripe writes in RAID 5. This was necessary because we observed that our previous model tended to underestimate measured results. To date, we have only validated these models against RAID systems with at most four disks. Here we validate our improved model, and also our existing models for other read and write configurations, against measurements taken from an eight disk RAID array.

1 Introduction

Choice of RAID level can critically affect the performance delivered by a storage system. Therefore the ability to predict RAID performance for RAID systems of different sizes and configurations is crucial. Modern Service Level Agreements require effective performance predictions must provide the ability to reason not only about mean response times, but also higher moments and percentiles of response time.

Previous RAID models [4, 7, 11, 16, 17] approximate only the mean response time of the system. All RAID models that we develop approximate the full response time distribution, from which moments and percentiles can be calculated. In [9], we introduced analytical queueing network models of RAID 01 and 5, the two most commonly used RAID levels. We extended these models in [8] through the use of a multiclass queueing network to allow heterogeneous workload streams of read and write requests. In both cases, we validated these models against device measurements from a real-life RAID system. This demonstrated the accuracy of our models and also suggested some areas in which they could be more representative.

In this paper, we present improvements to our existing RAID 5 models for those cases where large partial stripe writes follow one or more full stripe writes. We demonstrate the accuracy of this model and our other models [9, 8] by validating them against a real eight disk RAID system. This is an improvement over [9], where RAID 01 was only validated for one size of request (2 blocks) under the same load on a 4 disk array. Furthermore, constraints of validating on a four disk array meant that there was only one validation for each size of RAID 5 write request (small partial, large partial and full-stripe). On an eight disk array there are three different configurations representing each of small and large partial stripe writes.

In [8] the model was extended to allow mixed streams of read and write requests. Specifically we studied the cases of a request stream made up of the same amount of read and write requests, and weighted in either direction with proportions of 75% and

*Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2BZ, United Kingdom. Email: {asl102,njd200,wjk}@doc.ic.ac.uk

25% for each. We also studied the four disk array under different loads and for a wide range of request sizes for all types of arrival streams and observed certain trends that we discussed in that paper. By carrying out similar validations on an eight disk array, we hope to see to what degree the previous observations were anomalous, and to discuss possible reasons and model changes if the trends are still visible.

The remainder of this paper is organised as follows: in Section 2 we include a summary of our existing models, before presenting the improved RAID 5 large partial stripe write model in Section 3. We then validate our models more comprehensively by comparing the analytical results with measurements from an eight disk array for a variety of request sizes in Section 4.

2 RAID Model

Our RAID model is developed in a bottom-up hierarchical fashion. We begin by modelling each disk drive in the array as a single M/G/1 queue. An important subtlety that needs to be taken into account in the service time distribution is that modern disks are zoned, with more sectors on the outer tracks than inner tracks. Therefore, a random request is more likely to be directed to a sector on an outer track, and it is also faster to transfer data on a track closer to the circumference than the centre of the disk.

The service time density of an access to a random location on a single zoned disk is the convolution of the seek time, rotational latency and data transfer time probability density functions. We use the seek time and rotational latency distributions defined in [19] and the data transfer time distribution from [9]. We denote the random variables of seek time, rotational latency and k-block transfer time as S, R and T_k respectively. The response time distribution of the M/G/1 queue is obtained by numerically inverting [1] the corresponding Pollaczek-Khintchine transform equation [6].

We then abstract the RAID as a fork-join queueing network [3] of M/G/1 queues. In an N-queue fork-join network, (see Fig. 1), each incoming job is split into N subtasks at the fork point. Each of these subtasks queues for service at a parallel service node before joining a queue for the join point. When all N subtasks in the job are at the head of their respective join queues, they rejoin (synchronise) at the join point.

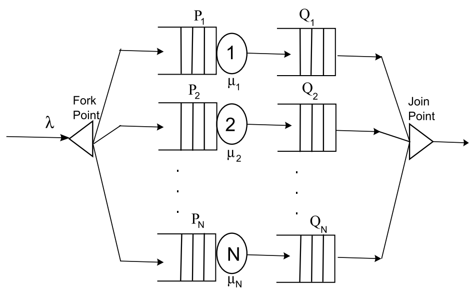


Figure 1: Fork-join queueing model

It is difficult, however, to model job response times in a fork-join synchronisation analytically. Indeed, exact analytical results only exist for the mean response time of a two server system consisting of homogeneous M/M/1 queues [12]. Approximations for mean response times for M/M/1 and M/G/1 fork-join queues are more abundant [12, 14, 15, 16, 18] but such results do not permit higher moments or full response time distributions to be calculated. Therefore, we have previously presented [9] an approach using the maximum order statistic [5, 10] to derive an approximation to the cumulative distribution function of a fork-join queue's response time. This was inspired by [7], which defined an approximation of the fork-join queue that enables the calculation of both the mean and further moments of response time.

The standard fork-join network directly models the behaviour of a RAID system in only a small number of cases (e.g. full stripe I/O operations in RAID 0). Consequently, the fork-join model must be tailored to support the full range of I/O access patterns that occur when performing read or write operations of different sizes on different RAID levels. Our initial model is designed to accept a homogeneous stream of I/O requests of a given size and type (RAID 01 or 5, read or write). We further assume that all the service time distributions on all disks are identically distributed. For the sake of notational simplicity, let $W_d(t, \gamma, \frac{1}{\mu})$ define the cdf of the response time distribution of a single M/G/1 queue (disk), γ is the arrival rate at an individual disk and μ is the mean service rate. We assume there are n disks in the array and that the arrival rate of logical I/O requests to the disk array as a whole is λ. In [9], these models are introduced and we summarise them here.

The cdf of the response time for a b-block read from RAID 01 is:

$$W_{\text{read}}(t) = \begin{cases} \left(W_d\left(t, \frac{\lambda b}{n}, E[R] + E[S] + E[T_1]\right)\right)^b & \text{if } b < n \\ \left(W_d\left(t, \lambda, E[R] + E[S] + E[T_{\frac{b}{n}}]\right)\right)^n & \text{otherwise} \end{cases}$$

Similarly, the cdf of the response time of a b-block mirrored write to RAID 01 is:

$$W_{\text{write}}(t) = \begin{cases} \left(W_d\left(t, \frac{2\lambda b}{n}, E[R] + E[S] + E[T_1]\right)\right)^{2b} & \text{if } 2b < n \\ \left(W_d\left(t, \lambda, E[R] + E[S] + E[T_{\frac{2b}{n}}]\right)\right)^n & \text{otherwise} \end{cases}$$

The cdf of the response time for a b-block read from RAID 5 is:

$$W_{\text{read}}(t) = \begin{cases} \left(W_d\left(t, \frac{\lambda b}{n}, E[R] + E[S] + E[T_1]\right)\right)^b & \text{if } b < n \\ \left(W_d\left(t, \lambda, E[R] + E[S] + E[T_{\frac{b}{n}}]\right)\right)^n & \text{otherwise} \end{cases}$$

Due to parity calculation a RAID 5 write request is modelled differently for different sized requests. The simplest RAID 5 write request is one which consists only of a number of complete stripes (i.e. $b \bmod (n-1) = 0$). In this case, computation of parity does not require pre-reading of existing data and so the only operation is to write to all disks. The cdf of request response time is therefore defined as:

$$W_{\text{write}}(t) = \left(W_d\left(t, \lambda, E[R] + E[S] + E[T_{\frac{b}{n-1}}]\right)\right)^n$$

A write request involving a partial stripe write will consist of two separate requests. The first is a pre-read for the calculation of the new parity. Then when all the parity pre-reads are completed and the new parity calculated, the second request, a partial stripe and new parity write request, is issued. Therefore, for partial stripe writes, we define a mean service time and density as the average of the service time (mean or density) of the parity pre-read and write request that follows.We note that these two subtasks are not independent. Indeed, we assume that they are highly dependent, and therefore the overall response time of a write request will be:

$$W_{\text{write}}(t) = P(2W \le t) = P\left(W \le \left(\frac{t}{2}\right)\right)$$

If a request consists of $b < \frac{n-1}{2}$ blocks (i.e. a small partial stripe write), the pre-read involves reading the old parity and data that will be replaced for parity calculation, then writing the new data to the same disks. The write will start after the last pre-read completes, so one disk will need to complete a full rotation (R_{max}) to return to the same sector. The overall response time cdf is therefore:

$$W_{\mathrm{write}}(t) \quad = \quad \left(W_d\left(\frac{t}{2}, \frac{2\lambda(b+1)}{n}, \frac{(2b+1)(E[R]+E[S])+R_{max}}{2(b+1)} + E[T_1]\right)\right)^{b+1}$$

The pdfs of both seek time and rotational latency must then be scaled accordingly to conform to the mean service times above:

$$f'(t) = \begin{cases} \frac{1}{2(b+1)} & \text{if } t = 0 \\ \frac{2b+1}{2(b+1)} f(t) & \text{otherwise} \end{cases}$$

where $f(t)$ is the probability density function of seek time or rotational latency.

For a large partial stripe write, $\frac{n-1}{2} \leq b < n-1$, the new parity is calculated by pre-reading from the disks that will not be written to and XORing it with the new data. Therefore at some point in the request each disk in the array is written to. The cdf of request response time is:

$$W_{\mathrm{write}}(t) = \left(W_d\left(\frac{t}{2}, \lambda, E[R]+E[S]+E[T_1]\right)\right)^{n/2}$$

If $b > n-1$ and $0 < b \bmod (n-1) < \frac{n-1}{2}$, at least one full stripe write will occur followed by a small partial stripe write. The first request consists of the full stripe writes and the pre-read, and the second is the partial stripe write. Let $b_{mod} = b \bmod (n-1)$. The cdf of request response time is:

$$W_{\mathrm{write}}(t) \quad = \quad \left(W_d\left(\frac{t}{2}, \frac{\lambda(n+b_{mod}+1)}{n}, \frac{(n+b_{mod})(E[R]+E[S])+R_{max}}{n+b_{mod}+1} + E[T_{\frac{k}{2}+\frac{b_{mod}+1}{n}}]\right)\right)^{\frac{n+b_{mod}+1}{2}}$$

In [9], our RAID models assumed homogeneous arrival streams. In [8] we used multiclass queues to generalise these models for heterogeneous streams composed of both reads and writes. The resulting request response time cdf of a RAID system with a mixed arrival stream of read and write requests is defined as:

$$W(x) = p_{read}W_{\mathrm{read}}(x) + (1-p_{read})W_{\mathrm{write}}(x)$$

where p_{read} is the probability that a request is a read.

We note that the arrival rate to the disk array defined for each type of request must be modified to take the combined stream into account. For RAID 01 the arrival rate at each disk becomes:

$$\frac{\lambda(p_{read}\min(b,n) + (1-p_{read})\min(2b,n))}{n}$$

On RAID 5, the arrival rate at each disk is:

$$p_{read}\lambda\frac{\min(b,n)}{n} + (1-p_{read})\gamma$$

where γ is the arrival rate at each disk in the array in the case that $p_{read} = 0$.

3 Improved Large Partial Stripe Write Model

Our validation work in [8] suggested that the model for large partial stripe following full stripe writes (where $\frac{n-1}{2} \leq b \bmod (n-1) < n-1$) could be improved as it tended to underestimate the measured results. Furthermore, on the eight disk array the measurements showed that as the size of the partial stripe increases the mean response time decreases, whereas in the analytical model [9] mean response time increases.

In order to improve the analytical model we analysed the physical behaviour on the disk when a large partial stripe write follows a full stripe write. Specifically, we focused on the amount of seeking each disk must do between the time that a partial stripe parity pre-read completes and the partial stripe write begins. The fewer disks that are pre-read ($n - b_{mod} - 1$), the more likely that the pre-read will complete before the remaining $b_{mod} + 1$ disks complete their respective full stripe writes. If any of the $b_{mod} + 1$ disks complete before the pre-read has completed servicing then that disk must wait to write the new data or parity. In this time, that disk may start servicing the next request in its queue, or just rotate away from the desired position. Henceforth, when the pre-read eventually completes, those disks will have to re-seek back causing additional seek and rotational latency. However, if the pre-read completes first then, when another disk completes its full stripe write, it can immediately write the new data or parity for the large partial stripe write without any additional seeking. We accordingly approximate the probability of the $b_{mod} + 1$ disks having to seek as $\frac{n-b_{mod}-1}{n}$.

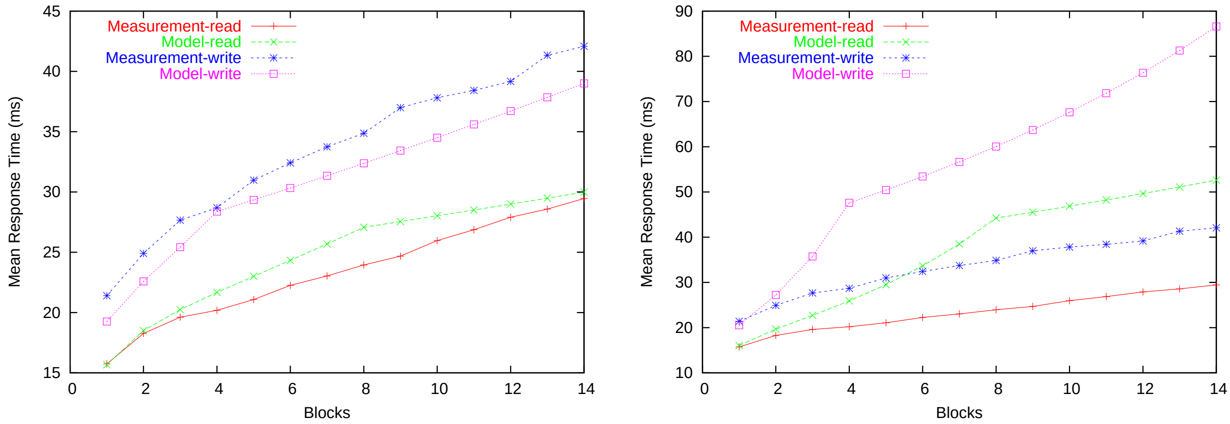
However, as the number of full stripes written increases this relationship becomes less relevant. This is because each disk will take different amounts of time to write the (larger amount of) full stripe data and the additional pre-read time on some disks will be insignificant in comparison. The effect of zoning amplifies these differences. As the number of full stripes (k) increases, the disk that finishes first is less likely to depend on whether there was an additional pre-read on that disk, and it is more likely that all the disks will need to re-seek. Therefore, we define the probability of seeking as $1 - \frac{b_{mod}-1}{nk}$. Since all disks have to seek initially for the start of the full-stripe write, the mean seek time becomes $\left(1 - \frac{b_{mod}-1}{2nk}\right)(E[R] + E[S])$. All other parameters in the model remain the same as the previous model, so the cdf or request response time is:

$$W_{\text{write}}(t) = \left(W_d \left(\frac{t}{2}, \frac{\lambda(n + b_{mod} + 1)}{n}, \left(1 - \frac{b_{mod} - 1}{2nk}\right)(E[R] + E[S]) + E[T_{\frac{k+1}{2}}] \right) \right)^{\frac{n+b_{mod}+1}{2}}$$

4 Validation

We demonstrate the accuracy of our models by validating them against a real eight disk RAID system. This is an improvement over [9], where RAID 01 was only validated for one size of request (2 blocks) on a four disk array, and also over [8] where both our RAID 01 and RAID 5 models were only validated against a four disk system.

Our experimental platform consists of an Infortrend A16F-G2430 RAID system containing eight Seagate ST3500630NS disks. Each disk has 60801 cylinders. A sector is 512 bytes and we have approximated, based on measurements from the disk drive, that the time to write a single physical sector on the innermost and outermost tracks are 0.012064ms (t_{max}) and 0.005976ms (t_{min}) respectively. The stripe width on the array is configured as 128KB, which we define as the block size. Therefore there are



(a) $\lambda = 0.01$ request/ms

(b) $\lambda = 0.03$ requests/ms

Figure 2: Comparison of mean response time against block size for RAID 01 for different values of λ.

256 sectors per block. The time for a full disk revolution is 8.33ms. A track to track seek takes 0.8ms and a full-stroke seek requires 17ms for a read request; the same measurements are 1ms and 18ms respectively for write requests [13].

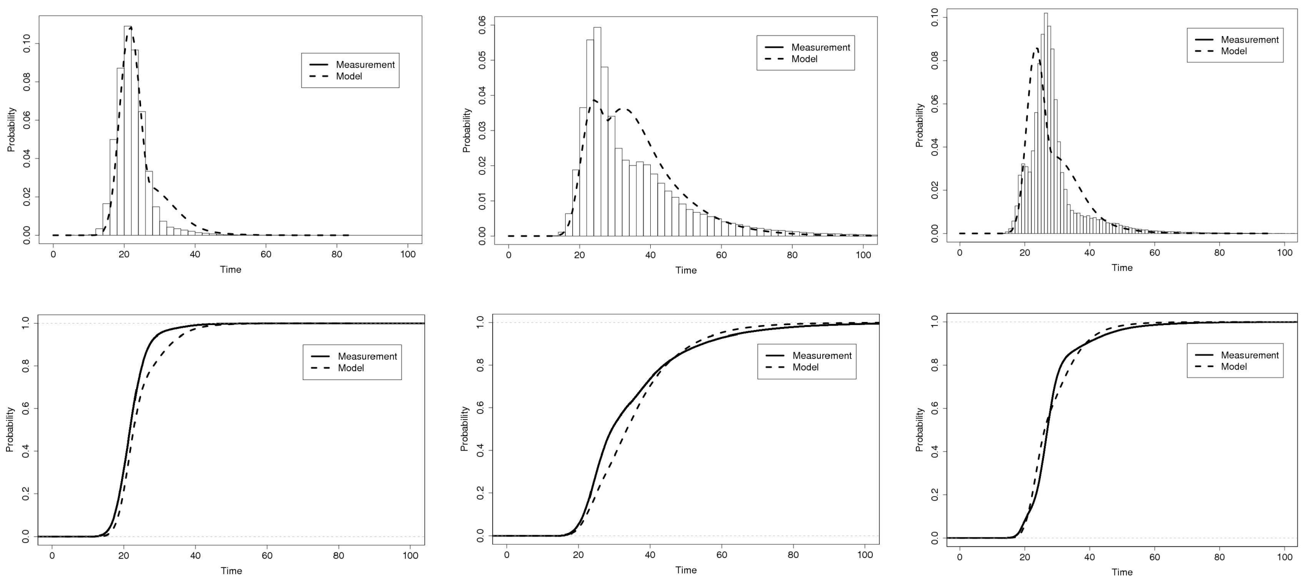
To obtain response time measurements from this system, we implemented a benchmarking program that issues read and write requests using a master process and a number of child processes. These child processes are responsible for issuing and timing I/O requests, leaving the master free to spawn further processes without the need for it to wait for previously-issued operations to complete.

In order to validate the analytical model effectively, it was necessary to minimise the effects of buffering and caching as these are not currently represented in the model. We therefore disabled the RAID system's write-back cache, set the read-ahead buffer to 0 and opened the device with the `O_DIRECT` flag set. For each of the experiments presented below, 100 000 requests were issued and the resulting means, variances, pdfs and cdfs of the response times were calculated using the statistical package `R`.

4.1 RAID 01

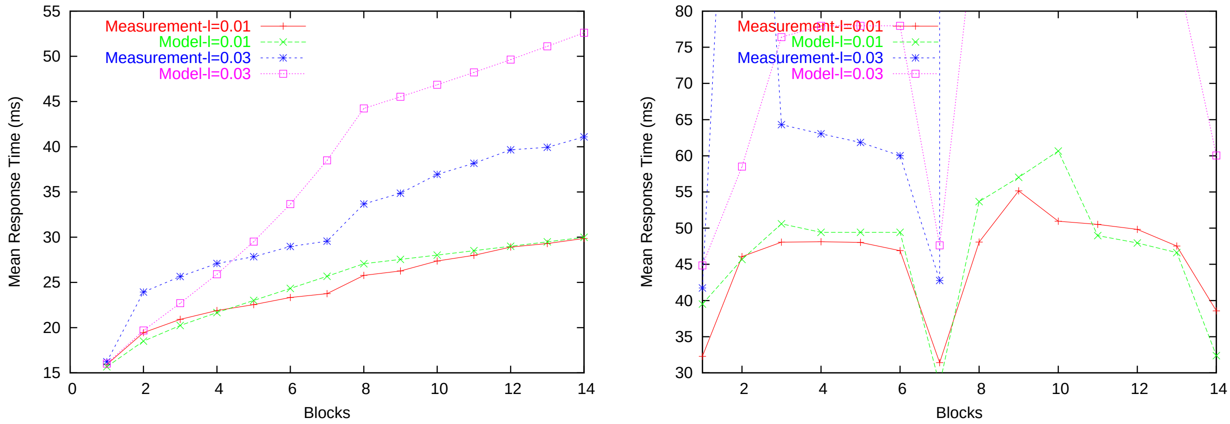
Fig. 2 shows measured and modelled mean response times of reads and writes for RAID 01 for two different values of λ – a light load of $\lambda = 0.01$ requests/ms (Fig. 2(a)) and a heavy load of $\lambda = 0.03$ requests/ms (Fig. 2(b)). For requests under a light load, agreement between model and measurement is excellent. However under heavy load the model tends to increasingly overestimate. The RAID controller re-orders jobs in a queue for optimal performance [2], so a longer queue enables more re-ordering. This is not represented in our model yet, hence the disparity between model and measurement.

Fig. 3 compares pdfs and cdfs for some randomly chosen parameters. Interestingly, even if the measured and modelled cdfs do not have excellent agreement, their pdfs show some similar trends.



(a) 100% read requests, $b = 6$, $\lambda = 0.01$ (b) 100% write requests, $b = 3$, $\lambda = 0.03$ (c) 100% write requests, $b = 4$, $\lambda = 0.01$

Figure 3: RAID 01 b-block request response time pdfs and cdfs for arrival streams of reads or writes with rate λ requests/ms.



(a) RAID 5 read requests

(b) RAID 5 write requests

Figure 4: Comparison of mean response time against block size for RAID 5 for different values of λ.

4.2 RAID 5

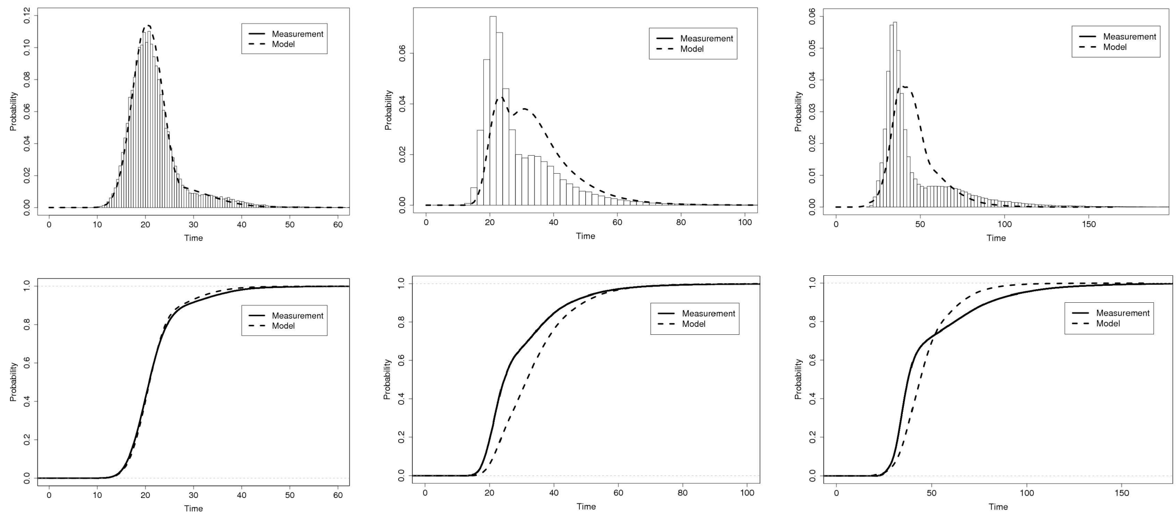
Fig. 4(a) shows measured and modelled mean response times for RAID 5 reads under light and heavy loads. Similar to its RAID 01 equivalents, agreement is excellent for light load, but under heavier load for larger block sizes, the model increasingly overestimates the measurements. Fig. 4(b) shows measured and modelled results for RAID 5 writes under light and heavy loads. The dips for both measurement and model at 7 and 14 blocks occur because these are full stripe writes with no slow parity pre-reads. For light load there is good agreement between model and measurement. For a heavier load, both measurement and model quickly show signs of saturation.

Fig. 5 compares pdfs and cdfs for some randomly chosen parameters. The modelled pdf in Fig. 5(b) displays the bimodal nature of the measured result, but not the peak of the maximum value.

4.3 Mixed Reads and Writes

Fig. 6 shows measured and modelled mean response times for arrival streams with varying proportions of reads and writes for RAID 01 for two different values of λ (0.01 and 0.03), while Fig. 8 shows the same for RAID 5. In both cases, we again observe good agreement between measured and modelled results. Fig. 7 displays a selection of full pdf and cdf results for RAID 01 mixed reads and writes, while Fig. 9 contains a further selection for RAID 5. Particularly noteworthy is Fig. 9(a), in which the model accurately captures the bimodal distribution of the measured results.

We were particularly interested in determining if some apparently spurious measurement results in [8] could be reproduced on the eight disk array. For RAID 5 mixed reads and writes we observed extremely long mean response times for 2-blocks requests for all three mixes (25% reads, 50% reads and 75% reads) which were much larger than the times for 2-block reads or 2-block writes and were not predicted by the model. In Table 4 we again observe this phenomenon, and indeed see that it is even

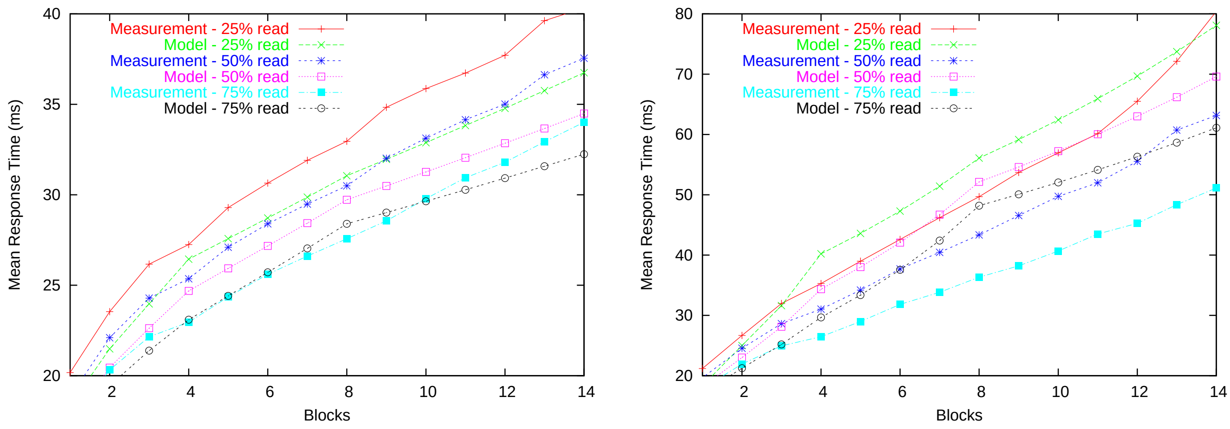


(a) 100% read requests, $b = 4$, $\lambda = 0.01$

(b) 100% read requests, $b = 6$, $\lambda = 0.03$

(c) 100% write requests, $b = 13$, $\lambda = 0.01$

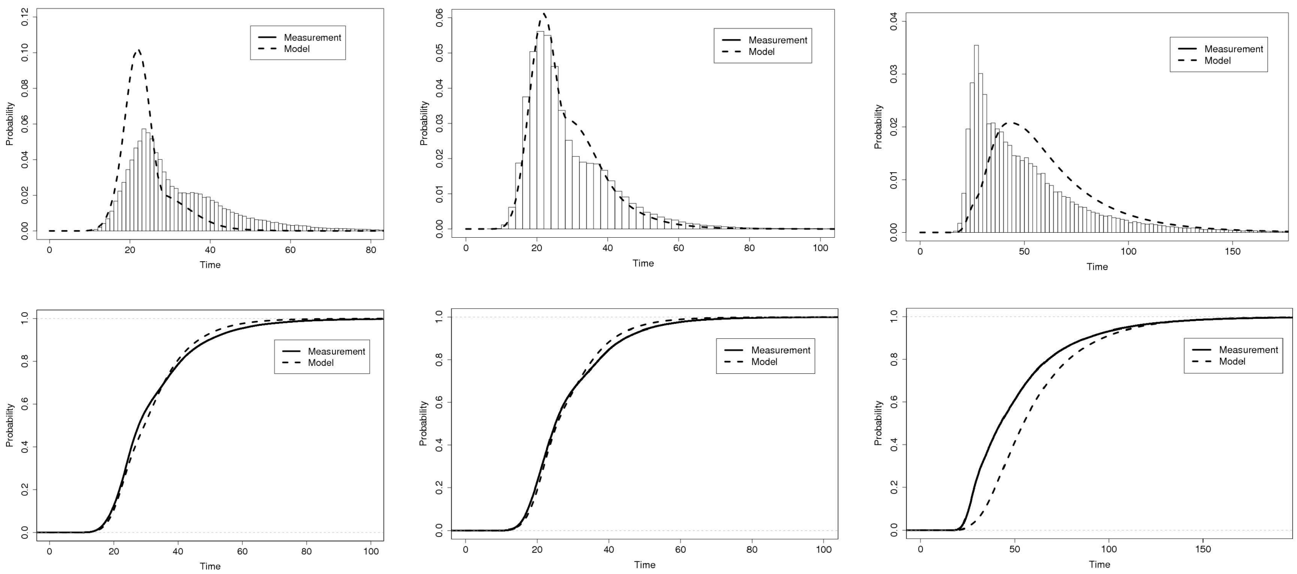
Figure 5: RAID 5 b-block request response time pdfs and cdfs for arrival streams of reads or writes with rate λ requests/ms.



(a) $\lambda = 0.01$ request/ms

(b) $\lambda = 0.03$ requests/ms

Figure 6: Comparison of mean response time against block size for RAID 01 for mixed arrival streams with different values of λ.

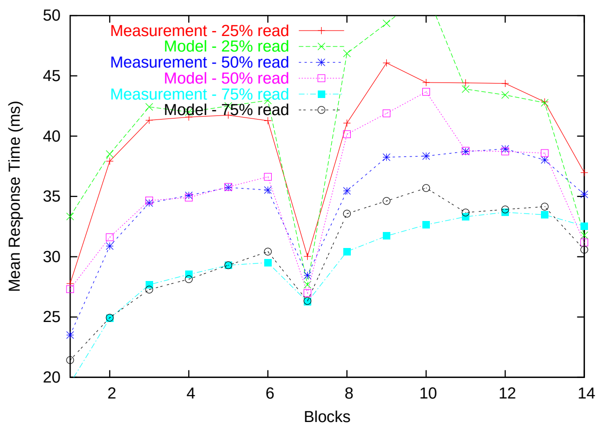


(a) 25% read requests, 75% write requests, $b = 3$, $\lambda = 0.03$

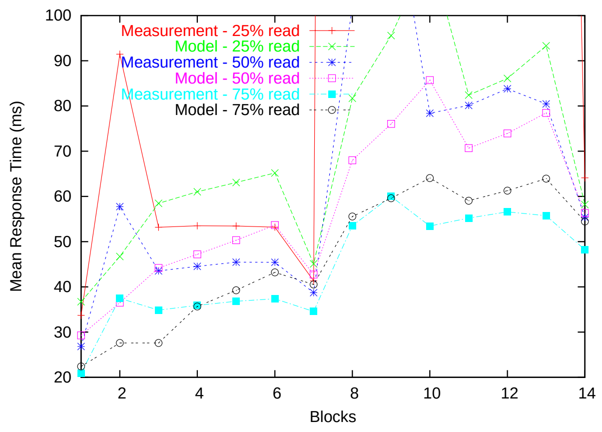
(b) 50% read requests, 50% write requests, $b = 3$, $\lambda = 0.03$

(c) 75% read requests, 25% write requests, $b = 14$, $\lambda = 0.03$

Figure 7: RAID 01 b-block request response time pdfs and cdfs for arrival streams of mixed reads and writes with rate λ requests/ms.

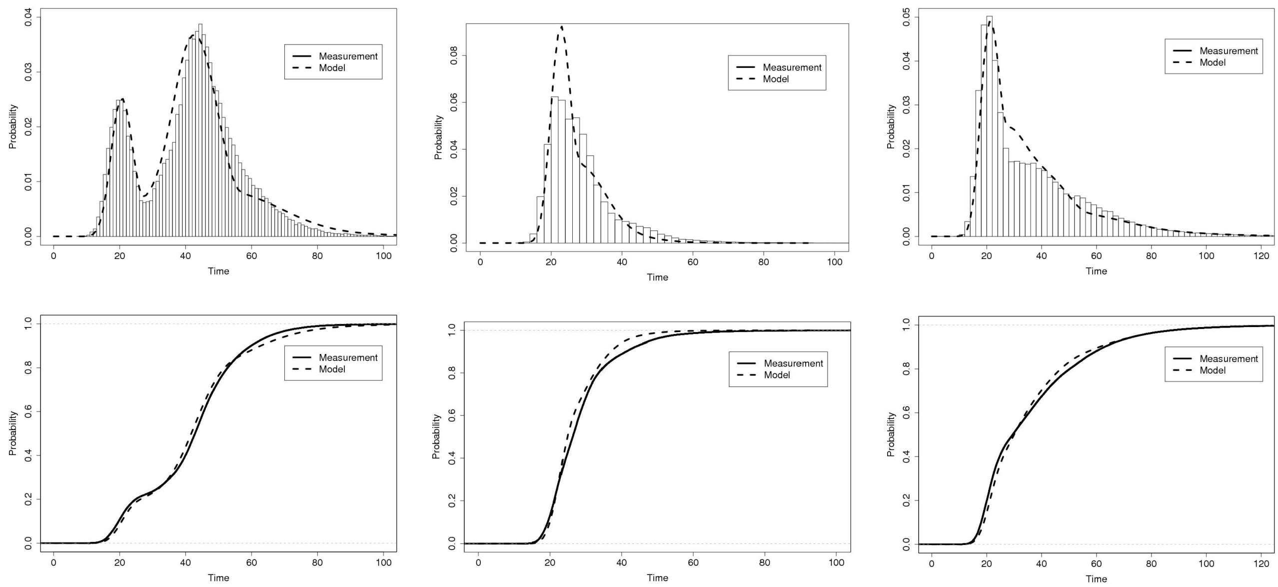


(a) $\lambda = 0.01$ request/ms



(b) $\lambda = 0.03$ requests/ms

Figure 8: Comparison of mean response time against block size for RAID 5 for mixed arrival streams with different values of λ.



(a) 25% read requests, 75% write requests, $b = 4$, $\lambda = 0.01$

(b) 50% read requests, 50% write requests, $b = 7$, $\lambda = 0.01$

(c) 75% read requests, 25% write requests, $b = 4$, $\lambda = 0.03$

Figure 9: RAID 5 b-block request response time pdfs and cdfs for arrival streams of mixed reads and writes with rate λ requests/ms.

more pronounced for 8- and 9-block transfers at $\lambda = 0.03$, suggesting it was not just an artifact of the four disk configuration. Further investigation suggests that it is a result of constraining all operations to begin at the start of a stripe, leading to unequal load on some disks, as when this restriction is relaxed it does not occur.

5 Conclusion

In this paper we have presented an improved performance model for RAID systems capable of calculating full request response time distributions. In particular, we have improved the RAID 5 large partial stripe following full stripe write model to more closely accurately observed behaviour. We validated our models for RAID 01 and 5 for reads, writes and mixtures of the two on a real-life RAID array with eight disks.

There are a number features which we still need to model in order to have a comprehensive model capable of representing real I/O workloads. Firstly, caching is not yet supported in our model. Secondly, we would like to support sequential as well as random I/O, to better model the effects of locality. Thirdly, we currently constrain the alignment of RAID 5 write requests to start at the beginning of a stripe in all cases. In the future, we would like to allow for requests that start with a partial stripe, followed by further data. Preliminary investigations suggest that this also remedies some of the unusual measurements for mixed reads and writes on RAID 5. Fourthly, all our models assume fixed request sizes and we would like to extend them to incorporate distributions of block sizes. Fifthly, we need to model the effect of the re-ordering of requests by the RAID array when greater load is experienced. Finally, we have assumed Markovian arrivals in our model, and have generated request streams that conform to this assumption for our measurements. We intend to compare the model response times with response times generated from real I/O traces.

Acknowledgements

We are grateful to Peter Harrison and Soraya Zertal for helpful discussions. This work is supported by EPSRC research grant “Intelligent Performance Optimisation of Virtualised Data Storage Systems” (iPODS) (EP/F010192/1), with industrial partners IBM and Reuters.

References

[1] J. Abate and W. Whitt. The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems Theory and Applications*, 10(1-2):5–88, 1992.

[2] D. Anderson. You don’t know jack about disks. *Queue*, 1(4):20–30, 2003.

[3] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson. RAID: High-Performance, Reliable Secondary Storage. *ACM Computing Surveys*, 26(2):145–185, June 1994.

[4] S. Chen and D. Towsley. A performance evaluation of RAID architectures. *IEEE Transactions on Computers*, 45(10):1116–1130, 1996.

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

[6] P. G. Harrison and N. M. Patel. *Performance Modelling of Communication Networks and Computer Architectures*. Addison-Wesley, 1993.

[7] P. G. Harrison and S. Zertal. Queueing models of RAID systems with maxima of waiting times. *Performance Evaluation*, 64(7-8):664–689, August 2007.

[8] A. S. Lebrecht, N. J. Dingle, and W. J. Knottenbelt. Modelling and validation of response times in zoned RAID. In *16th IEEE International Symposium on Modeling, Analysis, and Simulationof Computer and Telecommunication Systems (MASCOTS '08)*, September 2008. To appear.

[9] A. S. Lebrecht, N. J. Dingle, and W. J. Knottenbelt. A response time distribution model for zoned RAID. In *15th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA)*, June 2008.

[10] A. S. Lebrecht and W. J. Knottenbelt. Response time approximations in fork-join queues. In *23rd UK Performance Engineering Workshop (UKPEW)*, July 2007.

[11] E. K. Lee. *Performance Modeling and Analysis of Disk Arrays*. PhD thesis, University of California at Berkeley, 1993.

[12] 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.

[13] Seagate. Barracuda ES Data Sheet, 2007. http://www.seagate.com/docs/pdf/datasheet/ disc/ds_barracuda_es.pdf.

[14] A. Thomasian and A. N. Tantawi. Approximate solutions for M/G/1 fork/join synchronization. In *Proc. WSC '94*, pages 361–368, 1994.

[15] E. Varki. Mean value technique for closed fork-join networks. In *Proc. ACM SIGMETRICS*, pages 103–112, 1999.

[16] E. Varki. Response time analysis of parallel computer and storage systems. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1146–1161, 2001.

[17] E. Varki, A. Merchant, J. Xu, and X. Qiu. Issues and challenges in the performance analysis of real disk arrays. *IEEE Transactions on Parallel and Distributed Systems*, 15(6):559–574, June 2004.

[18] S. Varma and A. M. Makowski. Interpolation approximations for symmetric fork-join queues. In *Proc. Performance '93*, pages 245–265, 1994.

[19] S. Zertal and P. G. Harrison. Multi-RAID queueing model with zoned disks. In *High Performance Computing and Simulation Conference (HPCS'07)*, 2007.

Appendix

λ (ms^{-1})	# Blks	Reads Measured Mean (ms)	Reads Measured σ^2 (ms^2)	Reads Modelled Mean (ms)	Reads Modelled σ^2 (ms^2)	Writes Measured Mean (ms)	Writes Measured σ^2 (ms^2)	Writes Modelled Mean (ms)	Writes Modelled σ^2 (ms^2)
0.01	1	15.7	15.4	15.7	20.8	21.4	26.8	19.3	19.6
	2	18.3	13.0	18.5	18.0	24.9	38.8	22.6	27.3
	3	19.6	13.1	20.2	20.2	27.7	60.7	25.4	41.6
	4	20.2	15.2	21.7	24.6	28.7	77.2	28.4	58.4
	5	21.1	16.9	23.0	30.5	31.0	89.4	29.3	63.5
	6	22.3	21.4	24.3	37.2	32.4	102.5	30.3	68.9
	7	23.0	22.9	25.7	44.5	33.7	118.1	31.3	74.8
	8	24.0	27.7	27.1	52.0	34.9	132.5	32.4	80.9
	9	24.7	29.4	27.5	54.4	37.0	155.0	33.4	87.5
	10	26.0	34.4	28.0	56.8	37.8	164.2	34.5	94.4
	11	26.9	38.4	28.5	59.3	38.4	148.0	35.6	101.7
	12	27.9	42.1	29.0	61.8	39.2	147.3	36.7	109.5
	13	28.6	45.1	29.5	64.5	41.3	170.4	37.9	117.7
	14	29.5	50.0	30.0	67.2	42.1	173.5	39.0	126.3
0.03	1	15.7	15.4	16.0	25.2	21.4	26.8	20.5	35.2
	2	18.3	13.0	19.7	31.7	24.9	38.8	27.2	80.0
	3	19.6	13.1	22.7	47.9	27.7	60.7	35.7	154.6
	4	20.2	15.2	25.9	70.6	28.7	77.2	47.6	283.1
	5	21.1	16.9	29.5	99.1	31.0	89.4	50.4	322.1
	6	22.3	21.4	33.7	134.2	32.4	102.5	53.4	366.9
	7	23.0	22.9	38.5	179.0	33.7	118.1	56.6	418.1
	8	24.0	27.7	44.2	239.2	34.9	132.5	60.1	476.9
	9	24.7	29.4	45.5	255.1	37.0	155.0	63.7	544.6
	10	26.0	34.4	46.9	272.3	37.8	164.2	67.6	622.6
	11	26.0	38.4	48.2	290.5	38.4	148.0	71.8	712.9
	12	27.9	42.1	49.6	310.1	39.2	147.3	76.4	817.4
	13	28.6	45.0	51.1	331.0	41.3	170.4	81.3	939.2
	14	29.5	50.0	52.6	353.3	42.1	173.5	86.6	1081.8

Table 1: Mean and variance of request response time for RAID 01 reads and writes against measured results.

λ (ms^{-1})	# Blks	Reads				Writes			
		Measured		Modelled		Measured		Modelled	
		Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)
0.01	1	15.9	17.0	15.7	20.8	32.3	129.2	39.5	192.0
	2	19.5	32.4	18.5	18.0	46.1	566.2	45.7	193.8
	3	20.9	34.2	20.2	20.2	48.1	132.0	50.6	232.5
	4	21.9	34.1	21.7	24.6	48.1	128.4	49.4	201.6
	5	22.6	34.3	23.0	30.5	48.0	124.4	49.4	201.6
	6	23.3	35.1	24.3	37.2	46.9	118.9	49.4	201.6
	7	23.8	35.6	25.7	44.5	31.4	83.7	28.4	58.4
	8	25.8	52.7	27.1	52.0	48.1	1042.2	53.7	269.6
	9	26.3	56.2	27.5	54.4	55.2	1464.3	57.0	312.6
	10	27.4	64.8	28.0	56.8	51.0	699.5	60.7	362.8
	11	28.0	68.8	28.5	59.2	50.5	699.7	49.0	213.1
	12	28.9	73.7	29.0	61.8	49.8	668.5	48.0	204.6
	13	29.3	73.9	29.5	64.5	47.5	590.5	46.6	193.6
	14	29.9	78.5	30.0	67.2	38.6	121.8	32.4	80.9
0.03	1	16.2	21.5	16.0	25.2	41.7	408.1	44.9	346.8
	2	23.9	106.7	19.7	31.7	152.6	11040.6	58.5	573.4
	3	25.7	116.6	22.7	47.9	64.3	539.6	76.4	1036.4
	4	27.1	129.2	25.9	70.6	63.0	524.6	78.0	1052.9
	5	27.9	135.0	29.5	99.0	61.9	517.8	78.0	1052.9
	6	29.0	148.2	33.7	134.2	60.0	537.7	78.0	1052.9
	7	29.6	153.8	38.5	179.0	42.8	516.9	47.6	283.1
	8	33.7	196.8	44.2	239.1	sat	sat	96.9	1701.3
	9	34.9	225.3	45.5	255.1	sat	sat	119.3	2628.9
	10	37.0	247.4	46.9	272.2	sat	sat	152.9	4399.6
	11	38.2	281.1	48.2	290.5	sat	sat	91.2	1388.5
	12	39.7	295.3	49.6	310.0	sat	sat	89.2	1300.0
	13	39.9	298.0	51.1	330.9	sat	sat	85.4	1167.2
	14	41.1	321.6	52.6	353.2	sat	sat	60.1	476.8

Table 2: Mean and variance of request response time for RAID 5 reads and writes against measured results.

λ	#	25% Reads, 75% Writes				50% Reads, 50% Writes				75% Reads, 25% Writes			
		Measured		Modelled		Measured		Modelled		Measured		Modelled	
(ms^{-1})	Blks	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)
0.01	1	20.2	30.2	18.3	21.8	18.8	30.5	17.4	22.6	17.3	25.4	16.6	22.3
	2	23.5	41.0	21.5	26.4	22.1	40.6	20.4	24.4	20.3	32.0	19.5	21.6
	3	26.2	60.6	24.0	37.8	24.3	55.6	22.6	32.6	22.2	41.4	21.4	26.5
	4	27.2	73.5	26.4	52.1	25.4	66.0	24.7	43.7	23.0	49.2	23.1	34.3
	5	29.3	90.2	27.6	57.6	27.1	82.9	25.9	49.7	24.4	61.0	24.4	40.4
	6	30.7	104.5	28.7	64.0	28.4	95.4	27.2	56.7	25.6	71.0	25.7	47.6
	7	31.9	120.3	29.9	71.1	29.5	106.5	28.4	64.6	26.6	78.0	27.0	55.7
	8	33.0	130.7	31.1	79.0	30.5	115.7	29.7	73.5	27.6	86.6	28.4	64.5
	9	34.8	154.1	32.0	85.7	32.0	137.3	30.5	79.6	28.6	96.1	29.0	69.1
	10	35.9	163.6	32.9	92.9	33.1	143.0	31.3	86.1	29.8	103.8	29.6	74.1
	11	36.7	156.1	33.8	100.6	34.2	147.7	32.1	93.1	30.9	113.6	30.3	79.3
	12	37.7	164.1	34.8	108.7	35.0	151.2	32.9	100.6	31.8	114.9	30.9	84.9
	13	39.6	186.2	35.8	117.5	36.6	176.9	33.7	108.6	32.9	132.5	31.6	90.9
	14	40.3	187.3	36.8	126.8	37.6	181.3	34.5	117.1	34.0	141.0	32.2	97.2
0.03	1	21.2	49.1	18.3	21.8	19.7	46.4	18.2	32.3	17.8	35.1	17.1	29.2
	2	26.7	101.8	25.0	68.7	24.6	85.4	23.0	56.2	21.9	61.0	21.2	43.6
	3	32.0	189.6	31.6	126.8	28.6	140.7	28.1	98.4	24.9	91.2	25.2	71.6
	4	35.3	301.2	40.2	217.2	31.0	199.2	34.4	159.4	26.5	118.0	29.7	110.7
	5	39.0	405.3	43.6	256.0	34.2	269.1	38.0	197.0	29.0	160.0	33.4	144.2
	6	42.6	529.8	47.3	308.1	37.7	372.2	42.1	247.3	31.8	222.8	37.6	188.6
	7	46.2	671.2	51.4	375.1	40.5	458.3	46.7	317.5	33.9	273.3	42.4	250.8
	8	49.7	836.0	56.1	464.3	43.3	592.7	52.1	420.5	36.3	341.9	48.2	345.5
	9	53.7	1012.3	59.2	534.2	46.6	685.4	54.6	482.5	38.2	388.8	50.1	389.5
	10	57.0	1240.0	62.4	615.9	49.7	812.5	57.2	555.3	40.7	451.2	52.1	440.7
	11	60.2	1491.2	65.9	711.8	52.0	944.3	60.0	641.1	43.5	551.2	54.1	500.6
	12	65.5	2472.3	69.7	824.6	55.5	1212.1	63.0	742.4	45.3	615.0	56.3	570.9
	13	72.1	3206.3	73.7	957.9	60.7	1739.7	66.2	862.8	48.4	788.1	58.6	653.8
	14	80.4	5305.8	78.1	1116.3	63.2	1824.8	69.6	1006.3	51.1	896.7	61.1	752.0

Table 3: Comparison of mean response times and variances for mixed read and write request streams for RAID 01.

λ (ms^{-1})	# Blks	25% Reads, 75% Writes				50% Reads, 50% Writes				75% Reads, 25% Writes			
		Measured		Modelled		Measured		Modelled		Measured		Modelled	
		Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)	Mean (ms)	σ^2 (ms^2)
0.01	1	27.8	130.8	33.3	241.9	23.5	109.1	27.3	227.1	19.5	68.2	21.4	152.1
	2	37.9	453.6	38.5	260.2	30.9	312.4	31.6	247.9	24.9	173.2	25.0	164.8
	3	41.3	233.8	42.4	304.0	34.5	258.4	34.7	284.1	27.7	189.9	27.3	185.7
	4	41.6	227.5	42.0	261.2	35.1	250.3	34.9	244.9	28.5	185.9	28.1	163.0
	5	41.7	222.1	42.5	258.5	35.7	247.1	35.8	244.5	29.3	189.4	29.3	166.5
	6	41.3	205.6	43.0	257.4	35.5	227.7	36.6	246.0	29.5	174.1	30.4	171.3
	7	30.0	92.2	27.7	55.4	28.5	94.3	27.0	52.1	26.3	73.5	26.3	48.5
	8	41.1	641.9	46.9	323.4	35.5	398.6	40.2	302.9	30.4	216.6	33.6	211.4
	9	46.1	961.2	49.4	368.0	38.3	576.4	41.9	339.1	31.7	269.7	34.6	232.6
	10	44.5	569.8	52.0	418.9	38.4	419.2	43.7	379.2	32.7	241.4	35.7	255.2
	11	44.4	562.5	43.9	214.7	38.7	415.2	38.8	191.1	33.3	253.0	33.7	139.9
	12	44.4	547.0	43.4	196.8	38.9	407.7	38.7	173.3	33.7	242.7	33.9	129.7
	13	42.8	474.3	42.8	179.1	38.0	351.5	38.6	156.6	33.5	216.6	34.2	120.3
	14	37.0	145.3	31.8	78.6	35.2	150.9	31.2	75.5	32.6	126.5	30.6	71.7
0.03	1	33.7	337.2	36.7	355.6	26.8	235.0	29.3	295.7	21.0	123.3	22.4	181.8
	2	91.5	4961.0	46.7	535.7	57.8	2138.8	36.5	414.7	37.5	804.9	27.6	240.3
	3	53.2	597.0	58.5	863.5	43.5	545.4	44.2	612.8	34.9	395.3	27.6	240.3
	4	53.5	604.0	61.0	888.4	44.5	551.1	47.2	641.3	35.9	397.1	35.7	360.1
	5	53.5	603.2	63.1	920.7	45.5	575.8	50.3	693.2	36.8	407.4	39.2	411.0
	6	53.2	621.4	65.2	960.8	45.4	560.3	53.7	757.3	37.4	402.0	43.2	474.1
	7	41.3	479.6	45.1	256.2	38.7	411.1	42.7	229.7	34.6	301.3	40.5	203.9
	8	1341.1	695702	81.7	1467.7	102.3	10900	68.0	1129.0	53.5	1884.4	55.6	712.7
	9	2250.1	2177073	95.6	2057.0	130.2	16333	76.0	1456.8	60.0	2562.2	59.6	851.7
	10	170.2	60233	114.0	2993.3	78.4	3753.9	85.7	1907.0	53.5	1357.5	64.1	1020.1
	11	193.7	74681	82.4	1077.8	80.1	4047.0	70.7	743.3	55.2	1456.8	59.0	496.3
	12	350.1	186247	86.1	1390.2	83.8	5213.8	73.9	833.3	56.6	1587.7	61.29	517.8
	13	444.7	211185	93.3	2669.0	80.5	4935.4	78.5	1116.3	55.7	1511.8	63.9	578.3
	14	64.1	3934.3	58.2	456.3	55.4	1182.5	56.3	428.9	48.2	677.6	54.5	394.6

Table 4: Comparison of mean response times and variances for mixed read and write request streams for RAID 5.