Multi-RAID Queueing Model with Zoned Disks

Soraya Zertal[1] and Peter Harrison[2]
[1] PRiSM, Université de Versailles, 45 Av. des Etats-Unis, 78000 Versailles, France
Zertal@prism.uvsq.fr
[2] Imperial College London, South Kensington Campus, London SW7 2AZ, UK
pgh@doc.ic.ac.uk

Abstract— **A queueing model is developed for a multi-RAID storage system implemented on modern zoned disks, using fine, accurate access time functions. An extension of a previous analytical model that utilizes Fork-Join composition of M/G/1 queues, it describes zoning directly in terms of the probability distributions or moments of the model's components, such as seek time, rotational latency and data transfer time. These quantities are calculated directly using the principles of operation of the hardware. This is in contrast to estimating them from simulations and theoretical bounds, as in previous zoned disk models. The resulting multi-RAID model turns out to be accurate, when its performance predictions, characterized here by the mean of queueing and response times, are compared with simulation, and also scalable; not only for the zoned technology but also for alternate ones.**

keywords : Multi-RAID, Zoned disks, Fork-Join, M/G/1 queues, I/O modelling, Simulation.

I. Introduction

Computer applications become ever more data intensive. To satisfy their QoS requirements in terms of capacity, performance and availability, RAID (Redundant Arrays of Independent Disks) are commonly used. To describe and predict RAID performance, several analytical and simulation models have appeared in the literature since the introduction of RAID [2] but their objective is always one and only one RAID configuration per disk array. However, it has been shown that with the continous evolution of data access to disk arrays, it is necessary to provide different RAID configurations on the same array to adapt the data storage to user requirements on access time and availability. This leads to better exploitation of the storage system's space and improved performance of its usage. With the introduction of this type of multi-RAID system, none of the available RAID models is capable of describing and predicting its performance effectively.

We introduced a new analytical queueing methodology for this purpose in [6], [25] and this is still the only one addressing such a complex disk array, to our best knowledge. Any analytical performance modelling of storage systems is concerned with expressing mathematically the details of its devices' technology. This is not yet the case for our multi-RAID model since, to date, it cannot account for zoned disks. The aim of this paper is to extend our previous modelling study [8] by taking into account both the fine details of modern zoning disk technology and more complex, but more representative, access time functions. The result is an accurate and scalable multi-RAID model.

In the rest of the paper, section 2 summarizes RAID systems and modelling in this area. Section 3 details the new, analytical model, describing zoned disks and the aforementioned access time functions, whereas Section 4 describes the simulation procedures. Section 5 discusses and compares the numerical results obtained from the analytical model and the simulation. Finally, section 6 concludes the paper and suggests future research directions.

II. Overview of the RAID system and related work

A RAID storage system consists of a disk system manager and a collection (array) of independent disks. The disk system manager is a software component of the RAID controller. It receives logical requests from the multiple system users, at different rates, subdivides the data into blocks and distributes them across the disks. Consequently, for each logical request, it generates a number of physical requests and sends them to the associated disks which receive requests at different rates. Finally, the disk system manager waits for responses from each requested disk to construct the (logical) response to send to the user. The request subdivision-distribution process is performed according to the data/redundancy pattern over the disks. When various data placement schemes [2] coexist with dynamic selection of the current redundancy pattern, in order to provide storage space and access time optimizations, we obtain a dynamic Multi-RAID [24]. Requests' independent executions on such asynchronous disks lead to Fork-Join-type modelling problems, described and analysed in [8].

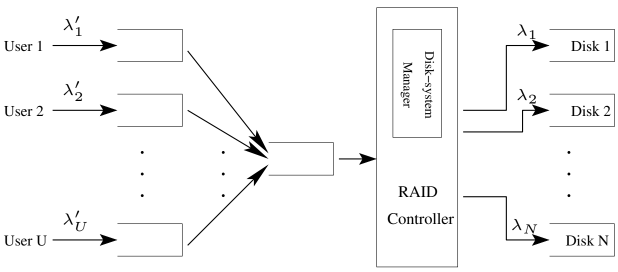

Fig. 1. Requests flow in a RAID storage system

Modelling of RAID systems is the subject of several studies. For a few examples, we can cite the approximation of access latency [11], queueing models for RAID3 and RAID5 [1], RAID queueing models in different execution modes – normal, degraded and recovery [13], [14], reconstruction and failure tolerance [5]

Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)

and caching and controller optimization [3], [23]. All of these models are appropriate for different RAID levels but just one at a time. Thus, none of them is appropriate for a multi-RAID system. To our knowledge, only the model presented in [8] can handle analytically all the characteristics of a multi-RAID system and can be used to predict its performance. In fact, on a multi-RAID, the unbalanced workload leads to a difference between disks' waiting times and the use of asynchronous disks introduces a significant difference between the seek times. The RAID is then presented as a collection of M/G/1 queues, one for each disk, interacting so as to account for *synchronization* between the disks. This interaction may be approximated by estimating the mean of the maximum of the response times among the disks involved in a logical access. In fact, an exact solution for this type of Fork-Join problem was obtained for Poisson arrivals and exponential service times, together with an approximation for other distributions in [6], [8]. A detailed evaluation of such approximations is the subject of [7]. However, in this multi-RAID model, it was assumed that all disk cylinders are identical and the seek time is calculated using a simple function as derived in [18].

Recently, a new disk organisation emerged called *zoned disk technology*, in which the number of sectors per track is variable. Consecutive cylinders are collected into groups, called *zones*, such that within each zone, the track capacity (number of sectors) and the transfer rate are fixed. However, these two parameters decrease from the outer to the inner zones. These disks have become very popular due to their greater storage capacitiy and transfer rate. Their average rotational latency is constant but the variable seek and transfer times necessitate more complex calculations in terms of the assumed statistical workload and disk behaviour in a storage model [16], [17].

RAID modelling relies strongly on the analysis of Fork-Join queueing networks. Until now, research has mainly been concerned with approximations, performance bounds and calculations based on simulations. In [21], [20], a performance bound is calculated for a closed Fork-Join network. In [15], the Fork-Join response time for homogeneous processes with exponential service time distributions is calculated for two processes and approximated for more, based on simulations and theoretical bounds. In [22], Fork-Join queueing systems with general service times are approximated using interpolation and in [19], the approximation of similar systems is based solely on preliminary simulations.

III. Analytical model

The entire RAID model that we address is based on a collection of M/G/1 queues with various extensions to account for the Fork-Join nature of the parallel disk accesses corresponding to a logical request. The response time of each physical request, to an individual disk, is composed of four components: the time spent waiting to start service in the disk queue (Q), the seek time (S), the rotational latency (R) and the transfer time, which we separate into two components, t and T, corresponding to transfer from the disk's cylinder to its buffer and from this buffer via the bus, respectively. The model we develop uses the notation shown in Table I.

Parameter	Description
N	Number of disks in the storage system.
C	Number of cylinders on a disk.
SEC	Number of sectors on the disk.
SEC_c	Number of sectors on cylinder c.
spb	Number of sectors per block.
B	Logical request size (transfer block).
Q_i	Queueing time at disk i.
D_i	Seek distance on a disk i.
S_i	Seek time on a disk i.
R_i	Rotational latency on the same cylinder.
R_{MAX}	Full disk rotation time.
t	Block transfer time between the disk's buffer and its cylinder.
T	Bus transfer time of one block.
λ	Logical request arrival rate to the storage system.
p_i	Probability that disk i is used.
λ_i	Physical request arrival rate to disk i.
λ_{Rj}	Physical request arrival rate to the RAIDj area.
λ_{iRj}	Physical request arrival rate to a RAIDj area on disk i.
P_{raid_j}	$RAIDj$ area's proportion in the whole storage system space.
$Z_r(i)$	Read response time on disk i.
$Z_w(i)$	Write response time on disk i.
$\overline{Z_r}$	Mean response time for a read.
$\overline{Z_w}$	Mean response time for a write.
$\overline{Z}$	Mean response time for any request.
p_w	Probability that a request is a write.
p_r	Probability that a request is a read.
p_s	Probability of a sequential access.

TABLE I: Notation for the RAID model's parameters

A. Maximum of random variables

Suppose a task forks into a number of subtasks that are processed in parallel independently. The task's completion instant is that of the last subtask to complete processing, whereupon the subtasks combine (join) to re-form the original task. The Fork-Join time of the task, i.e. the time elapsed between the fork instant and the join instant, is therefore the maximum of the subtasks' processing times. In [6], the following result for the moments of Fork-Join times in a Markovian environment was derived:

Proposition 1: The kth moment $M_n(\boldsymbol{\alpha}, k)$ of the maximum of $n \geq 1$ independent, negative exponential random variables with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ is defined by the recurrence

$$M_n(\boldsymbol{\alpha}, k) = \frac{k}{\sum_{j=1}^{n} \alpha_j} M_n(\boldsymbol{\alpha}, k-1) + \frac{\sum_{j=1}^{n} \alpha_j M_{n-1}(\boldsymbol{\alpha}_{\setminus j}, k)}{\sum_{j=1}^{n} \alpha_j}$$

for $n \geq 1$ and $M_0(\boldsymbol{\epsilon}, k) = 0$, for all $k \geq 1$, with $M_n(\boldsymbol{\alpha}, 0) = 1$ for all $n \geq 0$.
where $\boldsymbol{\alpha}_{\setminus j} = (\alpha_1, \ldots, \alpha_{j-1}, \alpha_{j+1}, \ldots, \alpha_m)$
In the special case that all the parameters of the exponential distributions are equal ($\forall j,\ \alpha_j = \alpha$) :

$$M_n(\boldsymbol{\alpha}, 1) = \frac{1}{\alpha} \sum_{m=1}^{n} \frac{1}{m} \quad and \quad M_n(\boldsymbol{\alpha}, 2) = \frac{1}{\alpha^2} \sum_{m=1}^{n} \frac{1}{m^2}$$

The logical request access time is defined as the maximum of all its physical request access times. We require the value of this quantity, assuming that the physical request access times are independent.

For generally distributed random variables, it is shown in [8] that the expected value of the maximum of n independent, non-negative random variables with means $\mathbf{m} = (m_1, \ldots, m_n)$, $\boldsymbol{\alpha} = (m_1^{-1}, \ldots, m_n^{-1})$ and second moments $\mathbf{M} = (M_1, \ldots, M_n)$ is approximated by the function $I(n, \boldsymbol{\alpha}, \mathbf{M})$ defined by the recurrence, for $k = 2, \ldots, n$, with $I(1, \alpha_1, M_1) = 1/\alpha_1$

$$\begin{aligned} I(k, \boldsymbol{\alpha}, \mathbf{M}) &= \frac{1}{k} \sum_{i=1}^{k} I(k-1, \boldsymbol{\alpha}_{\setminus i}, \mathbf{M}_{\setminus i}) \\ &\quad + \alpha_i M_i L_{k-1}(\boldsymbol{\alpha}_{\setminus i}, \alpha_i)/2 \qquad (1) \end{aligned}$$

where $L_{k-1}(\boldsymbol{\alpha}_{\setminus i}, s)$ is the Laplace transform of the probability density function of the maximum of $k-1$ *exponential* random variables.

This result is exact if all the random variables are exponential. Notice that, when exact, all the summands give the same result. When approximate, the result is the average obtained by picking each of the k random variables in turn as the last in the sequence, and maximizing this and the maximum of the rest.

In the special case that all the parameters are equal $\forall i$, $\alpha_i = \alpha$ and $M_i = M$, proposition 1 gives the result

$$L_{k-1}(\boldsymbol{\alpha}, \alpha) = 1/k$$

so that

$$I(k, \boldsymbol{\alpha}, \mathbf{M}) = I(k-1, \boldsymbol{\alpha}, \mathbf{M}) + \frac{M\alpha}{2k}$$

$$I(k, \boldsymbol{\alpha}, \mathbf{M}) = 1/\alpha + (M\alpha/2) \sum_{i=2}^{k} 1/i$$

For each type of access (read or write), RAID variant and request size, the number of participating disks k is computed.

B. Mean response Time

Each disk is modelled by an M/G/1 queue of physical requests. It serves read/write requests and parity pre-read/update requests. Each physical request relates to one data block and its response time is composed of four terms, as formulated below. We use n overbars to indicate an nth moment.
1. **Queueing time** is calculated using the Pollaczek-Khinchin formulae[9], extended to handle multiple classes:

$$E[Q_i] = \frac{\sum_{j=1,5} \lambda_{iRj} \overline{\overline{X_{iRj}}}}{2(1-\rho_i)} \qquad (2)$$

where, refering to Table I,
- $X_i = S_i + R_i + t_i$ is the service time on disk i;
- t_i is the transfer time for one block from the track to the buffer of disk i;
- $\overline{\overline{X_{iRj}}}$ is the second moment of service time on $RAID_j$ area on disk i;
- $\rho_i = \lambda_i \overline{X_i}$ is the traffic intensity on disk i.

2. **Seek time**, depends on the distance D_i between the current position of the device's read/write head and the target position. In this paper, we use a more complex, but more accurate, formula [12] as follows :

$$S_i(D) = \begin{cases} 0 & \text{if } D_i = 0 \\ a\sqrt{D_i} + b(D_i - 1) + c & \text{otherwise} \end{cases} \qquad (3)$$

where a, b, c are hardware-related constants :

$$\begin{aligned} a &= (-10 \times MinSeek + 15 \times AvgSeek \\ &\quad -5 \times MaxSeek)/(3 \times \sqrt{C}) \\ b &= (7 \times MinSeek - 15 \times AvgSeek \\ &\quad +8 \times MaxSeek)/(3 \times C) \\ c &= MinSeek \end{aligned}$$

The hardware parameters *MinSeek, AvgSeek* and *MaxSeek* are respectively the seek time from one track to the adjacent one, the mean seek time and the seek time from the inner track to the outer. We assume that the incoming logical requests' addresses are independent random variables, uniformly distributed over the disk-address space. Since C is large, the distance D can be well approximated by a continuous random variable. Assuming that the number of sectors (and hence blocks) per track varies linearly with the cylinder number, the density function of D can be shown to be :

$$f_D(x) = A + Gx + Ex^3 \quad (0 \leq x \leq C-1) \qquad (4)$$

where we define the constants:

$$\begin{aligned} A &= \frac{V(C-1)}{3\gamma^2} \\ G &= -\frac{V + \beta^2(C-1)^2}{3\gamma^2} \end{aligned}$$

$$
\begin{aligned}
E &= \frac{\beta^2}{3\gamma^2} \\
V &= 6\alpha^2 + 6\alpha\beta(C-1) + 2\beta^2(C-1)^2 \\
\gamma &= \alpha(C-1) + \beta(C-1)^2/2 \\
\alpha &= SEC_{C-1}/spb, \\
& \quad \text{the number of blocks on an innermost track} \\
\beta &= SEC_0/spb - SEC_{C-1}/spb, \\
& \quad \text{the difference between the number of} \\
& \quad \text{blocks on outermost and innermost tracks.}
\end{aligned}
$$

For simplicity, we have assumed that all disks have the same hardware parameters, including a, b, c, C. However, it would be easy to extend our model to heterogeneous devices.

3. **Rotational latency** is a random variable with uniform distribution on $[0, R_{MAX}]$, with density function :

$$f_R(x) = \frac{1}{R_{MAX}} \qquad 0 \le x \le R_{MAX} \qquad (5)$$

4. **Single block transfer time** is composed of a bus transfer time T and a cylinder/device_buffer transfer time t. Assuming negligible contention, T depends only on the bus bandwidth. Previously, t was considered fixed, but this assumption is no longer valid on multi-zoned disks. In fact, t depends on the position of the accessed cylinder on the disk. There are more sectors on outer zones leading to a reduced block transfer time. Assuming track-alignment, the transfer time of a block on a cylinder c is calculated as:

$$\frac{spb * R_{max}}{SEC_c}$$

C. Moments

The first three moments of the four parameters above are needed in the analytical model and calculated as :

1. **Queueing time :** Multi-zoning has no effect on the queueing time formula itself – it just affects the server utilization, service time variance, and hence the queueing time values given by that formula. Thus the moments' expressions are unchanged from [8].

2. **Seek time :** This is the parameter most affected by multi-zoning. Its new moments are given by :

$$
\begin{aligned}
\overline{S} &= (c-b) + aM_{1/2} + bM_1 \\
\overline{\overline{S}} &= (c-b)^2 + 2a(c-b)M_{1/2} \\
&\quad + [a^2 + 2b(c-b)]M_1 + 2abM_{3/2} + b^2M_2 \\
\overline{\overline{\overline{S}}} &= (c-b)^3 + a(c-b)^2M_{1/2} \\
&\quad + 3(c-b)[a^2 + b(c-b)]M_1 \\
&\quad + [a^3 + 6ab(c-b)]M_{3/2} \\
&\quad + 3[a^2b + b^2(c-b)]M_2 + 3ab^2M_{5/2} + b^3M_3
\end{aligned}
$$

where the nth moment of D is, for $n \ge 0$ (not necessarily integer):

$$M_n = (C-1)^{n+1}\left[\frac{A}{n+1} + \frac{G(C-1)}{n+2} + \frac{E(C-1)^3}{n+4}\right]$$

3. **Rotational latency :** Multi zoning has no effect on the rotational latency; thus the moments' expressions are unchanged from [8].

4. **Cylinder/device_buffer transfer time :** It is not constant and its first three moments can be shown to be :

$$\overline{t} = \frac{spb * R_{max} * (C-1)}{SEC}$$

$$\overline{\overline{t}} = \frac{spb^2 * R_{max}^2}{SEC} \times \Sigma_{j=0}^{C-1}\frac{1}{SEC_j}$$

$$\overline{\overline{\overline{t}}} = \frac{spb^3 * R_{max}^3}{SEC} \times \Sigma_{j=0}^{C-1}\frac{1}{SEC_j^2}$$

From the above, we calculate the moments of the positioning time Y and hence of the service time X:

$$\overline{X} = \overline{Y} + \overline{t}$$

$$\overline{\overline{X}} = \overline{\overline{Y}} + \overline{\overline{t}} + 2\overline{Y}\,\overline{t}$$

$$\overline{\overline{\overline{X}}} = \overline{\overline{\overline{Y}}} + \overline{\overline{\overline{t}}} + 3\overline{\overline{Y}}\,\overline{t} + 3\overline{Y}\,\overline{\overline{t}}$$

IV. Simulation

To evaluate our multi-RAID model for zoned disks, we adapted the simulator of [8] to handle the new disk geometry and access time functions. The simulator is written in C and is composed of three main parts: a logical request generator, a logical to physical mapping core and a simulation engine. The hardware parameters are obtained from a library, which is separated from the execution routines for flexibility and scalability purposes. Two modules were especially affected by the modification : the address mapping and the access time calculation modules. Experiments were carried out using different combinations of both workload and architecture configuration parameters.

The RAID array modelled is composed of 16 disks connected to an ultra wide SCSI bus. The characteristics of the zoned disk used in the simulations are summarized in table II. Additional details on this device can be consulted in [4].

Parameter	value
Formatted capacity	36,74 GB
Sectors/device (SEC)	$18,37 \times 10^7$
Rotation	10000 rpm
Cylinders (C_{max})	29950
Min Seek	0,4 - 0,6 ms
Avg Seek	4,5 - 5 ms
Max Seek	11 - 12 ms
Data Heads	4
Zones number	18

TABLE II: Disk Characteristics - Fujitsu-MAN3367

Simulations were run for a warm-up period of 300000 logical request arrivals, then for a further 700000 arrivals, during which the measurements were gathered.

V. Results and discussion

To compare the adapted Multi-RAID model (with zoned disks) against the simulation, we plotted their respective mean queueing and response times. We validated our model according to four main parameters :
1. **Request size :** Figures 2 and 3 show the effect of the request size, in terms of the number of blocks per logical request (B), on the mean queueing and response times respectively. The RAID01 variant was chosen for this experiment in view of its popularity due to a lower cost of disks. The access request environment was exclusive read to isolate the basic disk access behaviour from any complex write execution scheme. Good agreement is apparent on both figures.

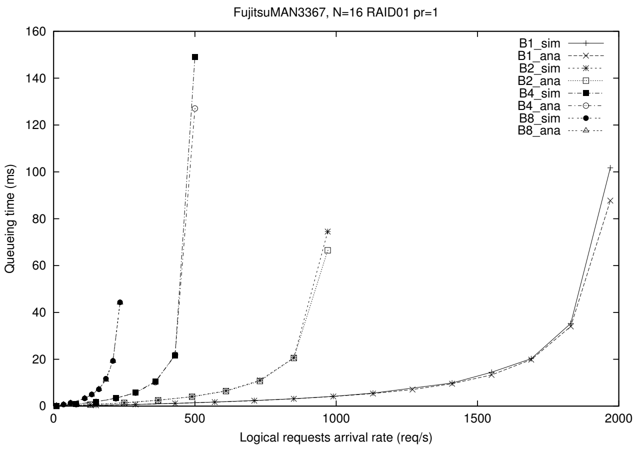

Fig. 2. Queueing time (RAID01,pr=1)

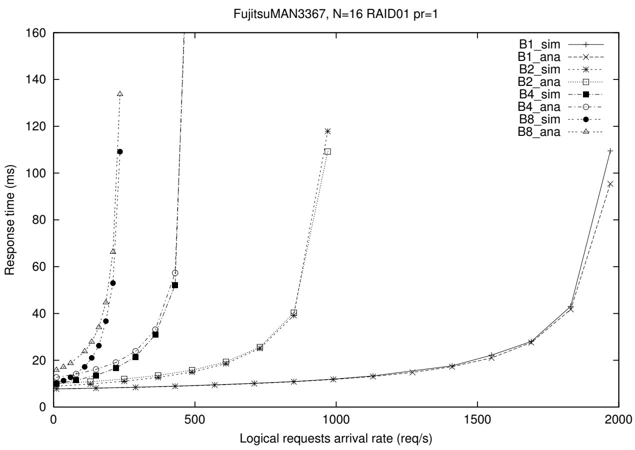

Fig. 3. Response time (RAID01,pr=1)

2. **Request type :** In figures 4 and 5, we see the effect of the request type (read/write) on the queueing and response times respectively, in a RAID01 environment with small requests (B=1). We focused on the small request size (4KB blocks) because it is the size of 96% of requests in an OLTP workload [10], typically associated with large storage systems. In addition to the good agreement, we notice the effect of the double disk access generated by every write request on the mean response time.

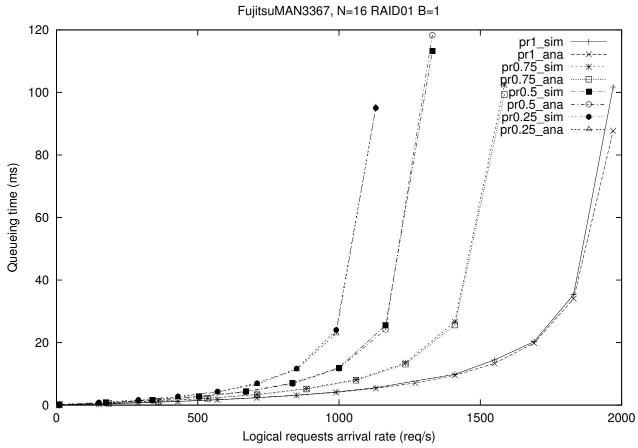

Fig. 4. Queueing time (RAID01,B=1)

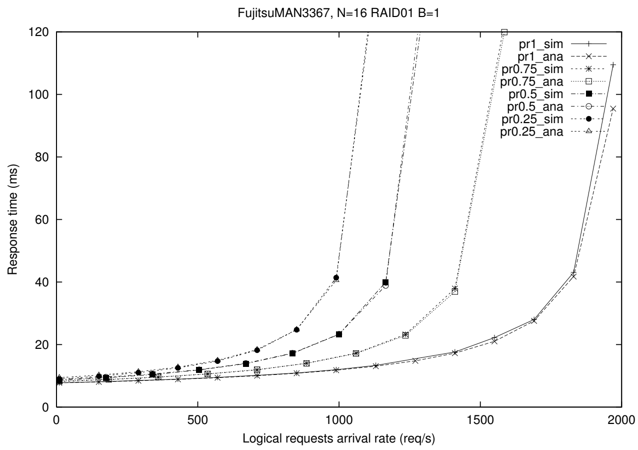

Fig. 5. Response time (RAID01,B=1)

3. **RAID variant :** Figures 6 and 7 show the effect of the RAID variant on the queueing and response times respectively. In fact, these figures show the RAID5 performance, considering some combinations of request type, to be superior to (i.e. with lower queueing and response times than) that of RAID01 on figures 4 and 5. Notice again the good agreement with simulation.

The original model [6], [25] was motivated by the need for an analytical model of a Multi-RAID storage system[1], when controlling asynchronous disks with uniformly distributed data. Figures 8 and 9 respectively show the mean queueing and response times for a mixed workload (75% reads) and a Multi-RAID (mixed RAID) system with two different mixture ratios : a RAID01 oriented system (75% of RAID01) and a RAID5 oriented system (75% of RAID5). Also shown on these figures are the corresponding exclusive RAID01 and exclusive RAID5 results as bounds. The agreement found confirms that the model remains

[1]Where many RAID schemes coexist on the same storage space. Our study is limited to the RAID01 and RAID5, the most commonly used ones.

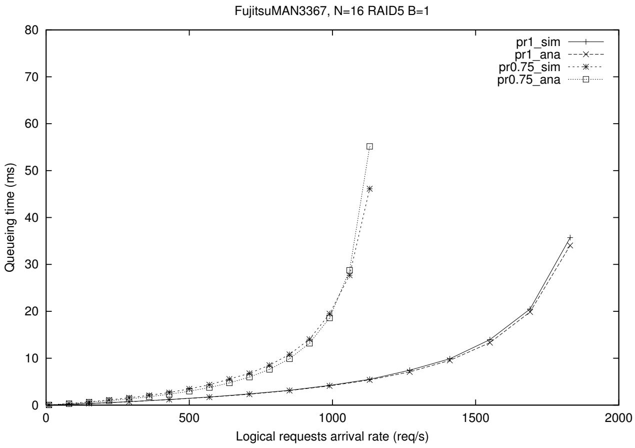

Fig. 6. Queueing time (RAID5,B=1)

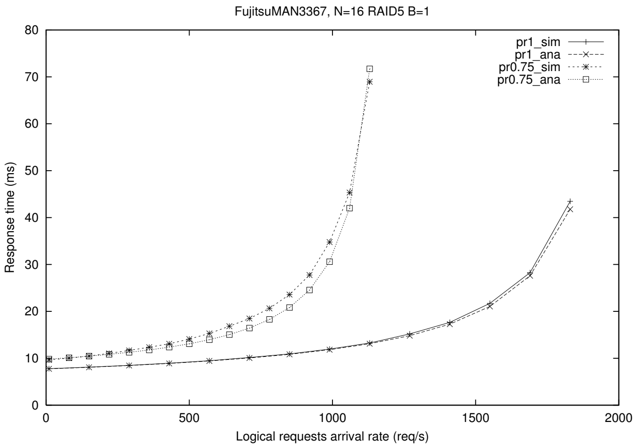

Fig. 7. Response time (RAID5,B=1)

valid and accurate for Multi-RAID storage systems when extended to modern zoned disks.

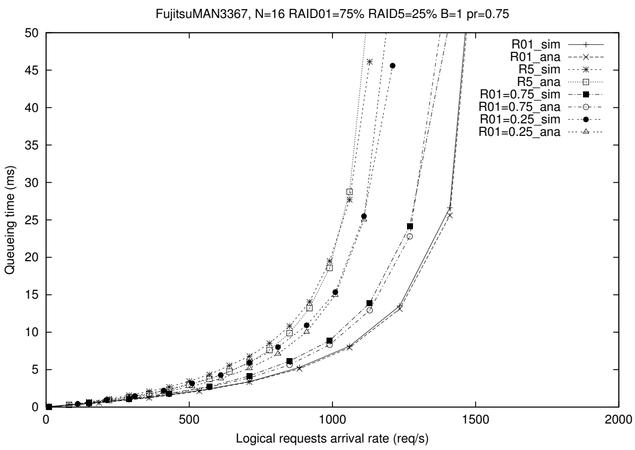

Fig. 8. Queueing time (R01/R5)

In general, we see good agreement between the analytical results and the simulation, as well as the decreasing penalizing effect of the small RAID5 writes on mean response time, as its usage percentage decreases in the mixed RAID system.

4. **Quantitative model accuracy:** We estimated how accurate are the various parts of our model for multi-zoned disks by analysing each component of the access time separately. We can see, in tables III and IV, the excellent precision of the first two moments of the seek distance (D), the first three moments of the seek time (S) and the rotational latency (R), respectively.

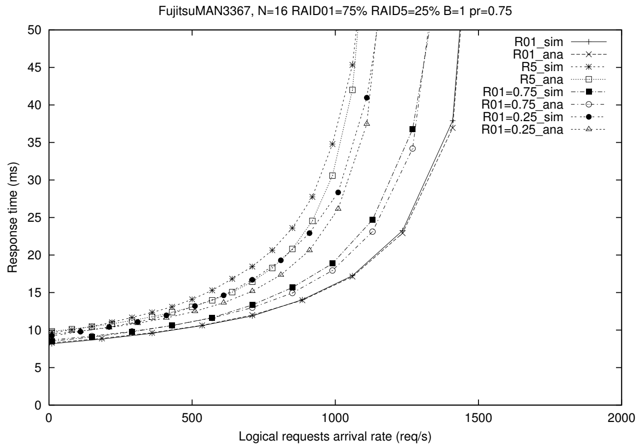

Fig. 9. Response time (R01/R5)

Moment	Model	Simulation	% err
1	9866.27	9795.23	0.72%
2	146 577 339.47	144 211 521.78	1.6%

TABLE III: Moment comparison:seek distance (D)

	Moment	Model	Simulation	% err
S	1	4.708	4.683	0.53 %
	2	28.54	28.24	1.06 %
	3	200.35	198.1	1.13 %
R	1	3.00	2.993	0.23 %
	2	12.0	11.949	0.42 %
	3	54.0	53.681	0.59 %

TABLE IV: Moments comparison:seek time (S) and rotational latency (R)

VI. Conclusion

We developed an intricate analytical model in [6], [25] for a multi-RAID storage system, based on queueing theory and taking into account the effect of synchronized Fork-Join operations. In this paper, we took a step further by adapting that model to modern, zoned disk technology. This is of great interest because the model is completely free of simulation and estimation-approximations – although obviously not of its approximating assumptions, which have been carefully checked [7]. All the constituent moments have been calculated directly, giving additional accuracy. We validated our new model against simulation results, considering different combinations of request sizes, request types and RAID variants in their mixtures. The excellent agreement we obtained suggests our model is a suitable tool to evaluate, in a short time compared to simulation, the performance of any configuration of a multi-RAID storage system, without requiring any approximated input parameters.

In the near future, we plan to compare our model with those developed by [15], [22], to estimate the accuracy obtained analytically as opposed to using interpolation and other approximations. We intend also to handle heterogeneous disks in the same storage system to increase the flexibility and scalability of our model.

References

[1] S. Chen and D. Towsley. A performance evaluation of raid architectures. *IEEE Transactions on Computers*, 45(10), October 1996.

[2] G. Gibson D. A. Patterson and R. H. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proceedings of SIGMOD Conference*, June 1988.

[3] J. Xu E. Varki, A. Merchant and X. Qiu. An integrated performance model of disk arrays. In *Proceedings of the International Symposium onModelling, Analysis and Simulationof Computer and Telecommunications Systems (MASCOTS)*, 2003.

[4] Fujitsu. *Disk Drives. Products/Maintenance Manual*. Fijitsu, 2001.

[5] C. Han and A. Thomasian. Performance of two disk failure tolerant disk arrays. In *International Symposium on performance Evaluation of computer and telecommunication systems*, 2003.

[6] P.G. Harrison and S. Zertal. Queueing models with maxima of service times. In *Proceedings of TOOLS Conference*, 2003.

[7] P.G. Harrison and S. Zertal. Calibration of a queueing model of raid systems. In *Proceeding of 2004 International Workshop on Practical Applications of Stochastic Modelling*, 2004.

[8] P.G. Harrison and S. Zertal. Queueing models of RAID systems with maxima of waiting times. *Performance Evaluation*, 2006. To appear.

[9] Ng Chee Hock. *Queueing Modelling Fundamentals*. John Wiley Publisher, 1996.

[10] P. Biswas K. K. Ramakrishnan and R. Karedla. Analysis of file I/O traces in commercial computing environments. In *Proc. Joint conf. on Measurement and Modelling of Computer Systems SIGMETRICS/PERFORMANCE*, June 1992.

[11] M. Y. Kim and A.N. Tantawi. Asynchronous disk interleaving: approximating access delays. *IEEE Transactions on Computers*, 40(7), 1991.

[12] E.K. Lee. *Perfomance modelling and analysis of disk arrays*. ph.D. thesis, University of California, Berkeley, USA, 1993.

[13] A. Merchant and P.S. Yu. An analytical model or reconstruction time in mirrored disks. *Performance evaluation*, 20, May 1994.

[14] A. Merchant and P.S. Yu. Analytic modeling of clustered raid with mapping based on nearly random permutation. *IEEE Transactions on Computers*, 45(3), March 1996.

[15] R. Nelson and A. N. Tantawi. Approximate analysis of fork-join synchronisation in parallel queues. In *IEEE Trans. Computers*, volume 37, June 1998.

[16] S. Christodoulakis P. Triantafillou and C. A. Georgiadis. A Comprehensive Analytical Performance Model for Disk Devices Under Random Workloads. *IEEE Transactions on Knowledge and data Engineering*, 14(1), 2002.

[17] Sangsoo Park and Heonshik Shin. *Rigourous Modeling of Disk Performance for Real-Time Applications*, volume 2986. Springer Berlin, 2004.

[18] C. Ruemmler and J. Wilkes. An introduction to disk drive modeling. *IEEE Computer*, 27(3), March 1994.

[19] A. Thomasian and A. N. Tantawi. Approximative solutions for M/G/1 Fork/Join synchronisation. In *Proceedings of the 1994 Winter Simulation Conference*, 1994.

[20] E. Varki. Mean Value Technique for Closed Fork-Join Networks. In *Proceedings of ACM SIGMETRICS conf. on Measurement and Modeling of Computer Systems*, May 1999.

[21] E. Varki and L. W. Dowdy. Analysis of Balanced Fork-Join Queueing Networks. In *Proceedings of ACM SIGMETRICS conf. on Measurement and Modeling of Computer Systems*, May 1996.

[22] S. Varma and A. M. Makowski. Interpolation approximations for symmetric fork/join queues. In *Performance Evaluation Journal*, volume 20, 1994.

[23] T.M. Wong and J. Wilkes. My cache or yours? making storage more exclusive. In *USENIX Ann. Technical Conference*, 2002.

[24] S. Zertal. *Dynamic redundancy mechanisms for storage customisation on multi disks storage systems*. ph.D. thesis, University of Versailles, France, January 2000.

[25] S. Zertal and P.G. Harrison. Multi-level raid storage system modelling. In *Proceedings of 2003 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (Spects)*, 2003.