An Analytic Performance Model of Disk Arrays

Edward K. Lee
University of California
571 Evans Hall
Berkeley, CA 94720
eklee@cs.berkeley.edu

Randy H. Katz
University of California
571 Evans Hall
Berkeley, CA 94720
randy@cs.berkeley.edu

Abstract

As disk arrays become widely used, tools for understanding and analyzing their performance become increasingly important. In particular, performance models can be invaluable in both configuring and designing disk arrays. Accurate analytic performance models are preferable to other types of models because they can be quickly evaluated, are applicable under a wide range of system and workload parameters, and can be manipulated by a range of mathematical techniques. Unfortunately, analytic performance models of disk arrays are difficult to formulate due to the presence of *queueing* and *fork-join synchronization*; a disk array request is broken up into independent disk requests which must all complete to satisfy the original request. In this paper, we develop and validate an analytic performance model for disk arrays. We derive simple equations for approximating their utilization, response time and throughput. We validate the analytic model via simulation, investigate the error introduced by each approximation used in deriving the analytic model, and examine the validity of some of the conclusions that can be drawn from the model.

1 Introduction

Disk arrays provide high I/O performance by striping data over multiple disks. High performance is achieved by servicing multiple I/O requests concurrently and by using several disks to service a single request in parallel. Given the increasing importance of disk arrays as high-performance secondary storage systems [7, 8, 12, 14, 15, 17], tools for understanding their performance become increasingly important. In particular, performance models, combined with a thorough understanding of an installation's workload, will be invaluable in both configuring and designing disk arrays. In general, accurate analytic performance models are preferable to other types of models, such as empirical and simulation, because they can be quickly evaluated, are applicable under a wide range of system and workload parameters, and can be manipulated by a range of mathematical techniques. Even when analytic models are not directly applicable to a particular system or workload, they are frequently useful for quickly analyzing general properties of the system, stimulating intuition and furthering understanding.

Unfortunately, analytic performance models of disk arrays are difficult to formulate due to the presence of queueing and fork-join synchronization; a disk array request is broken up into independent disk requests, all of which must complete to satisfy the disk array request. While systems with only of queueing *or* fork-join synchronization are frequently tractable given certain reasonable approximations, the combination of the two result in systems that are very difficult to analyze analytically. Exact analytic solutions for the two server fork-join queue given Poisson arrivals and independent service times currently exist [1, 4] but the k-server fork-join queue remains unsolved. Other related work in the field of disk array performance falls into four primary categories: (1) simulation studies [3, 12, 16], (2) analytic models that ignore queueing effects [2, 9, 17], (3) analytic models that ignore fork-join synchronization [8] and (4) restricted queueing models that deal with fork-join synchronization using specialized techniques not easily extended to modeling disk arrays [5, 13]. Most analytic queueing studies deal with general queueing systems rather than disk arrays in particular.

In this paper, we develop and validate an analytic performance model for asynchronous disk arrays. Our model is different from previous analytic models of disk arrays mentioned above for the following reasons. First, we use a closed queueing model with a fixed number of processes whereas previous analytic models of disk

arrays have used open queueing models with Poisson arrivals. A closed model more accurately models the synchronous I/O behavior of scientific, time-sharing and distributed systems. In such systems, processes tends to wait for previous I/O requests to complete before issuing new I/O requests, whereas in transaction based systems, I/O requests are issued randomly in time regardless of whether the previous I/O requests have completed. Second, to the best of our knowledge, we present the first analytic model for disk arrays that handles both the queueing at individual disks and the fork-join synchronization introduced by data striping. Previous analytic models handling both queueing and fork-join synchronization cannot easily be applied to disk arrays; these assume service times across servers (disks) are independent whereas in disk arrays, service times are very much dependent.

We have found the analytic model useful for comparative studies that focus on the relative performance of various disk array configurations. Lee [10] uses the analytic model presented here to derive a formula for the optimal size of data striping in disk arrays, providing an important application of the model. We are currently using the model to calculate quantitative price/performance metrics for disk arrays under a range of system and workload parameters. The model is also useful in identifying important factors in the performance of disk arrays and presents insights that are useful in characterizing the performance of real disk arrays. The model, of course, is not without limitations. Although the analytic model can be used to model both read and write requests to non-redundant disk arrays and read requests to a subclass of RAID's (Redundant Arrays of Inexpensive Disks) identified in Section 2, it currently does not model parity updates which occur during write requests to a RAID. Another limitation is that the workload model is not expressive enough to easily model real workloads. Thus, although the model can be used to accurately predict the performance of real disk arrays under many of the synthetic workloads used in this paper, its prediction of performance under real workloads may be highly approximate. A final limitation is that we only model the disk components of a disk array system. Real disk array systems have many other components such as strings, I/O controllers, I/O busses and CPU's that affect the performance of the system. We are currently addressing the above limitations in a work in preparation.

In the following sections, we first derive an exact expression for the utilization of the model system. Because the exact expression contains parameters that are difficult or impossible to compute, we analytically approximate the difficult parameters to make the expression more tractable. From the resulting approximate equation for utilization, we derive equations for response time and throughput. We then validate the analytic model via simulation and investigate the error introduced by each approximation made in deriving the analytic model. Next, we empirically verify certain conclusions that can be drawn from the analytic model and present an alternative empirical derivation of the analytic model. Finally, we derive and validate a simple extension to the basic analytic model which allows the modeling of variable sized request distributions.

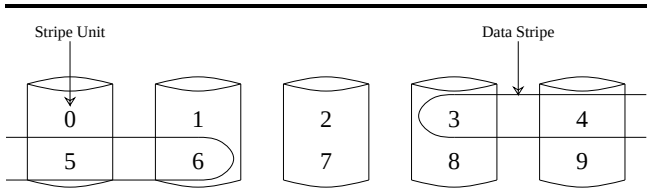

Figure 1: Data Striping in Disk Arrays.

Stripe unit is the unit of data interleaving, that is, the amount of data that is placed on a disk before data is placed on the next disk. Stripe units typically range from a sector to a track in size (512 bytes to 64 kilobytes). The figure illustrates a disk array with five disks with the first ten stripe units labeled.

Data stripe is a sequence of logically consecutive stripe units. A logical I/O request to a disk array corresponds to a data stripe. The figure illustrates a data stripe consisting of four stripe units spanning stripe units three through six.

2 The Modeled System

Our primary focus is on modeling non-redundant asynchronous disk arrays. Figure 1 illustrates the basic disk array of interest and the terms *stripe unit* and *data stripe*.

For readers who are familiar with the RAID taxonomy, we mention that the analytic model we develop can also be used to model reads for RAID level 5 disk arrays using the left-symmetric parity placement [11]. This is because the left-symmetric parity placement has the property that it does not disturb the *data* mapping illustrated in Figure 1.

3 The Analytic Model

In this section, we derive equations that approximate the read/write performance of non-redundant asyn-

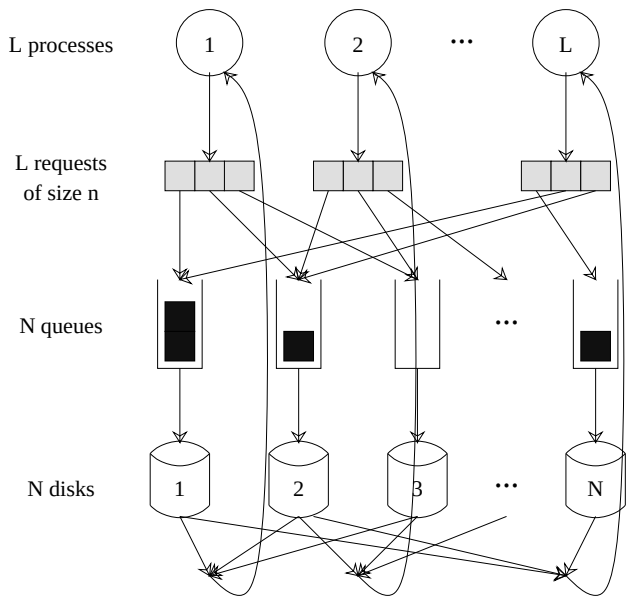

Figure 2: Closed Queuing Model for Disk Arrays. The parameters of the above system are as follows:

L	$\equiv$	Number of processes issuing requests.
N	$\equiv$	Number of disks.
n	$\equiv$	Request size in disks ($n \leq N$).
S	$\equiv$	Service time of a given disk request.

chronous disk arrays. Our approach is to derive the expected utilization of a given disk in the disk array. Because we are modeling a closed system where each disk plays a symmetric role with respect to each other, knowing the expected utilization of a given disk will allow us to compute the entire system's throughput and response time.

For the sake of convenience and clarity, we present the complete derivation of the analytic model before presenting the empirical validation of the model. However, the actual development of the model followed an iterative process consisting of an alternating sequence of analytical derivations and empirical validations. While some of the approximations made in this section, when separately presented from their empirical validation, may appear arbitrary, the approximations and results of the model are justified in Section 4.

3.1 The Model System

Consider the *closed* queueing system illustrated by Figure 2. The system consists of L processes, each of which issues, one at a time, an *array request* of size n stripe units. Each array request is broken up into n *disk requests* and the disk requests are queued round-robin starting from a randomly chosen disk. Each disk services a single disk request at a time in a FIFO manner. When all of the disk requests corresponding to an array request are serviced, the issuing process generates another array request, repeating the cycle. Note that two or more array requests may partially overlap on some of the disks, resulting in complex interactions. We sometimes refer to array requests simply as *requests*. In the derivation of the analytic model, we will assume that L and n are fixed. We will also assume that the processes do nothing but issue I/O requests. Later, we will extend the model to allow variable sized requests.

3.2 The Expected Utilization

In deriving the expected utilization of the model system, the following definitions will prove useful:

U	$\equiv$	Expected utilization of a given disk.
R	$\equiv$	Response time of a given array request.
W	$\equiv$	Disk idle (wait) time between disk requests.
Q	$\equiv$	Queue length at a given disk.
p_0	$\equiv$	Probability the queue at a given disk is empty when the disk finishes a disk request.
p	$\equiv$	n/N, probability of a request accessing a given disk.

If we visualize the activity at a given disk as an alternating sequence of busy periods of length S and idle periods of length W, the expected utilization of a given disk is,

$$U = \frac{E(S)}{E(S) + E(W)}. \tag{1}$$

Idle periods of length zero can occur and imply that another disk request is already waiting for service, $Q > 0$, when the current disk request finishes service.

Let r_0 denote the time between the end of service of a given disk request and the issuing of a new array request into the system. Let $r_i, i \in \{1, 2, \ldots\}$, denote the successive time intervals between successive issues of array requests numbered relative to r_0. Let M denote the number of array requests issued after a given disk finishes a disk request until, but excluding, the array request that accesses the given disk. Since each array request has probability p of accessing a given disk and disk requests are issued one at a time, M is a modified geometric random variable with $E(M) = 1/p - 1$. Figure 3 illustrates the above terms.

By conditioning on the queue length at the time a

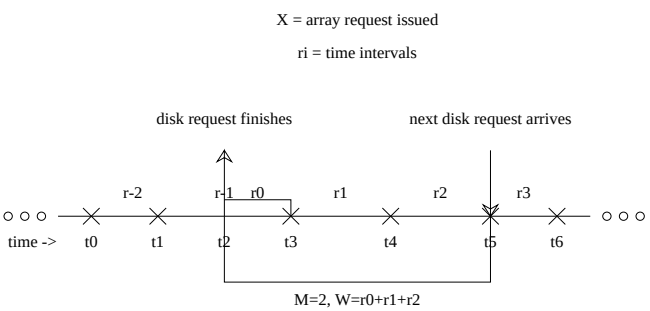

Figure 3: Time-line of Events at a Given Disk. After the disk request finishes service at time t_2, $M = 2$ array requests that do not access the given disk are issued at times t_3 and t_4 before an array request that accesses the given disk is issued at time t_5. The disk remains idle for a time period of $W = r_0 + r_1 + r_2$.

disk request finishes service, we can write,

$$\begin{aligned} E(W) &= P(Q>0)E(W|Q>0)+ \\ &\quad P(Q=0)E(W|Q=0), \\ E(W) &= (1-p_0)0 + p_0 E(\textstyle\sum_{i=0}^{M} r_i), \\ E(W) &= p_0(E(r_0) + E(\textstyle\sum_{i=1}^{M} r_i)). \end{aligned}$$

Substituting into Equation 1 we have,

$$U = \frac{E(S)}{E(S) + p_0(E(r_0) + E(\sum_{i=1}^{M} r_i))} \qquad (2)$$

Equation 2 is an exact, though not directly useful, equation for the expected utilization of the model system.

3.3 Approximating the Expected Utilization

In the previous section, we formulated an exact equation, Equation 2, for the expected utilization of the model system. Unfortunately, the exact equation consists of terms which are very difficult if not impossible to compute. In this section, we approximate components of Equation 2 to make it analytically tractable.

To simplify Equation 2, the first approximation we make is $E(\sum_{i=1}^{M} r_i) \simeq E(M)E(r_i) = (1/p - 1)E(R)/L$. From Little's Law, we know that the average time between successive issues of array requests is $E(R)/L$. Note also that M is a *stopping time*, that is, the event that the mth request misses the given disk, $M > m$, is independent of the random variables $r_{m+1}, r_{m+2}, \ldots$. Thus, from Wald's Equation, the above approximation would be exact if $r_i, i \in \{1, 2, \ldots\}$ were independently distributed with a common mean of $E(R)/L$. For the moment, we will take the above approximation as given, but later show via simulation that the above is an extremely good approximation.

The second approximation we make is to assume that $E(r_0) \simeq 0$. As motivation, we present the following observations concerning $E(r_0)$:

- $E(r_0) = 0$ implies that disk requests associated with the same array request finish at the same time and thus an array request is issued immediately whenever *any* disk request finishes.
- $E(r_0) = 0$ when $n = 1$, that is, when each array request consists of a single disk request, the completion of each disk request corresponds to the completion of the corresponding array request and, thus, the process that issued the disk request will immediately issue another array request.
- $E(r_0) \simeq 0$ when $n = N$, that is, when an array request always uses all the disks, disk requests associated with the same array request will tend to finish at close to the same time because all of the disks will be in very similar states and operate in a lock step fashion since disk service times are deterministic and disk requests across disks will be almost identical.
- Hopefully, $E(r_0)$ will not deviate too far from zero for values of n other than one or N.

The third and final approximation is $p_0 E(R)/E(S) \simeq 1$. This equation is true for $M/M/1$ systems and approximately holds at low to moderate loads for $M/G/1$ systems but the primary motivation for the approximation comes from empirical observations. Incorporating the three approximations, we can rewrite Equation 2 as,

$$U \simeq \frac{1}{1 + \frac{1}{L}(1/p - 1)}. \qquad (3)$$

Note that under the approximations we have made, *the expected utilization is insensitive to the disk service time distribution, S.*

The expected throughput in bytes per second can be written as,

$$T = \frac{UNB}{E(S)}, \qquad (4)$$

where B is the size of the stripe unit. Since this is a closed system, the expected response time can be expressed as,

$$E(R) = \frac{LnB}{T}, \qquad (5)$$

where $T/(nB)$ is simply the throughput in array requests per second. Future references to a specific analytic model will refer to the above equations and to Equation 3 in particular.

To briefly motivate the usefulness of the analytic model, consider the following applications of the model. An equation for the stripe unit size, B, that maximizes throughput can be computed from Equation 4 by differentiating it with respect to B and solving for local maxima. Another simple application is to solve Equation 5 with respect to N to get a feel for the number of disks needed to achieve a certain response time for a given workload.

4 Empirical Validation

In this section, we examine via simulation the accuracy of the analytic model and the error introduced by each approximation in the derivation of the analytic model. We also empirically examine the validity of the conclusions that can be drawn from the model. In particular, we will show that utilization is insensitive to the disk service time distribution and that utilization can be accurately represented by an equation of the form $U = 1/(1 + \frac{1}{L}f(p, N))$ where $f(p, N)$ represents an arbitrary function of p and N. Finally, we will extend the basic analytic model to handle variable sized requests and validate the result. Our basic approach is to compare the results of the analytic model to simulation using aggregate statistics which summarize errors and graphs which plot the analytic model versus the simulated data points.

4.1 Metrics for Error Analysis

When analyzing the errors in an analytic model versus simulation over a wide range of inputs, we must use aggregate statistics summarizing the errors because it is infeasible to compare directly the error resulting from each and every data point. The following defines standard statistical terms [6] and metrics that will be used in the rest of this paper. Note in particular the definitions of the three error metrics R^2, *max error* and *90 % error* which will be used to characterize errors in the analytic model.

response The result of a simulation run.

parameter A simulation variable that can influence the response.

factor A parameter that is being varied in a simulation study. In most studies, certain simulation parameters are held constant.

factor level A possible value for a factor. For example, the factor *request size* may have factor levels 4 KB, 8 KB, or 16 KB.

design An experimental design is constructed by specifying the factors, the factor levels used at each simulation run and the number of times the simulation is repeated with the same factor levels (usually using a different seed in the random number generator).

design point The factor levels for a single simulation run. For example, (number of disks = 8, request type = read, request size = 4 KB).

design set The set of design points used in an experimental design. For example, a design set can be constructed by taking a cross product of the following factor levels: number of disks $\in \{8, 16\}$, request type $\in$ {read, write}, request size $\in \{4, 8, 16\}$ KB.

The following defines the metrics R^2, *max error* and *90 % error*.

y_i	The ith simulated response.
$\hat{y_i}$	The ith analytically predicted response.
e_i	$y_i - \hat{y_i}$ (error of the ith input)
$\overline{y}$	Arithmetic mean of y_i's.
SSE	$\sum e_i^2$ (sum of squared errors)
SST	$\sum (y_i - \overline{y})^2$ (sum of squares total)
R^2	$(SST - SSE)/SST$ (coefficient of determination)
max error	max e_i (maximum error)
90 % error	The 90th percentile of e_i.

The coefficient of determination, R^2, is a measure of how closely the analytic model predicts the simulated response. A simple model predicting that the response is equal to the mean, $\overline{y}$, for all design points would have $R^2 = 0$. A model that perfectly predicts the simulated response would have $R^2 = 1$. R^2 can be interpreted as the fraction of "total variation" that is explained by a model. Although R^2 is a good measure for checking the overall accuracy of a model, it may be highly dependent upon the choice of the design set and may vary significantly for certain subsets of the design set. That is, an analytic model that is wildly inaccurate for a significant collection of design points can still have $R^2 \simeq 1$ if the model is accurate for the other design points. Because R^2 can be insensitive to large errors in the analytic model, the *max error* metric is useful as an indication of the worst predictions made by the model. The *90 % error* metric is a compromise between R^2 and *max error* which compensates for the extreme sensitivity of *max error* to outliers.

4.2 Experimental Design

We are interested in the following factors to validate our analytic model.

diskModel	Disk type.
N	Number of disks in the disk array.
L	Number of processes issuing I/O requests.
B	Size of the stripe unit (block size).
p	Request size as a fraction of the disk array.

The design set consists of the complete cross product of the following factor levels.

Factor	Factor Levels
diskModel	Lightning, Fujitsu, FutureDisk
N	2, 3, 4, 8, 16
L	1, 2, 4, 8, 16, 32
B	(1, 4, 16, 64) KB
p	$(1, 2, \ldots, N)/N$

The parameters for the three disk models listed above are described in Table 1. Note that the number of factor levels for p depend upon N and are expressed as multiples of $1/N$. This means, for example, that there are twice as many design points with $N = 4$ relative to $N = 2$. Thus, although we do not intrinsically value disk arrays with $N = 4$ more than disk arrays with $N = 2$, the design set implicitly assigns the former twice the importance of the latter. To compensate for this effect, we weigh all design points by the factor $1/N$ when calculating statistics.

We simulate each design point in the design set twice with two different random seeds to quantify the experimental error which intrinsically cannot be explained by any model. Thus, the total number of simulation runs is $2(3 \times 4 \times 6 \times (2 + 3 + 4 + 8 + 16)) = 4,752$. In each simulation run, the disks are rotationally synchronized, requests to the disk array are aligned on stripe unit boundaries, and L and p are held constant.

4.3 Validation of the Analytic Model

In this section, we examine via simulation the accuracy of the analytic model and the error introduced by each of the three approximations in the model of Section 3. Recall that the three approximations are as follows:

1. $E(\sum_{i=1}^{M} r_i) \simeq (1/p - 1)E(R)/L$,
2. $E(r_0) \simeq 0$,
3. $p_0 E(R)/E(S) \simeq 1$.

Consider the following definitions:

$$\hat{U} \equiv \frac{1}{1 + \frac{1}{L}(1/p - 1)}$$

$$\hat{U}_1 \equiv \frac{E(S)}{E(S) + p_0(E(r_0) + (1/p - 1)E(R)/L)}$$

$$\hat{U}_2 \equiv \frac{E(S)}{E(S) + p_0(\sum_{i=1}^{M} r_i)}$$

$$\hat{U}_3 \equiv \frac{1}{1 + \frac{1}{E(R)}(E(r_0) + \sum_{i=1}^{M} r_i)}$$

The variable $\hat{U}$ represents the analytic model derived in Section 3 by applying approximations 1, 2 and 3. The variables $\hat{U}_1$, $\hat{U}_2$ and $\hat{U}_3$ are submodels derived by applying only one of approximations 1, 2 or 3, respectively. By examining the errors in $\hat{U}_1$, $\hat{U}_2$ and $\hat{U}_3$, we can approximately study the errors introduced by each of the corresponding approximations.

The table below tabulates the error metrics discussed in Section 4.1 for the *best* empirical model, $\hat{U}$, $\hat{U}_1$, $\hat{U}_2$ and $\hat{U}_3$. The best empirical model is that model which minimizes the sum of squared errors or, equivalently, maximizes R^2 and is computed by averaging the responses of data points with the same design point. The error metrics for the best empirical model are useful for comparison purposes and give a feel for the magnitude of experimental errors that can not be explained by any model. The metric $1 - R^2$ is included for easier comparisons of models that have values of R^2 very close to one. Because we are more interested in relative rather than absolute errors, the error metrics are calculated for the logarithm of utilization rather than utilization directly. This means that for small errors, less than approximately 20 %, the *max error* and *90 % error* metrics can be interpreted as percentage deviations. For example, a *max error* of 0.0430 represents a 4.30% deviation from the *predicted* utilization.

Model	R^2	$1 - R^2$	*max err*	*90 % err*
BEST	0.9995	0.0005	0.0430	0.0133
$\hat{U}$	0.9814	0.0186	0.1863	0.0987
$\hat{U}_1$	0.9990	0.0010	0.0834	0.0192
$\hat{U}_2$	0.9888	0.0112	0.1676	0.0742
$\hat{U}_3$	0.9808	0.0192	0.1807	0.0951

The above table shows that 0.05 % of the variation in response is due to experimental errors which can not be explained by any model. The maximum experimental error is 4.30 %, and 90 % of all experimental errors are less than 1.33 %. The maximum error for the analytic model, $\hat{U}$, derived in Section 3 is 18.63 % with 90 % of errors less than 9.87 % of the predicted utilization. Finally, approximations 2 and 3 introduce most of the experimental errors, while approximation 1, as expected, introduces very few errors.

Figure 4 plots the response predicted by the analytic model $\hat{U}$ together with the 90 percentile intervals as a

	Lightning	Fujitsu	FutureDisk
bytes per sector	512	512	512
sectors per track	48	88	132
tracks per cylinder	14	20	20
cylinders per disk	949	1944	2500
revolution time	13.9 ms	11.1 ms	9.1 ms
single cylinder seek time	2.0 ms	2.0 ms	1.8 ms
average seek time	12.6 ms	11.0 ms	10.0 ms
max stroke seek time	25.0 ms	22.0 ms	20.0 ms
sustained transfer rate	1.8 MB/s	4.1 MB/s	7.4 MB/s

Table 1: Disk Model Parameters. Average-seek-time is the average time needed to seek between two equally randomly selected cylinders. Note that sustained-transfer-rate is a function of bytes-per-sector, sectors-per-track and revolution-time. *Lightning* is the IBM 0661 3.5" 320 MB SCSI disk drive, *Fujitsu* is the Fujitsu M2652H/S 5.25" 1.8 GB SCSI disk drive and *FutureDisk* is a hypothetical disk of the future created by projecting the parameters of the Fujitsu disk approximately three years into the future based on current trends in disk technology. The most dramatic improvements are in the bit and track density of the disks rather than in mechanical positioning times. Thus, disks in the future will have much higher sustained transfer rates but only marginally better positioning times. The seek profile for each disk is computed using the following formula:

$$seekTime(x) = \begin{cases} 0 & \text{if } x = 0 \\ a\sqrt{x-1} + b(x-1) + c & \text{if } x > 0 \end{cases}$$

where x is the seek distance in cylinders and a, b and c are chosen the satisfy the single-cylinder-seek-time, average-seek-time and max-stroke-seek-time constraints. The square root term in the above formula models the constant acceleration/deceleration period of the disk head and the linear term models the period after maximum disk head velocity is reached. If cylinders-per-track is greater than approximately 200, a, b and c can be approximated using the following formulas:

$$\begin{aligned} a &= (-10\ minSeek + 15\ avgSeek - 5\ maxSeek)/(3\sqrt{numCyl}) \\ b &= (7\ minSeek - 15\ avgSeek + 8\ maxSeek)/(3\ numCyl) \\ c &= minSeek \end{aligned}$$

where *minSeek, avgSeek, maxSeek* and *numCyl* correspond to the disk parameters single-cylinder-seek-time, average-seek-time, max-stroke-seek-time and cylinders-per-track, respectively. We have compared the model to the seek profile of the Amdahl 6380A published by Thisquen [18] and have found the model to closely approximate the seek profile of the actual disk. In practice, we have found the model to be well behaved, although care must be taken to check that a and b evaluate to positive numbers.

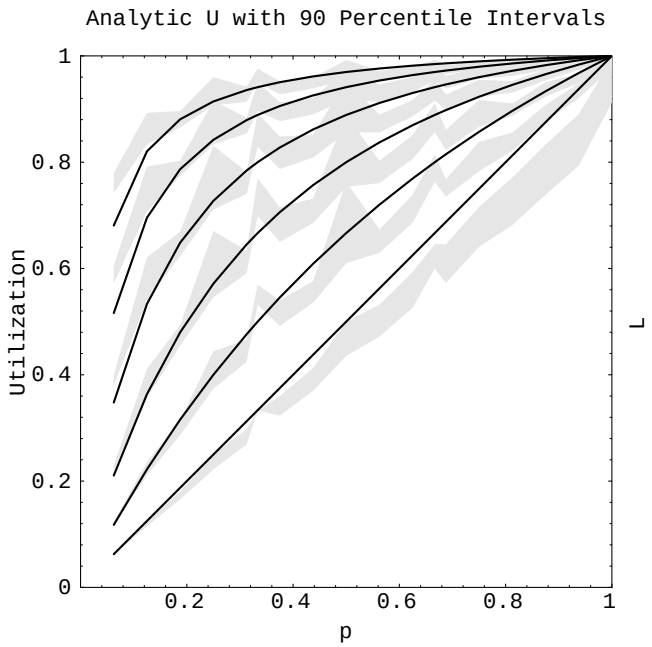

Figure 4: Analytic U with 90 Percentile Intervals as a Function of L and p. Each line and shaded region represents the analytically predicted response and empirically determined 90 percentile intervals, respectively, for a different value of L. From bottom to top the values for L are 1,2,4,8,16 and 32, respectively.

function of L and p, the only two factors considered important by the analytic model. In the figure, each 90 percentile interval encloses 90% of the weighted simulated responses for the given values of L and p. We have elected to give percentile intervals rather than confidence intervals because standard methods for calculating confidence intervals require that errors be normally distributed with a constant standard deviation. Certain transformations can be made on responses to satisfy the requirement but these transformations are often artificial. Percentile intervals, on the other hand, do not require any assumptions in the distribution of errors but do require a larger number of data points to calculate. Figure 4 shows that the analytically predicted utilization approximately passes through the 90 percentile intervals determined by simulation. As we will see later, the jaggedness of the percentile intervals is caused by the fact that the factor N, which is ignored by the analytic model, is significant in explaining variations in the response.

4.4 Validation of Model Properties

The previous section examined the errors in the analytic model, but regardless of the accuracy of the model, certain properties of the model may be valid where the model itself is not. This section will investigate the model's prediction that the utilization is independent of the disk service time distribution, S, or more specifically is dependent only on the factors L and p. We will also examine whether, as implied by the model, the utilization can be accurately modeled by an equation of the form $U = 1/(1+\frac{1}{L}f(p,N))$ where $f(p,N)$ represents an arbitrary function of p and N. Recall that $f(p,N)$ is equal to $1/p - 1$ in the analytic model.

4.4.1 Significance of Factors

The following table tabulates the error metrics for the best empirical models—models that maximize R^2—when certain factors are excluded. The first entry labeled *NONE* corresponds to the best empirical model that can be constructed when no factors are excluded and is identical to the model identified as *BEST* in the previous section. If the exclusion of a factor results in error metrics that are only slightly different from the error metrics of the *NONE* entry, this provides strong evidence that the factor can be safely ignored by the analytic model, at least over the range of factor levels investigated. The last two entries in the table illustrate the effects of excluding more than one factor at a time. As before, the metrics are calculated for the logarithm of utilization rather than for utilization directly.

Deleted Factors	R^2	$1-R^2$	*max err*	*90 % err*
NONE	0.9995	0.0005	0.0430	0.0133
diskModel	0.9990	0.0010	0.0570	0.0205
N	0.9936	0.0064	0.1653	0.0525
L	0.4238	0.5762	1.4311	0.4835
B	0.9986	0.0014	0.0789	0.0252
p	0.4285	0.5715	1.8733	0.5018
diskModel B	0.9983	0.0017	0.0843	0.0269
diskModel B N	0.9926	0.0074	0.1678	0.0577

As predicted by the analytic model, the above table illustrates that utilization is insensitive to the disk service time distribution, S, or more specifically to the two principle factors, *diskModel* and B, that determine S. Excluding both the factors *diskModel* and B result in error metrics *max error* = 8.43% and *90 % error* = 2.69% which compare favorably with the error metrics of the best case when no factors are excluded of *max error* = 4.30% and *90 % error* = 1.33%. Thus, we conclude that utilization is insensitive to the disk service time distribution, S.

The insensitivity of utilization to the disk service time distribution is hardly surprising given that we have a closed queueing system where processes perform nothing but I/O. Consider the following thought experiment. If we replaced all the disks with devices twice as fast but the same in other respects, we would expect throughput to exactly double but utilization to remain the same. Real systems are more complicated because changing disks or stripe unit sizes not only changes the mean of the service time distribution but also its shape. However, this simple thought experiment provides a valuable insight as to why utilization is insensitive to the disk service time distribution. It also provides a reason to believe that this is a generalizable property that also holds under a wide range of factor levels that we have not investigated in this paper.

Unlike excluding the factors *diskModel* and B, excluding factor N introduces significant errors. The errors, however, are not large enough to say that it is never acceptable to ignore N. Whether N can be ignored depends on the actual use of the model. If one is primarily interested in the effects of varying N, it may not be acceptable, but otherwise, it is probably acceptable. The table shows that the remaining factors, L and p, are clearly important and cannot be ignored.

4.4.2 Relationships Between N, L and p

The previous section identified the factors N, L and p as significant in formulating a model for the utilization of the modeled system. In this section, we empirically investigate the relationships between these factors. In particular, we examine whether, as implied by the model, the utilization can be accurately modeled by an equation of the form $U = 1/(1 + \frac{1}{L}f(p,N))$ where $f(p,N)$ represents an arbitrary function of p and N. We then search for values of $f(p,N)$ that result in accurate analytic models.

Figure 5 plots $\log(1/\overline{U}-1)$ versus $\log_2 L$ for the different factor levels of N and p where $\overline{U}$ is the geometric mean of the utilization over all design points with the same values of N, L and p. We use $\overline{U}$ rather than arbitrarily selecting design points with a specific value for *diskModel* and B to reduce experimental error. If $U \simeq 1/(1 + \frac{1}{L}f(p,N))$, a plot of $\log(1/\overline{U}-1)$ versus $\log_2 L$ should result in straight lines with a slope of -1. As is evident from the figure, this is approximately the case.

Now that we have verified that $U \simeq 1/(1 + \frac{1}{L}f(p,N))$, it remains to determine a good approximation for $f(p,N)$. Since f is a function only of p and N and not L, we can theoretically determine f by fixing L to any particular value. In particular, if $L = 1$, the resulting system has no queueing and it is evident that

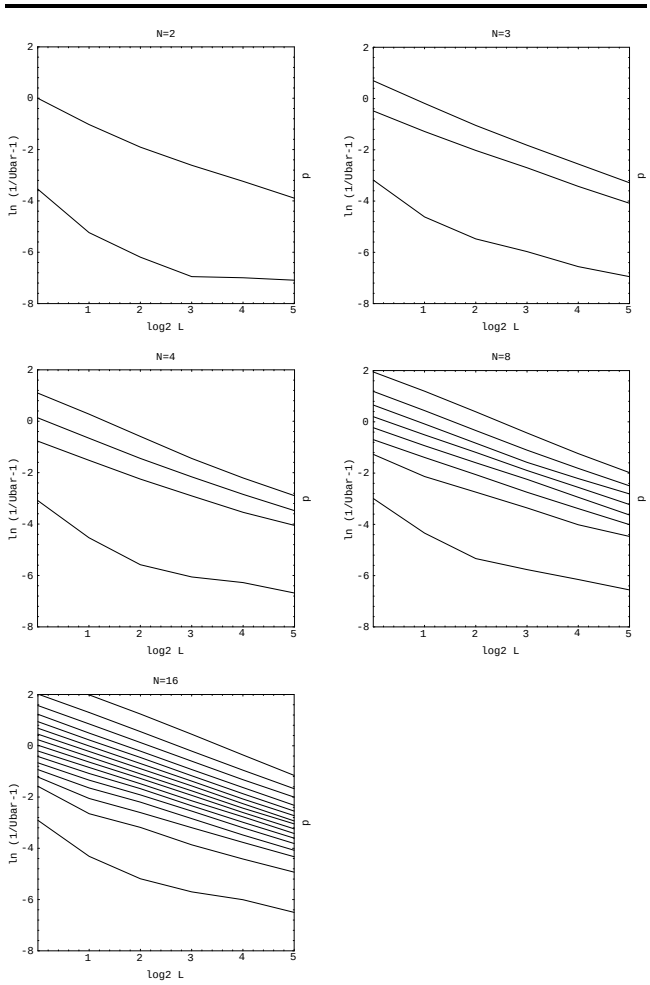

Figure 5: Plots of $\log(1/\overline{U}-1)$ vs. $\log_2 L$. Each plot corresponds to a different value of N. Within each plot, each line represents a plot of $\log(1/\overline{U}-1)$ versus $\log_2 L$ for a different value of p. From top to bottom the values for p are $1/N, 2/N, \ldots$, and N/N, respectively.

$U \simeq p$. Substituting, we have $p \simeq 1/(1 + \frac{1}{L}f(p,N))$ and solving for $f(p,N)$ we have $f(p,N) = 1/p - 1$. Note that this results in the same analytic model, $U \simeq 1/(1+\frac{1}{L}(1/p-1))$ derived in Section 3. The reader can look upon the above result as an alternative derivation of the analytic model based on empirical techniques.

Having rederived the analytical model above, two questions immediately arise:

1. Can we get a more accurate model by solving for utilization with $L = 2$ rather than $L = 1$? Theoretically, a solution to such a model would take into account a greater amount of the interaction between processes and should result in a more accurate model.

2. How good is the best model of the form $U \simeq 1/(1+\frac{1}{L}f(p,N))$?

The second question is easily answered empirically; we simply calculate the values of $f(p,N)$ which maximizes R^2 and examine the errors of the resulting model. The first question is more difficult to answer. Even for $L=2$, we must take into account both queueing and fork-join synchronization. We approximately model the system with $L=2$ as a discrete-time discrete-state Markov chain. If we assume that disk service times are constant, the number of states required to model the system is equal to the number of disks in the system. That is, the queue length at each disk is either zero or one, disks with a queue length of one are always consecutively located in the disk array, and the state where every disk has a queue can be merged with the state where no disk has a queue. We have formulated and solved such a model for arbitrary N. The solution is complex enough and would require sufficient explanation that it is not presented here. The model will be presented in a dissertation currently in preparation.

Let $\hat{U}_L$ represent the best empirical model of the form $U \simeq 1/(1+\frac{1}{L}f(p,N))$, let $\hat{U}_{L1} \equiv 1/(1+\frac{1}{L}(1/p-1))$, and let $\hat{U}_{L2}$ represent the approximate solution for utilization derived by assuming $L=2$. The table below tabulates the error metrics for $\hat{U}_L$, $\hat{U}_{L1}$ and $\hat{U}_{L2}$.

Model	R^2	$1-R^2$	*max err*	*90 % err*
$\hat{U}_L$	0.9929	0.0071	0.1299	0.0642
$\hat{U}_{L1}$	0.9814	0.0186	0.1863	0.0987
$\hat{U}_{L2}$	0.9814	0.0186	0.2176	0.0920

The error metrics of $\hat{U}_{L1}$, the analytic model derived in Section 3, compares favorably with the error metrics of $\hat{U}_L$, the best empirical model of the form $U \simeq 1/(1+\frac{1}{L}f(p,N))$. Somewhat surprisingly, The *max error* for $\hat{U}_{L2}$ is larger than that for $\hat{U}_{L1}$ although the *90 % error* is smaller. Although not visible from the table due to roundoff, R^2 for $\hat{U}_{L2}$ is slightly larger than that for $\hat{U}_{L1}$. We conclude that the additional complexities of $\hat{U}_{L2}$ does not, in general, merit its use over $\hat{U}_{L1}$.

4.5 Modeling Variable Request Sizes

In this section, we extend our model to handle variable sized workloads. Although we derived Equation 3 assuming a constant request size, it can be easily extended by noting that the parameter p is just the probability that a given request will access a given disk. Thus, given a workload which is f_1 fraction requests of size p_1 and $f_2 = 1-f_1$ fraction requests of size p_2, $p = f_1p_1 + f_2p_2$. In general, if x is the size of requests as a fraction of the disk array and $F(x)$ its corresponding cumulative distribution function. Then,

$$p = \int_0^1 x \; dF(x) = \overline{p} \tag{6}$$

where $\overline{p}$ is the average request size as a fraction of the disk array.

To validate the above result, consider the design set consisting of the following parameter and factor levels:

Parameter	Parameter Value
diskModel	Fujitsu
N	8
B	32 KB

Factor	Factor Levels
L	1, 2, 4, 8, 16, 32
p_1	$(2,\ldots,N)/N$
p_2	$(1,\ldots,p_1N-1)/N$
f_1	0.20, 0.40, 0.60, 0.80

Since we have shown in Section 4.4.1 that utilization is insensitive to *diskModel*, N and B, we hold them constant. In each simulation run, L processes randomly issue requests of size p_1, f_1 fraction of the time, and requests of size p_2, $1-f_1$ fraction of the time. Each design point in the design set is simulated twice with two different random seeds in order to quantify the experimental error. Thus, the total number of simulation runs is $2(6 \times 28 \times 4)) = 1344$. In each simulation run, the disks are rotationally synchronized, requests to the disk array are aligned on stripe unit boundaries and L is held constant.

The following are the error metrics from the experiment where $\hat{U} \equiv 1/(1+\frac{1}{L}(1/\overline{p}-1))$.

Model	R^2	$1-R^2$	*max err*	*90 % err*
BEST	0.9986	0.0014	0.0801	0.0192
$\hat{U}$	0.9969	0.0031	0.0934	0.0300

As can be seen from the table, the analytic model which only uses the factors L and $\overline{p}$ compares favorably with the best empirical model which uses all of the factors L, p_1, p_2 and f_1.

Figure 6 plots the response predicted by the analytic model $\hat{U}$ together with the maximum error intervals and individual data points as a function of L and $\overline{p}$. As can be seen from the figure, the analytically predicted response approximately passes through the simulated data points.

5 Summary and Future Work

We have derived and validated an analytic performance model for disk arrays. We initially modeled disk arrays as a closed queueing system consisting of a fixed

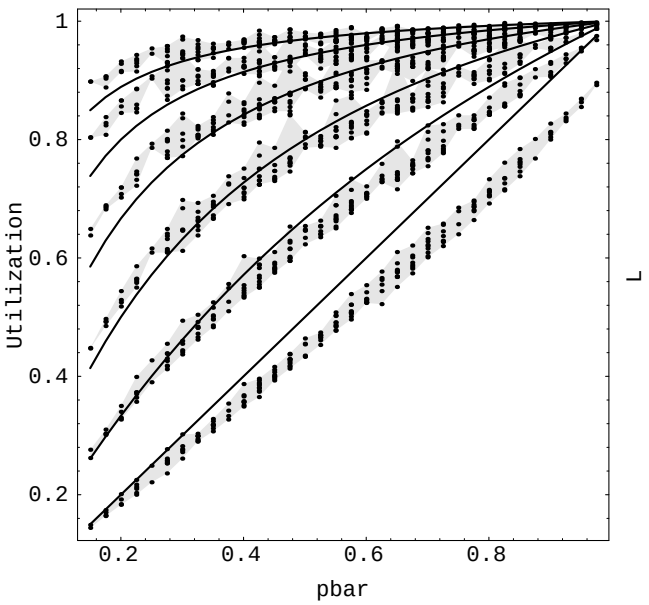

Figure 6: Analytic U with Maximum Error Intervals and Individual Data Points. In the above plot, $diskModel$ = Fujitsu, $N = 8$ and $B = 32$ KB. Each line, shaded region and set of datapoints corresponds to a different value of L. From bottom to top the values for L are 1,2,4,8,16 and 32, respectively.

number, L, of processes continuously issuing requests of a fixed size, p, to a disk array consisting of N disks. We then extended the model to handle variable sized requests. The resulting model predicts the expected utilization of the model system, U, as approximately $\frac{1}{1+\frac{1}{L}(1/\overline{p}-1)}$ where $\overline{p}$ is the average size of requests as a fraction of the number of disks in the disk array. We then directly derived the expected response time and throughput as a function of utilization. We showed via simulation that the simulated utilization is generally within $\pm 10\%$ of the utilization predicted by the analytic model. We also examined the error introduced by each approximation made in the derivation of the analytic model to better understand the validity of the approximations. Finally, we validated two results of the analytic model, namely that utilization is insensitive to the disk service time distribution and that utilization can be accurately represented by an equation of the form $U = 1/(1 + \frac{1}{L}f(p, N))$ where $f(p, N)$ represents an arbitrary function of p and N.

There are several major areas for future work with respect to the analytic model presented here. First, we plan to extend the workload model to handle something similar to CPU think time, where the processes, instead of simply issuing I/O requests, would alternate between computation and I/O. Second, we plan to extend the model to handle write requests to RAID's. Finally, we will apply the analytic model to solve problems in the design and configuration of disk arrays such as determining the optimal size for data striping, computing the price/performance ratios for various disk array and workload parameters, and quantifying tradeoffs between the performance and reliability of disk arrays.

6 Acknowledgements

We would like to thank our government and industrial affiliates, Array Technologies, DARPA/NASA (NAG2-591), DEC, Hewlett-Packard, IBM, Intel Scientific Computers, California MICRO, NSF (MIP 8715235), Seagate, Storage Tek, Sun Microsystems and Thinking Machines Corporation for their support. This material is based in part upon work supported by the National Science Foundation under Infrastructure Grant No. CDA-8722788.

References

[1] Francois Baccelli. Two parallel queues created by arrivals with two demands. Technical Report 426, INRIA—Rocquencourt France, 1985.

[2] Dina Bitton and Jim Gray. Disk shadowing. In *Proc. Very Large Data Bases*, pages 331–338, August 1988.

[3] Peter M. Chen and David A. Patterson. Maximizing performance in a striped disk array. In *Proc. International Symposium on Computer Architecture*, pages 322–331, May 1990.

[4] L. Flatto and S. Hahn. Two parallel queues created by arrivals with two demands i. *SIAM J. Appl. Math.*, 44:1041–1053, October 1984.

[5] Philip Heidelberger and Kishor S. Trivedi. Queueing network models for parallel processing with asynchronous tasks. *IEEE Trans. on Computers*, C-31:1099–1109, November 1982.

[6] Raj Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, Inc., 1991.

[7] R. H. Katz, G. A. Gibson, and D. A. Patterson. Disk system architectures for high performance computing. In *Proc. IEEE*, volume 77, pages 1842–1858, December 1989.

[8] Michelle Y. Kim. Synchronized disk interleaving. *IEEE Trans. on Computers*, C-35:978–988, November 1986.

[9] Michelle Y. Kim and Asser N. Tantawi. Asynchronous disk interleaving. Technical Report RC12497, IBM, January 1987.

[10] Edward K. Lee and Randy H. Katz. An analytic performance model of disk arrays and its application. Technical Report UCB/CSD 91/660, University of California at Berkeley, November 1991.

[11] Edward K. Lee and Randy H. Katz. Performance consequences of parity placement in disk arrays. In *Proc. ASPLOS*, pages 190–199, April 1991.

[12] Miron Livny, S. Khoshafian, and H. Boral. Multi-disk management algorithms. In *Proc. SIGMETRICS*, pages 69–77, May 1987.

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

[14] David A. Patterson, Peter M. Chen, Garth Gibson, and Randy H. Katz. Introduction to redundant arrays of inexpensive disks (RAID). In *Proc. IEEE COMPCON*, pages 112–117, Spring 1989.

[15] David A. Patterson, Garth Gibson, and Randy H. Katz. A case for redundant arrays of inexpensive disks (RAID). In *Proc. ACM SIGMOD*, pages 109–116, June 1988.

[16] A. L. Narasimha Reddy and Prithviraj Banerjee. An evaluation of multiple-disk I/O systems. *IEEE Trans. on Computers*, 38:1680–1690, December 1989.

[17] K. Salem and H. Garcia-Molina. Disk striping. In *Proc. IEEE Data Engineering*, pages 336–342, February 1986.

[18] J. Thisquen. Seek time measurements. Technical report, Amdahl Peripheral Products Division, May 1988.