A Performance Model of Disk Array Storage Systems

Elizabeth Varki* Shiela X. Wang
Department of Computer Science
University of New Hampshire
email: [varki,xw]@cs.unh.edu

Abstract

The distribution of data across multiple, parallel disks, known as data striping, is a popular method of data storage as it allows for faster access. In this paper, we develop an analytical model that estimates the mean response time, throughput, and mean queue length of data requests when data striping storage techniques are used. Our model is shown via simulations to provide a good approximation of the actual access time under a variety of system and workload parameters designed to emulate real-world conditions. Since analytical results can be readily computed (as opposed to empirical and simulation models), our model should prove useful to file and storage managers in their decisions with regard to the optimal placement of data on storage devices.

Index Terms: parallel input/output, RAID disks, performance evaluation, I/O performance, disk striping.

1 Introduction

Storage systems represent a growing market due to the enormous volumes of data generated and used by today's applications. To meet these storage demands, there have been many recent developments in the storage market. These include the development of *S*torage *A*rea *N*etworks (SANs) and *N*etwork *A*ttached *S*torage (NAS) which allow clients to by-pass the server and access storage devices directly. Regardless of the storage topology (i.e., a server-to-disk SCSI parallel bus architecture, or a SAN, or a NAS), almost all storage installations today contain *disk arrays.* Disk arrays provide improved I/O performance by distributing data across multiple disks in parallel. This organization of data that results in parallelism of I/O requests is called *striping*, and an array of disks organized in this format is called a disk array, or a *R*edundant *A*rray of *I*nexpensive *D*isks (RAID) [17]. In order to predict the cost and performance of today's computer and storage systems, it is necessary to understand the behavior of these disk array components and their interaction with the rest of the system.

In this work, we develop and validate a simple analytical performance model of disk arrays. Analytic models are useful for analyzing and understanding the properties of the system under study. Unlike empirical and simulations models, analytic models are inexpensive and can be evaluated very quickly. Analytic performance models are useful for predicting the performance measures of a system under a wide range of workload, operating system, and hardware parameters.

Due to the growing popularity of disk arrays in the past decade, several models of disk arrays have been developed. Each model assumes a different setting of some disk array policy. Kim and Tantawi[7] present an analytic method of approximating the sum of seek and rotational latencies for n disks having asynchronous interleaving. Chen and Towsley[3, 4] present a queueing model for the performance of RAID3 and RAID5 disks. Lee and Katz[10] present a queueing model of a non-redundant array of disks. Merchant and Yu[13, 14, 15, 16] present a number of queueing models for RAIDs in normal and recovery mode. Menon and Mattson[11, 12] present queueing models of RAID5 disks in normal, degraded, and rebuild modes. Thomasian and Menon[20, 21, 22] use a M/G/1 queueing model to analyze RAID5 disks. Kuratti and

*This research was supported in part by NSF under grant EIA-9974992

Sanders[8] compute the mean service time of an I/O request in a RAID5 for a transaction processing workload. None of these models predict the performance of the disk array in a workload specific manner. Also, many of these models are complex and do not scale well.

Previous work on analytical models of disk arrays typically study the disk system in isolation. Also, most previous analytical studies use open queueing models with Poisson arrivals to analyze disk arrays. The model developed here uses a closed queueing network with a fixed number of circulating processes in the system. Each process issues a I/O request and once the request is completed, a new request is generated and issued to the storage systems after some delay. A closed model is more appropriate to analyze such synchronous I/O requests where a process must block until the I/O request is complete. Extensive measurements of different UNIX systems show that synchronous requests account for $51-74\%$ of the total I/O workload on a system [18]. The only prior work (we are aware of) that uses closed networks to analyze disk arrays is the model developed by Lee and Katz [9]. However, their model assumes that the think time between I/O requests is zero and there are a constant number of outstanding requests to the storage system at all times. The model developed here analyzes the effect of non-zero think time. Our model is general and can be used to study the behavior of disk arrays regardless of the storage topology used. It can be used by file and storage managers to decide on the optimal data placement on storage devices.

Our disk array model is validated by comparing with simulated results. A limitation of this work is that the model is validated only for read workloads on RAID Level 5 disk arrays connected to the server using a SCSI parallel bus architecture. The behavior of read-modify and reconstruct writes is yet to be modeled. Nor have the performance modeling of disk arrays connected directly to SANs or network attached disk arrays been addressed here. We are currently addressing these limitations.

The remaining part of this paper is organized as follows: Section 2 presents the analytical model of disk arrays. Section 3 describes the hardware and simulation platform. Section 4 presents the model validation results, and conclusions and future work are given in Section 5.

2 The Disk Array Model

Disk arrays provide improved I/O performance by striping data across multiple disks in parallel. The data is interleaved by blocks which are referred to as *striping units*. A *logical* disk request issued by an application is divided into equally sized *physical* requests to the disk array. Each smaller physical request accesses a different disk, thereby reducing the amount of time to transfer the data. Parity is used in disk arrays to protect against disk failures. In RAID level 5, the parity is distributed across all the disks in the array. Thus, no single parity disk is a bottleneck and multiple small writes as well as multiple small read requests can be serviced in parallel.

Queueing network models are important tools in the design and analysis of computer and storage systems since they achieve a balance between accuracy and efficiency. The following subsections describe the queueing model of the disk array and give approximate equations for the mean performance measures of the modeled system.

2.1 The Queueing Model

The focus of this work is the modeling of synchronous I/O requests made to a disk array system. Processes issue I/O requests and block till the I/O request completes. Once the I/O request is completed, the blocked process is woken up. After a non-zero delay time, this process issues another I/O request and blocks and this cycle is repeated. Such systems are typically modeled by closed queueing networks. Figure 1 shows a high-level queueing model representation of the disk array system. Note that components like the device driver, array controller, and bus interconnects are not shown here. These can be added later or their effects can be included in the disk service time. The think time is modeled by a delay server (where there is no queueing). The disk array system is represented by a fork-join subsystem. Let K represent the number of disks in the disk array. Each of the K disks within the array is represented by a queueing center within the fork-join subsystem. Let s represent the striping unit size of the disk array. A request of size r arriving to the disk array system is striped (forked) across $k = \frac{r}{s}$ disks. It is assumed that the requests access data from adjoining disks of the array in a uniform

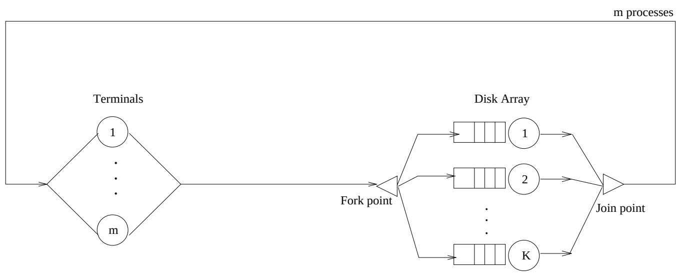

Figure 1: Queueing Model of the Storage System

pattern. The request is completed only after all its k sub-requests striped across the k disks are completed. A sub-request that completes "waits" for its siblings to finish. This waiting is represented by a join node in the fork-join model. The scheduling discipline at each of the disks is assumed to be first-in-first-out. Once all the sub-requests are serviced, the request completes and the corresponding process unblocks. This process, after a non-zero delay issues another I/O request and this cycle is repeated continuously.

The parallel fork-join structure is a classic problem that is extensively studied but has no general solution [25]. The synchronization constraints introduced by the forking and joining of sub-requests makes these networks non product-form [2]. Hence, none of the solution techniques developed for product-form networks can be used, making the modeling and analysis of parallel networks complicated. The modeling of disk arrays is further complicated by the fact that since a request is striped on any of the $k \leq K$ disks, two or more requests may partially overlap across some of the disks.

In [25], we have derived approximate mean performance measures for the queueing model described here. It is shown that the mean response time of a fork-join subsystem is approximately equal to:

$$R(m) \approx= S[O_k + A(m)]$$

where $R(m)$ represents the mean response time of the disk array subsystem when there are m processes in the closed network, S is the mean disk service time, O_kS represents the time taken to service the last sub-request, and $A(m)$ represents the mean number of waiting sub-requests in a disk queue seen by an arriving request. The term O_kS represents the mean service time of a request arriving to the disk array while the term $SA(m)$ represents the mean wait time of this arriving request. The mathematical derivation of this equation is beyond the scope of this paper and an interested reader is referred to [25].

Workload description	r : mean request size m : multiprogramming level (i.e., number of processes in the network). Z : think time (i.e., delay time)
Disk parameters	p : mean disk positioning time. t : mean disk transfer time. $S = p + t$: mean disk service time.
Disk Array parameters	K : number of disks in the array. s : striping unit size. $k = \frac{r}{s}$: number of disks that a request is striped onto. O_kS : mean of the k^{th} order statistic of sub-request service times. (i.e., the mean time taken to execute the last sub-request to complete execution.) The value of O_k depends on the distribution of disk service times.
Performance measures	R: mean response time of the disk array. X: mean throughput of the disk array. Q: mean number of requests in the disk array. A: mean number of sub-requests seen in a disk queue by an arriving sub-request.

Table 1: Notation

The notation used in the paper is summarized in Table 1. The next subsection shows how the values of S, O_k, and $A(m)$ are found for a disk array system under a specific workload.

2.2 Derivation of Disk Parameters

The parameter S represents the mean disk service time, which is the time required to service a sub-request issued to a disk. There are three intrinsic components of disk service time, namely, the *seek* time (the time taken to position the arm to the correct cylinder), the *rotational* latency (the time taken for the requested sector to rotate under the head), and the *transfer* time, t (the time taken for the actual transfer of the data). The sum of seek time and rotational latency is referred to as the *positioning* time, p. In addition, the disk service time may include bus *contention* time during which period data cannot be transferred due to absence of an I/O path. There are several papers that address the modeling of disk service times. These models differ in the assumptions made and the disk details modeled. Any of these models can be used to derive the disk service time of a disk array. In this work, we use the disk service time model presented by Shriver[19] and Barve et. al. [1] since their model addresses the dependence of disk service times on the submitted workload. In particular, they model disk service times under three types of workload, namely, random uniform access across the cylinders, sequential access with runs, and random uniform access across a restricted, contiguous subset of the cylinders. Their model is flexible and parameters like caching, controller overhead, and disk scheduling policy can be easily incorporated depending on the degree of accuracy required. Also, their model is valid for multi-zoned disks.

The parameter O_k refers to the scaling factor by which S must be increased in order to compute the disk service time of the last sub-request to finish. Since a request issued to a disk array is completed only when all the sub-requests finish, the service time of the last sub-request effectively represents the service time of the request. If there are k sub-requests, the service time of the last request is equal to the k^{th} order statistic of sub-request service times [23]. In order to compute the mean value of the k^{th} order statistic of sub-request service times, it is necessary to know the distribution of sub-request service times. Kim and Tantawi [7] show that positioning time can be approximated by a normal distribution. Under these conditions, the approximate mean value of the maximum of the positioning times of k disks is given by $p + \sigma\sqrt{2\log k}$, where σ is the coefficient of variation of the positioning time distribution. The transfer time of a disk is dependent on the disk transfer rate and the sub-request size and has a uniform distribution. Thus,

$$O_k S \approx= p + \sigma\sqrt{2\log k} + t.$$

The parameter $A(m)$ refers to the number of waiting sub-requests in a disk queue. Thus, $A(m)S$ gives the average wait time experienced by an arriving sub-request. In [25], it is shown that the mean number of requests in a disk array seen by an arriving request is approximately equal to the average number of requests in the disk array queue when the multiprogramming level of the network is one less (i.e., $A_{darray}(m) \approx= Q(m-1)$). Every request arriving to a disk array stripes into k sub-requests which are assigned to adjoining disks. For example, consider a request that is to be striped onto 3 disks arriving at a disk array consisting of 24 disks. Suppose the starting address of the request is disk 5. Then this request can be striped onto disks 5, 6, and 7. If the parity stripe unit for this stripe is on disks 6 or 7, the request will be striped onto disks 5, 6, 8 or 5, 7, 8, respectively. If the starting address is 23, the request will be striped onto disks 23, 24, and 1. It is assumed that requests access data throughout the disk array uniformly. Thus, the probability that a sub-request is assigned to a particular disk is given by $\frac{k}{K}$. For example, consider the cases when disk 6 will be accessed if it is given that $k = 3$. Disk 6 will be accessed if the starting address of the request is on disks 4, 5, or 6. Thus, for every access to a disk array, disk 6 will be accessed $\frac{k}{K}$ times. However, if the parity stripe unit for a particular row is on disks 4 or 5, then disk 6 will also be accessed for requests starting on disk 3. But, the parity stripe unit could also be stored on disk 6 and the request would then stripe on disks 4,5,7 or 5,7, 9. In this case, disk 6 is not accessed during a read operation. The last two cases effectively cancel each other out in the case of RAID level 5 disks where the parity is uniformly distributed. Thus, the effect of parity placement does not affect the probability of access to disk 6 in the case of Read only workloads under normal mode operation. For every access to the disk array, disk 6 will be accessed $\frac{k}{K}$ times. It follows that $A(m) = \frac{k}{K}Q(m-1)$.

The next subsection uses these parameters to compute the mean performance measures of a disk array.

2.3 Computation of Disk Array Performance Measures

The disk array performance measures are derived using the technique given in [24] which is briefly explained here. (Note that in [24], this technique is demonstrated only for fork-join systems where every request stripes into exactly K sub-requests which are assigned to unique disks. Also, the service time distributions in [24] are assumed to be exponential.) The technique involves iteratively solving the equations

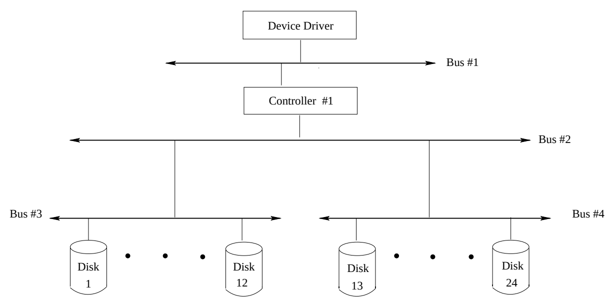

Figure 2: HP C2247 Disk Drive Configuration

$$R(m) \approx= O_k S + \frac{k}{K} SQ(m-1).$$

From Little's law, the throughput and queue length of the disk array are given by:

$$X(m) = \frac{m}{R(m)+Z}$$

$$Q(m) = R(m) * X(m)$$

for $m = 1, 2, \cdots$ where the initialization of the iteration is $Q_i(0) = 0$ for the disk array subsystem. The model is validated using simulations in the next section.

3 Simulation Platform

Disksim [5], a detailed, strongly validated disk simulator is used to validate our model. The simulator accurately models zoned recording, spare regions, disk caches, bus delays, and controller overheads. In this section, the storage system and the workload used in the simulations are described.

3.1 Hardware Description

For this work, the simulator was configured to model the the HP C2240 series of disk drives [6]. Table 2 summarizes the hardware specifications of the disk.

The measured simulations are performed on a RAID level 5 disk array consisting of 24 HP C2240A disks connected to the main system via a SCSI bus architecture. The configuration is shown in Figure 2. The parity blocks of the disk array have the same size as the striping unit and are rotated between the disks. The RAID level 5 disk array is arranged into 3 groups of 8 disks each. Thus, the parity stripe unit consists of 7 data striping units and 1 parity striping unit. Thus, the effective capacity of the disk array is equivalent to the capacity of 21 disks. For all the simulations, it is assumed that the parity is rotated in the left symmetric format [9] among the disks. However, our model is not dependent on the organization of parity units since the effect of parity is captured in the probability of disk accesses. A change in the organization of parity would require only a change in the computation of probability of disk accesses.

Disk capacity	1.05 GB
Rotation Speed (in rpms)	5400
Data Surfaces	13
Cylinders	2051
Sectors	2054864
Zones	8
Sectors/Track	56-96
Interface	SCSI-2

Table 2: HP C2247 Disk Drive Specifications

3.2 Workload Description

We use synthetically generated random read-only workloads to validate our model. The workload parameters we varied are:

- m : the multiprogramming level (which is the number of processes circulating in the closed disk array network). The number of processes is fixed during each simulation run. Each process repeatedly issues an I/O request, waits for its completion, spends some non-zero delay time, and then issues another I/O request. For this work, we ran simulations with multiprogramming levels ranging from 1 to 20.
- r : the request size, which is fixed for each simulation run. The distribution of request sizes used in the simulations are:
  1. an exponential distribution with a mean of 128KB.
  2. a normal distribution with a mean of 512KB.
  3. a normal distribution with a mean of 4MB.

4. an uniform distribution with a mean 0f 2MB.

The requests are uniformly distributed throughout the disk array. The exponential workload represent workloads generated by time-sharing and transaction-processing applications. The normal distributions represent workloads generated by multimedia and scientific applications.

- k : the number of disks across which the data that the I/O request is accessing is striped. This parameter is fixed during each simulation run. The striping unit sizes are adjusted according to the request size so that the data are striped across 1 to 21 disks.

In all the simulations, data are randomly read from the disk array. The disk scheduling discipline is set to first-in-first-out. The multiprogramming levels, request sizes, and stripe unit sizes are varied in each simulation run. Two sets of simulations are run, the first set with no delay time (i.e., there is a constant number of outstanding requests in the disk array) and the second set with delay time. The delay times are fixed and are drawn from an exponential distribution with a mean of 30 time units.

4 Model Validation

The primary performance metric used for comparing our model predictions is the mean response time of the disk array subsystem. The mean response time of a request issued to a disk array is defined to be the time from the moment the request is issued until the time its last sub-request finishes. The throughput and mean queue length of the disk array can be easily computed using the equations given earlier.

Figures 3 and 4 plot the disk array response time curves for varying request sizes striped across $k = 1, 2, \cdots, 21$ disks as the multiprogramming level changes. The simulations are set such that the simulated mean response time estimates are accurate within 1.0 time units at 95% confidence. As the graphs show, the model response time curves are very close to the simulated curves. The model response time estimates are, on average, within 6% of the simulated response time values.

The graphs clearly indicate that request size and multiprogramming levels have a significant impact on the performance of a disk array. For all request sizes, as the multiprogramming level increases the benefit from striping decreases. It is better to reduce the amount of data striping via an increase in the striping unit size so that more I/O requests may be serviced concurrently. The graphs also indicate that the addition of delay between requests has the same affect as decreasing the multiprogramming level at the disk array. More generally, the disk utilization has significant impact on the performance.

5 Conclusions

We have developed an analytical performance model of disk arrays under synchronous I/O workloads. The system is modeled as a close network with a delay server. The model is validated for read requests and the model estimates are found to be very close to the simulated estimates.

Currently, we are extending the model to handle write requests. We are also planning to incorporate the effects of caching and bus contention. The advantage of our model is that it is flexible. As shown here, it is relatively easy to incorporate the results of prior disk modeling work into our model. We plan to model the physical components of the device, such as the caches and the controllers and then model the disk array by composing these component models together. The validity of the model will be tested using synthetic workloads and traces of real workloads.

References

[1] Barve, R., Shriver, E., Gibbons, P.B., Hillyer, B.H., Matias, Y., Vitter, J.S., "Modeling and optimizing I/O throughput of multiple disks on a bus", Proceedings of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Atlanta, GA, 1999.

[2] Baskett, F., Chandy, K.M., Muntz, R.R., Palacios, F.G. "Open, closed, and mixed networks of queues with different classes of customers", Journal of the

Association of Computing Machinery, vol. 22, no. 2, April 1975, pp. 248 – 260.

[3] Chen, S., Towsley, D. "The design and evaluation of RAID 5 and parity striping disk array architectures", Journal of Parallel and Distributed Computing, 17, 1993, pp. 58 – 74.

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

[5] Ganger, G.R., *System-oriented evaluation of I/O subsystem performance*, Ph.D. thesis, University of Michigan, Ann Arbor, MI, June 1995.

[6] Hewlett-Packard Company, "HP C2240 series 3.5-inch SCSI-2 disk drive, Technical Reference Manual", Part number 5960-8346, Edition 2, April 1992.

[7] Kim, M. Y., Tantawi, A. N. "Asynchronous disk interleaving: approximating access delays", IEEE Transactions on Computers, vol. 40, no.7, July 1991, pp. 801 – 810.

[8] Kuratti, A., Sanders, W.H., "Performance analysis of the RAID 5 disk array", Proceedings of International Computer Performance and Dependability Symposium, April 1995, pp. 236 – 245.

[9] Lee, E. K., Katz, R. H., "Performance consequences of parity placement in disk arrays", ACM International Conference on Architectural Support for Programming LAnguages and Operating Systems, 1991, pp. 190-199.

[10] Lee, E. K., Katz, R. H. "An analytic performance model of disk arrays" Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, May, 1993.

[11] Menon, J., Mattson, D., "Performance of disk arrays in Transaction processing environments", Proceedings of 12th International Conference on Distributed Computing Systems, IEEE Computer Society Press, Los Alamitos, CA 1992.

[12] Menon, J., "Performance of RAID5 disk arrays with read and write caching", Distributed and Parallel Databases, 2(3), July 1994, pp. 261 – 293.

[13] Merchant, A., Yu, P. S., "Performance analysis of a dual striping strategy for replicated disk arrays", Proceedings of 2nd International Conference on Parallel and Distributed Information Systems, 1993.

[14] Merchant, A., Yu, P. S., "An analytical model of reconstruction time in mirrored disks", Performance Evaluation, 20(1-3), May 1994.

[15] Merchant, A., Yu, P. S. "Analytic modeling and comparisons of striping strategies for replicated disk arrays", IEEE Transaction on Computers, vol. 44, no. 3, March 1995, pp. 419 – 433.

[16] Merchant, A., Yu, P. S. "Analytic modeling of clustered RAID with mapping based on nearly random permutation", IEEE Transaction on Computers, 45(3), March 1996, pp. 367 – 373.

[17] Patterson, D., Gibson, G., Katz, R.H., "A case for redundant arrays of inexpensive disks (RAID)", Proceedings ACM SIGMOD Conference Management of Data, June 1988.

[18] Ruemmler, C., Wilkes, J., "UNIX disk access patterns", Winter USENIX Conference, January 1993, pp. 405 - 420.

[19] Shriver, E., *Performance modeling for realistic storage devices*, Ph.D Thesis, New York University, 1997.

[20] Thomasian, A., "Priority queueing in RAID5 disk arrays", RC 19734, IBM I.J. Watson Research Center, Yorktown Heights, NY, October 1994.

[21] Thomasian, A., Menon, J., "Performance analysis of RAID5 disk arrays with a vacationing server model for rebuild mode operation", Proceedings of the 10th International Conference on Data Engineering, February 1994, pp. 111 – 119.

[22] Thomasian, A., Menon, J., "Approximate analysis for fork-join synchronization in RAID5", Computer Systems: Science and Eng., 1997.

[23] Trivedi, K.S., *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, 1982.

[24] Varki, E., "Mean value technique for closed fork-join networks", Proceedings of ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Atlanta, GA, May 1999, pp. 103 – 112. Also, Performance Evaluation Review, 27, 1.

[25] Varki, E., "Response time bounds of parallel computer and storage systems", submitted to IEEE Transactions on Parallel and Distributed Systems, 2000.

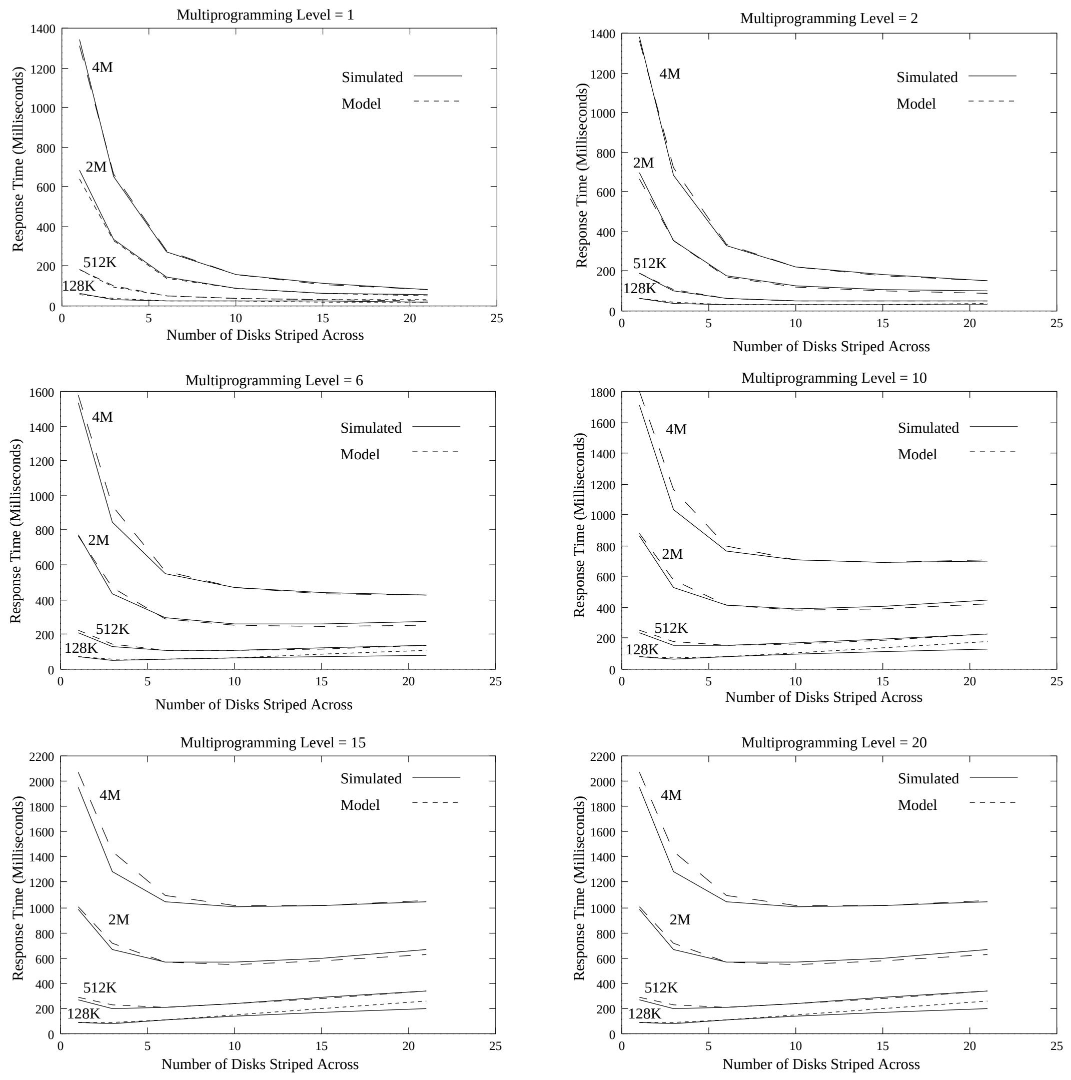

Figure 3: RT of Disk Array when Delay Time = 0

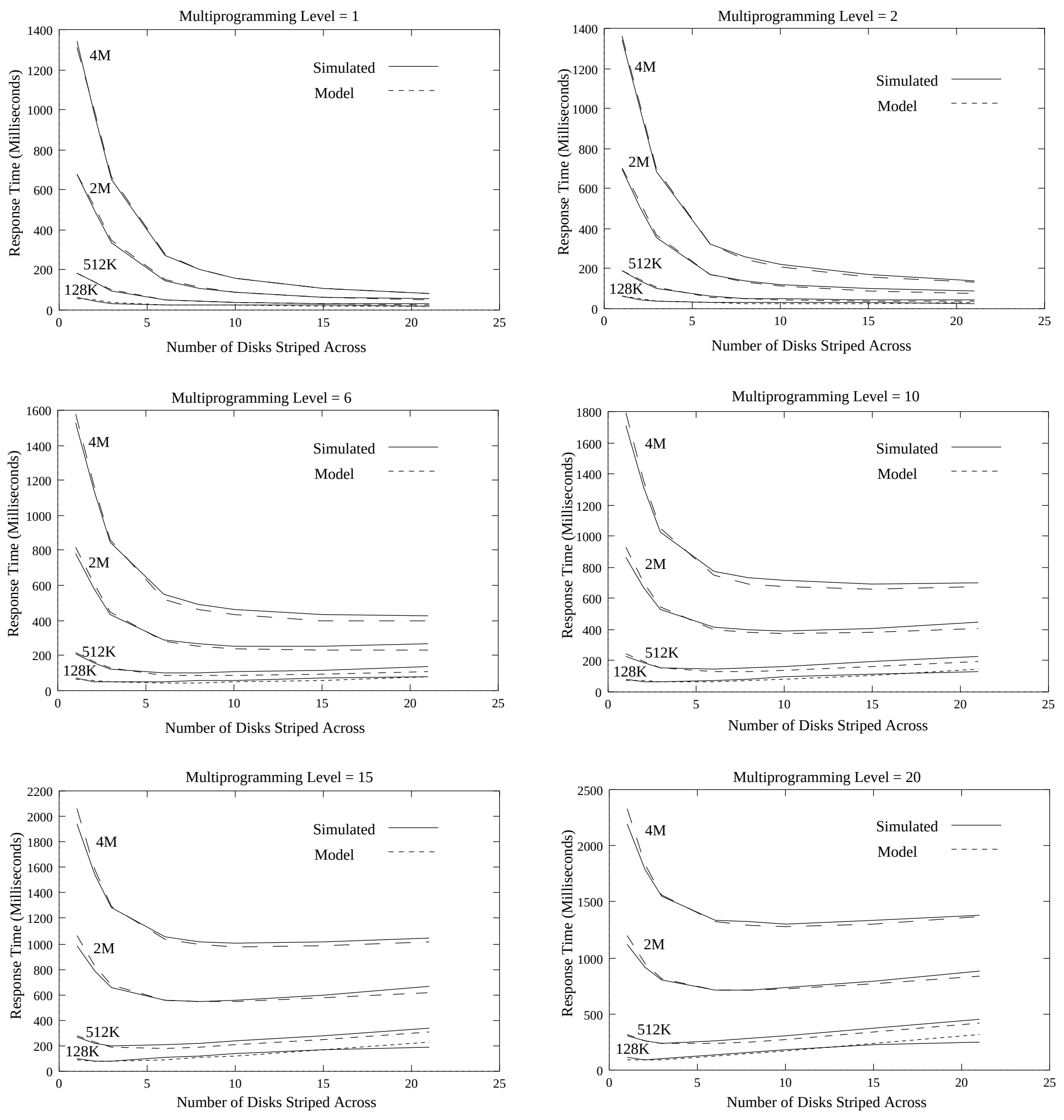

Figure 4: Response Time of the Disk Array when Delay Time = 30 time units