Proceedings of the IASTED International Conference
Parallel and Distributed Computing and Systems (PDCS '99)
November 3-6, 1999, Cambridge, Massachusetts - USA

AN ADAPTIVE MODEL FOR PARALEL I/O PROCESSING

Athena I. Vakali and Georgios I. Papadimitriou

Department of Informatics,
Aristotle University of Thessaloniki, Box 888,
54006 Thessaloniki, Greece.

Abstract

Modern I/O subsystems have increased capacity and density but their performance has not been improved accordingly. To address this problem we have developed a new model that is based on feedback information provided to the I/O subsystem controller. The presented model is applied to a multiple disk drive subsystem which serves requests in parallel. Under conventional servicing, each request refers to a specific drive and is placed on the corresponding disk drive queue in order to be serviced. The proposed feedback-based model redirects requests among disk drives towards performance gain. The feedback is evaluated by a metric identified by the queue length per disk drive. The request servicing in a parallel disk drive subsystem is simulated and simulation runs measure both conventional and feedback-based servicing. The simulation results validate the presented model and prove that it shows an important improvement in both seek and servicing times compared with the conventional request servicing model.

Keywords: I/O subsystems, parallel I/O, secondary storage, disk drive performance, adaptive models.

1. Introduction

Modern I/O subsystems are reinforced with quite efficient mechanisms implemented as policies that perform scheduling, reordering of I/O requests or read-ahead. The current complicated storage systems infrastructure hardens the development of analytic as well as simulation models. Disk controller has been considered as the most suitable component for hosting storage systems policies and current technology provides efficient controllers with respect to the disk drive's functionality. Most disk controllers are reinforced with self-managing techniques through standard interfaces used on standard systems without software modifications [1].

This paper presents a new model in solution to the problems of I/O bottleneck and I/O request servicing. Our approach is based on the following important issues:

- most current storage systems support multiple drives and a queue of requests is associated with each drive,
- the response time could be improved by redirecting requests to idle drives or to drives with lighter load,
- the information provided by each disk drive's queue could be used as feedback in order to perform the request redirection,
- the storage system could be self-managed and the workload could be served more efficiently.

The remainder of the paper is organized as follows. The next section presents the multiple disks I/O storage subsystem configuration and identifies the most crucial performance factors. Section 3 introduces the proposed feedback model and analyzes the request servicing policies as well as the performance metrics and their impact on improving the storage system responsiveness and functionality. Section 4 presents the simulation results and discusses the performance gain in the presented model. Finally, conclusions are summarised in Section 5.

2. Multiple disk drives I/O Subsystem

Several I/O subsystems have been suggested in modern parallel and distributed systems. Most of these assumes the hierarchical memory model proposed in [2] where an abstract machine consists of a set of processors interconnected via a high-speed network and each processor access an appropriate I/O controller. Each of these controllers manages a set of disk drives. The controller is responsible for managing and directing read/write requests to the queues of the disk drives. Most modern magnetic disks have an embedded Small Computer Systems Interconnect (SCSI) controller. Here, we concentrate on a multiple disks subsystem where disk drives are managed by a common I/O controller. Disk drives are of the same type and have similar configuration requirements (Figure 1).

2.1. Device and Workload Characteristics

A typical data storage hierarchy includes main memory, magnetic disks and possibly tape drives or tertiary storage

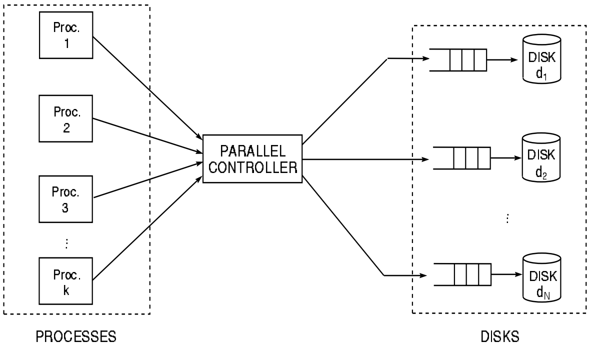

Figure 1: Multiple Disks Subsystem model.

devices. The host operating system guides an I/O request to the disk drive controller via the appropriate disk drivers. The controller manages the storage and retrieval of data to and from the disk mechanism and performs mappings between incoming logical addresses and the physical disk location that stores the data [4]. The storage subsystem comprises of N individual disk drives which store information and each one is considered as an independent drive. Each disk drive is equipped with one read/write head per disk surface and the head moves to the appropriate cylinder location in order to serve the current request. *Reading* data is satisfied by accessing the disk which contains the requested data whereas *writing* is performed by the disk specified by the file system (as explained next). Disks serve requests in parallel in order to exploit the system's responsiveness. Disk drives have associated queues that contain requests waiting to be serviced. Current controller interfaces support command queueing, which permits the reordering of queued requests in order to improve disk performance (e.g. SCSI-2).

Requests arrive to the system randomly by various independent processes. Some requests arrive while others are being serviced, and so queues are created in each disk drive. Requests arrival rate could be either constant or independent and exponentially distributed or bursty. The disk controller commands the drives to serve the requests posed by the file system. A typical request consists of the following attributes:

- *Device*: the id of the disk drive to serve the request,
- *Operation*: either R(ead) or W(rite),
- *Start Location*: the physical address where the data are(will be) located,
- *Size*: the amount of data(in MBytes) to be read(written),
- *Arrival Time*: the time when the request arrived at the controller.

According to the above request pattern, the controller will direct each request to the appropriate drive in order to be served. As depicted in Figure 1, each disk drive in the subsystem is an individual "server" and has its own queue. Therefore, the disk controller is the common server whose service is required by all I/O operations.

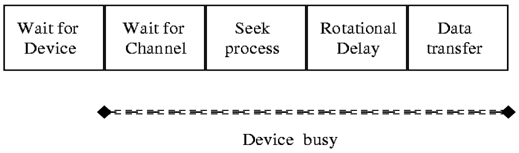

Figure 2: Timing for a Disk I/O Transfer.

2.2. Performance Metrics

Given the arrangement of disk surfaces and read/write heads, the time required for a particular I/O operation involves mainly the following actions (Figure 2) [3]:

- wait in queue : time spent in queue waiting for the drive to be free for servicing the I/O,
- wait for channel : time spent waiting for the channel to be free such that the seek and sector information can be sent down,
- seek time : time spent to move the appropriate head to the appropriate cylinder,
- latency time : time spent for the required sector to rotate around to the location of the head,
- transfer time : time to perform the actual data transfer.

Disk performance is measured by specific metrics based on the above times spent at each phase of the request servicing process. The overall time for executing and completing a user request consists of command overhead, seek time, rotational delay and data transfer time. Command overhead has been reducing due to the acceleration of disk controller's chips and mechanisms. Therefore, the service time of a request in the disk mechanism is a function of the seek time(ST), the rotational latency(RL) and the transfer time(TT) whereas queue delay must be considered also for the evaluation of the overall service time [5, 6]. The most widely used formula for evaluating the expected service time involves these time metrics and it is expressed by :

$$\mathrm{E}[ServiceTime] = \mathrm{E}[ST] + \mathrm{E}[RL] + \mathrm{E}[TT] \quad (1)$$

where $\mathrm{E}[ST]$ refers to the expected seek time, $\mathrm{E}[RL]$ refers to the expected rotational delay and $\mathrm{E}[TT]$ refers to the expected transferring time.

Seeking is a major performance factor and several expressions have been suggested for expected seek time evaluation. While seeking, the read/write head arm is involved in the operations of speedup, coast, slowdown and settle, successively in order to reach the requested location. The speedup time will be the dominant factor for short seeks whereas the coast is the dominant factor for long seeks. The following function has been used widely for the approximate evaluation of the seek time :

$$Seek\_Time(dist) = \begin{cases} 0 & \text{if } dist = 0 \\ a + b\sqrt{dist} & \text{if } 0 < dist < cutoff \\ c + d\ dist & \text{if } dist \geq cutoff \end{cases} \quad (2)$$

where a, b, c, d and $cutoff$ are device-specific parameters and $dist$ is the number of cylinders to be traveled. Furthermore, a closed formula has been derived for the expected seek time ($\mathrm{E}[ST]$) under random uniform access [5].

The expected rotational delay is evaluated by $\mathrm{E}[RL] \approx \frac{Revolution\_Time}{2}$ for randomly distributed requests. The transfer time depends on the amount of data to be transfered and is evaluated by $\mathrm{E}[TT] = \frac{Request\_Size}{Transfer\_Rate}$ under a constant $Transfer\_Rate$.

3. The Feedback Model

Our work is based on the idea of redirecting requests at the disk controller, based on provided feedback in order to improve disks performance. Controller is suggested as a better suited place for the task of reorganizing data [1].

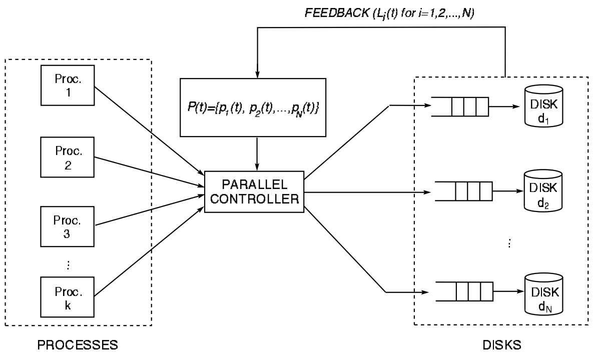

Figure 3: The Feedback-Based I/O subsystem.

Here, the controller doesn't remain a static design where data are placed as directed by the file system. Instead the controller becomes a dynamic tool which efficiently re-directs the requests to the physical medium according to the feedback information. Figure 3 presents the structure of our feedback-based model which revises the basic multiple disk drives model (presented in Figure 1) according to the proposed feedback. Each request is a either a read or a write process guided to the controller.

Our model is based on the distinction of the requests by their type (Read or Write). The feedback information is used in order to redirect the write requests from heavy loaded disk drive queues to other disk queues which remain idle or are lightly loaded. As presented in Figure 4, the feedback-based controller alters the original request pattern such that the attributes of device and start location are adapted to the systems load in case of writing. Therefore, write service is performed by indirecting request pattern within the controller to allow data relocation such that the service time is improved.

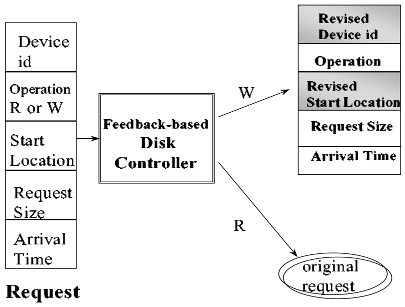

Figure 4: the Feedback-Based Servicing policy.

4. Simulation Results

We have built an event based simulator to validate the feedback-based I/O servicing model. In order to study the performance of the proposed model we have implemented the conventional model and two types of models for the feedback-based model, namely the deterministic and the probabilistic model. According to the deterministic model requests are redirected by determining the minimum loaded disk (used as feedback) each time, whereas under the probabilistic model the disk choice depends on the disks probability distribution (as described in the previous section).

Each simulation run considered arrivals of more than 500,000 requests over the simulated time. The simulation model was studied for an I/O subsystem of $2, 4, \ldots, 10$ disk drives. Each disk is configured by the characteristics proposed in [4, 5] for the *HP 97560* disk drive. The workload is characterized by the arrival process, the request rate, bursty arrivals and the fraction of read and write processes. The read/write ration was also a parameter for the simulation process and there are different arrival sets depending on the probability of having reads to vary within the range $0.1, \ldots, 0.9$.

The proposed feedback-based model showed to be beneficial in all cases when compared to the conventional I/O servicing model. Figures 5 and 6 present the improvement rates of the deterministic and probabilistic over the conventional model, respectively. These rates refer to the improvement in service time as evaluated by equation 1. The curves represent the results when reading probability was $0.1, 0.3, 0.5, 0.7, 0.9$, under I/O subsystems of $2, 4, 6, 8, 10$ disk drives. As it was expected, the most beneficial improvements in service time result when having low read ratio and many parallel disk drives, since there is an increased exploitation of the service parallelism and

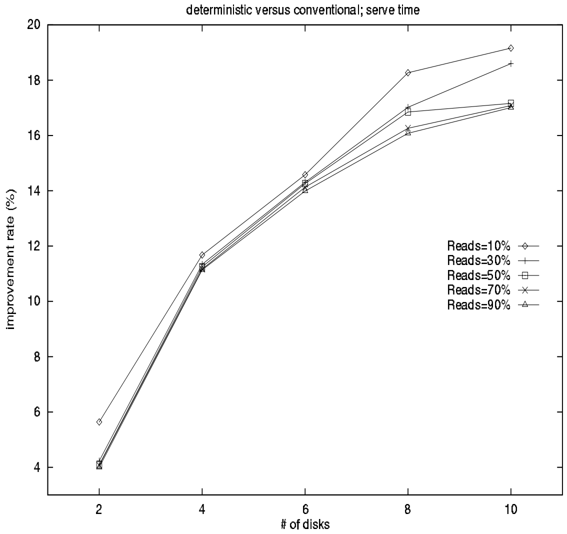

Figure 5: Deterministic over conventional.

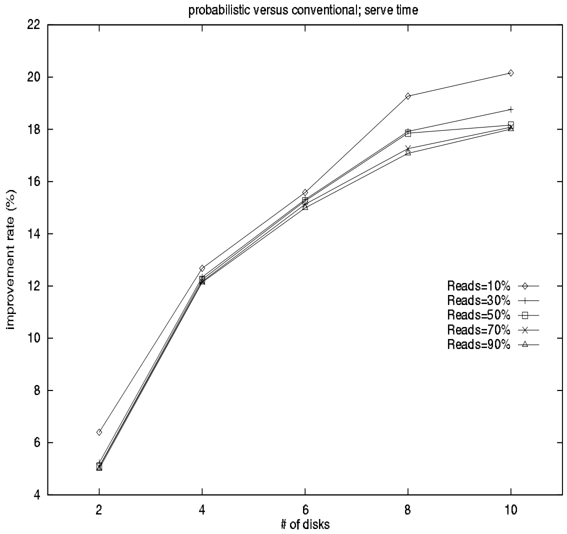

Figure 6: Probabilistic over conventional.

load balancing. More specifically, the improvement rates for the deterministic over the conventional model vary between 4% (2 disks, 0.9 read ratio) and 19% (10 disks, 0.1 read ratio), whereas the corresponding improvement rates for the probabilistic over the conventional model vary between 5% (2 disks, 0.9 read ratio) and more than 20% (10 disks, 0.1 read ratio).

5. Conclusions

The presented paper provided a new I/O servicing model in a parallel multiple I/O subsystem. The proposed model have introduced a request servicing redirection, based on disk queue information used as feedback. The redirection concerns write requests and the I/O controller is responsible for the new model implementation. The performance analysis proves that the proposed model improves the I/O servicing process and this is also documented by a developed simulation model. Simulation runs for heavy disk loads have been presented and indicative results are demonstrated. Service time is considerably benefited by the feedback-based model at rates over than 20%.

Future work could expand this model in order to include different disk technology configurations, along with data redundancy schemes (e.g. [7]. This expansion could be quite useful for investigating the influence of disk parameters to the feedback-based processing. Also, we could adopt different load estimations such as expected seek or rotational delays at the disk queues. Disk caches could also be added to the model structure in order to further study the caching influence to the I/O servicing performed under the proposed feedback-based model.

References

[1] R. English and A. Stepanov: “Loge : A Self-Organizing Disk Controller", *HPL-91-179*, HP Labs, Technical Report, Dec. 1991.

[2] J. Vitter and E. Shriver: “Algorithms for Parallel Memory I,II", *Department of Computer Science*, Brown University, Technical Report CS-90-21, Sep. 1990.

[3] S.W. Ng: “Advances in Disk Technology - Performance Issues", *IEEE Computer*, Vol.31, No.5, pp.75-81, 1998.

[4] C. Ruemmler and J. Wilkes: “An Introduction to Disk Drive Modeling", *IEEE Computer*, Vol.27, No.3, pp.17-28, 1994.

[5] E. Shriver: “Performance modeling for realistic storage devices", *Ph.D. Thesis*, Department of Computer Science, New York University, May 1997.

[6] E. Shriver, A. Merchant and J. Wilkes: “An Analytic model for disk drives with readahead caches and request reordering", *ACM SIGMETRICS’98*, Conference Proceedings, pp.182-191, Jun 1998.

[7] Vakali A. and Manolopoulos Y.: “An Exact Analysis on Expected Seeks in Mirrored Disks", *Information Processing Letters*, Vol.61, No.6, pp.323-329, 1997.