A New Approach to the Design of High Performance Multiple Disk Subsystems: Dynamic Load Balancing Schemes

A.I. Vakali, G.I. Papadimitriou, and A.S. Pomportsis

Department of Informatics
Aristotle University of Thessaloniki, Greece
{avakali,gp,apombo}@csd.auth.gr

Abstract. The performance of storage subsystems has not followed the rapid improvements in processors technology, despite the increased capacity and density in storage medium. Here, we introduce a new model based on the idea of enhancing the I/O subsystem controller capabilities by dynamic load balancing on a storage subsystem of multiple disk drives. The request servicing is modified such that each request is directed to the most appropriate disk drive towards servicing performance improvement. The redirection is performed by a proposed algorithm which considers the disk drive queues and the disk drives "popularity". The proposed request servicing has been simulated and the load balancing approach has been shown quite effective as compared to conventional request servicing.

1 Introduction

Modern I/O subsystems are equiped with efficient policies such as scheduling, reordering of I/O requests or read-ahead. The management of a multiple disk subsystem is usually governed by a controller which is responsible for request servicing and storage system functionality. Nowdays most controllers in disk subsystems come with self-managing techniques for request servicing and are easily adapted standard systems without major software modifications [1].

We propose a load balancing approach towards reducing the difference between processor speed and disk servicing time. A similar model was proposed in [5] where the request redirection has been proven quite beneficial. Here, we formulate and extended the earlier approach by proposing an effective load balancing scheme under a specific probability updating scheme. The next section presents the multiple disks I/O storage subsystem model whereas Section 3 introduces the proposed load balancing approach Section 4 presents the simulation results and conclusions are summarised in Section 5.

2 The Storage Model

One of the most common memory models is the hierarchical memory model proposed in [6] 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 I/O controllers manages a set of disk drives [2]. The storage subsystem comprises of D individual and independent disk drives. Disk drives have associated queues that contain requests waiting to be serviced. Requests arrive to the system randomly by various independent processes. The total 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 [3,4]. The most widely used formula for evaluating the expected service time involves these time metrics and it is expressed by :

$$\mathrm{E}[ServiceTime] = \mathrm{E}[ST] + \mathrm{E}[RL] + \mathrm{E}[TT] \tag{1}$$

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. Formulae for these times have been given in [3,4].

3 The Dynamic Load Balancing Approach

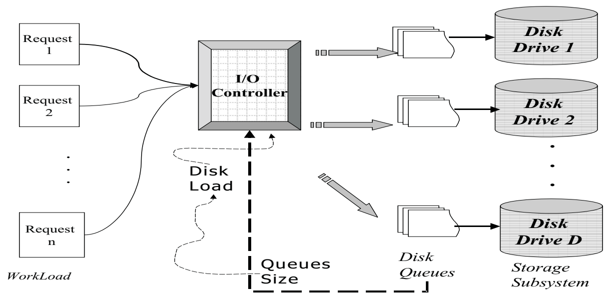

Fig. 1. The Dynamic Load Balancing Model.

In our approach 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 load information. Figure 1 presents the structure of our dynamic load balancing model which alters the conventional multiple disk drives model accordingly. Writing is performed by indirecting request pattern within the controller to allow data relocation such that the service time is improved. At each time slot t, the dynamic load balancing based controller contains a probability distribution $P(t)$ over the set of disks. Thus, $P(t) = \{p_1(t), \ldots, p_D(t)\}$, where D is the number of disks. The probability distribution $P(t)$ is updated at each time slot by taking into account the estimated load of each disk drive. The load is estimated by using the disk queue length since the queue size is an indicative metric for each disk load.

Definition : The load $L_i(t)$ of disk d_i $(i = 1, \ldots, D)$ is evaluated in by :

$$L_i(t) = \max\{Q_i(t), \epsilon\} \quad (2)$$

where $Q_i(t)$ is the number of requests waiting in the queue of disk d_i, while ϵ is a positive real number in the neighborhood of 0.
The probability $p_i(t)$ of redirecting a write request to a disk d_i is inversely proportional to the load $L_i(t)$ of this disk.
At any time slot t, for any two disks d_i and d_j, we have:

$$\frac{p_i(t)}{p_j(t)} = \frac{L_j(t)}{L_i(t)}$$

since the choice between two disks i and j for servicing a write request will be made according to their loads $L_i(t)$ and $L_j(t)$. The disk with the heavier load has smaller probability than the other disk with the lighter load. Since at any given time slot t, it holds $\sum_{i=1}^{D} p_i = 1$, we are expressing all p_is $(i = 1, 2, \ldots, D)$ in terms of one specific disk, namely the i disk drive. From the above, we derive that the choice probability of each disk d_i $(i = 1, \ldots, D)$ will be :

$$p_i(t) = \frac{\frac{1}{L_i(t)}}{\sum_{k=1}^{D} \frac{1}{L_k(t)}} \quad (3)$$

4 Experimentation - Results

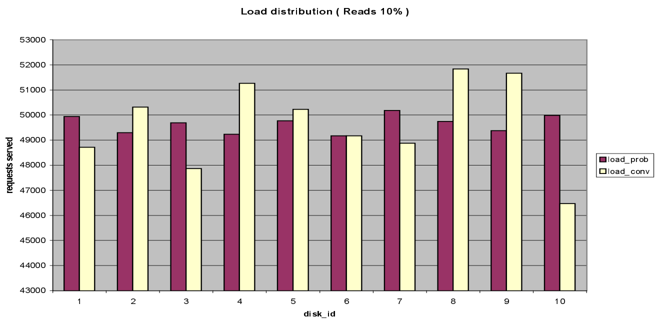

Fig. 2. Load distributions, 10% reads (Probabilistic versus conventional).

In order to study the performance of the proposed model we have simulated the conventional model and two types of models for the dynamic load balancing model, namely the deterministic and the probabilistic model. 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 [2,3] for the *HP 97560* disk drive. The read/write ratio is a parameter varying in the range $0.1, \ldots, 0.9$. The proposed model has been showed to be beneficial in all cases when compared to the

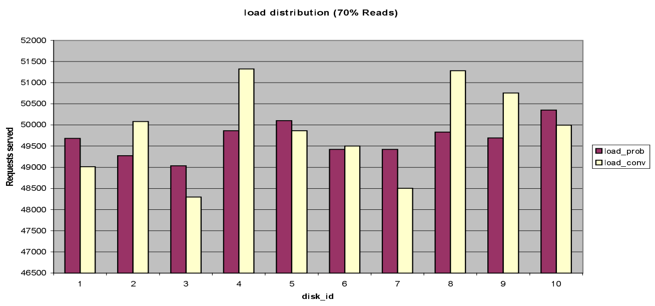

Fig. 3. Load distributions, 70% reads (Probabilistic versus conventional).

conventional I/O servicing model. Figures 2 and 3 depict the load distribution under indicative loads serviced by both the conventional and the proposed dynamic load balancing probabilistic model. Figure 2 depicts the load distribution for a parallel I/O subsystem of 10 disk drives, under both the proposed and the conventional model under 0.1 read ratio and Figure 3 presents the same bars for a 0.7 read ratio. These histograms depict the load variation for each of these models and they represent the beneficial load distribution of the proposed dynamic load balancing model.

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 the load estimation and characterization metric. The redirection concerns write requests and the I/O controller is responsible for the new model implementation. Simulation runs for heavy disk loads have been presented and indicative results are demonstrated.

References

1. English R., Stepanov A. : Loge : A Self-Organizing Disk Controller. HP Labs, Technical Report, **HPL-91-179**, (1991).
2. Ruemmler C., Wilkes J.: ‘An Introduction to Disk Drive Modeling, IEEE Computer, **27** (1994) No.3, 17-28.
3. Shriver E. : Performance modeling for realistic storage devices, Department of Computer Science, New York University, **Ph.D. Thesis** (1997).
4. Shriver E., Merchant A. and Wilkes J. : An Analytic model for disk drives with readahead caches and request reordering. ACM SIGMETRICS’98, Conference Proceedings, (1998) 182-191.
5. Vakali A.I., Papadimitriou G. I. : An Adaptive Model for Parallel I/O Processing. Proceedings of the IASTED International Conference, Parallel and Distributed Computing and Systems, **PDCS 99** (1999) 139-142.
6. Vitter J., Shriver E.: Algorithms for Parallel Memory I,II, Department of Computer Science, Brown University, Technical Report **CS-90-21** (1990).