論文 / 著書情報
Article / Book Information

Title	Consideration of disk segmentation and disk sequencingfor efficient use of disk resource on heterogeneous disk environment
Author	Makoto Kataigi, Dai Kobayashi, Haruo Yokota
Journal/Book name	Proceedings of the Second IEEE International Conference on Digital Information Management, Vol. , No. , pp. 577-579
Issue date	2007, 10
DOI	http://dx.doi.org/10.1109/ICDIM.2007.4444286
URL	http://www.ieee.org/index.html
Copyright	(c)2007 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other users, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works.
Note	このファイルは著者（最終）版です。 This file is author (final) version.

Consideration of disk segmentation and disk sequencing for efficient use of disk resource on heterogeneous disk environment

Makoto Kataigi
Tokyo Institute of Technology
mkataigi@de.cs.titech.ac.jp

Dai Kobayashi
Tokyo Institute of Technology
Japan Society for the Promotion of Science
daik@de.cs.titech.ac.jp

Haruo Yokota
Tokyo Institute of Technology
yokota@cs.titech.ac.jp

Abstract

Recently, it is needed to have high performance and use capacity efficiently at the same time on the heterogeneous disk environment. But it is difficult to do so because the performance and capacity of each disk are different. We think to extend AOD method for this aim. This method uses the difference of access load between primary data and backup data and balance access load and data amount simultaneously on the homogeneous environment. In this paper, we consider to maximize usable capacity by the position of backup and the disk sequencing. Furthermore, we arrange the primary data in order to balance the performance based on its Bandwidth to Space Ratio.

1 Introduction

Recently, the number of large storage systems constructed by many HDDs is increasing because of enlargement of data. Usually, these systems need to have a high performance and reliability. So, these systems balance access load because average response time of HDD drops off by load imbalance and data amount of each disk because the recovery methods become inefficient if it is imbalance.

However, the counter-operations are occurred when we use both methods simultaneously. We proposed Adaptive Overlapped Declustering (AOD) [7] to solve this problem. This method leverages the difference of access pattern between primary data and backup data on the environment that the systems have backup data. We put primary data to balance access load and backup data on the neighbor (left and right) disks primary is data put on. We balance the data amount by dividing the backup data to 2 parts on the left and the right adjacent disks.

On the other hand, the HDDs which are components of storage systems have become higher performance and larger capacity [5]. But we use old model HDDs or HDDs limited the performance and capacity at other old HDDs on a method kind of AOD. So we cannot use the new disk's resource efficiently. This is because the existing methods didn't care of the heterogeneous disk environment; these methods are created for the homogeneous disk environment.

In this paper, we suggest the method that use AOD to balance both performance and capacity simultaneously in heterogeneous disk environment. There is also a research to use both performance and capacity efficiently on the environment which Bandwidth to Space Ratio (BSR) of each disk is balanced [3]. We adopt the BSR method to AOD for efficient use of performance.

In this research, we aim to resolve the problem that performance and capacity cannot be used efficiently on the heterogeneous disk environment. There are many kinds of performance such as disk and network throughput, in this research we consider the performance as throughput bottlenecked. Firstly, we allocate backup area to solve capacity difference. Secondly, we think about disk sequencing to use capacity efficiently. Finally, we allocate primary area to settle performance difference. We think to segmentalize disk area to some sub-area such as disk left, right, primary and backup area. Positioning of real data is future work.

The rest of this paper is organized as follows. We consider the method to make the capacity utilization of the system higher in Section 2, and the method to make the performance utilization ratio of the system higher in Section 3. Finally in Section 4, we introduce some related work.

The parameters used in this paper are shown in Table1.

2 The efficient use of capacity

In this section, we aim to use capacity efficiently by disk area segmentation and disk sequencing. Especially, we think to reduce the capacity of the area which backup data cannot be put on both left disk and right disk (called

Table 1. Parameters at Disk n

C_n	Capacity ($C_n = L_n + R_n + G_n$)
L_n, R_n	Capacity of left and right area
D_n, G_n	Capacity can use and cannot use
B_n	Bandwidth
P_n	Capacity of primary area
P_{L_n}, P_{R_n}	Primary capacity put on left and right area

N : Disk number in this system
For simplify, we write disk $n\,(> N)$ as disk $n \, mod \, N$, and disk $n\,(\leq 0)$ as disk $(n + \alpha N) \, mod \, N$.

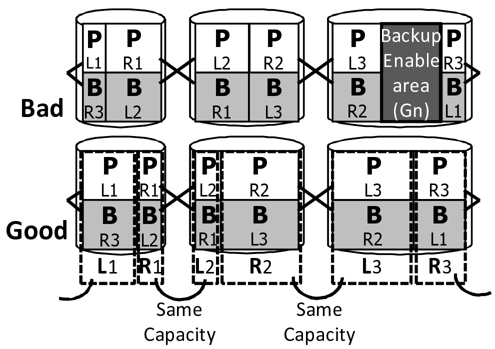

Figure 1. The capacity relation

backup-enable area) (Fig 1). Firstly, we segmentalize disk area to disk left area whose backup data is on the left disk and right area whose backup data is on the right disk, and adjust the capacity of each left and right area for minimizing the capacity of backup-enable area. Secondly, in Section 2.1 the disk sequence is settled, but it relates to the capacity of backup-enable area, so in Section 2.2 we consider about the disk sequencing to minimize the capacity of backup-enable area.

2.1 Disk segmentation

Firstly, we consider the way to minimize the capacity of the backup-enable area by the disk segmentation. There is a relation that the capacities between adjacent disks are equal because the primary data and backup data have to be the same capacity (Fig 1). Of course, the disk capacities relate to the disk areas capacities. Therefore, we get the following formulas (1).

$$L_n = R_{n-1}, \qquad C_n = L_n + R_n + G_n \tag{1}$$

So we have to minimize the capacity of backup-enable area to keep the condition satisfied.

To minimize the capacity of backup-enable area, we **calculate the capacity** of each area by the using formulas (1). And we create a minimum backup-enable area on the disk which have the area whose capacity is grater than the disk capacity. We also take the **border condition** that the capacity of the last disk's right area is same as that of the first disk's left area into account. The disk segmentation algorithm is shown as following Table 2.

Table 2. Disk Segmentation Algorithm

	Input	$\{C_n\}$ the array of disk capacities
	Output	$\{L_n\}$ the array of left area's capacities
Initi-alize	Step1.	$n = 2, G_k = 0$ for all k
	Step2.	$L_1(\alpha) = \alpha \; (0 \leq \alpha \leq C_1)$
Calculate capacities	Step3.	$L_n(\alpha) = C_{n-1} - G_{n-1} - L_{n-1}(\alpha)$
	Step4.	If $L_n(0) > C_n$, then $G_{n-1} += L_n(0) - C_n$, $n = n - 1$, go to Step3.
	Step5.	If $n < N$, then $n = n + 1$ go to Step3.
Border condition	Step6.	When N is odd, then $\alpha = \frac{1}{2} L_n(0)$
	Step7.	When N is even
	Step7.1	If $L_N(0) \neq 0$, then $G_N += L_N(0)$
	Step7.2	$\alpha = \frac{1}{2}(C_1 - G_1)$

2.2 Disk sequencing

The disk sequencing depends on the capacity of backup-enable area because there is a relation between the capacities of both neighbor disks. So then we think about how to reduce the capacity of the backup-enable area by disk sequencing.

We can take the disk array whose disk capacity increases at first, and then decrease as a proper sequencing. And we take where we insert a new disk to satisfy the proper disk sequencing into account. The disk insertion algorithm shown as following Table3. Of course, we can still use this algorithm when extremely big capacity disk added.

Table 3. New Disk Insertion Algorithm

Input	D_{new} the capacity of new disk $\{D_n\}$ the array of disk capacity
Output	position to insert new disk
Step1.	From the disk which has minimum capacity, search the positions which satisfy $D_l \leq D_{new} \leq D_{l+1}$ (call L) or $D_m \leq D_{new} \leq D_{m+1}$ (call M).
Step2.	calculate $\delta_c(N)$ for insertion at both L and M.
Step3.	Insert new disk at the position $\lvert\delta_c(N)\rvert$ is small.

3 To have a high performance

In this section, we aim to have a high performance after adjusting of the capacity L_n and R_n. In order to have high performance, just considering about the primary area is enough, because the required performance of backup data in AOD is zero. So we adjust BSR for primary area (called **BSRp**) to achieve as good performance as possible.

There is a relation of primary area's capacities between adjacent disks which is shown in the followings formulas (2).

$$P_{L_n} = R_{n-1} - P_{R_{n-1}}, \qquad P_n = P_{L_n} + P_{R_n} \quad (2)$$

We calculate the primary area's capacity of each disk by using these formulas and if the that capacity is grater than the disk capacity, we have to readjust the primary capacity from that disk to other disks while keeping the BSRp balanced. The algorithm is shown in Table 4.

Table 4. Primary Determination Algorithm

Input	$\{R_n\}$ the array of the right area's capacity
	$\{B_n\}$ the array of the disk bandwidth
Output	$\{P_n\}$ the array of the primary capacity
Step1.	$n = 1$
Step2.	$P_m = \frac{1}{2} D_{all} * \frac{B_m}{B_{all}}$ for all m
Step3.	we search the minimum $\delta_{bsr}(n, m)$ for $m \in [1..N]$ and that is δ_{bsr}^{min}, $m = min$
Step4.	If $\delta_{bsr}^{min} \geq 0$, then go to Step8.
Step5.	For $m \in [n..min]$, $P_m = P_m + \lvert\delta_{bsr}^{min}\rvert * \frac{B_m}{\sum_{k=n}^{min} B_k}$
Step6.	For $m \in [min+1..n+N]$, $P_m = P_m - \lvert\delta_{bsr}^{min}\rvert * \frac{B_m}{\sum_{k=min+1}^{n+N} B_k}$
Step7.	go to Step3.
Step8.	$n = n + 1$
Step9.	if $n \leq N$, then go to Step2., otherwise end.

4 Related work

Recently, there are many researches focus on having high performance and using capacity efficiently on the heterogeneous disk environment. For example, in [1, 2, 6] they applied extended striping or RAID to use on the heterogeneous disk environment. In these researches, they assumed BSR of each disk is almost the same. But in our research, we don't need this.

In [8], they made BSR of virtual disks become equal by summing up some disks. In [4], they used the difference of access pattern. They put the high accessed data on the high performance disk and the low accessed data on the low performance disk. But these methods can not be always used.

In [3], they focus BSR and defined the data position to reduce the deviation between the BSR of disk and the BSR of data on it. The experiment's results showed that higher utilization ratio of both bandwidth and capacity took on the environment, which is constructed from disks that have the same BSR. But, the result is not good on the environment in which the disks don't have the same BSR.

5 Summary and future work

In this paper, we thought the method to have high performance and use capacity efficiently on the heterogeneous disk environment. We used AOD in this environment for this aim. To use capacity efficiently, we segmented disks to left and right areas, and we propose the method how to define the ratio of left and right. And we propose the disk sequencing to minimize the capacity backup data cannot be putted on. Finally we adjust the allocation of primary area to have high performance. Especially, we propose the method to adjust primary BSR for high performance.

For future work, we are considering to estimate the utilization rate of capacity and performance on the realistic disk pattern such as when new disks (usually it has larger capacity than old one) are added. And we are also thinking about the arrangement and the rearrangement methods of real data in such situation that we need to have where to put or move data when the access pattern is changed or when new disks are added.

Acknowledgments

This work is partially supported by the JST CREST, MEXT of Japanese Government via Grant-in-Aid for Scientific Research #19024028 and the MEXT 21st Century COE Program.

References

[1] T. Cortes and J. Labarta. A case for heterogeneous disk arrays. *IEEE International Conference on Cluster Computing (Cluster'00)*, page 319, 2000.

[2] T. Cortes and J. Labarta. Extending heterogeneity to RAID level 5. In *Proceedings of the 2001 USENIX Technical Conference*, pages 119–132, Boston, 2001. USENIX Association.

[3] A. Dan and D. Sitaram. An online video placement policy based on bandwidth to space ratio (bsr). *SIGMOD Rec.*, 24(2):376–385, 1995.

[4] J.-W. Ding and Y.-M. Huang. Resource-based striping: An efficient striping strategy for video servers using heterogeneous disk-subsystems. *Multimedia Tools and Applications*, 19(1):29–51, 1 2003.

[5] E. Grochowski and R. F. Hoyt. Future trends in hard disk drives. In *IEEE TRANSACTIONS ON MAGNETICS*, volume 32, MAY 1996.

[6] Y. Wang and D. H.-C. Du. Weighted striping in multimedia servers. In *International Conference on Multimedia Computing and Systems*, pages 102–109, 1997.

[7] A. Watanabe and H. Yokota. Adaptive overlapped declustering: A highly available data-placement method balancing access load and space utilization. pages 828–839. 21st International Conference on Data Engineering (ICDE'05), 2005.

[8] R. Zimmermann. *Continuous media placement and scheduling in heterogeneous disk storage systems*. PhD thesis, 1998. Adviser-Shahram Ghandeharizadeh.