Title

Author1, Author2, Author3, Author4, Author5

1 Introduction

To be finished.

2 Trace data and Analysis Methodology

2.1 System structure

The system provides services to users in the form of cloud disk. Users can use cloud disk in a manner similar to local disk, for example, as a system disk or data disk. In the system, each cloud disk segment is divided into fixed-size segments (32GB). The system is divided into several clusters. The architecture of the cluster is shown in figure 1, and the overall architecture is shown in figure 2. The bottom layer of each cluster is stored as an append-only distributed file system. The Block service layer abstracts the append-only files into fixed-size blocks for users by means of log-structure-merge. The Block server is responsible for pulling the segment metadata and passing the data between users and segments in a proxy manner. All read and write traffic to segment goes through the corresponding block server. The master node stores the block server information corresponding to the segment and provides control services for the block server, such as the addition, deletion and error recovery of the block server.

2.2 Existing Problems

As is known to all, the throughput of each device, or rather, each segment varies greatly with time. In the current structure, the throughput of block servers may be imbalanced after a period of time without rebalacing. The imbalance will lead to the following consequences:

- Some block servers may reach the maximum throughput. Cloud disk application providers have to add new servers even new clusters;
- Long tail effect may appear, i.e., some devices may suffer from high latency and slow reads and writes.

In summary, analyses of I/O pattern of cloud disks and efforts to improve the situation are necessary.

2.3 Analysis Methods

Since the current situation is not good enough, we need to analyse the trace as well as schedule the cluster to improve the situation. We use a simulator to simulate the scheduling process and output the schedule result.

In order to analyse if the scheduling strategy is good, we need to define an index to measure the level of load balance. Here we choose the variance of read throughput between block servers as a measurement. As the variance changes over time, we use median of variances over time as a measurement when we want to judge the schedule effect over a period of time. Due to the fact that the movement of a segment employs computation resource as well as time, we define the number of moving as the schedule cost. Thus, the target of rebalance is to minimize read throughput and schedule cost.

2.4 Trace Data

We use trace of clusters in business environment for simulation. We select 6 clusters, including ESSD clusters, SSD clusters and efficient clusters in a certain available zone for analysis. Since there are clusters with all levels of performance, the analysis is comprehensive.

Considering the size of data set and simulation efficiency, the grain fineness of trace data is 1 minute, i.e., we can know the read and write throughput per minute from the trace data. If the time range of trace data is too long, analysis will be difficult. Thus, we select a representative time range, which is from 2020-2-10 17:00 to 2020-2-10 19:00 (UTC+8:00).

Each item of trace data includes six fields: time, device, segment, read throughput, write throughput and

host. These data are sufficient for analysis and scheduling simulation.

3 Analysis

3.1 Trace analysis

In this chapter, we mainly introduce some phenomena and understandings observed through trace analysis. Since the average write traffic of the cluster in trace is much higher than the read traffic, we believe that the write traffic in the cloud environment is a more scarce resource. Therefore, the following paper explores the load balancing problem centering on the write traffic.

3.1.1 Unbalanced network traffic distribution

Cluster	Block server	Write traffic
AY306L	48	1.03E+13
AY251Z	96	1.99E+13
AY272T	82	2.35E+13
AY306O	95	1.90E+13
AY336D	96	2.05E+13
AY272M	95	1.99E+13

Table 1: Cumulative write traffic

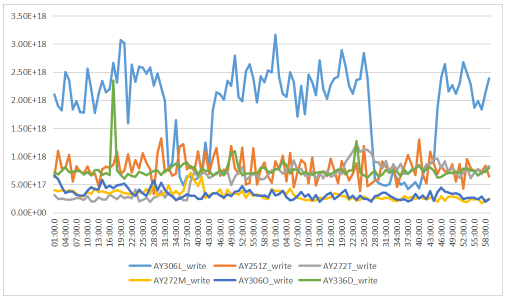


Figure 1: Unbalance in different clusters

We made a statistical analysis on the access and load balancing of the six clusters. First, we counted the cumulative write traffic of the six clusters in trace for two hours(Table 1). We then calculated the variance of write traffic of different block servers in each cluster(Figure 1).Tthe vertical axis is the traffic variance of block server in the cluster, and the horizontal axis is the time, a serious imbalance can be obeserved.

3.1.2 Impact of Group User

First, we counted the cumulative traffic distribution in the segment. We observed that cloud disk of the same user may be similar in the write traffic characteristics. When the user scale is large and cloud disk has a large write traffic, it may have a great impact on the overall balance degree of the cluster. This kind of phenomenon appear in cluster AY272T and AY336D, their segment accumulated traffic and block server cumulative traffic as shown in figure,we can observe the four highest traffic segment belong to the same cloud disk, and their write traffic is very high, the cloud disk flow curve as shown in figure,It can be found that it is similar to the cluster balance degree curve characteristics, so it can be inferred that the balance degree fluctuation of cluster AY306L is affected by the cloud disk.

3.1.3 Write Traffic Distribution of Segments

First we delete cluster AY272T, AY336D from Figure 1 to eliminate impact of group user.The write traffic variance of ESSD in the figure is greater than that of efficient disk. Based on this observation, we guess the cluster's traffic balance degree may be associated with the cloud disk performance, namely the high performance disk is more likely to cause imbalance phenomenon.

We observed the changes of block server traffic in different clusters and found that most of the write traffic fluctuations of block servers are not very large, but there are significant differences among different block servers(Figure **??**).Occasionally there is a large fluctuation, but it is affected by few segments of very high write traffic disk.Therefore, we assume that the segment traffic distribution is unbalanced, and this imbalance leads to traffic differences between block servers.To confirm our hypothesis,we calculated the cumulative write traffic of all segments(Figure **??**), which is like the zipF distribution .

Since the traffic difference between different segments may be large, the cluster balance degree is mainly affected by a small part of segments with higher traffic ranking.In order to observe the proportion of these segments in different clusters, we intercepted the segments whose cumulative traffic was greater than 1/10 of the average traffic of block server for observation(Figure 2). (Figure 2). It can be found that the curve of the high-

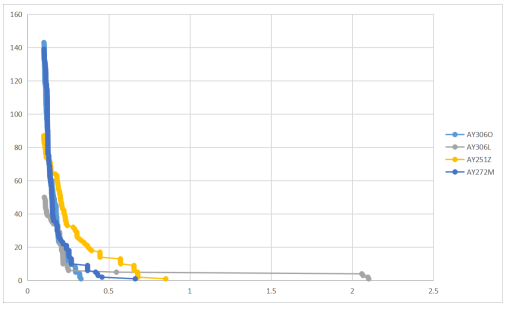


Figure 2: Segment traffic distribution of different clusters

performance disk approaches 0 at a slower speed, which indicates Among the few segments with the highest network traffic 1. High performance clusters have higher peaks 2. High performance clusters Large degree of dispersion As a result, the placement of these segments is more likely to cause imbalance

3.2 Impact of Simple Scheduling Strategies

In this section, we introduce a simple segment-based balancing method based on historical records and greedy strategy, which move segments periodically to rebalance the throughput of block servers. Moreover, the influence of different parameters on the scheduling results will be analyzed.

3.2.1 Introduction to Simple Scheduling Strategies

We first define a *simple greedy strategy*, which performs a round of schedule at a specified time interval (to be specified by the administrator). When the time to schedule arrives, the scheduler will calculate the historical average on every segment over a period of time. Then, the scheduler will traverse segments by historical records from high to low. The scheduler will move the segment to the block server which has the lowest throughput. Obviously, schedule frequency and the time length of historical record will make an impact on the schedule result.

In the sections below, we will discuss the impact of several parameters (e.g. knowledge of future throughput, proportion of scheduling segments). All these topics will be discussed on the *simple greedy strategy* as an example.

3.2.2 Impact of Greedy Algorithm

We simulate simple greedy algorithm on different clusters. We can find that simple greedy strategy can reduce the variance in some degree, but there's still some distance to the upper limit (Table 2). What's worse, sometimes, negative optimizations may occur (Figure 3). The reason is that average of historical records can not describe the pattern of the cluster properly.

Cluster	Type	No Schedule	Greedy	Future
AY306L	ESSD	2.13E+18	6.51E+17	5.36E+17
AY251Z	ESSD	7.68E+17	5.53E+17	1.99E+17
AY272T	SSD	6.87E+17	1.10E+17	3.02E+16
AY306O	Efficient	3.08E+17	1.14E+17	3.29E+06
AY336D	Efficient	7.27E+17	1.12E+17	3.35E+06
AY272M	Efficient	2.97E+17	7.75E+16	3.52E+06

Table 2: Median of cluster variance

For example, I/O pattern of many segments is periodic. In cluster AY251Z, the I/O pattern of a device, which has 16 segments and accounts for 9.2% flow of the whole cluster, is in a cycle of 20 minutes. However,

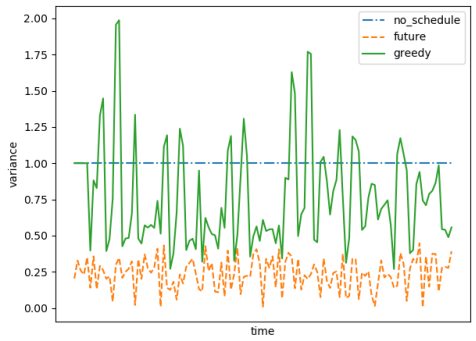


Figure 3: Result of Simple Greedy Strategy on AY251Z

Figure 4 indicates that these segments are in busy only in a short period of the cycle. If the observation range of history is too low, prediction will differ greatly from the actual state. However, if we raise the observation range, in some segments, local characteristics will be missed, though the periodic problem may be solved. As a result, average of historical records can not describe the pattern of the cluster properly. To improve the schedule result, detailed I/O pattern should be extracted for scheduling.

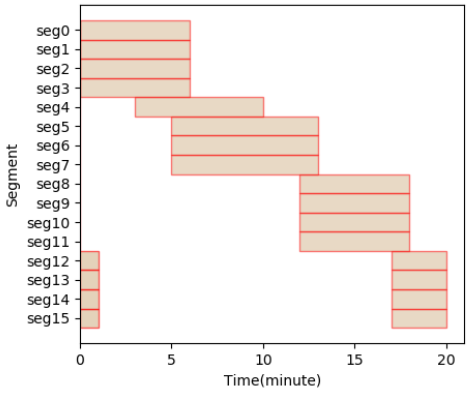


Figure 4: Busy Time of Segments in a Certain Device

3.2.3 Impact of Knowledge of Future Throughput

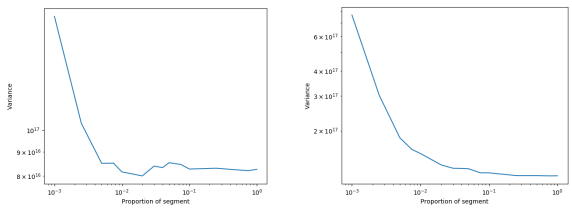
Scheduling effect not only depends on strategy itself, but also depends on the schedulability of the cluster. If we can know the future throughput of each segment, the scheduling effect will be improved. Hence, we can analyze the best case of greedy algorithms. We can use the following method to attempt to reach an approximate upper bound for any greedy schedule strategy. In the *simple greedy strategy*, we pretend to know the future throughput of each segment, and execute scheduling with the future throughput. Besides, schedule should be frequent

enough, or we will deviate from upper bound.

Table 2 shows the initial state and the approximate upper bound of any greedy algorithm. Note that, there is little potential for ESSD clusters to improve, while SSD clusters and efficient clusters have great potential to improve. As was stated in section 3.1.3, ESSD cloud disks have better performance and allow higher throughput. Thus, users are more likely to read and write more to the disk and the smallest unit of scheduling will be larger. This makes scheduling algorithm less effective on clusters with higher performance.

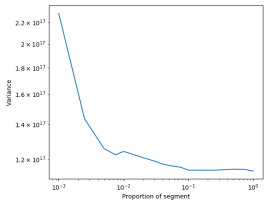
3.2.4 Impact of Proportion of Segments to Schedule

According to the 80/20 rule, we have a reason to believe that few segments contributes most to the variance of the entire cluster. Hence, we define a *partial greedy strategy*, which only replace segments with the most throughput in each round of schedule. Evidently, the proportion of segments to schedule each time will also make an impact on the schedule result. Figure 5 shows the relation between variance and the schedule proportion of *partial greedy strategy*. Schedule interval and observation length of history are both 5 minutes.

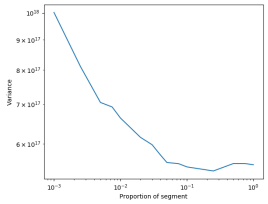


(a) AY336D

(b) AY272T



(c) AY306O



(d) AY251Z

Figure 5: Variance of Cluster with Different Schedule Proportion

From the figure, we can learn that from the point of view of results, there is no significant difference between implementing greedy schedule for all the segments and for only 10% of all segments. Few segments contributes most to the variance. Data from multiple clusters support this result.

References