Scalable Tape Archiver for Satellite Image Database and its Performance Analysis with Access Logs – Hot Declustering and Hot Replication –

Toshihiro NEMOTO and Masaru KITSUREGAWA
Institute of Industrial Science, University of Tokyo, JAPAN

Abstract

Recently, global environmental studies have become very important around the world. Repeated observations of wide areas of the earth at the same time makes satellite images useful. For understanding earth's environment it is necessary to use a substantial amount of the temporally sequenced satellite images. At the Institute of Industrial Science, University of Tokyo, we have been building a global environmental digital library whose main contents are earth surface images from NOAA environmental satellite images and GMS images received at our institute.

In this paper, we describe our satellite image archive system. We focus on a scalable tape archiver (STA), which is a key component of our archive system. We explain two orthogonal file/cassette replacement strategies, declustering and replicating, adopted for the STA and show their effectiveness. After their basic performance is shown, we describe a performance analysis using access traces taken at our archive system. The proposed schemes named *hot declustering* and *hot replication*, work effectively in our STA in daily operations.

Introduction

Recently, global environmental problems have attracted strong attention around the world. Continuous observation over a wide area of the earth is helpful in understanding environmental changes of the earth and earth scientists usually analyze enormous amount of consecutive satellite images for this purpose. Satellite data archives therefore play an important role in this process. At the Institute of Industrial Science, University of Tokyo, we have been building a global environmental digital library whose main contents are earth surface images from NOAA (National Oceanic and Atmospheric Administration) environmental satellite and GMS (Geostationary Meteorological Satellite) received at our institute. We started the ingestion of the satellite images more than fifteen years ago. Their total number is now around 60,000 scenes, which amount to 6 TB.

We have been studying the large scale tertiary storage system named *scalable tape archiver (STA)* to store the satellite images[3]. We believe that next generation tertiary storage should employ commodity components as much as possible in order to decrease systems costs. The STA consists of a number of small commodity tape archivers and tape migration units between adjacent archivers. It is easy to add element archivers, or remove them from the system dynamically at the customer site. The system automatically migrates cassettes from the original system to a newly added archiver so that the system, as a whole, will always be load-balanced.

The STA uses two orthogonal file/cassette replacement schemes, declustering and clustering. Declustering means redistribution of frequently accessed tape cassettes (hot tape cassettes) evenly over the STA by tape migration. We call it *hot declustering*. Clustering means making replicas of frequently accessed files (hot files) and placing them at the tail of tapes. We call it *hot replication*. In hot replication, replicated hot files are clustered at the tail of tapes, which could significantly reduce the seek cost. Hot declustering distributes hot tapes over the element archivers as evenly as possible, that is, it contributes to equalize the workload among the element archivers. Thus the declustering and clustering of hot files plays a very important role boosting the performance of our system. Both of these schemes are based on the heat of the file and the cassette tape. The original concept of heat and temperature was introduced for file management on disk arrays by G. Copeland et al[10]. G. Weikum et al. extended the research to dynamic relocation of data on disk arrays when the file is expanded[11]. As far as the authors know, heat-based file relocation research has been confined to secondary storage devices. Because of the many differences between disk arrays and tape archivers, which we will discuss in this paper, we have derived our own heat based schemes for tape archivers.

There have been several research efforts to improve the file system on tertiary storage. S. Christodoulakis et al. studied optimal static data placement in tertiary storage libraries to minimize tape exchanges and seeks[4]. L. T. Chen et al. proposed optimal data partitioning algorithms for a spatial temporal database[5]. They also tried

to rearrange the fragments to reduce the seek length. B. K. Hillyer et al. described I/O scheduling algorithms for serpentine tape drives[6]. Striping techniques applied to tape archivers for the improvement of bandwidth were explained by A. L. Drapeau et al.[7] and L. Golubchik et al[8]. We proposed a partial migration scheme to move only a necessary part of a file from a tertiary storage[9]. There are several studies on optimization of tertiary file accesses. As mentioned above, tape migration mechanisms plays a very important role for both hot declustering and hot replication. However, the previous papers have not investigated such tape migration systems, as far as the authors know.

In this paper, we first describe our satellite imagery archive system. Then we introduce the STA. We define heat and temperature for the STA and then we show the declustering and clustering schemes of the STA based on the heat and temperature. We will explain the effectiveness of hot declustering through tape migration scheme briefly. Tape transport mechanism using additional hardware was discussed in our previous paper[3]. In this paper, we will analyze the performance of our archiving system using the trace of around 460,000 accesses against our satellite imagery archive system from around the world through the Internet. The simulation study shows that the hot declustering can reduce response time considerably. It is more effective compared with cache disk. In addition, we show that hot replication at tape tail can reduce the response time. Using the tape drives which can load or eject a tape without rewinding, seek time can be shortened considerably when hot files are placed together at the tail of tapes. We simulate the basic performance of hot replication using synthetic data. In addition we confirm the effectiveness of clustering of hot data using the access trace of our satellite image database. Hot replication also improves the performance of STA significantly.

Satellite imagery archive system

Figure 1 shows the overview of our satellite image archiving system. Two antennas are used for direct reception of satellite images from NOAA environmental satellite and from GMS respectively. Satellite images are automatically received about 8 to 13 times a day from the NOAA satellite and 24 times a day from GMS and transferred to a SPARCcenter 2000E file server during reception. The images are registered into a database management system together with information about reception date and time, satellite name, observed area, receiving station name, original image file name, file size and reduced file name. We are also starting to archive NOAA images received at Bangkok, Thailand. These images are also transfered and registered into our database system.

The images are ingested into 8mm-based STA.

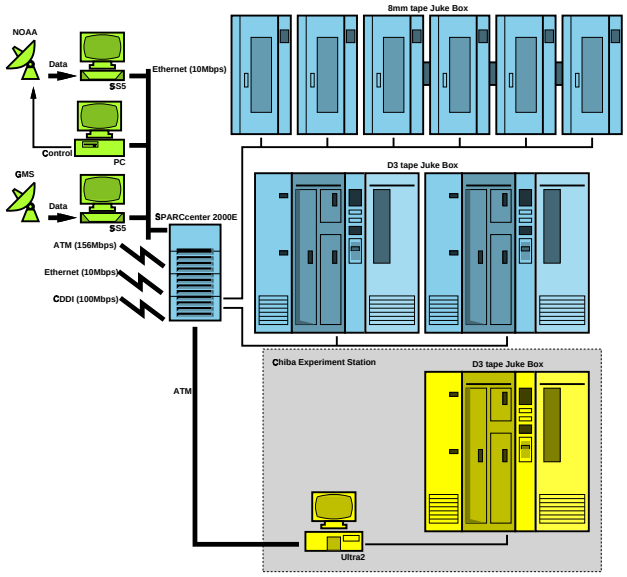

Figure 1. Satellite imagery archive system

The digital archive project started in 1994. Very recently we have procured a D3-based archiver. The images are stored redundantly.

Ftp, gopher and HTTP server daemons run continuously on the file server. In addition to many researchers in ecology, biology, hydrology, oceanography and meteorology, anybody can access to the satellite images through the Internet. Through WWW, users can search for the desired images by providing the system with meta information such as date, time, satellite name, receiving station, observed point and so on (figure 2). Since CGI scripts translate the user's query into SQL, it is easy to retrieve desired images for users unfamiliar with database systems. Through this interface users can access all images including those received at Bangkok.

Architecture of scalable tape archiver

The scalable tape archiver (STA) is composed of a number of small size tape archivers (element archivers) and tape migration units connecting any two adjacent element archivers. Figure 3 shows the organization of the STA using an 8mm tape jukebox, NTH-200B, as the element archiver. The experimental STA, composed of four NTH-200B's, has already been constructed and we are using the system to serve the global environment research database. The NTH-200B has two Exabyte 8505 tape drives, tape handler robotics and a tape rack with 200 slots. It also has a controller for its own tape handler robotics and for the tape cassette migration unit. The host computer sends commands for holding, releasing and moving a tape cassette to

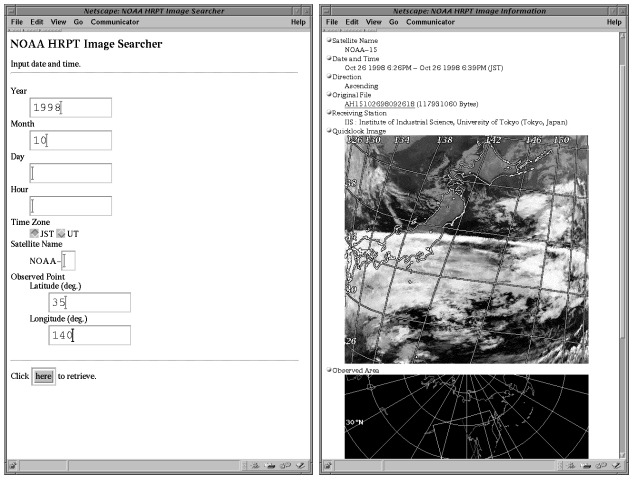

Figure 2. User interface of satellite imagery archive system

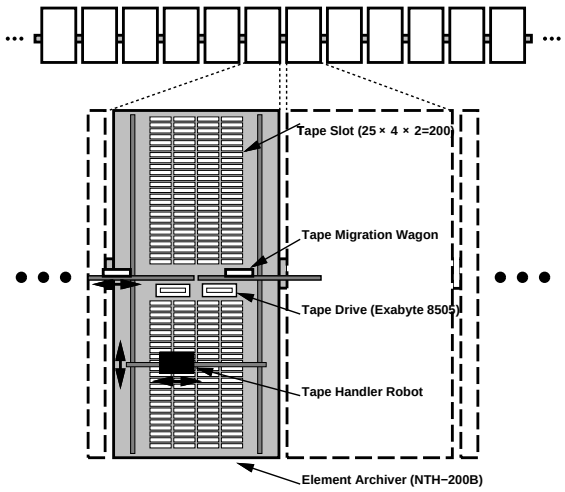

Figure 3. Scalable tape archiver

the controller and receives the status of the element archiver through an RS-232C port. The tape handler robotics takes a tape cassette from a slot in the rack and loads the cassette into the drive or unloads a cassette and returns it to its assigned slot. It also places, or picks cassettes from the wagon, which is used for cassette pass-through between element archivers. The Exabyte 8505's are connected to the host computer through a SCSI bus.

Tape migration mechanisms

Tape cassette migration is executed as follows.

1. The tape migration unit brings the wagon back into the source element archiver, if the wagon is not currently there.

Figure 4. Photograph of scalable tape archiver

2. The tape handler robot in the source element archiver takes the cassette to migrate from a slot or a drive.
3. The tape handler robot places the cassette in the wagon of the tape migration unit.
4. The tape migration unit transfers the wagon from the source element archiver to the destination element archiver.
5. The tape handler robot in the destination element archiver picks up the cassette from the wagon, and places the cassette into the specified slot or drive.

These steps are coordinated so that the counterweight of the tape handler robot does not interfere with the movements of the tape migration unit.

Heat based performance improvement schemes

Temperature, heat, and normalized heat

First, we describe the heat and temperature metrics[10]. Heat is the access frequency for a tape or an element archiver over some period of time. The heat of data is its access frequency, the heat of a tape is the accumulated heat of the data in it and the heat of an element archiver is the accumulated heat of the tapes in it. The temperature of data is defined as the heat of data divided by its size. Originally, temperature was introduced to evaluate the cost performance ratio of data migration in disk arrays. We can transfer heat more efficiently by choosing data with a higher temperature. In tape archivers, however, the cost of each tape migration is always the same, and the temperature of a tape is therefore synonymous with the heat of the tape. In addition, when different archivers have different numbers of drives, it is necessary to normalize the

heat by dividing it by the number of drives. As heat is the abstraction of the object load, it should be normalized depending on its serving capability. In our experimental system, initially two drives are available to each element archiver. However once a drive fails, the serving capability is halved. In order to handle this situation, we redefine its heat as twice the original. Then the load balancer detects the heat imbalance and starts to migrate hot tapes to an element archiver which has lower heat, namely ones with higher serving capability. Heat in disk arrays usually need not be normalized since in general all the component disks are the same kind.

Declustering of hot tapes by migration : *Hot declustering*

High access locality hinders efficient use of the archivers. If hot tapes are concentrated on a few element archivers, the hot element archivers may receive too many tape access requests leaving the cold element archivers underutilized. To reduce the concentration of accesses and to improve efficient use of the resources, it is necessary to scatter the frequently accessed tapes around the STA. For this purpose, two load balancing mechanisms, foreground migration and background migration are introduced into the STA.

Foreground migration. When a new access request is issued for a tape in an element archiver where all drives are currently in use, migrating the requested tape cassette to another element archiver which has a free drive can reduce the response time significantly. We call such migration foreground migration. There are several alternatives for selecting the destination archiver. In [3], four basic destination selection policies were examined: random, space balancing, heat balancing, and distance minimizing. In order for this paper to be self-contained, the policies are briefly explained here again. Random policy selects the destination element archiver at random. Space balancing selects the element archiver in which the number of tape cassettes is smallest. Heat balancing selects the element archiver whose heat is lowest and the nearest element archiver is selected in distance minimizing. We found that there was no significant difference among them. In the following experiments, when there are several candidate element archivers to which a tape cassette can be migrated, the element archiver that has the lowest heat is selected as the destination.

Background migration. Usually the size of the file on tertiary storage system tends to be large. In our satellite imagery database system, each image from NOAA and GMS is around 100 MB. Thus once the reading or writing of the data begins, it takes a relatively long time to move the data compared with the tape handling time of the robotics. During that time drives are busy but the robotics are idle. When the tape handler robots and migration units in both the source and destination element archivers are idle, we can migrate tape cassettes between element archivers so that the heat of the archivers becomes uniform. We call such migrations background migrations. In background migration the cassette is always migrated from the element archiver which has more cassettes to the one holding fewer cassettes. We call the sending archiver the source archiver and the receiving archiver the destination archiver. A cassette for migration is selected as follows. When the heat of the source element archiver is higher than that of the destination archiver, a hot tape is selected for migration. A cold tape is selected in the opposite case.

When more than two background migrations can be executed at the same time, the pair of element archivers whose inventory differs most is selected first. If there is no difference in cassette inventory, then the pair of element archivers whose heat difference is largest is selected. Therefore two basic policies for selecting source and destination; heat emphasizing and space emphasizing are examined in [3]. The heat emphasizing selects the pair of element archivers whose heat difference is largest and the space emphasizing selects the pair of element archivers whose cassette inventory differs the most. Even smaller differences are detected between the policies. Sensitivity control is also an important issue. If the heat balancer is too sensitive to the heat difference, too many migrations occur. Such unnecessary migration degrades the performance. Through extensive simulations, we determined that background migration should be invoked if the heat difference between two archivers is more than 20% or if the difference in number of free slots is greater than three.

Hot data clustering at the tail of a tape : *Hot replication*

Tape drive with multiple load/eject zones. Most of the current commercial tape drives have to rewind a tape when they eject it because the directory information is stored at the physical beginning of the tape. Thus the time to rewind the tape occupies a large percentage of total access time. To solve this problem, tape drives possessing multiple load/eject zones have been developed[12, 13]. In these tape drives the directory information can be stored in each of the load/eject zones. Accordingly these tape drives can load and eject tapes without rewinding.

Clustering of hot files. If frequently accessed files are clustered into adjacent areas, the seek time can be reduced significantly. But it is difficult to know whether the data is hot or not when it is generated. We adopt the following hot data clustering method. During data loading, the system does not completely fill each tape with original data. Some amount of the tail of each tape is reserved for the replicas

Table 1. Simulation Parameters

Element archiver	
Number of element archivers	16
Maximum number of cassettes in an element archiver	200
Number of drives in an element archiver	2
Drive	
Tape load time	35sec
Seek speed	25 MB/sec
Read/Write speed	0.5 MB/sec
Tape eject time	20sec
Tape handler robot and tape migration unit	
Robot move time	2sec
Robot move time with holding and placing cassette	14sec
Wagon unit move time	9sec

Table 2. Initial tape cassette distribution

Element No.	1 ··· 5	6 ··· 11	12 ··· 16
Hot Tapes	8 ··· 8	88 ··· 88	8 ··· 8
Cold Tapes	182 ··· 182	102 ··· 102	182 ··· 182
Total	190 ··· 190	190 ··· 190	190 ··· 190

of hot data. In our implementation, 20% of the tape is used for hot replication. When the tape drive is free, the hot data on cache disks are replicated onto that reserved free area. Normal requests have a higher priority than replication requests. This replication on the tail of a tape is only performed when some of the tape drives are idle. A hot file can be replicated onto the tail of any tape with available space. Usually we do not have to load a new cassette for replication. The cassette used for the previous request can be used for replication. Thus replication can be done efficiently. We can cluster the hot files on the tail of tapes, and the seek time is reduced considerably when the tape drive with multiple load/eject zones is used. In addition, we can overwrite the replicas of hot files in that area without modifying the original files if the access locality changes.

Performance evaluation of hot declustering through tape migration

Simulation parameters

To evaluate the performance of the STA, we execute simulations to measure response time, which is defined as the time from the issue of a request until completion of reading

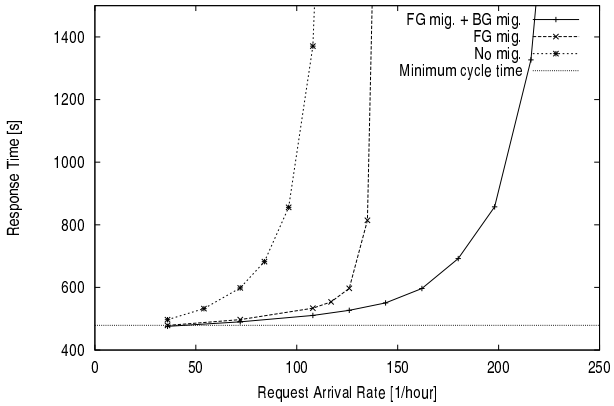

Figure 5. Average response time of initial 50,000 accesses

of the requested data. The simulation parameters shown in table 1 are based on measured values for the experimental STA using the NTH-200B element archivers described in the previous section. We assume that each tape has 4.8 GB of data in it. The size of each file is 100 MB in all the simulations. The read/write time of one 100 MB file is 200 seconds, accordingly, the minimal cycle time is 479 seconds[1] on average. The interval time of request arrival depends on a negative exponential distribution. Because the destination element archiver should have a vacant slot for the migrated cassette, we selected 95% as the load factor. The STA consists of sixteen element archivers. The access locality follows an 80/20 rule, that is 80% of the accesses are to 20% of the tapes. The initial distribution of the tapes in the STA is shown in table 2. The distance of each foreground migration is limited to five element archivers and that of each background migration is limited to one unless slated otherwise.

Effectiveness of foreground and background migration

Figure 5 shows the average response time after 50,000 accesses from the initial tape distribution. Compared to the result of no migration, response time is significantly reduced when only foreground migration is introduced into the STA. The background migration mechanism further improves the performance. When only foreground migration is employed, the response time sharply increases while it moderately increases for the strategy with both foreground and background migration. This is because the foreground migration does not care about space balancing. Busy archivers migrate hot tapes to the idle archivers.

[1] Robot move time + robot move time with holding and placing cassette + drive setup time + average seek time + read/write time + average seek time (for rewinding) + tape eject time + robot move time + robot move time with holding and placing cassette)

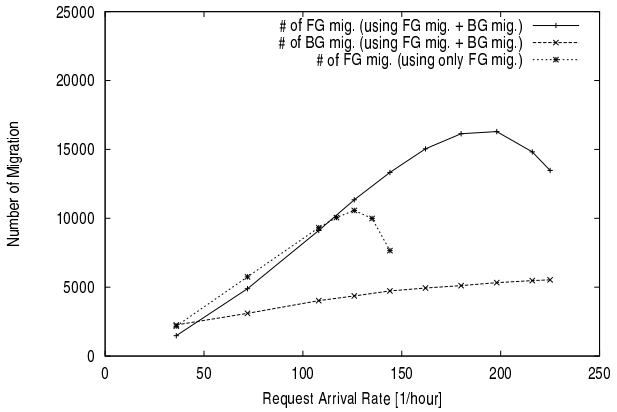

Figure 6. Number of migration of initial 50,000 accesses

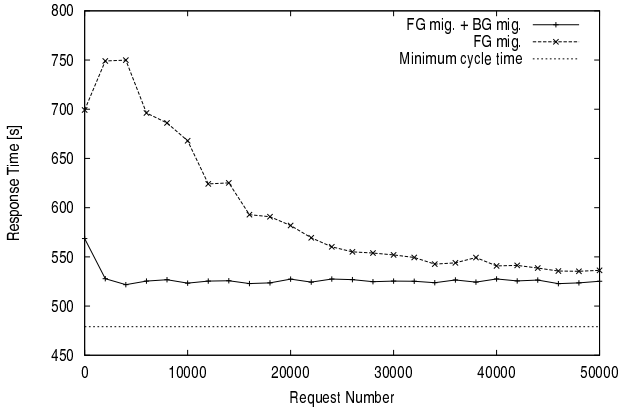

Figure 7. Average response time for intervals of 2,000 accesses

Since there is no background migration, free space on cold archivers can easily become full. Once it becomes full, foreground migrations hardly occur. If background migration in addition to the foreground migration is employed, the load balancer detects the space imbalance and the migration of cold tape from a cold archiver to a hot archiver is immediately invoked. This increases the performance.

Figure 6 shows the number of foreground and background migrations corresponding to figure 5. As the arrival rate increases, the number of migrations also increases. However the number of foreground migration starts to decrease after a certain arrival rate. This is because for such high request arrival rate, most of the drives are busy serving requests from its own element archiver, and cannot serve the requests from the other archivers.

Figure 7 shows the average response time at intervals of 2,000 accesses where the request arrival rate is 126 requests per hour. Using background migration makes

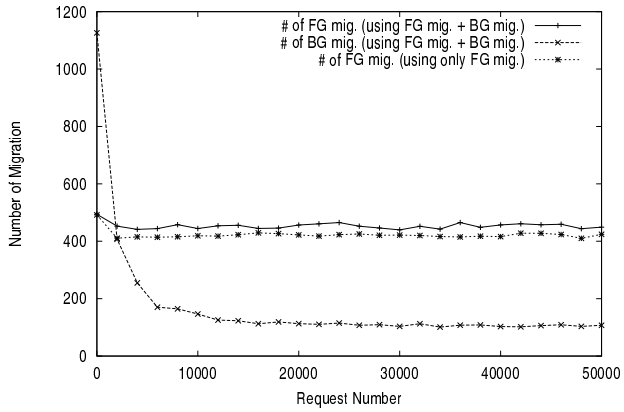

Figure 8. Number of migration for intervals of 2,000 accesses

it possible to track the changing of access locality quickly. It can be seen that convergence is more rapid. Figure 8 shows the number of migrations for the same horizontal axis. In the strategy using both foreground and background migrations, very frequent background migrations occurring early contribute to the heat balancing. On the other hand, in the foreground-only strategy the response time at first increases a lot and thereafter slowly converges to around 600 seconds. This happens for the following reason. Because initially every element archiver has enough empty slots to accept the in-migrated cassettes, foreground migration works well. However, after a while, all the empty slots in the cold archivers are used up. That is, hot tapes occupy the free slots on the cold element archivers close to the hot archivers, since the cold element archivers accept all the hot tapes transferred by foreground migration but it is infrequent that the cold archiver with fewer hot tapes migrates cassettes back to the hot archivers. Thus the cold element archivers no longer have any empty slot to accept new cassettes. Due to this, the response time increases and the number of foreground migrations drops down until around 4,000 accesses. After that, the STA slowly converges to the stable state. This is caused by the migration from the cold element archiver without vacant slots to the hot archivers. But this migration is very infrequent since cold archivers contain only a few hot blocks. Thus the convergence is much slower than in the strategy with background migration.

On the other hand, when both foreground migration and background migration are employed, background migration can produce empty slots in cold archivers by transferring back a cassette to a hot archiver with many empty slots through the space balancing technique. Thus background migration significantly accelerates convergence.

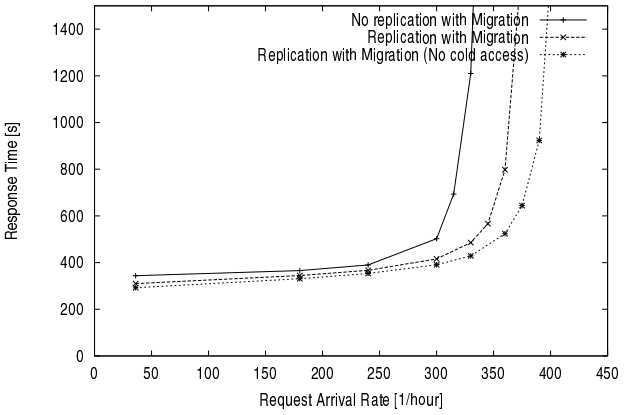

Figure 9. Average response time of initial 50,000 accesses with hot declustering

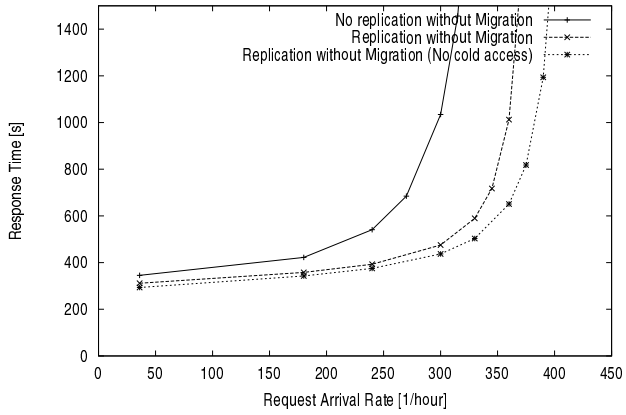

Figure 10. Average response time of initial 50,000 accesses without hot declustering

Performance evaluation of hot replication

Simulation parameters

In this section, we evaluate the performance of the hot replication scheme. The simulation parameters of STA are the same as in the previous case. We assume that all of the drives can load/eject tapes without rewinding. Two kinds of data, hot data and cold data, are stored in the STA and they follow the 90/10 rule. The size of each data is 100 MB. The capacity of each tape is 7 GB and 4.8 GB space from the beginning of each tape is used for original data and the tail area is used for the replicas of hot data. At the beginning of this simulation, the hot data has already been replicated and no replication is executed during the simulation.

In this simulation we adopt the following scheduling algorithms. First we try to find the element archiver which has a free drive and then determine the tape which receives the largest number of service requests. All the requests issued for that tape are scheduled to minimize the total seek and are served at one time. If the element archiver possessing an idle drive has no tape to be served, the tape receiving the largest number of waiting requests in the other element archiver is migrated and served. Here, the foreground migration mechanism is also expected to help hot replication. When the requests to hot files are accessed, replicas are accessed instead of original files.

Effectiveness of hot replication

Figure 9 and figure 10 show the average response time of 50,000 accesses from the beginning of the simulation in the case where hot declustering is employed and in the case where hot declustering is not employed. Each case has three lines: the first curve is the case for no replication. The second is the case for replication. By employing hot replication, the response time is much improved. The last curve is denoted as "No Cold Access". In the first two curves, 90/10 rule is employed, but in the last curve 100/10 rule is adopted. That is, all the requests are given to hot data, which means seek is minimized. Thus the third curve can be regarded as an upper bound. For 90/10 distribution, the performance of hot replication approaches the third curve.

No matter whether hot declustering is employed or not, it is shown that hot replication can largely improve the performance. The average seek length is reduced much when the request arrival rate is low and high. When the request arrival rate is low, the element archivers almost always have an idle drive and the requests are served one by one. In this situation the replicas are always accessible and then the seek length is shortened. The difference of the seek length between the case where the hot files have replicas and the case with hot replication and without replication is around 1000 MB[2]. It takes 40 second to seek 1000 MB. This corresponds to the difference of response time at the same request arrival rate at figure 9 and figure 10 when the request arrival rate is 36 requests per hour.

Figure 11 and figure 12 also show the average response time for 50,000 accesses from the beginning of the simulation in the case where hot declustering is employed and case where hot declustering is not employed. The simulation parameters in these figures are the same as those of figure 9 and figure 10, except for the size of the archived file. In these simulations the size of each file is 20 MB, which is the average size of compressed GMS satellite images. In these situations hot replication is more effective. The seek time occupies a large part of the response time because the read/write time is small when files are com-

[2]The seek speed of Exabyte drive is 25 MB/sec, which is much faster than the read speed, 0.5 MB/sec

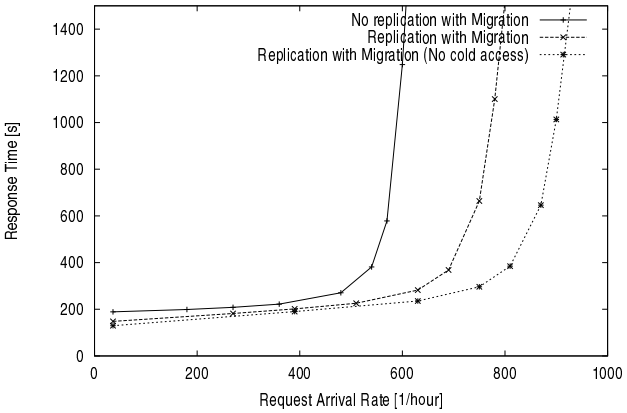

Figure 11. Average response time of initial 50,000 accesses with hot declustering

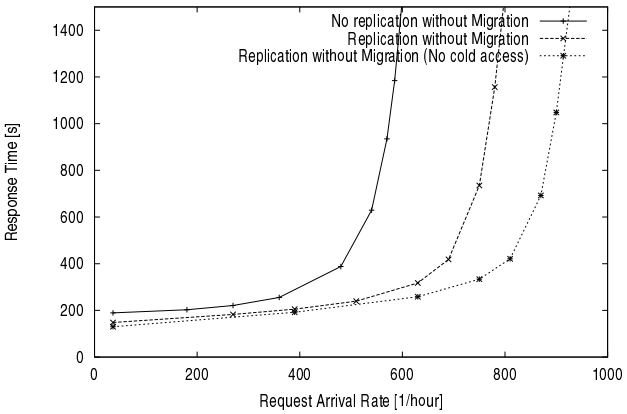

Figure 12. Average response time of initial 50,000 accesses without hot declustering

pressed. Hot replication reduces the average seek time. Accordingly hot replication can improve the performance further when the size of archived files is smaller.

Performance evaluation with access trace of satellite image database

Satellite image database system

The earth surface images from satellite NOAA have been received for about fifteen years at the Institute of Industrial Science, University of Tokyo since 1983 and the reception of the images observed by satellite GMS was begun in 1995. All of the received images are stored in our experimental STA which serves requests from earth scientists at Japanese universities and also from international users. Currently 29,800 images from NOAA, which are 3 TB in size and 29,000 images from GMS, 3 TB in size are archived. A commercial RDBMS manages the information related to the archived images such as satellite names, file names and sizes of archived images, file names of the reduced images (called quick look images for browsing stored on disk array). The RDBMS is connected to the HTTP server so that all of the images can be retrieved through the Internet via World Wide Web (WWW). The HTTP server invokes CGI scripts to make inquiries to the DBMS, to retrieve the images, to display the image information, to migrate images from tapes to disks, to send images to clients and so on. Currently only the sites connected by wide band network such as ATM are permitted to access the original raw image because of their size. However, anybody can access to the quick look images on the disk array through the Internet. They are also available by gopher and ftp.

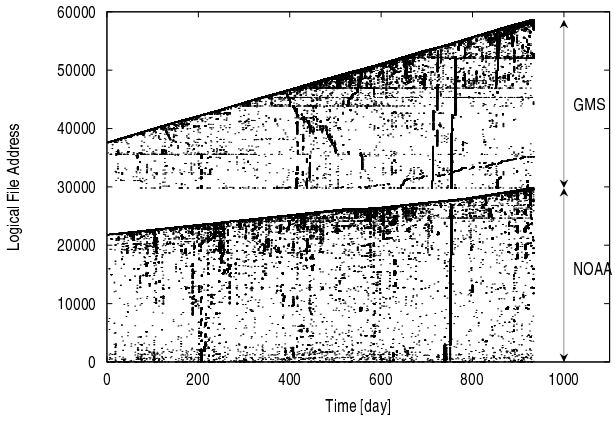

Figure 13. Distribution of requests for satellite image database

Access trace log

Figure 13 shows the distribution of access requests for the quick look images on our satellite image database from April 1996 to October 1998. The total number of requests is about 461,000, which consists of 49,000 requests through ftp, 215,000 through gopher and 197,000 through WWW approximately. The horizontal axis in figure 13 represents the elapsed days since April 1st, 1996. The vertical axis denotes the logical file addresses. A dot in the figure represents an access request. The image files are grouped into two subgroups for NOAA and GMS and are numbered in the order of their reception. The NOAA images are numbered from 0 to 29,799 and the GMS images from 29,800 to 58,636.

We found that most requests fall into two classes.

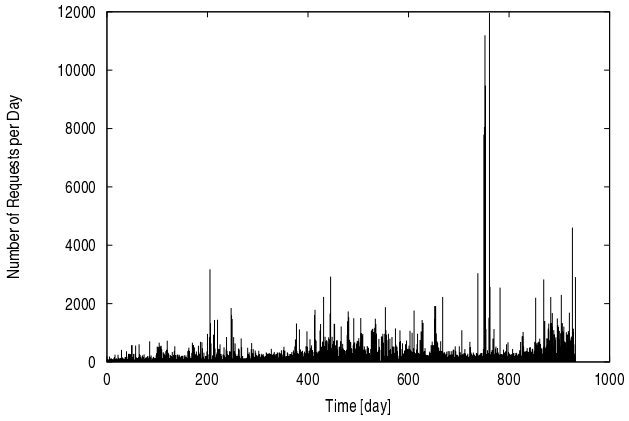

Figure 14. Number of requests per day for satellite image database

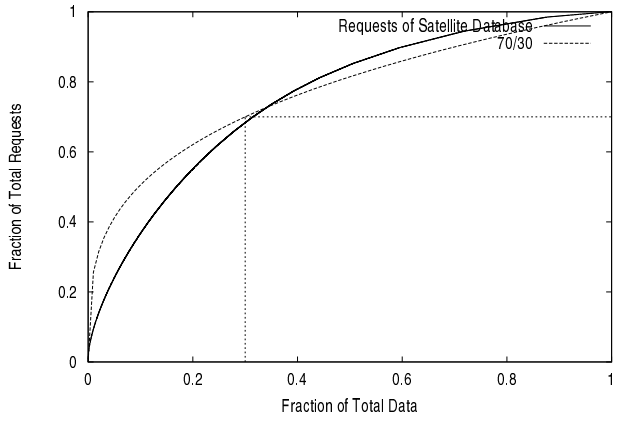

Figure 15. Access locality of satellite image database

One of them is a group of requests issued for the latest images just after the images are received. This group is represented by the two lines in figure 13. The other is a group of requests where a few clients accessed a series of image files in a short period. This causes a sequential access to a lot of archived images. In figure 14 they are shown by vertical lines. Figure 14 indicates the number of the access requests per day. There are some days when more than 10,000 requests were received. Most such busy days correspond to the vertical lines in figure 13. The access locality of the requests for our satellite image database is shown in figure 15. The curve following 70/30 rule shown in [10] is also indicated. In our satellite image database, approximately 70% of the requests are directed at 30% of the data. However, there is a difference between these two lines. The curve shown in [10] indicates that the hottest data receive much more requests. Our satellite image database has milder distribution.

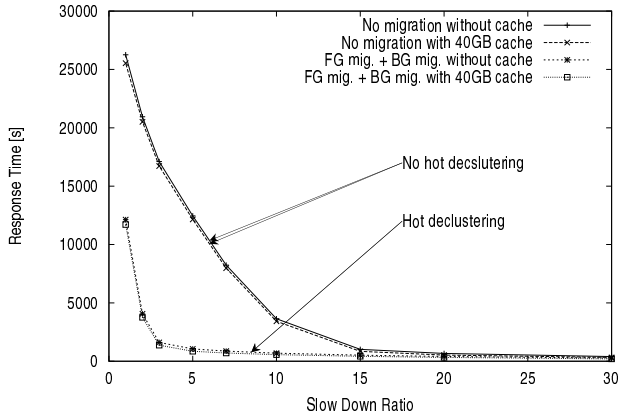

Figure 16. Average response time of requests for satellite image database

Performance evaluation of hot declustering using access trace

Simulation parameters. We performed simulations to evaluate the performance of the declustering schemes using real traces against the satellite image database. In these simulations, we assume that the access requests for the quick look images are issued against the corresponding original satellite images. The distribution of the requests for the quick look images is not necessary the same as that for the original images. However, we assume that these two distributions do not differ much. Each original data has its own reduced image and some characteristics such as that the most recent images receive the most requests and the series of images observed at a certain period are accessed together are common to both cases.

In these simulations, in addition to the 461,000 requests via WWW, gopher and ftp, we also simulated around 28,000 write requests caused by the newly received images. Thus the total number of the requests issued for STA in these simulations is 489,000. As for the request arrival rate, since the file size of the quick look image is much smaller than that of the original image, we simulated the low traffic situation by expanding the interval of the requests. We call this degree of expansion “slow down ratio”. The location of image files on tapes is equal among the real archivers and the simulations. All files are divided into two groups, NOAA files and GMS files, based on their observation satellite and they are stored separately chronologically. The NOAA files are stored on 570 tapes. The first 548 tapes are 112m long (5 GB capacity) without compression and the rest of them are 160m long (7 GB capacity) with compression. The GMS files are stored on 94 160m long

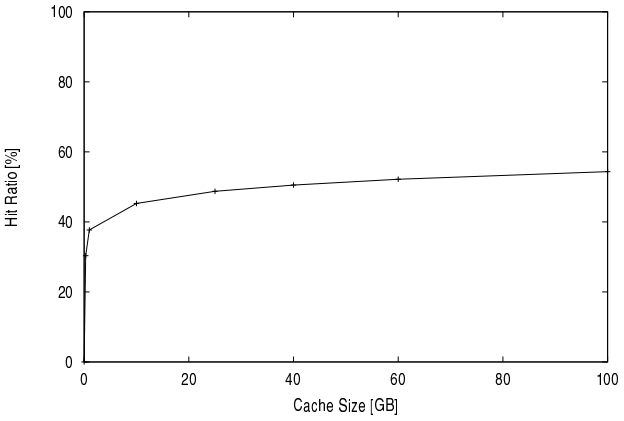

Figure 17. Cache hit ratio of requests for satellite image database

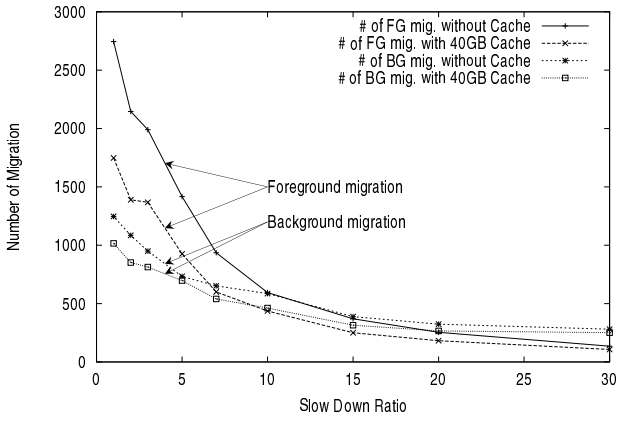

Figure 18. Number of migrations by requests for satellite image database

tapes with compression. Compression is not very helpful for NOAA, is useful to the GMS data. We assumed that NOAA data is not very compressible, assigning a value of 0.66 to it, but GMS files, being more compressible, were assigned a value of 0.2.

The STA consists of four NTH-200B element archivers. In the initial situation, the tapes storing the NOAA files are placed on the first, second and third element archivers and those for the GMS files are placed on the fourth element archiver. We also execute the simulation in the case where the system includes a 40 GB cache disk. The transfer rate of the cache disk is 10 MB/sec and it follows an LRU data replacement policy. The parameters of archivers are the same as those shown in table 1.

Simulation results. Figure 16 shows the average response time of 450,000 accesses from the initial state. The horizontal axis shows the slow down ratio, the ratio to expand the request interval time. Regardless of the existence of the cache disk, tape migration is significantly effective reducing the response time. It can be seen that hot declustering can reduce the response time to one third of that of non-hot clustering case. Tape migration is much more effective in reducing the response time than the cache disk. Figure 17 shows the relation between the size of the cache disk and cache hit ratio. The hit ratio saturates quickly. We can see that 40 GB cache disk is sufficiently large. Even with a large cache disk, we cannot reduce the hit ratio, thus cannot improve the response time. In short, hot declustering through tape migration is more effective than the cache disk. In addition, using both the tape migration and the cache disk together further improves the average response time.

Figure 18 represents the total number of migrations during the simulation. The number of foreground migrations is small when the request arrival rate is low, that is, when the slow down ratio is high because the STA receives a new request infrequently. When the request arrival rate becomes high, that is, when the slow down ratio is small, foreground migrations occurred more frequently, which contribute to the performance improvement. The number of background migrations also increases as the request arrival rate becomes high because the background migrations are executed to compensate for the space imbalance caused by the foreground migrations.

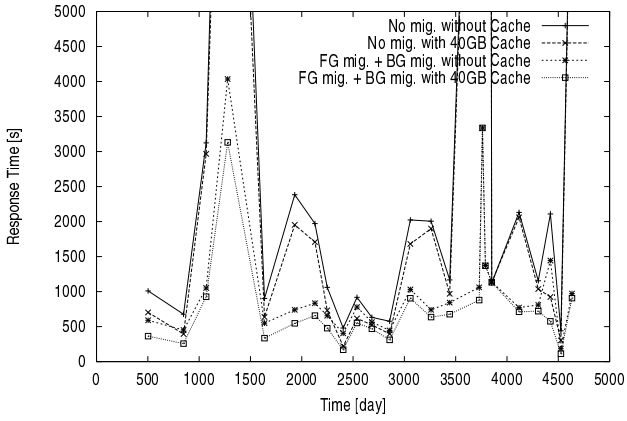

Figure 19. Average response time for intervals of 20,000 requests for satellite image database

Figure 19 shows the average response time for intervals of 20,000 requests when the slow down ratio is five. In the period around 1200th day, the period around 2000th day and the period after 3000th day, the response time is very large when hot declustering is not employed. At these times, the tape migration mechanism significantly

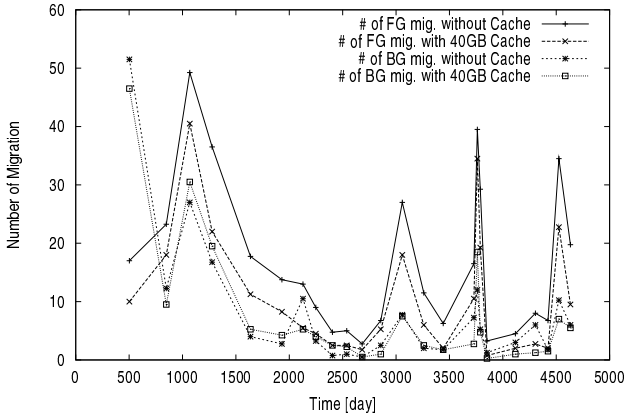

Figure 20. Number of migrations for intervals of 20,000 requests for satellite image database

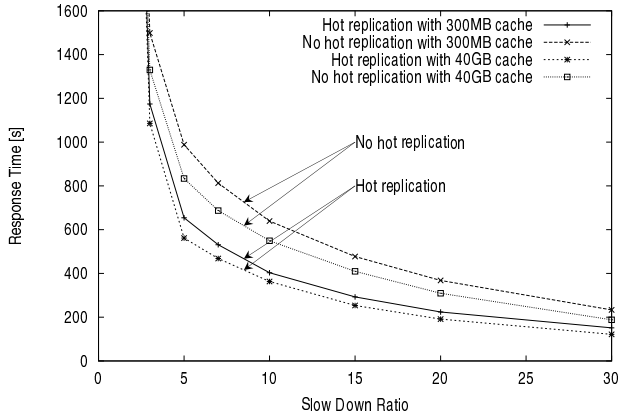

Figure 21. Average response time of 450,000 accesses with hot declustering

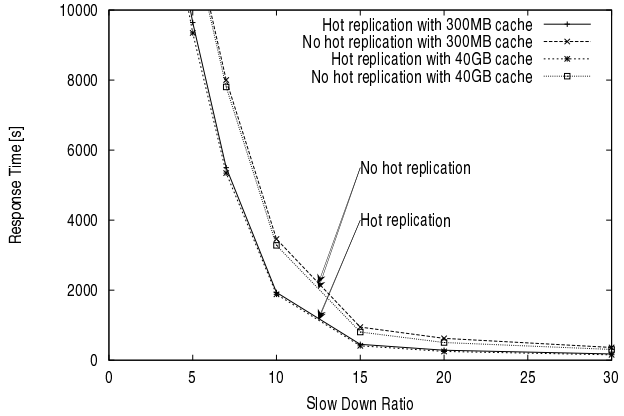

Figure 22. Average response time of 450,000 accesses without hot declustering

decreases the average response time, while the cache disk is not effective. Those periods corresponds to the vertical lines shown in figure 13, which means that many consecutive files were requested. Thus the cache disk does not work well. On the other hand, those files are stored consecutively over several tapes and such tapes are stored in the same element archiver. Thus foreground migration, through which adjacent archivers' drives can be used, helps to improve the performance. Around 2500th day, the migration is not effective while the cache disk reduces the response time. At this time, a small number of files, especially the latest ones, are requested repeatedly. Thus, cache disk rather than tape migration, works well.

Figure 20 shows the number of the tape migrations for intervals of 20,000 requests when the slow down ratio is five. A lot of foreground migrations occurred around 1200th, 2000th and 3000th day. They reduce the response time. Many background migrations are executed at the beginning of the simulation, which helps largely to resolve initial heat imbalance by equalizing tapes in each element archiver. After that, the number of background migrations changes proportionally to the number of foreground migrations. The same phenomenon is found in figure 18.

Performance evaluation of hot replication

Simulation parameters. In this section, we evaluate the performance of hot replication scheme using the access trace of 489,000 requests issued for quick look images in our satellite image database. We assumed that every tape drive can load/eject a tape without rewinding. In this simulation, All of the NOAA files and the GMS files are stored in 160m (7 GB) tapes. The first 5 GB areas are used to store the original data and the remaining 2 GB at the tail of the first 548 tapes storing NOAA files are reserved for replicas of hot files. At the beginning of the simulation, the NOAA files and GMS files received before April 1st, 1996 are stored on the tapes.

There are no replicas of hot files at the start of the simulation. During the simulation, after the file has been accessed five times, it is regarded as hot. The hot file on the cache disk is replicated onto the tail of the tape in an idle drive. The replicas must be after the 5 GB point on the tapes. The hot data are not replicated onto tapes which do not have at least 5 GB of data. To minimize the replication cost, the replication does occur when the candidate file is not on cache disk or when space for replication cannot be found on tapes in idle drives. The other parameters are the same as those described in the previous section.

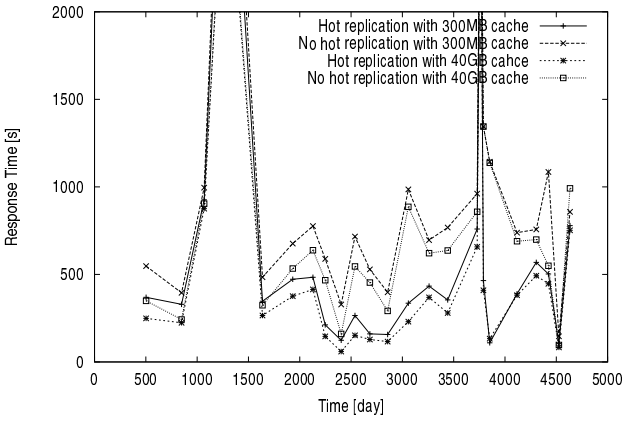

Figure 23. Average response time for intervals of 20,000 requests with hot declustering

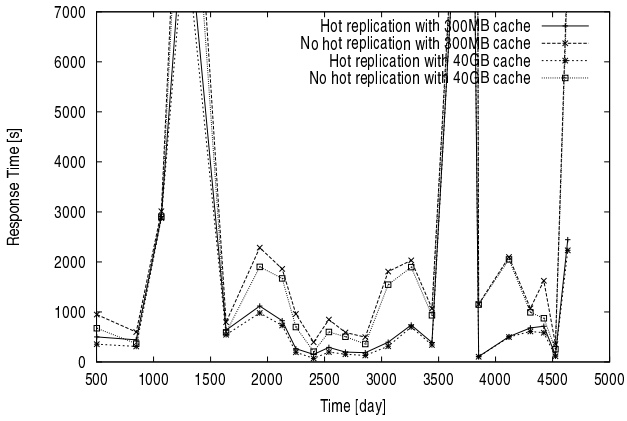

Figure 24. Average response time for intervals of 20,000 requests without hot declustering

Simulation results. Figure 21 and figure 22 show the average response time of 450,000 requests from the initial state with and without hot declustering respectively. In both figures, we compare the systems which include 40 GB disk and the system with only 300 MB. The hot replication reduces the response time regardless of tape migration. It can be seen that hot replication reduces the response time by $30 \sim 50\%$ depending on the slow down ratio for the non-hot declustering case (figure 22). By employing both hot clustering and hot replication, we can further improve the performance as shown in figure 21. Using hot replication in addition to the cache disk can further improve the performance even if the size of the disk cache is sufficient. The performance improvement of hot replication is not so large compared with the results obtained using synthetic data in the previous section. This is due to the fact that access locality is rather weak in the real access trace. We can see that the hot replication is more sensitive to access locality compared with hot declustering.

Figure 23 and figure 24 show the average response time for intervals of 20,000 requests with and without hot declustering. The slow down ratio is set to five. These figures also prove that hot replication can improve the performance of the STA regardless of the existence of cache disk and tape migration. We can see that the response time with hot replication is almost always lower than that without hot replication, which means hot replication is useful in improving performance.

Conclusion

In this paper, we described our satellite imagery archive system using the STA and examined the effectiveness of the clustering and declustering schemes of hot files in the STA through extensive simulations by injecting the request log of more than 460,000 actual accesses. These requests are roughly divided into two categories, requests for latest images and for consecutive accesses.

The STA can handle the requests very efficiently. *Hot declustering* through tape migration significantly improves the performance. Foreground migration and background migration works well to equalize the workload on the system. Newly injected hot images are automatically placed so that the system as a whole is balanced instead of just placing it at the head of empty slot list. The *hot replication* method also improves the performance by exploiting access locality, such as frequent accesses to cloud free images. It reduces the seek cost by clustering the hot files at the tail of tapes. It was shown that hot declustering is more effective than cache disk.

Hot declustering is also very effective for batch access to a series of images. Requested cassettes are dynamically migrated to inactive drives of adjacent element archivers. Thus the requests can be served in parallel by activating multiple drive units, which was made possible by cassette migration mechanism.

Hot declustering can reduce the response time to 10% of that of the system without hot declustering at maximum. Hot replication improves the response time by $30 \sim 50\%$ depending on the request arrival rate. By employing both, we can considerably improve the performance.

Acknowledgment

Prof. Takagi at Science University of Tokyo originally designed the satellite image receiving antenna system. NCL cooperation helped us to design and manufacture the scalable tape archiver hardware based on our specification.

References

[1] B. Kobler, J. Berbert, P. Caulk, and P. C. Hariharan. "Architecture and desing of storage and data management for the NASA Earth observing system data and information system (EOSDIS)". In *Proceedings of Fourteenth IEEE Symposium on Mass Storage Systems*, pages 65–76, Monterey, California, September 1995.

[2] M. Stonebraker, J. Frew, and J. Dozier. "The SEQUOIA 2000 architecture and implementation strategy". Technical report, SEQUOIA 2000 Technical Report 93/23, University of California, Berkeley, CA, 1993.

[3] T. Nemoto, M. Kitsuregawa, and M. Takagi. "Simulation studies of the cassette migration activities in a scalable tape archiver". In *Proceedings of The Fifth International Conference on Database Systems for Advanced Applications*, pages 461–470, Melbourne, Australia, April 1997.

[4] S. Christodoulakis, P. Triantafillou, and F. A. Zioga. Principles of optimally placing data in tertiary storage libraries. In *Proceedings of the 23rd VLDB Conference*, pages 236–245, Athenes, Greece, August 1997.

[5] L. T. Chen, D. Rotem, A. Shoshani, B. Drach, M. Keating, and S. Louis. "Optimizing tertiary storage organization and access for spacio-temporal datasets". In *Fourth NASA Goddard Conference on Mass Storage Systems and Technologies*, Collage Park, Maryland, March 1995.

[6] B. K. Hillyer and A. Silberschatz. "Random I/O scheduling in online tertiary storage". In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 195–204, Montreal, Canada, June 1996.

[7] A. L. Drapeau and R. H. Katz. "Striped tape arrays". In *Proceedings of Twelfth IEEE Symposium on Mass Storage Systems*, pages 257–265, Montrey, California, April 1993.

[8] L. Golubchik, R. Muntz, and R. W. Watson. "Analysis of striping techniques in robotic storage libraries". In *Proceedings of Fourteenth IEEE Symposium on Mass Storage Systems*, pages 225–238, Monterey, California, September 1995.

[9] K. Sako, T. Nemoto, M. Kitsuregawa, and M. Takagi. "Partial migration in an 8mm tape based tertiary storage file system and its performance evaluation through satellite image processing applications". In *Proceedings of 6th International Conference on Information Systems and Management of Data*, pages 178–190, Bombey, India, 1995.

[10] G. Copeland, W. Alexander, E. Boughter, and T. Keller. "Data placement in Bubba". In *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, pages 99–109, Chicago, Illinois, June 1988.

[11] G. Weikum, P. Zabback, and P Scheuermann. "Dynamic file allocation in disk arrays". In *Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data*, pages 406–415, Denver, Colorado, May 1991.

[12] "Ampex DST 312". http://www.ampex.com/html/dst312.html.

[13] "GY-2120". http://www.sony.co.jp/ProductsPark/Professional/DataArchive/BC2/BC2-1/GY2120/index.html.