Three-State Disk Model for High Quality and Energy Efficient Streaming Media Servers

Zhihui Du[1+], Wenjun Fan[2], Yunpeng Chai[3]
[1]Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, 100084, Beijing, China
+Corresponding Author's Email: duzh@tsinghua.edu.cn
[2]Departamento de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid, 28040, Madrid, Spain
Email: efan@dit.upm.es
[3]School of Information, Renmin University of China, 100872, Beijing, China

***Abstract*—Energy conservation and emission reduction is an increasingly prominent and global issue in green computing. Among the various components of a streaming media server, the storage system is the biggest power consumer. In this paper, a Three-State Disk Model (3SDM) is proposed to conserve energy for streaming media servers without losing quality. According to the load threshold, the disks are dynamically divided into three states: overload, normal and standby. With the requests arriving and departing, the disk state transition among these three states. The purpose of 3SDM is to skew the load among the disks to achieve high quality and energy efficiency for streaming media applications. The load of disks in overload state will move to disks in normal state to improve the quality of service (QoS) level. The load of disks in normal state will be packed together to switch some disks into standby state to save energy. The key problem here is to identify the blocks that need migrating among disks. A sliding window replacement (SWR) algorithm is developed for this purpose, which calculates the block weight based on the request frequency falling within the window of a block. Employing a validated simulator, this paper evaluates the SWR algorithm for conventional disks based on the proposed 3SDM model. The results show that this scheme is able to yield energy efficient streaming media servers.**

Keywords—Energy Conservation; Streaming Media Servers; Conventional Disk;Green computing

I. INTRODUCTION

Energy conservation has been extensively studied in green computing and in the context of data centers, because the power consumption has already been a major problem for large scale data centers. Among various components of a data center, storage consumes a large portion of the energy, and storage demand is increasing by 60% annually [1]. For increasingly popular streaming media applications, more and more storage spaces are used for streaming media services. In a typical data center, the storage device usually accounts for 27% of the total electricity consumed [2]. In some disk-dominated storage system, the portion is even higher, approximately 86% [3]. With the increasing need of storage spaces of storage system, this rate will continue to grow. Therefore, the power consumption of the disks in streaming media servers plays a significant role in green computing.

In this research, we focus on designing a new model of disk state transition based on conventional disks to save energy for streaming media servers. A new data movement rule based on the disk state transition model is proposed to implement the data exchange and distribution. The slide window replacement algorithm calculates the block visit popularity, and the blocks of video migrate through the disks in terms of the visit popularity for storage system's energy conservation and load balance.

This paper is organized as follows: In section 2 we briefly describe the background and the related work. In section 3, we propose the energy saving model. In section 4, we propose the data replacement scheme. In section 5, the simulation results of different energy-efficient data layout algorithms are shown. Finally, section 6 gives the conclusion and discussion of future work.

II. BACKGROUND AND RELATED WORK

A. Disk Power Models

Modern disk model uses three power modes: active, idle, and standby. In the active mode, the platters rotate at full speed for head seek, read or write a sector. In idle mode, the platters also spin at the full speed, but there is not any disk activity coming up. So, when the disk is in the idle mode, it consumes most energy but never provides any service. Thus, it's not necessary to distinguish between active mode and idle mode, since in these two modes the disk runs at full speed. When in standby mode, the disk does not work and can't serve any request. In order to provide service, the disk has to go through a cold start to spin up into active mode that stirs up extra power and time.

Gurumuthi et al. [4] have proposed dynamic multi-speed disks which can spin at different lower speeds to increase the opportunities to save energy. The shifting cost of dynamic multi-speed disks between different rotational speeds is smaller compared with that of conventional disks between standby and active mode. Unfortunately it is not clear that the multi-speed disk product can be widely deployed soon because of its prohibitive manufacturing cost.

B. Disk Power Management

The goal of disk power management (DPM) is to save energy by switching disks to low-power mode whenever possible without adversely affecting performance.

For conventional disks, the most common DPM algorithm is Fixed Threshold (FT) [5], in which, a disk is transitioned to low-power mode after a fixed threshold time has elapsed since the last visit. The threshold is usually set to a break-even time, which is the period that a disk would have to be in low-power mode to conserve the same energy consumed by transitioning the disk down and back up to active mode. FT is usually used by other energy-efficient schemes as the DPM algorithm.

For multi-speed disks, Carrera et al. [11] proposed switching the speeds of multi-speed disks based on the observed load. Gurumurthi et al. [4] suggested using changes in the average response time and the disk request queue's length to drive dynamic disk-speed transitions.

C. Related cache replacement algorithm

Traditional disk replacement algorithms mainly include LRU, MRU, LFU, LRU-K 2Q, MQ, etc.

Least Recently Used (LRU): discards the least recently used items first. This algorithm requires keeping track of what was used when, which is expensive if one wants to make sure the algorithm always discards the least recently used item. LRU algorithm is easy to be polluted by a scanning visit, make no distinction of the page access frequency, and let it unable to meet requirements.

Most Recently Used (MRU): MRU algorithm is exactly the opposite of LRU algorithm. It deletes the most recently used items first. When a file is being repeatedly accessed in a looping sequential reference pattern [6], MRU is the best replacement algorithm. In [7], it is pointed out that for random access patterns and repeated scans over large datasets, which sometimes is known as cyclic access patterns, MRU replacement algorithms have more hits ratio than LRU because of their tendency to retain older data. MRU algorithms are most useful in situations where the older item is more likely to be accessed.

Least Frequently Used (LFU): LFU counts how many times an item is needed. Those that are used least often are discarded first. Some items visited many times in a short time, but it is difficult to replace them when they no longer accessed. That is the "cache pollution". Besides, LFU operating cost too much. So in practice, applications will use improved LFU Algorithm, such as LFU-Aging, an item is accessed, its previous visits multiply by a parameter less than 1, so that different access time of visits with different weights.

LRU-K [8] algorithm: LRU-K algorithm records the visiting time of the item be accessed in the K previous times, if an item is accessed less than K times, then the first K of the visit prior to the time recorded as a great value. When item is missing, select the smallest number of visits and the largest LRU value of item to replace. First Look up the item visited only 1 time, select the one with the largest LRU value to replace, if there is no item visited only 1 time, then look up these accessed 2 times, and so on. LRU-K algorithm can replace the items which have few visited times and are long time no use from the buffer, but the LRU-K algorithm is actually a similar LFU algorithm still cannot avoid the problem of pollution buffer.

2Q algorithm [9]: 2Q algorithm maintains a LRU queue, and two FIFO queues: Q_{in}, Q_{out}. Q_{in} saves the item that is in the buffer accessed only 1 time, Q_{out} saves the item replaced. LRU queue saves the item which is accessed more than 1 time. If the request item is in the LRU queue, the visit time of the item plus 1, and if it is in the Q_{in} or Q_{out}, move it to the LRU queue, otherwise, replace the item in the Q_{in}, and add the replaced item to the Q_{out}.

MQ (Multi - Queue) algorithm [10]: MQ algorithm maintains the number of LRU queues Q_0, Q_1, Q_2, ···, Q_{n-1}; for $i < j$, the items in Q_j have a larger life cycle than the items in Q_i. And it also maintains a FIFO queue Q_{out}, for recording the visited times of the replaced the items. With the passage of time, moving the inactive block from a higher level queue to a lower level queue in order to modify the queues, for deleting the items which have a number of visited times but no longer recently accessed, which keep these items still staying in the higher level queue. The MQ algorithm comprehensively considers three properties that include the Minimal Lifetime, Frequency-based Priority and Temporal Frequency, which lets the MQ has a better second level buffer caches hit rate.

Compared with reference [15], this paper provides a new disk model and the corresponding algorithm is developed based on this model.

Different algorithms are prevalent in the various research systems. This is true in currently available products and should continue in the future.

III. THREE-STATE DISK ENERGY SAVING MODEL

The streaming media systems in this paper refer to the common commercial streaming media systems, e.g. You-Tube. These kinds of systems provide real VOD services and contain movies, TV programs, video clips, and many other kinds of video contents.

In this section, we introduce the power management and power model in our streaming media storage system. On one hand, we study the Fixed Threshold (FT) techniques since we use the FT as our DPM algorithm. On the other hand, we introduce our energy saving model, the Three-States Disk Model (3SDM), for conventional disk-based streaming media storage system, and designs energy-efficient disk state transition and data exchange algorithms based on this model. Furthermore, we offer the theoretical energy consumption algorithm to be reference.

A. Study of FT Algorithm

In the current disk energy saving environment, FT is the basis of DPM algorithms. As a review here, we briefly outline FT algorithm as Fig. 1.

We set the length of break-even time as BT in the FT algorithm. λ is the user request rate. The mean length of user

sessions is $\bar{L}$. The disk spends the length of t_d to spin down from active mode to standby mode, and take the length of t_u to spin up from standby mode to active mode. The power of each disk's high-power mode is P_h and the power of each disk's low-power mode is P_l. E_d and E_u are respectively the consumed energy of transitioning a disk down to low-power mode and switching it up to high-power mode.

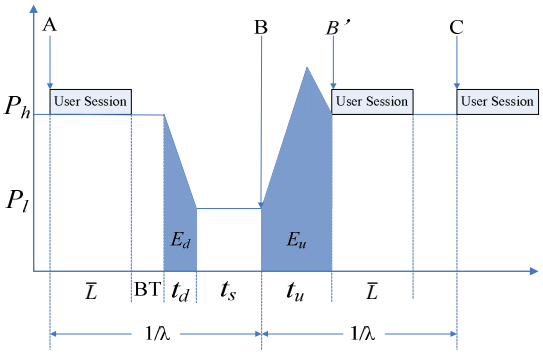

Fig. 1. FT Algorithm

As shown in Fig. 1, A, B and C are 3 adjacent user sessions. Firstly, the disk operates in high power mode for user session A, after the time duration of $\bar{L}$, if there is no request in the length of BT, the disk spins down to the low power mode. Secondly, as another request arrives, the disk spins up again to provide service. But the disk can serve the request only after that it has finished shutting down and then has started up. So B will be served from the time B'. And the disk will continue to serve the next request, if the request such as C arrives before the disk starts switching into standby mode. We find that the disk stay in standby mode between two sessions. So, in this time duration, the disk is saving energy. To simplify discussion, we set this time duration is t_s.

In order to save more energy, an obvious scheme is to extend the length of t_s. Therefore, our target is to allow the disks presenting in the active mode to work as long as possible and the disks working in the standby mode to sleep to the top of their bend.

B. *Introduction of 3SDM*

As we consider the QoS of data center, we know that some disks will be exhausted because of overload. Therefore, in our 3SDM, the disk has three states: overload, normal, and standby, which is set by the real-time disk load. Inspired by the feature of finite-state machine (FSM), which composes of a finite number of states, transitions between those states, and actions, similar to a flow graph in which one can inspect the way logic runs when certain conditions are met. We adopt this concept to express our basic idea on energy saving for streaming media servers. Fig. 2 depicts a view of the disk state transition among these three states.

With the request arriving and increasing, the disk spins up from the standby mode to normal active mode. If the request is too heavy beyond the disk load threshold, the disk will convert to overload state. Another state transition direction is from overload to normal, and then from normal to standby. The disk state changes from overload to normal that can improve the QoS. The disk in normal state spins down to the standby mode that can conserve energy. Therefore, the disk state transition direction "overload -> normal -> standby" is our design target. It does offer a useful means for structuring our research and discussion.

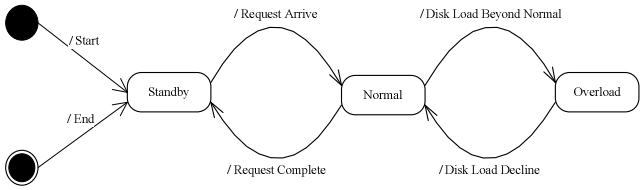

Fig. 2. Disk State Transition

In subsequent sections, we delve into the technical detail and address more subtle research.

C. *Implementation of 3SDM*

To keep things simple, let's assume that 3SDM dynamically divides all disks into overload disk group (ODG), normal disk group (NDG) and standby disk group (SDG) according to their real-time loads. The disks in ODG always work at full speed and never spin down but approximate overload, so they can not satisfy majority user requests very well; the disks in NDG also work at full speed with normal load and never spin, they can serve majority user requests very well; the disks in SDG is designed to mainly stay at standby mode to save energy. The diagram in Fig. 3 introduces the 3SDM model.

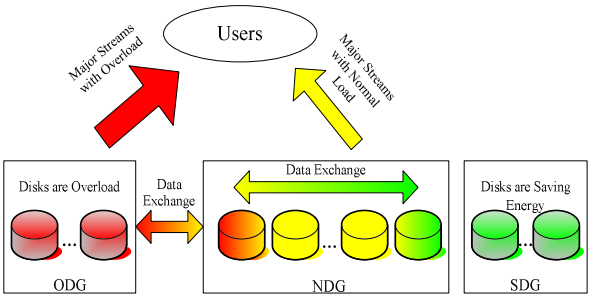

Fig. 3. The 3SDM Model

The target of 3SDM is to maximize energy saving while insure that ODG and NDG can support enough user requests with the least data migration. The key for maximizing energy saving is to keep all standby disks staying in the sleeping mode and retain as many standby disks as possible. Furthermore, load balance is another significant consideration. The disks in ODG assume a high load and cannot provide better service. The hottest data in overload disks have to be swapped into the normal disks. Last but not least, 3SDM extremely distribute the data of NDG in order to make the hot with normal load disks much hotter and the cold normal disks decreasing much colder.

In our previous work [15], we offered a disk energy conservation model, where disks are separated into two groups: one group shoulders the major load, and the other group provides the minor load. However, the disks in the major load

group may be in standby mode, whereas the disks in the minor load group can be in the normal or even overload mode. Instead, we clearly outline the three disk states in our current model.

We use the disk full load transmission bandwidth as the system full load threshold. The disk full load threshold (DFLT) is defined by equation (1).

$$DFLT = \frac{1000ms}{DST + DRT + \left(\frac{VB}{DITR}\right)} * VB \quad (1)$$

Table I describes these parameters in equation (1). DST is 3.4ms, DRT is 2.0ms, VB is 40KB/s and DITR is 60MB/s. So, the DFLT is 6.6MB/s in our system. In practical operation, the system is hard to reach the theoretical full load, so we set 5 MB/s as the overload threshold.

TABLE I. IBM ULTRASTAR 36Z15 DISK PARAMETERS

Parameters	Value
Standard Interface	SCSI
Disk Rotation Speed	15000RPM
High Power(Read/Write/Seek)	13.5W
Low Power(Standby)	2.5W
Spinup Time(Standby to Active)	10.9secs
Spinup Energy(Standby to Active)	135J
Spindown Time(Active to Standby)	1.5secs
Spindown Energy(Active to Standby)	13J
Break-even Time(BT)	15.2secs
Disk Average Seek time	3.4ms
Disk Average Rotate time	2.0ms
Disk Inner Transfer Rate	60MB/s
Video Bitrate	40KB/s

The temperature of data block is reflected by Block Weight (BW), which can be calculated as (2), in which AN_i is the total access number of Block i within a past fixed time window. δ is significantly larger than 1 if Block i is the video prefix that is the first block of the video file, otherwise is equal to 1. $Size_i$ is the size of block i, and we give the time duration of each block a value of two minutes. So the block size is approximate 2MB, if the video bitrate is 128kbps.

$$BW_i = \delta * \frac{AN_i}{Size_i} \quad (2)$$

D. Dynamic Disk State Transition

1) Calculating System Load

The system load (SL) is calculated as (3), in which N is the total block number in the system, CLP is the computing load period, AC_i is the access count of block i in the computing load period, $Size_i$ is the size of block i, which has mentioned before.

$$SL = \frac{\sum_{i=0}^{N-1}\left(AC_i * Size_i\right)}{CLP} \quad (3)$$

In our system, we set the value of CLP with 60000ms. Obviously, the value of CLP can be changed to other parameter.

2) The Algorithm of Dynamic Disk State Transition

Step 1. According to the SL calculated by (3), and the DFLT calculated by (1), 3SDM determine how many disks should be active to satisfy the request.

$$N_{active} = \frac{SL}{DFLT} \quad (4)$$

Step 2. Calculating the average block weight of all disks and then ranking these disks in descending order. The first N_{active} disks will be chosen as active disks, while others should be in standby states.

Step 3. Setting the disks load beyond the disk full load threshold as the overload disk, and the other active disks are normal disks.

Step 4. The hotter normal disk number and the colder normal disk number are calculated by equation (5) and (6).

$$N_{normal-colder} = \left\lfloor \frac{N_{normal}}{2} \right\rfloor + N_{normal} \bmod 2 \quad (5)$$

$$N_{normal-hotter} = N_{normal} - N_{normal-colder} \quad (6)$$

3) The Algorithm of Dynamic Data Exchanging

Step 1. Sorting all video blocks based on block weight in descending order, known as the Block Weight List (BWL).

Step 2. Calculating Middle Block Weight (MBW) by (7) and (8), in which len (BWL) stands for the length of the BWL. Therefore, MBW is the approximate minimum value of video block weight in active disks.

$$k = \left\lfloor \frac{N_{active}}{N_{total}} * len\left(BWL\right) \right\rfloor \quad (7)$$

$$MBW = Block\ Weight\ of\ BWL\left(k\right) \quad (8)$$

Step 3. If there are video blocks with higher Block Weight than MWV in SDG, they should be added into the block swap request waiting queue (BSRWQ). However, we never force any disk in SDG to wake up just for data exchanging, which will happen until the disk mode becomes active.

Step 4. The first independent thread called LoadBalance swaps video blocks between overload disks and the hottest normal disks. The hottest video blocks in these overload disks will be swapped with the coldest video blocks in this normal disk.

Step 5. The second independent thread called ExtremeDistribution, which swaps video blocks between normal disks. The hottest video blocks in the first hottest normal disk will be swapped with the coldest video blocks in the first coldest normal disk, and the hottest video blocks in the second hottest normal disk will be swapped with the coldest video blocks in the second coldest normal disk, etc.

E. Theoretical Algorithm of System Energy Saving

The theoretical algorithm of system energy saving is designed to report a greatest lower bound of energy saving and a least upper bound of energy consumption for reference.

The theoretical energy consumption (TEC) is calculated by the equation (9).

$$TEC = N_{active} * serviceTime * Power_{activeDisk} \quad (9)$$

N_{active} comes from the equation (4), service time is the length of service, and $Power_{activeDisk}$ is the power when a disk stays in active mode.

In consideration of the cache effect, the least upper bound of energy consumption (LUBEC) is calculated by the equation (10), in which δ_{Cache} is smaller than 1.

$$LUBEC = \delta_{Cache} * TEC \quad (10)$$

Therefore, greatest lower bound of energy saving (GLBES) can be calculated by equation (11)

$$GLBES = \frac{energyConsume - LUBEC}{energyConsume} \quad (11)$$

IV. BLOCK REPLACEMENT ALGORITHM

From the description made above, we can find that the block weight plays an important role in this model. Both the disk state transition and the data migration need to sort the block according to the block weight.

In our previous work, we didn't focus on prediction algorithm for data migration. Getting more and more energy conservation is a big inspiration for us. So, we presume the prediction has a vital role, which can conserve more energy. Here, we propose a Sliding Window Replacement (SWR) algorithm to improve the accuracy of the prediction, and the ultimate target is to enhance the energy saving efficiency.

Commonly, we use LFU as the replacement algorithm. However, the user's interests are alternated dramatically, and the requests exhibit very strong temporal locality. Therefore, we set a sliding window to count the access frequency falling into the window. Obviously, if the length of sliding window is the whole service duration, the frequency is the same with LFU.

We regard the user request as a time series:

$$F_0, F_1, F_2, \cdots [F_{k+1}, \cdots, F_{k+m}], \cdots$$

The sliding window whose length is m, along the time series forward sliding, counts the temporal frequency.

The access number (AN) with SWR is calculated by the equation (12)

$$AN_i = AN_{i,j} + \delta_{History} * AN_{i,j-1} \quad (12)$$

$AN_{i, j-1}$ is the access number of Block i in the last sliding window period. $\delta_{History}$ is important to set the history access impact factor. If $\delta_{History}$ equal to 0, AN_i is only the temporal frequency, and is not affected by the history frequency information.

V. EVALUATION METHODOLOGY

In This section, we first introduce the test bed, and then use different methods to evaluate our model and algorithm.

A. Test Bed

We enhanced the widely used DiskSim simulator [12] by adding the support for common DPM algorithms FT. The specifications for the disk used in our study are similar to that of the IBM Ultrastar 36Z15 [13]. The key parameters are shown in Table I.

In this section, we choose FT, PDC[5], 3SDM online and the theoretical algorithm as the evaluation targets. FT supplies energy saving lower limit for energy-efficient data layout algorithms, while the theoretical algorithm provides the approximate upper limit.

B. Real-system Trace

Our experiments use two real-system traces for the purpose of testing the effects of various algorithms in this environment.

First, a 24 hours' real-system trace from the CCTV (China Central Television) VOD system in January 2005 is adopted. Its user arrival rate is 0.198 per second, and its mean session length of all users is 188.65 seconds. Due to the drastic fluctuation of system load, the mean value of online user number is much higher in the evening than in the early morning.

Another 24 hours' real-system trace is obtained from UUSee systems [14]. The user arrival rate is 0.0591 per second, and the mean session length is 1094.48 seconds.

Thus, the CCTV VOD trace represents short-session dominated streaming media applications, similar to YouTube, while the UUSee trace stands for long-session dominated movies-on-demand applications.

C. Different Energy Saving Model

In energy saving aspect, in Fig. 4, we can see that FT always has the worst energy saving efficiency. Compared with FT, PDC has some improvement, however, which is limited. Its energy-saving is only 2.2 times of that of FT for CCTV VOD trace, and 2.6 times for UUSee trace. The energy saving effect of the 3SDM online algorithm in 24 hours is 10.69 times of FT and 4.9 times of PDC for CCTV VOD trace, and is 6.92 times of FT and 2.72 times of PDC for UUSee trace.

The 3SDM online algorithm outperforms other online algorithms because it keeps enough disks always be staying in standby mode. Admittedly, the theoretic algorithm has the best effect, and keeps saved energy up to more than 85% at any cases for it doesn't need to consider disk state transition and data migration.

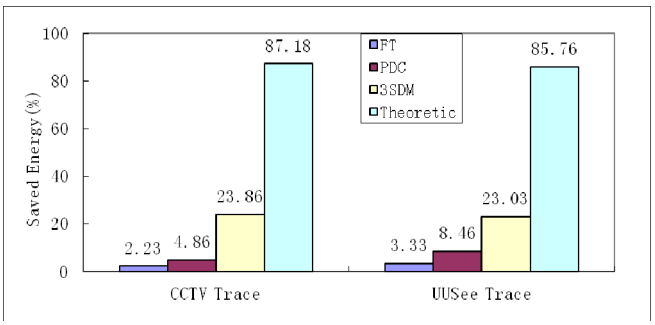

Fig. 4. Saved Energy with Different Saving Model

D. *Different Length of Sliding Window*

Fig. 5 shows the different sizes of sliding window affecting the energy saving efficiency. In this test, we set $\delta_{History} = 0$, so the history information will not impact on the current access statistics and the size of the sliding window is the unique impact factor.

From the column chart, we find that one hour size sliding window have the best energy saving efficiency, since both the length of CCTV trace and UUSee trace are just 24 hours and every hour the user's interest is relatively stable.

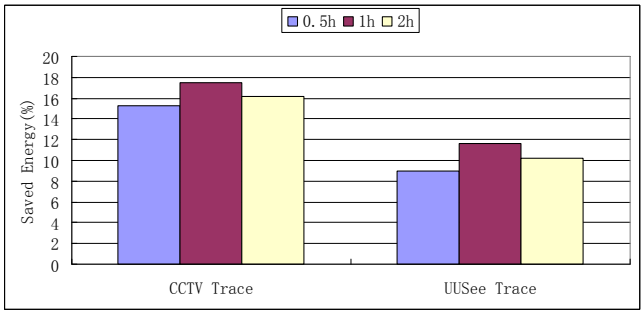

Fig. 5. Different Size of Sliding Window

E. *Different History Access Impact Factor*

SWR is applied to calculate the block weight. In our system, we set the length of the sliding window to 1 hour, based on the last part discussed that 1 hour sliding window have the best energy conservation efficiency.

Table II, Fig. 6, and Fig. 7 show the tendency both of saved energy and mean startup delay going with the increasing value of $\delta_{History}$.

We change the history access impact factor $\delta_{History}$ to observe the energy saving situation and QoS. The result is shown as Table II. When $\delta_{History}$ equals to 1, SWR is just like the LFU. With the increasing value of $\delta_{History}$, the saved energy of UUSee trace climbs up and arrives to the peak at the value 1.1, but the saved energy of CCTV trace is rising up to 26.08% that is not the energy saving peak yet. Based on the result, we infer that the history access information can improve the block hit ratio, and the CCTV trace is affected more than the UUSee trace by the history access impact factor so far.

TABLE II. DIFFERENT HISTORY ACCESS IMPACT FACTOR

CCTV Trace						
$\delta_{History}$	1	1.1	1.2	1.3	1.4	1.5
Saved Energy(%)	23.86	23.94	24.21	25.05	25.61	26.08
Mean Startup Delay(ms)	303.19	334.11	351.59	385.09	390.79	405.23
UUSee Trace						
$\delta_{History}$	1	1.1	1.2	1.3	1.4	1.5
Saved Energy(%)	23.05	24.70	25.09	25.10	24.90	24.90
Mean Startup Delay(ms)	358.44	411.79	432.94	421.67	428.65	446.93

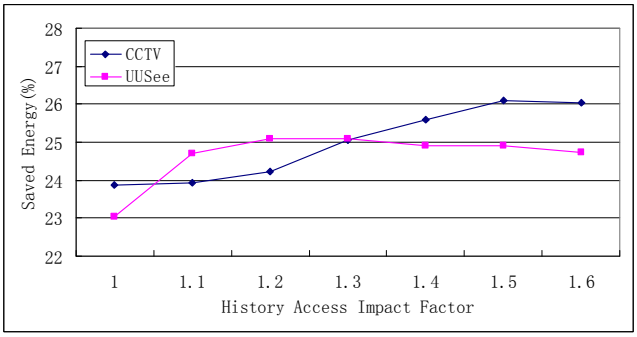

Fig. 6. Energy Saving of Different History Access Impact Factor

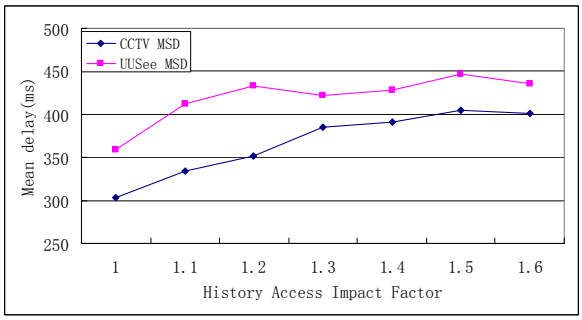

Fig. 7. UUSee QoS of Different History Access Impact Factor

As far as the QoS, we find that the mean startup delay is extended. This is easy to understand, because more energy saving means more disks are in standby mode, so when a request to the sleeping disk which have to wake up to serve

the request, which will occur the delay. Therefore, a balance between energy saving and QoS is another significant issue.

F. System Load Analysis

Fig. 8 and Fig. 10 present the system load of CCTV and UUSee trace, Fig. 9 and Fig. 11 demonstrate the corresponding active disk number.

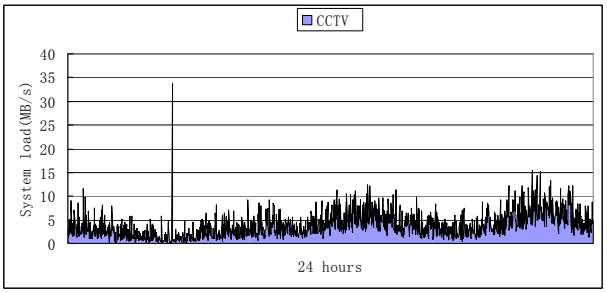

Fig. 8. CCTV 24 hours system load

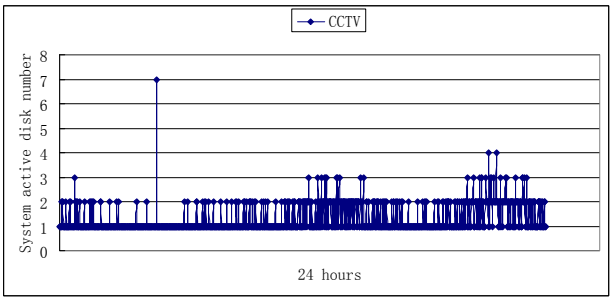

Fig. 9. CCTV 24 hours system active disk number

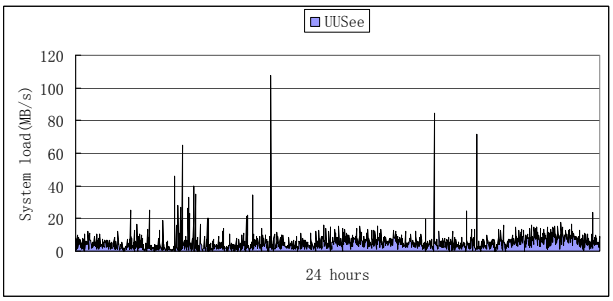

Fig. 10. UUSee 24 hours system load

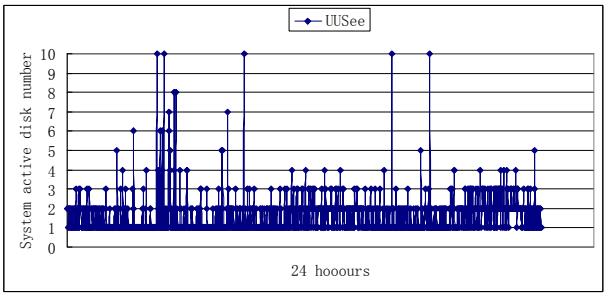

Fig. 11. UUSee 24 hours system active disk number

When the system load aggrandizes, the system active disk number dynamically increases to serve the requests. However, if the load surpasses the disk overload threshold, the system just can provide the total number of disks to satisfy the request. From Fig. 8 and Fig. 10, we find that the load exceeds 60MB/s sometimes, however, our system full load is 5*10=50MB/s, implying that only 10 disks can provide service.

With the effect of 3SDM, the active disk number is relatively stable. With the system running, the active number in CCTV system is 1, and in the UUSee system, the active number is 2.

G. Disk Load Analysis

We also monitor each disk load variations with both the CCTV trace and the UUSee trace. The result is shown as in Fig. 12 and Fig. 13. We can find that the system load is distributed on each disk of the storage system at the beginning of the test. Since user's requests are stable at first. After the stable phase, most of the requests focus on just several disks, because 3SDM migrates the most popular data blocks in to the hot active disks which can serve the request without overload.

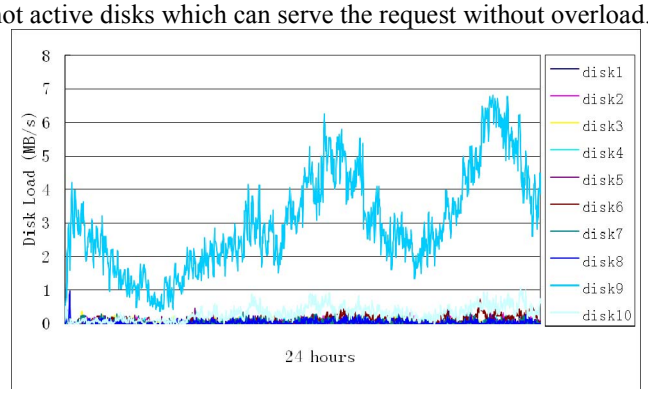

Fig. 12. Each disk load situation with CCTV 24 hours trace

From Fig. 12, disk 9 owns the most popular data blocks, and thus it bears the main load of the system. However, other disks also support some requests since some "cold" data blocks will be accessed but won't be migrated into disk 9.

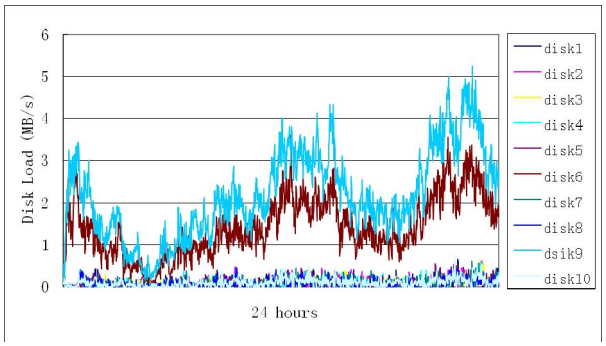

Fig. 13. Each disk load situation with UUSee 24 hours trace

From Fig. 13, we observe that there are two disks supporting the main services, which is consistent with the description of last part.

Therefore, how many active disks exist in the system that depends on the load from the request and the data block

distribution on the storage system. If the requested data blocks only distribute in the same disk, and the disk also bears the total request load, then the energy saving efficiency will be quite spectacular. However, if the total request load is not too heavy, but the requested data blocks are dispersed in many disks, these disks have to spin up to serve the request, and then the energy saving efficiency will be low. The goal of data migration is just to skew the load into several disks to centralize the popular data blocks and then to save more energy.

H. Special Access Pattern Trace

In order to test the performance of 3SDM, we generated a 60 minutes' trace and the requests all refer to one video. Under this condition, the result of the saved energy is 73.01%, the mean startup delay is 20.62ms and the mean delay jitter is 21.71ms. The energy saving effect dramatically approach to the theoretical algorithm result since there is only one disk in the active mode to provide service.

This spectacular energy saving result will occur such as the phenomenon like the new movie put into the homepage of the video website. If we can use these kinds of information, we can predict the request and improve the energy saving efficiency.

VI. CONCLUSION AND FUTURE WORK

In this paper, the disks of storage system are regrouping, based on the 3SDM model, into three state groups according to the system dynamic load. The disk states are overload, normal or standby. The design target of 3SDM urges the overload disks to change its state to normal for enhancing the QoS. With regard to normal disk, the 3SDM makes it switch into standby mode to save more energy.

For data block migration, we propose the SWR algorithm. The length of sliding window and the history access impact factor are two changeable parameters. With the better configuration of these two parameters, 3SDM could save more energy and provide better service.

There is room for further research and improvement. Different methods can be adopted to improve the block replacement algorithm, such as block lifetime, frequency-based priority and temporal frequency. Furthermore, we can explore better prediction methods and apply other information to calculate the block weight, such as user information, video website homepage information, and the video class.

ACKNOWLEDGMENT

This research is supported in part by the National Natural Science Foundation of China (No.61272087, No. 61202115, No. 61073008, No. 60773148 and No.60503039), Beijing Natural Science Foundation (No. 4082016 and No.4122039).

REFERENCES

[1] B. Moore, "Taking the Data Center Power and Cooling Chal-lenge," Energy User News, August 2002.

[2] C. Weddle, M. Oldham, J. Qian, A. A. Wang, P. Reiher and G. Kuenning, "PARAID: A gear-shifting power-aware RAID," ACM Trans. Storage, vol. 3, no. 3, pp. 245-260, October 2007.

[3] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton and J. Wilkes, "Hibernator: Helping Disk Arrays Sleep through the Winter," ACM Symp. Operating Systems Principles, pp. 177-190, Oct. 2005.

[4] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir and H. Franke, "DRPM: Dynamic speed control for power management in server class disks," Proc. 30th Annual Int'l Symp. Computer Architecture, pp. 169-179, June 2003.

[5] E. Pinheiro and R. Bianchini, "Energy Conservation Techniques for Disk Array-Based Servers," Proc. 18th Int'l Conf. Supercomputing, pp. 68-78, June 2004.

[6] Hong-Tai Chou and David J. Dewitt, "An Evaluation of Buffer Management Strategies for Relational Database Systems," Proceeding of the 11th international conference on Very Large Data Bases-Volume, pp. 127-141, August 1985.

[7] Shaul Dar, Michael J. Franklin, Björn Þór Jónsson, Divesh Srivastava, and Michael Tan, "Semantic Data Caching and Replacement," Proceeding of the 22th International Conf. on Very Large Data Bases, pp. 330-341, September 1996.

[8] Elizabeth J. O'Neil, Patrick E. O'Neil, and Gerhard Weikum, "The LRU- K Page replacement algorithm for database disk buffering," Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 297-306, May 1993.

[9] T. Johnson, D. Shasha, "2Q: a low overhead high performance buffer management replacement algorithm," Proceedings of the 20th International Conference on Very Large Data Bases, pp.439-450, Septeber 1994.

[10] Yuanyuan Zhou, James Philbin, "The multi-queue replacement algorithm for second level buffer caches," Proceeding of the General Track: 2002 USENIX Annual Technical Conference, pp. 91-104, June 2001.

[11] E. V. Carrera, E. Pinheiro and R. Bianchini, "Conserving Disk Energy in Network Servers," Proc. 17th Int'l Conf. Supercomputing, pp. 86-97, June 2003.

[12] G. Ganger, B. Worthington, and Y. Patt, The DiskSim simula-tion environment (v4.0). http://www.pdl.cmu.edu/DiskSim. September, 2009.

[13] Hitachi, Ultrastar 36Z15 Datasheet. http://www1.hitachigst.com/hdd/ultra/ul36z15.htm, March, 2011.

[14] X. Xiao, Y. Shi, Q. Zhang, J. Shen, and Y. Gao, "Toward Systematical Data Scheduling for Layered Streaming in Peer-to-Peer Networks: Can We Go Farther?," IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 5, pp.685-697, May 2010.

[15] Chai, Y.; Du, Z.; Bader, D.; Qin, X.; "Efficient Data Migration to Conserve Energy in Streaming Media Storage Systems . Parallel and Distributed Systems," IEEE Transactions on. Nov. 2012 (vol. 23 no. 11) pp. 2081-2093