ARTICLE IN PRESS

Simulation Modelling Practice and Theory xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud

Zhihui Du [a,*], Wenjun Fan [b], Yunpeng Chai [c], Yinong Chen [d]

[a] Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, 100084 Beijing, China
[b] Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[c] Key Laboratory of Data Engineering and Knowledge Engineering of Ministry of Education, School of Information, Renmin University of China, 100872 Beijing, China
[d] School of Computing, Informatics and Decision Systems Eng., Arizona State University, Tempe, AZ 85287, USA

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Prediction algorithm
Energy-efficient cloud computing
Conventional disk
Energy conservation

ABSTRACT

One of the major challenges in cloud computing and data centers is the energy conservation and emission reduction. Accurate prediction algorithms are essential for building energy efficient storage systems in cloud computing. In this paper, we first propose a Three-State Disk Model (3SDM), which can describe the service quality and energy consumption states of a storage system accurately. Based on this model, we develop a method for achieving energy conservation without losing quality by skewing the workload among the disks to transmit the disk states of a storage system. The efficiency of this method is highly dependent on the accuracy of the information predicting the blocks to be accessed and the blocks not be accessed in the near future. We develop a priori information and sliding window based prediction (PISWP) algorithm by taking advantage of the priori information about human behavior and selecting suitable size of sliding window. The PISWP method targets at streaming media applications, but we also check its efficiency on other two applications, news in webpage and new tool released. Disksim, an established storage system simulator, is applied in our experiments to verify the effect of our method for various users' traces. The results show that this prediction method can bring a high degree energy saving for storage systems in cloud computing environment.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

A group of leading computer scientists from UC Berkeley's RAD Lab published a white paper that lays out a broad road map for cloud computing [1] in 2009. The authors made an interesting table describing 10 obstacles to the success and opportunities of cloud computing. Admittedly, Cloud could integrate resource to reduce cost and improve performance. It also needs larger energy consumption. Performance is important, but now energy efficiency is even more important. Cloud computing, with all the layers of Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and data center, has developed into a major computing environment, in both industry and academia [2]. In this paper, our research focuses on IaaS and data center layers of cloud computing. Energy conservation has been extensively studied in green computing and in the context of data centers, because the power related costs in storage systems are growing in importance over time. Among various components of a data center, storage consumes a large portion of the energy, and storage demand is increasing by 60% annually [3]. In a typical data center, the storage device usually accounts for 27% of the total

* Corresponding author. Tel.: +86 10 62782530; fax: +86 10 62771138.
E-mail addresses: duzh@tsinghua.edu.cn (Z. Du), efan@dit.upm.es (W. Fan).

1569-190X/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.simpat.2013.06.002

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

electricity consumed [4]. In some disk-dominated storage system, the portion is even higher, approximately 86% [5]. With the increasing need of storage spaces of storage system, this rate will continue to grow.

As the cloud computing and data center industry grows increasingly obsessed with energy efficiency, cloud computing presents a compelling opportunity to reduce data center power bills. There are powerful economic factors pushing us towards cloud computing. One of the major reasons is the more efficient use of power by cloud computing providers.

In 2011, Mark Aggar, senior director of technology strategy of Microsoft presented that for every 100 W of power consumed in a typical data center, fewer than 3 W are associated with actual computing [6]. In other words, a typical data center consumes more than 30 times the energy that it uses to perform computation. Most of the remaining energy is wasted due to server under utilization. Power and cooling inefficiencies account for the rest. It is conceivable that large cloud deployments require thousands of physical machines and megawatts of power. The wasted power will be many times of the typical data center. Therefore, there is a need to create an energy-efficient cloud computing system to utilize the strengths of the data center and reduces the amounts of carbon emissions.

All in all, energy efficiency is imperative to cloud computing and its data centers.

According to knowledge of Web 2.0 [7], we are able to obtain a large amount of data about the user's behavior which could be used to predict the user's future action. Recommendation system is just based on this kind of knowledge, and we also could use it on server-side for disk energy saving. As is well known, many video website like You-Tube apply the concept of Web 2.0, so we could obtain the user's visiting information and then employing it to make energy efficiency. And the next step, we want to extend our visual field to propose a pervasive prediction model aiming at different application under Web 2.0 to achieve high quality and energy efficiency.

In this research, we focus on designing a new prediction algorithm based on priori information from user's behavior. With this method, we could forecast which data will be visited in the closed future exactly. In this paper, we employ conventional disk which never consume energy when it stops working. So, the basic idea of our energy saving scheme is: keep the non-working period as long as possible for all the disks to save energy by taking advantage the accurate prediction information to make this possible.

This paper is organized as follows: In Section 2 we briefly describe the background and the related work. In Section 3, we study the power management knowledge and introduce the Three-State Disk Model. In Section 4, we propose our new prediction algorithm. In Section 5, we provide the detail of the disk grouping scheme. In Section 6, the simulation results of different scenarios are shown. Finally, Section 7 gives the conclusion and discussion of future work.

2. Background and related work

2.1. 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 is 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 cannot 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. [8] 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.

2.2. 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 disk, the most common DPM algorithm is Fixed Threshold (FT) [9], 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 transforming 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. [10] proposed switching the speeds of multi-speed disks based on the observed load. Gurumurthi et al. [8] suggested using changes in the average response time and the disk request queue's length to drive dynamic disk-speed transitions.

2.3. Related work of disk energy efficiency

Gurumurthi et al. [11] have proposed multi-speed disk to save energy much more efficiently with shorter power mode switching time. Zhu et al. [5] have put forward Hibernator energy management system based on more practical multi-speed

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

disks. Nevertheless, these kinds of multi-speed disks are still far from being massive commercial deployment because of their poor cost-performance ratio.

Colarelli and Grunwald [29] introduce MAID to save energy. They make some disks to cache the most popular blocks, but this will result to data redundancy, because the disk cache stores a copy of the most popular blocks. Zhu and Zhou [12] have tried to conserve energy by caching parts of the content in memory in order to create opportunity for some disks to switch or keep into low-power mode. However, several aspects should be taken into seriously consideration for streaming media application. First, the storage space of the media library is tremendous compared with the memory space. Second, the users' interests change very rapidly. So, the energy saving effect of those memory cache replacement algorithms is limited.

Based on the multi-speed disk, Pinheiro and Bianchini [9] propose PDC to save energy. PDC uses MQ [13] replacement to migrate data through disks according to the visit popularity, and divides disks into different layout levels in accordance with the popularity of storage contents by a descending order. This makes the disks with low popularity content gain greater opportunities to remain idle so that they can be transitioned to low-power mode. However, for the streaming media storage, this scheme makes large data migration and brings the extra burden to storage system.

In recent years, researchers proposed to use extra buffer disks to improve energy-efficiency. In [27], the authors proposed an architecture of cluster storage systems, in which each I/O node manages multiple disks – one buffer disk and several data disks. The key idea of this design is to redirect disk requests from data disks to the buffer disk. In 2011, Manzanares et al. proposed an energy-aware prefetching strategy (PRE-BUD). With the support of disk buffers, this strategy can be used for economically attractive and environmentally friendly parallel I/O systems [28]. They compared PRE-BUD strategy with both disk configurations against three existing strategies, and the results of the experiment show that their proposal can reduce energy consumption by 30–50%.

In [20], Chai et al. proposed an efficient data migration technology (EESDC) to conserve energy for streaming media computing environment. First, the authors divide these disks into two groups as the disk grouping scheme. Second, the author did not focus on prediction algorithm for data migration. As we all know, block weight plays an important role in data migration, which would affect the disks grouping directly. So, the prediction has a vital role, if it can forecast the block weight before the closing future visiting, which could result in conserving more energy.

In [21], the authors provided a Three-State Disk Model (3SDM), because the authors thought that QoS must be considered when the streaming media server is overload. 3SDM divides the disks into three groups, and 3SDM employed a sliding window replacement (SWR) algorithm for data block weight computing. SWR is a prediction algorithm for data block. Both the disk state transition and the data migration need to sort the block according to the block weight. And the authors discovered if the size of sliding window and the history access impact factor were set optimal values, we could achieved several percent points more energy conservation. However, 3SDM is just limited to streaming media servers. Furthermore, the energy efficiency brought by SWR still has much room to improve.

2.4. Related cache replacement algorithm

Traditional disk replacement algorithms mainly include LRU, MRU, LFU, LRU-*K* 2*Q*, 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 [14], MRU is the best replacement algorithm. In [15], 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* [16] 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.

2*Q* algorithm [17]: 2*Q* 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}.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

MQ (Multi-Queue) algorithm [13]: 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.

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

2.5. Energy-efficient cloud computing

Cloud computing has many advantages to provide energy efficient storage system. The article [18], reviewed the usage of methods and technologies used for energy-efficient operation of computer hardware and network infrastructure by 2009. And the author presented that the key current technology for energy-efficient operation of servers in data centers is virtualization. VMs that encapsulate virtualized services can be moved, copied, created and deleted depending on management decisions.

However, this method depends on the application. If the application is streaming media service, it will be invalid. Because streaming media servers requires large scale distributed storage systems whatever they are virtual or not.

In 2012, researchers from the University of Melbourne in Australia have come to the conclusion that cloud computing is not always the greenest option on the storage and processing as well as the software level [19]. The authors argued that most studies seeking an answer to a similar question about the "green" nature of the cloud have only looked at the datacenter's energy consumption and have thus failed to include the important issue of energy use during data transfer. They suggest that the transport of data to and from datacenters, particularly since public cloud center might be a continent away, uses quite a bit more energy overall than simply storing data locally.

Admittedly, including the energy cost of data transport is a reasonable way to consider energy efficient cloud computing. But it is depends on the application scenario. If the application is compute-intensive, so the power consumption by data transport could be much less than the energy cost by data computing in the data center.

In addition, considering processors, disks, networks and cooling systems are the main power consumer in data center, and have an effect on each other. Thus, in order to maximize energy efficiency, it is necessary to consider the interaction among these power consumers when energy reduction methods are applied. Furthermore, processors' policy can determine the power utility by other power consumer. In 2009, Zikos and Karatza focused on energy consumed by processors [26]. They examined three local resource allocation policies that are based on shortest queue, in a cluster with heterogeneous servers. The simulation results illustrated that the differences among the policies depend on system load and there is a trade-off between performance and energy consumption.

3. Power management and Three-State Disk Model

In this section, on one hand, we studied the Fixed Threshold (FT) techniques since we use the FT as our DPM algorithm. On the other hand, we introduced the three states of disk, for conventional disk-based storage system.

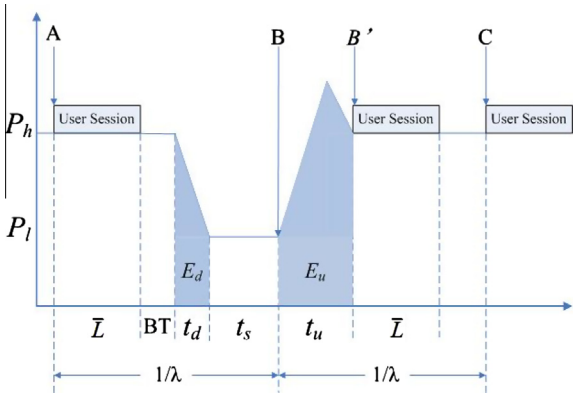

Fig. 1. FT algorithm.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

3.1. 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 $\overline{L}$. 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.

As shown in Fig. 1, A, B and C are three adjacent user sessions. Firstly, the disk operates in high power mode for user session A, after the time duration of $\overline{L}$, 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.

3.2. Three-State Disk Model

As is well-known, conventional disk has three modes: standby, idle, active. In this paper, as we mentioned before, we did not distinguish the idle mode and active mode. However, we must consider the QoS when the storage system provides some kinds of service. We know that some disks in the data center will be exhausted because of overload.

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 storage system.

The new disk three states are: overload, normal, and standby, which is set by the real-time disk load. Fig. 2 depicts a view of the disk state transition among these three states.

We specify that the state of disk never jump from standby to overload or vice verse. 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. And vice verse, 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. We express this direction by the blue arrow line in Fig. 2. It does offer a useful means for structuring our research and discussion.

The key for maximizing energy saving is to keep all standby disks staying in the sleeping mode as long as possible and meanwhile keep as small overload disks as possible.

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

4. Prediction algorithm

In this section, we develop a new prediction algorithm for data block replacement.

Our precious paper [20] did not study prediction algorithm for data migration. However, we found that the block weight plays an important role in data migration, which would affect the disks grouping directly. Thus, we believe the prediction has a vital role, which can conserve more energy. Another paper [21] employed a sliding window replacement algorithm for data

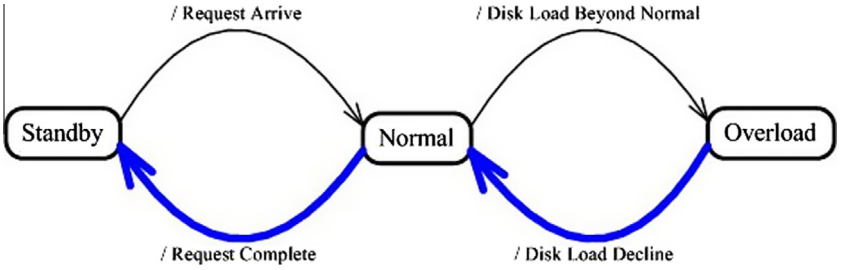

Fig. 2. Disk state transition.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

block weight computing. Both the disk state transition and the data migration need to sort the block according to the block weight. And we discovered if the size of sliding window and the history access impact factor were set as suitable values, we could achieved several percent points more energy conservation.

Getting more and more energy conservation is a big motivation for us. And it is conceivable that a good prediction algorithm could improve more energy efficiency.

In the following parts, we proposed a new prediction algorithm with priori information to improve the accuracy of the prediction, and the ultimate target is to enhance the energy saving efficiency. At the same time, the prediction algorithm can also work even without priori information.

4.1. Priori information and sliding window based prediction (PISWP) Algorithm

Caches use the temporal and special localities to improve the performance. This concept and different placement and replacement policies have been widely researched and taught in classrooms. The focus of our research is on better prediction algorithms making use of the social computing and Web 2.0 platforms, which tend to interact much more with the end user and make the end-user an integral part of our replacement policies. As such, Web 2.0 sites provide users with information storage, creation, and dissemination capabilities that were not possible in the environment now known as "Web 1.0". In this paper, we will not focus on how to get the user's information and to get what kind of information. We assume the priori information we got is valid and which will work for our experiments.

In this paper, we considered this kind of scenario: a coming popular content is uploaded to website, and there will be many users to access this new content. This kind of content maybe a new movie uploaded by a Youtube user, perhaps a new tool released by the developers from Microsoft, and also probably some information offered by the recommend system corresponding with user's interesting.

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.

We regard the user request as a time series:

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

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

Furthermore, we considered the priori information that a coming popular content will be posted online for a period. And in this period, these data blocks of this web content will be quite popular.

The access number (AN) with PISWP is calculated by the following equation:

$$AN_i = AN_{i,j} * \theta_{PI}^{\eta_{switch}|T_{end}-T_{cur}|} + \delta_{History} * AN_{i,j-1} \quad (1)$$

$AN_{i,j-1}$ is the access number of Block i in the last sliding window period. $\delta_{History}$ sets 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. θ_{PI} is the priori information impact factor. η_{switch} is an on–off switch to control the equation state for different applications. T_{end} means when the content will be withdrawn to offline, and T_{cur} represents the system current time.

In the next part, we will discuss how this equation can work for different scenarios.

4.2. PISWP works for different scenarios

First, we will discuss the scenario which has priori information such as new movie on line, new tools released and interesting web content recommended.

On this scenario, η_{switch} should be set to 1. T_{end} is set the value of time of withdrawing to offline. The unit of the value of $|T_{end} - T_{cur}|$ depends on the application. If the new content will stay online for a long time, it could set day to the unit, or if that just stay for several days, it also could set hour as the unit. We can normalize it based on how often we need to give a prediction value. The value of θ_{PI} must be larger than 1, we set it as 2 in our experiments. Before T_{end}, the data block weight of the pop content has a larger number scale than the nonpop data block. After T_{end}, θ_{PI} will set to 1 and η_{switch} will set to 0, and the data block weight will go back to the same number scale with other data blocks.

The block weight will accumulate in one sliding window, but after the system time crossing into another sliding window, the block weight will set to 0 and start to accumulate again. So, $AN_{i,j-1}$ is an important auxiliary element to block weight.

If the duration of $|T_{end} - T_{cur}|$ stays in one sliding window, it is very easy to block weight calculating, But sometimes, the duration of $|T_{end} - T_{cur}|$ will cross at least two sliding windows, actually, it is very easy too. Just like we mentioned above, the block weight will start to accumulate from 0 again for a new sliding window, if the content is still online, then θ_{PI} will go onto affect the block weight until T_{end}. And of course, if the duration of $|T_{end} - T_{cur}|$ is equal to the size of the sliding window, and then this algorithm will affect the block weight just in the sliding window.

Second, on the scenario without priori information, θ_{PI} should be 1 and η_{switch} should be 0, and Eq. (1) will become the original SWR that we used in [21]. Obviously, on this scenario, if the length of sliding window is the whole service duration, the frequency is the same with LFU.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

Something must be explained here, PISWP includes SWR and SWR includes LFU, but SWR and LFU never include PISWP. Just thinking this kind of situation: the new content is just post online or withdrawn offline. The PISWP could immediately increase or decrease the data block weight, but SWR is going to reflect against this condition until at least current sliding window, and LFU even need more time duration to increase or decrease the data block weight.

Another difference between PISWP and LFU is that PISWP could abandon the afterheat of data block when the online period is finished. Nevertheless, in LFU, even though the online period is end, the weight of the data block which is offline still has a high afterheat. So, LFU need a long time to reflect the situation that the online content has been offline.

Therefore, our new prediction algorithm not only works for calculating the block weight with priori information, but also works for counting the normal block weight without priori information.

5. Disk grouping scheme

The PISWR determines the way of data block distribution, and data block distribution directly impact on disk grouping. So, disk grouping is indirectly determined by PISWR. In other words, if the prediction algorithm has a bad result, the disk grouping will yield poor energy efficiency.

In work [20], a disk energy conservation model is proposed, 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 3SDM [21]. In this article we still applied 3SDM but we modified it. In the following parts we will provide the detail of our disk grouping scheme–3SDM.

5.1. Employment of 3SDM

To keep the discussion simple, we assume that 3SDM dynamically groups 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 they still cannot satisfy majority of user requests very well. The disks in NDG also work at full speed with normal load, and 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.

The 3SDM dynamically divides disks into ODG, NDG and SDG according to system load, and then performs data exchanging among disks in real time. The target of 3SDM is to maximize energy saving while insuring 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 order to improve the efficiency for energy conservation, the real-time data exchanging should have the following features: (i) do not force any disk in SDG to wake up only for data exchanging. (ii) Choose the "hottest" data blocks in disks which will spin down and become standby disk at once for data emigration. (iii) Select the "hottest" data blocks in ODG and the "coldest" data blocks in normal disks for data exchange to maintain the system load balance. (iv) The data blocks in NDG are extreme distribution, the hotter blocks move into the hotter normal disks and the colder data blocks move into the colder normal disks. (v) The data exchanging will happen in real time to adapt to the constant changing of user interest.

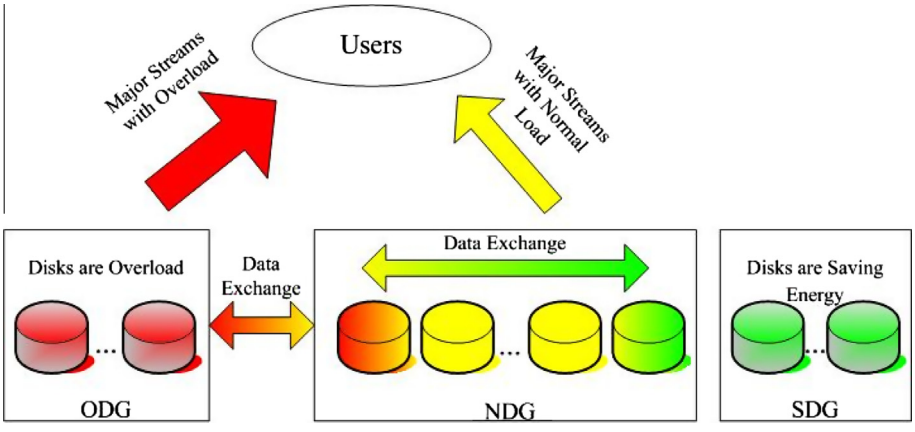

Fig. 3. The 3SDM model.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

(vi) The amount of exchanged data will be as small as possible in order to save the scarce disk access bandwidth in storage systems.

We want to mention that we used almost the same method of data transfer that was employed in [20]. In our consideration, the cost of data transfer has various aspects. The most important two cost aspects are bandwidth and energy. For the bandwidth, we do not force any disk in SDG to wake up only for data exchanging. And the amount of exchanged data will be as small as possible in order to save the scarce disk access bandwidth in storage systems. Thus, the data exchange never occupies the bandwidth excessively. For the energy aspect, there will be energy consumption as long as the disk is spinning, no matter whether the data transfers or not.

In conclusion, the energy saving is the target, the load balance is the performance guarantee, and the data extreme distribution is the scheme. Therefore, our aim could be represented by the following equation:

$$maxT(SDG), \quad minT(ODG), \quad s.tBR_{ODG+NDG} \geqslant BR_{SyetemLoad} \tag{2}$$

$BR_{ODG+NDG}$ means the bandwidth ratio of active disks, $BR_{SyetemLoad}$ represents the bandwidth ration of system load. $T(SDG)$ is the time duration when the disks stay in standby state, and in the same way, $T(ODG)$ means the time duration when the disks stay in overload state.

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

$$DFLT = \frac{1000\ \text{ms}}{DST + DRT + \left(\frac{VB}{DITR}\right)} * VB \tag{3}$$

Table 1 describes these parameters in Eq. (3). DST is 3.4 ms, DRT is 2.0 ms, VB is 40 KB/s and $DITR$ is 60 MB/s. So, the $DFLT$ is 6.6 MB/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.

5.2. Dynamic disk state transition

5.2.1. The algorithm of dynamic disk state transition

Step 1. Examine how many disks are active, and set the number N_{active}. We can achieve this aim by observing the state of the disk.

Step 2. Setting the disks whose load is beyond the full-load threshold as the overload disk, and then setting the other active disks as normal disks. Thus, we can obtain the number of normal disks, which we use N_{normal} to represent.

Step 3. It is assumed that the normal disks are going to be evenly divided into two groups. We have N_{normal} from the last step. Hence, using Eq. (4), we are able to obtain the colder disk number. After that, we have N_{normal} and N_{colder}, Thus, using Eq. (5), we are going to have the hotter normal disk number.

$$N_{\text{colder}} = \left\lfloor \frac{N_{\text{normal}}}{2} \right\rfloor + N_{\text{normal}} \bmod 2 \tag{4}$$

$$N_{\text{hotter}} = N_{\text{normal}} - N_{\text{colder}} \tag{5}$$

5.2.2. 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).

Table 1
IBM Ultrastar 36Z15 disk parameters.

Parameters	Values
Standard interface	SCSI
Disk rotation speed	15,000 RPM
High power (read/write/seek)	13.5 W
Low power (standby)	2.5 W
Spinup time	10.9 s
Spinup energy	135 J
Spindown time	1.5 s
Spindown energy	13 J
Break-even time (BT)	15.2 s
Disk average seek time	3.4 ms
Disk average rotate time	2.0 ms
Disk inner transfer rate	60 MS/s
Video bitrate	40 KB/s

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

Step 2. Calculating Middle Block Weight (MBW) by (6) and (7), in which len (BWL) stands for the length of the BWL. Therefore, MBW is the approximate minimum value of block weight in active disks.

$$k = \left\lfloor \frac{N_{\text{active}}}{N_{\text{total}}} * \text{len}(\text{BWL}) \right\rfloor \tag{6}$$

$$\text{MBW} = \text{Block Weight of BWL}(k) \tag{7}$$

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. It sorts all disks based on system load in descending order, known as the Disk Weight List (DWL). LoadBlance swaps blocks between overload disks and the DWL(m), in which m is set by Eq. (8). The hottest blocks in these overload disks will be swapped with the coldest blocks in DWL(m).

$$m = \frac{N_{\text{active}} - N_{\text{colder}}}{N_{\text{active}}} \tag{8}$$

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

5.2.3. 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 following equations:

$$\text{SL} = \frac{\sum_{i=0}^{N-1} (AC_i * Size_i)}{\text{CLP}} \tag{9}$$

$$\text{TEC} = \frac{\text{SL}}{\text{DFLT}} * serviceTime * Power_{activeDisk} \tag{10}$$

The system load (SL) is calculated as (9), 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. In our system, we set the value of CLP with 60,000 ms. Obviously, the value of CLP can be changed to other parameter, 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 Eq. (11), in which δ_{Cache} is smaller than 1.

$$\text{LUBEC} = \delta_{\text{Cache}} * \text{TEC} \tag{11}$$

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

$$\text{GLBES} = \frac{\text{energyConsume-LUBEC}}{\text{energyConsume}} \tag{12}$$

6. Evaluation methodology

In this section, we will introduce the test bed. And then we first discuss the result of two normal traces, after that we will present several more traces with priori information, and we could see how much energy efficiency improved.

6.1. Test bed

We enhanced the widely used DiskSim simulator [22] 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 [23]. The key parameters are shown in Table 1.

In this section, we choose FT, PDC [9], PISWP 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.

6.2. Real-system trace

Our experiments use two real-system traces for the purpose of testing the effects of various algorithms in this environment. We assume the streaming media content stores in the cloud, and users watch these video by accessing the server in

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

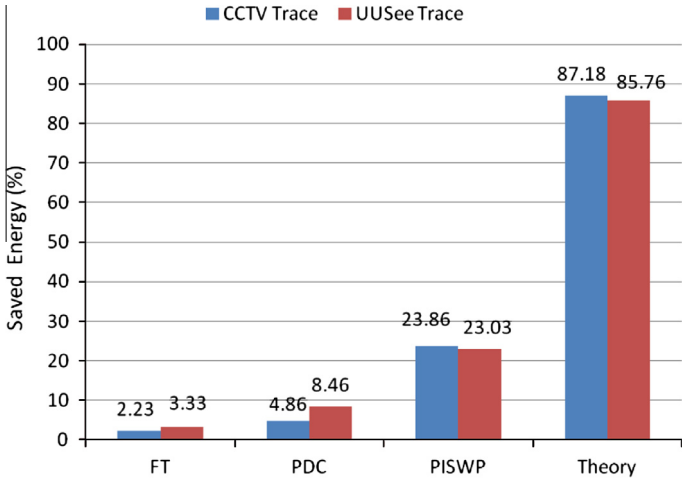

Fig. 4. Saved energy with different saving algorithms.

cloud. We want to mention that video server is a subset of cloud system. A cloud system consists of many components and subsystems, and the video server is one of them. So, our method and experiments still related to cloud system.

First, a 24 h' 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 s. 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 h' real-system trace is obtained from UUSee systems [24]. The user arrival rate is 0.0591 per second, and the mean session length is 1094.48 s.

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.

6.3. 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 PISWP online algorithm in 24 h 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 PISWP 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 does not need to consider disk state transition and data migration. Because the traces have not any priori information, thus now PISWP without priori information is just like SWR.

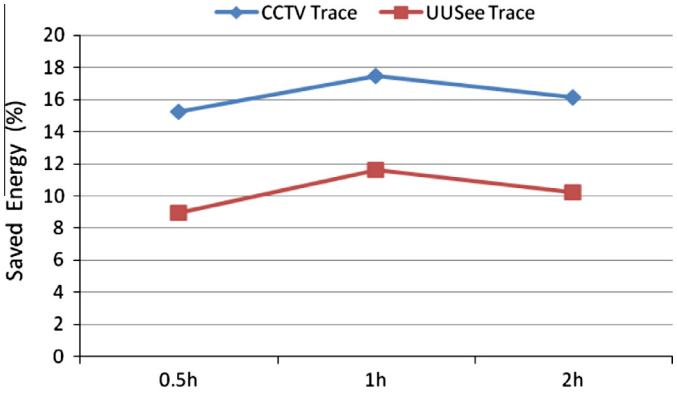

Fig. 5. Different size of sliding window.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

Table 2
Different history access impact factor.

$\delta_{History}$	1	1.1	1.2	1.3	1.4	1.5
CCTV trace						
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						
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

6.4. 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 h and every hour the user's interest is relatively stable.

6.5. Different history access impact factor

In these two traces, there is no priori information, so PISWP is just like SWR [21].

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

Table 2 and Figs. 6 and 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 2. When $\delta_{History}$ equals to 1, SWR is just like the LFU. With the increasing value of $\delta_{History}$ the saved energy of UUSee

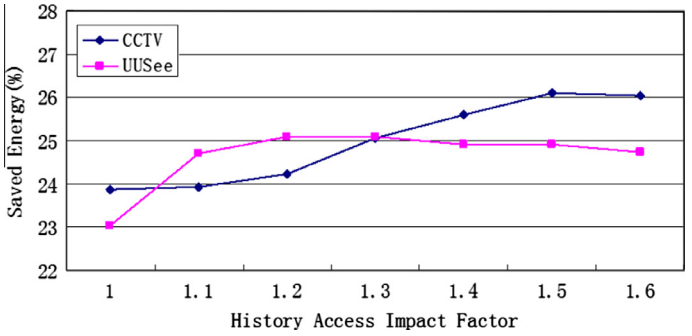

Fig. 6. Energy saving of different history access impact factor.

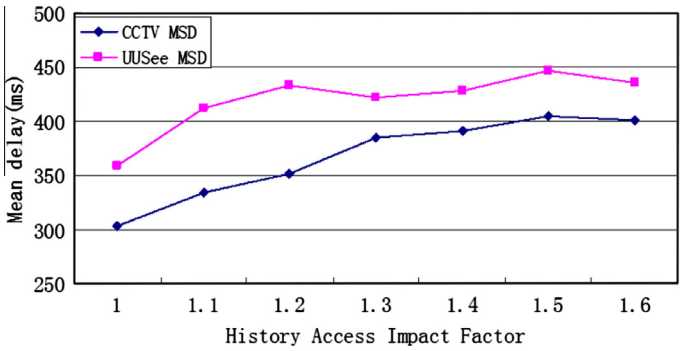

Fig. 7. QoS of different history access impact factor.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

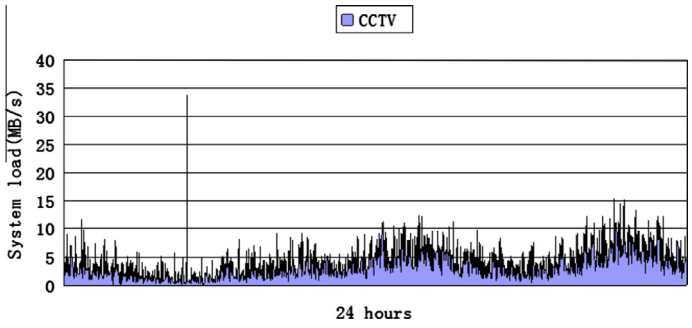

Fig. 8. CCTV 24 h system load.

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.

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.

6.6. System load analysis

Figs. 8 and 10 present the system load of CCTV and UUSee trace, Figs. 9 and 11 demonstrate the corresponding 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 Figs. 8 and 10, we find that the load exceeds 60 MB/s sometimes, however, our system full load is $5 * 10 = 50$ MB/s, implying that only 10 disks can provide service.

With the effect of PISWP, 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.

6.7. 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 Figs. 12 and 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 PISWP migrates the most popular data blocks into the hot active disks which can serve the request without overload.

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 will not be migrated into disk 9.

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 depends on the load from the request and the data block distribution on the storage system. If the requested data blocks distribute in the same disk only, 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,

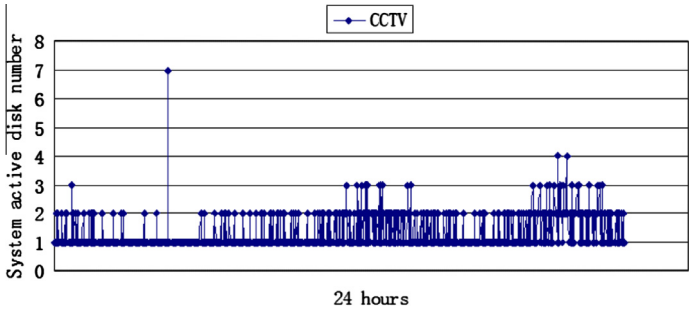

Fig. 9. CCTV 24 h system active disk number.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

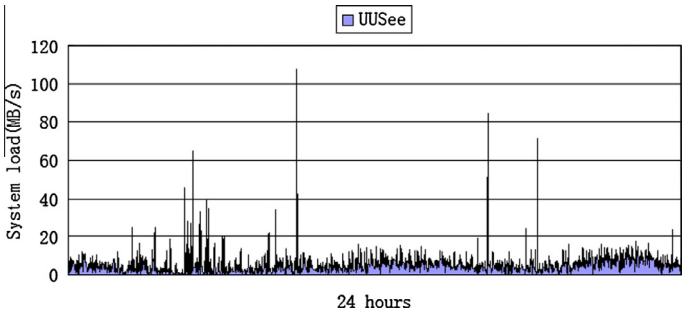

Fig. 10. UUSee 24 h system load.

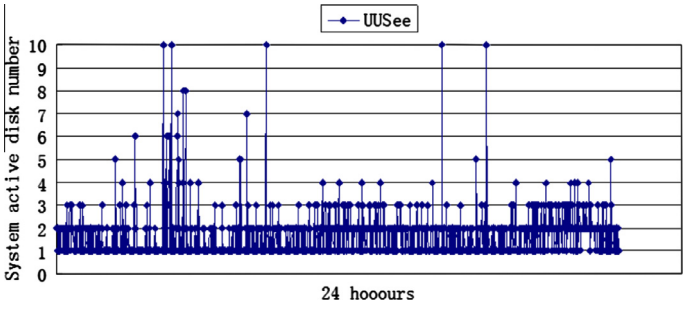

Fig. 11. UUSee 24 h system active disk number.

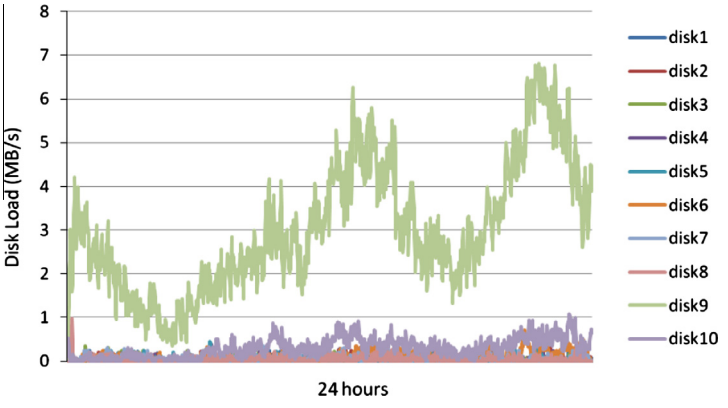

Fig. 12. Each disk load situation with CCTV 24 h trace.

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.

6.8. Priori information works for CCTV and UUSee

First, we generated a 60 min' 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 20.62 ms and the mean delay jitter is 21.71 ms. The energy saving effect dramatically approach to the theoretical algorithm result since there is only one disk in the active mode to provide service.

Inspired by this spectacular energy saving result, we conceive 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.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

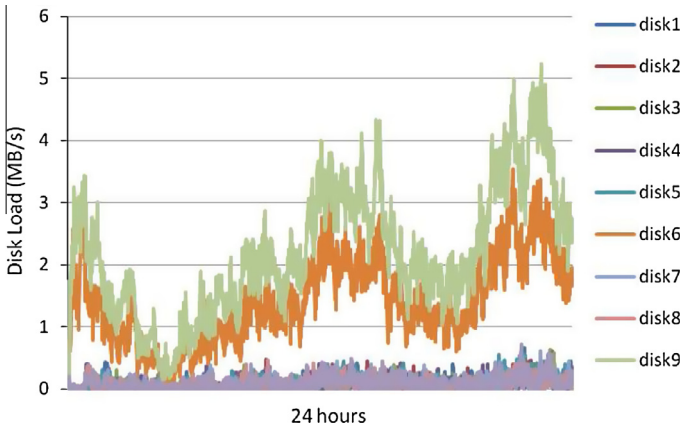

Fig. 13. Each disk load situation with UUSee 24 h trace.

Table 3
Priori information works for CCTV and UUSee.

	CCTV	UUSee
Saved energy (%)	31.256	64.529
Mean startup delay (ms)	384.876	289.396

Thus, we modified CCTV trace and UUSee trace. A middle user session (169 s) video was plugged into CCTV traces, and two long user sessions (1094 s and 1600 s) videos were inserted into UUSee trace. All of them were online for 24 h as a new video on web. And the result of experiment as Table 3.

UUSee with priori information got better energy efficiency than CCTV. Several reasons could take responsibility for this result: UUSee has two new videos online, and these two videos owned long user's session; the access of the new video in CCTV is sparse over the whole trace, but the access of the new video in UUSee is quite concentrated.

Fig. 14 shows the saved energy percent under different saving model with the traces that have priori information. It is clear that PISWP has better energy efficiency than PDC and FT. Table 4 reports the QoS under different saving model with priori information. For CCTV, PISWP has the worst QoS but which is still acceptable. However, PISWP's QoS is much better than PDC's.

In short, there are two requirements supporting PISWP works. First, the new content will be accessed. Second, the access must be concentrated. If the access from user does not meet these two requirements, it means PISWP failed. However, as a warrant, just like any plane has more than engines, PISWP can switch to be SWR, which can even provide an acceptable energy efficiency and QoS.

In these two traces, we set $\delta_{History}$ to 1, so that SWR is just like LFU, and we do not distinguish the difference between PISWP and LFU, because the new video has 24 h online duration.

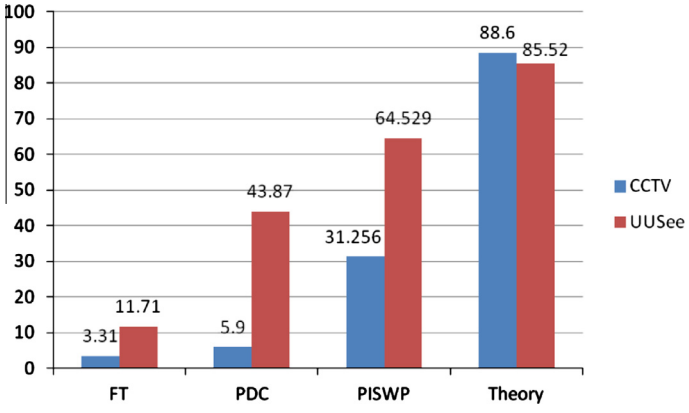

Fig. 14. Different saving algorithms work with CCTV and UUSee PI traces.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

Table 4
Mean startup delay.

	FT	PDC	PISWP
CCTV (ms)	204.59	305.47	384.87
UUSee (ms)	161.04	1289.274	289.396

Table 5
IPSWP works on other traces.

	webFileServer	toolReleaseServer
Saved energy (%)	42.76	24.37
Mean startup delay (ms)	165.52	363.18
Theory saved energy (%)	89.86	87.01

6.9. Other traces

In this part, we employ two other traces: toolReleaseServer and webFileServer. We assume both of the two servers are deployed in the cloud, and users want to read the news and download the new tools must access the server in cloud.

The trace of toolReleaseServer is imitated by the Mirosoft tool release server trace, and we analogy the idea and manufactured the trace and keep the trace works in our testbed. The webFileServer trace is inspired by one of the trace from [25], Web/SQL Server. We also built it and make it works in our testbed.

For webFileServer, we consider the application scenario is: one piece of news is just uploaded to a portal. The size of the web file is small, but the access from Internet is concentrated. The user's session will be short for many people probably make it at a quick glance.

The toolReleaseServer trace represents another kind of scenarios. We assume a new tool released by the developer. It will be downloaded by the user who is interested in. Thus, the concentrated degree of this trace will be lower than the news. However, users will access and download it. As the result, the user's session will be much longer than skimming news. In addition, the size of the tool should also be bigger than a webpage.

Both of them are of 24 h' trace and the result of experiment as follows:

From Table 5, we can observe that the webFileServer trace conserves much more energy than the toolReleaseServe and offers a better QoS as well.

Of course, webFileServer should have a better QoS, because it needs short HTTP access. However, why does it also have better energy efficiency? The reason is analyzed last part: The webFileServer in providing news can absorb much more concentrated accesses.

Again, in these two traces, we set $\delta_{History}$ to 1, so that SWR is just like LFU, and we also did not distinguish the difference between PISWP and LFU because we assume the new tool and news have 24 h online duration. So, in order to discuss the dispose of afterheat between PISWP and LFU, we make another two traces, toolReleaseServer and webFileServer, and both of them are 2 h long. And both of them have new content online from 100 s to 600 s. In other words, these new content will be online for 10 min.

Fig. 15 shows the saved energy percent under different saving model with the traces that have priori information. We could find that PISWP can save more energy than LFU. That is because PISWP could quickly dispose the afterheat of the data

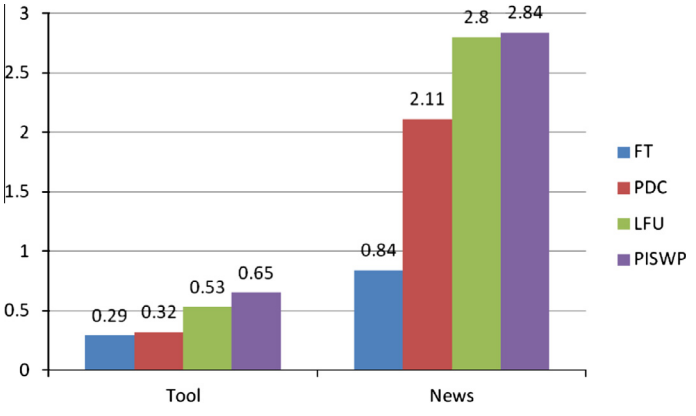

Fig. 15. Different saving algorithms work with tool releasing and news PI traces.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

Table 6
Mean startup delay.

	FT	PDC	LFU	PISWP
News (ms)	38.81	73.27	63.29	69.73
Tool (ms)	43.35	115.03	127.45	111.03

Table 7
Hit rate of traces.

Traces	CCTV	UUSee	News	Tool
Hit rate (%)	87.57	99.98	82.77	73.39
Energy saving (%)	31.25	64.52	2.84	0.65
Mean startup delay (ms)	384.87	289.396	69.73	111.03

block which have been offline, but LFU cannot. So, it is obvious that PISWP is different with LFU and much more energy efficiency than LFU. However, the saved energy percent is not very high, and the reason is the duration of trace is only 2 h. It can save much power if the trace is long enough.

Table 6 represents that PISWP also has a high QoS, for releasing tool, it is better than PDC and LFU, and for news, it is better than PDC and a little worse than LFU.

In a nutshell, if the server has a "concentrated" access in the closed future, the PISWP will give you a surprise to energy efficiency and has a good QoS.

6.10. Hit rate

In this part, we present the hit rate after PISWP is employed. We have the experiment with the traces of CCTV, UUSee, News and Tool under PISWP, and their data block access hit rates are in Table 7. We want to explicate that the hit rate is for the whole duration of each trace. Because we cannot assume when the user will access the data blocks that we already put into the hot disk, perhaps in the next slide window, or the slide window after the next one. Hence, it does not have much meaning whether we consider the hit rate of each slide window.

From this table, we find the energy efficiency is consistent with the hit rate. In other words, the higher the hit rate is, the better the energy efficiency will be. It is correct because the users behavior often exist locality. They always access the data blocks that stay in the hot disk, and thus it can reduce data transmit and disk state switch so that the system get the high energy efficiency.

For the stream media system, we assume the deviation threshold of hit rate is 10%. Thus, the deviation of the hit rate of CCTV trace is below 10%, while that of UUSee trace is above 10%. Although the priori information of CCTV trace did not bring a higher hit rate, it still had a fair energy efficient and a low mean startup delay.

On the other hand, when we consider the other two traces, we assume the deviation threshold of hit rate is 20%. Hence, the deviation of the hit rate of webFileServer is above 20%, while that of toolReleaseServer is below 20%. It is actually that the priori information about toolReleaseServer only yield the hit rate 73.39% and even the small energy saving 0.65%, but it did not bring a bad QoS, because the mean startup delay is 111.03 ms, which is well acceptable.

7. Conclusion and future work

In this paper, a new prediction algorithm, PISWP is proposed for energy-efficiency storage system in cloud. The algorithm is based on priori information and sliding window extracted from social networks and Web 2.0 sites. We reported the disk grouping scheme, a 3SDM model, which has three disk states according to the system dynamic load. The disk states are overload, normal, and standby (sleep). The design target of 3SDM is to push the overload disks to change its state to normal for enhancing the QoS. For the disks in normal state, the 3SDM algorithm explores the way to transit them into standby mode to further save energy.

PISWP predication is the key of the 3SDM algorithm. The prediction of PISWP will result in the disk state or disk grouping. If the PISWP offered an exact forecast, 3SDM could allow much more disk to stay in sleep state, and create better energy efficiency.

The accuracy of PISWP depends on the applications too. We target more energy saving in typical usages based on the input from usage patterns obtained from the Web 2.0 sites.

In the case of no priori information available, PISWP will work in the same way as 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 can still save more energy and provide better service.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002

In addition, the accuracy of priori information is another significant issue. If the priori information is not so accurate, we may not observe an obvious improvement of energy efficiency as the experiment of toolReleaseServer. On the other hand, the priori information is not a major factor, even when it is not accurate. The inferior priori information can lower the improvement of energy efficiency and quality of service. However, the impact is limited, and the result is still acceptable if the prior information is not accurate.

To further improve the energy efficiency, we are exploiting other applications just like news on webFileServer that can also trigger the PISWP's exiting energy efficiency function. Further, we will consider how to ensure the accuracy of priori information in different applications, even though it may need knowledge of other domain knowledge.

Acknowledgements

This research is supported in part by the National Natural Science Foundation of China (Nos. 61272087, 61202115, 61073008, 60773148 and 60503039) and Beijing Natural Science Foundation (Nos. 4082016 and 4122039).

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, Above the clouds: a Berkeley view of cloud computing, Communications of the ACM 53 (4) (2010).
[2] Yinong Chen, W.T. Tsai, Service-Oriented Computing and Web Software Integration, third ed., Kendall Hunt Publishing, 2011.
[3] B. Moore, Taking the data center power and cooling challenge, Energy User News, 2002.
[4] C. Weddle, M. Oldham, J. Qian, A.A. Wang, P. Reiher, G. Kuenning, PARAID: a gear-shifting power-aware RAID, ACM Transactions on Storage 3 (3) (2007) 245–260.
[5] Q. Zhu, Z. Chen, L. Tan, Y. Zhou, K. Keeton, J. Wilkes, Hibernator: helping disk arrays sleep through the winter, ACM Symp. Operating Systems Principles (2005) 177–190.
[6] Mark Aggar, The IT Energy Efficiency Imperative, A White Paper for Decision Makers on the Urgency and Benefits of Embracing IT Energy Efficiency Principles and Practices, Microsoft Corporation, June 2011.
[7] Tim O'Reilly (2005-09-30). What Is Web 2.0. O'Reilly Network (Retrieved 06.08.06).
[8] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir, H. Franke, DRPM: dynamic speed control for power management in server class disks, in: Proc. 30th Annual Intl Symp. Computer Architecture, June 2003, pp. 169–179.
[9] E. Pinheiro, R. Bianchini, Energy conservation techniques for disk array-based servers, in: Proc. 18th Intl Conf. Supercomputing, June 2004, pp. 68–78.
[10] E.V. Carrera, E. Pinheiro, R. Bianchini, Conserving disk energy in network servers, in: Proc. 17th Intl Conf. Supercomputing, June 2003, pp. 86–97.
[11] S. Gurumurthi, A. Sivasubramaniam, M. Kandemir and H. Franke, DRPM: dynamic speed control for power management in server class disks, in: Proc. 30th Annual Intl Symp. Computer Architecture, June 2003, pp. 169–179.
[12] Q. Zhu, Y. Zhou, Power-aware storage cache management, IEEE Transactions on Computers 54 (5) (2005) 587–602.
[13] Yuanyuan Zhou, James Philbin, The multi-queue replacement algorithm for second level buffer caches, in: Proceeding of the General Track: 2002 USENIX Annual Technical Conference, 2001, pp. 91–104.
[14] Hong-Tai Chou, David J. Dewitt, An evaluation of buffer management strategies for relational database systems, in: Proceeding of the 11th international conference on Very Large Data Bases-Volume, August 1985, pp. 127–141.
[15] Shaul Dar, Michael J. Franklin, Bjrn r Jnsson, Divesh Srivastava, Michael Tan, Semantic data caching and replacement, in: Proceeding of the 22th International Conf. on Very Large Data Bases, September 1996, pp. 330–341.
[16] Elizabeth J. ONeil, Patrick E. ONeil, Gerhard Weikum, The LRU-*K* page replacement algorithm for database disk buffering, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (1993) 297–306.
[17] 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 (1994) 439–450.
[18] Andreas Berl, Erol Gelenbe, Marco Di Girolamo, Giovanni Giuliani, Hermann De Meer, Minh Quan Dang, Kostas Pentikousis, Energy-efficient cloud computing, The Computer Journal 53 (7) (2010) 1045–1051.
[19] J. Baliga, R.W.A. Ayre, K. Hinton, R.S. Tucker, Green cloud computing: balancing energy in processing, storage, and transport, Proceedings of the IEEE 99 (1) (2011) 149–167.
[20] Y. Chai, Zhihui Du, D. Bader, X. Qin, Efficient data migration to conserve energy in streaming media storage systems, IEEE Transactions on Parallel and Distributed Systems 23 (11) (2012) 2081–2093.
[21] Zhihui Du, Wenjun Fan, Wenpeng Chai, Three-State Disk Model for High Quality and Energy Efficient Streaming Media Servers, in: The 11th International Symposium on Autonomous Decentralized System, 2013.
[22] G. Ganger, B. Worthington, Y. Patt, The DiskSim Simulation Environment (v4.0), September 2009. <http://www.pdl.cmu.edu/DiskSim>.
[23] Hitachi, Ultrastar 36Z15 Datasheet, March 2011. <http://www1.hitachigst.com/hdd/ultra/ul36z15.htm>.
[24] X. Xiao, Y. Shi, Q. Zhang, J. Shen, Y. Gao, Toward systematical data scheduling for layered streaming in peer-to-peer networks: can we go farther?, IEEE Transactions on Parallel and Distributed Systems 21 (5) (2010) 685–697
[25] Dushyanth Narayanan, Austin Donnelly, Write off-loading: practical power management for enterprise storage, in: 6th USENIX Conference on File and Storage Technologies, 2008, pp.253–267.
[26] S. Zikos, H. Karatza, Performance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times, Simulation Modelling Practice and Theory 19 (1) (2011) 239250.
[27] X.-J. Ruan, S. Yin, A. Manzanares, J. Xie, Z.-Y. Ding, J. Majors, X. Qin, ECOS: an energy-efficient cluster storage system, in: Proc. 28th Int'l Performance Computing and Communications Conf. (IPCCC), Phoenix, Arizona, December 2009.
[28] A. Manzanares, X. Qin, X.-J. Ruan, S. Yin, PRE-BUD: prefetching for energy-efficient parallel I/O systems with buffer disks, ACM Transactions on Storage 7 (1) (2011) (Article 3).
[29] D. Colarelli, D. Grunwald, Massive Arrays of Idle Disks for Storage Archives, Proc. ACM/IEEE Conf. Supercomputing, 2002, pp. 1-11.

Please cite this article in press as: Z. Du et al., Priori information and sliding window based prediction algorithm for energy-efficient storage systems in cloud, Simulat. Modell. Pract. Theory (2013), http://dx.doi.org/10.1016/j.simpat.2013.06.002