Marquette University e-Publications@Marquette

Mathematics, Statistics and Computer Science Faculty Research and Publications Mathematics, Statistics and Computer Science, Department of

12-6-2013

Using Intelligent Prefetching to Reduce the Energy Consumption of a Large-scale Storage System

Brian Romoser Texas State University - San Marcos

Ziliang Zong Texas State University - San Marcos

Ribel Fares Texas State University - San Marcos

Joal Wood Texas State University - San Marcos

Rong Ge Marquette University, rong.ge@marquette.edu

Accepted version. Published as part of the proceedings of the conference, 2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC), 2013. DOI. © 2013 Institute of Electrical and Electronic Engineers (IEEE). Used with permission.

Marquette University

e-Publications@Marquette

Department of Mathematics, Statistics, and Computer Science Faculty Research and Publications/College of Arts and Sciences

This paper is NOT THE PUBLISHED VERSION; but the author's final, peer-reviewed manuscript. The published version may be accessed by following the link in the citation below.

2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC), Vol. xx, No. x (2013): 1-9. DOI. This article is © Institute of Electrical and Electronic Engineers (IEEE) and permission has been granted for this version to appear in <u>e-Publications@Marquette</u>. Institute of Electrical and Electronic Engineers (IEEE) does not grant permission for this article to be further copied/distributed or hosted elsewhere without the express permission from Institute of Electrical and Electronic Engineers (IEEE).

Using intelligent prefetching to reduce the energy consumption of a large-scale storage system

Rong Ge Department of Mathematics, Statistics, and Computer Science, Marquette University, Milwaukee, WI Brian Romoser Computer Science Department, Texas State University, San Marcos, TX Ziliang Zong Computer Science Department, Texas State University, San Marcos, TX Ribel Fares Computer Science Department, Texas State University, San Marcos, TX Joal Wood Computer Science Department, Texas State University, San Marcos, TX

ABSTRACT:

Many high performance large-scale storage systems will experience significant workload increases as their user base and content availability grow over time. The U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) center hosts one such system that has recently undergone a period of rapid growth as its user population grew nearly 400% in just about three years. When administrators of these massive storage systems face the challenge of meeting the demands of an ever increasing number of requests, the easiest solution is to integrate more advanced hardware to existing systems. However, additional investment in

hardware may significantly increase the system cost as well as daily power consumption. In this paper, we present evidence that well-selected software level optimization is capable of achieving comparable levels of performance without the cost and power consumption overhead caused by physically expanding the system. Specifically, we develop intelligent prefetching algorithms that are suitable for the unique workloads and user behaviors of the world's largest satellite images distribution system managed by USGS EROS. Our experimental results, derived from real-world traces with over five million requests sent by users around the globe, show that the EROS hybrid storage system could maintain the same performance with over 30% of energy savings by utilizing our proposed prefetching algorithms, compared to the alternative solution of doubling the size of the current FTP server farm.

SECTION I. INTRODUCTION

Massive scale storage systems often face difficulties on two fronts as time passes. First, they must grapple with the problem of data expansion as new, more current information is added to a repository alongside many older but still valuable pieces of information. Second, as the system and its provided services gain more attention and popularity, user population will increase constantly over time. This will inevitably demand higher performance for existing systems. Administrators of massive storage systems cannot attempt to solve one issue but ignore the other, as both are integral components in allowing unhindered growth.

The USGS EROS is the United States' official repository of civilian terrestrial imagery and it hosts the world's largest satellite images distribution system containing many petabytes of satellite data. Every day, high resolution images of the Earth are taken by the orbiting Landsat satellites and added to EROS' ever-expanding library of satellite data. In October 2008, EROS opened their vast amounts of archived satellite images to the public for free, which prompts a large and continuous growth of the user base of EROS system. In about three years, the number of monthly requests for images received by EROS has nearly quadrupled, as shown in Figure 1. This number is expected to grow constantly in the future. To provide high quality service for a rapidly growing number of customers, EROS needs to improve the performance of its existing system, either by purchasing more hardware or exploring possible software optimization. Expanding the storage capacity is a simple way of improving performance, but it will also increase system cost and consume more power due to the increased number of devices. A software solution, while more complex and difficult to implement, could achieve the same level of performance without purchasing additional hardware. In this paper, we demonstrate our study on utilizing intelligent prefetching techniques to significantly reduce the power consumption and cost of the world's largest satellite images distribution system without degrading performance.



Fig. 1 The monthly satellite images download requests of EROS grow by almost 400% during Oct.2008 - Feb.20ll.

Before we discuss the details of the proposed prefetching algorithms, it will be helpful to explain how the EROS satellite image distribution system works. An illustration of the EROS system is shown in Figure 2. Users interact with the EROS system through the EarthExplorer¹ and the Global Visualization Viewer² web portals where they may select one or many images for download. If the requested images are available at the FTP server, a

download link will be immediately created and provided to the user. This will appear as an instantaneous procedure to the user and is the ideal response time that should be sought after for every request submitted to EROS. If, however, an image is not present in the public FTP server, a new image processing workorder will be initiated. To complete a workorder, the raw data captured by the Landsat satellites needs to be found at magnetic tape storage first then sent through a pipeline where a series of imaging processing and optimization algorithms are applied in order to produce the download-ready version. This process is both CPU-intensive and IO-intensive, which takes approximately 20 minutes in optimal conditions. Once the final version of a missing image is generated, it will be placed in the FTP server and the user will be notified of its availability via email. As a result, a noticeable delay will be incurred if a user requested image is not currently on the FTP server. Therefore, improving EROS system performance can be accomplished by reducing the miss rate of FTP servers.



Fig. 2 An overview of the EROS satellite images distribution system.

TABLE I EROS Long Term Archival Storage System

Tier	Model	Capacity	Hardware	Bus Interface
1	Sun/Oracle F5100	100 TB	SSD	SAS/FC
2	IBM DS3400	1 PB	HDD	SATA
3	Sun/Oracle T10K	10 PB	Tape	Infiniband

Table I outlines the tiered EROS Long Term Archival Storage System as specified in a recent EROS report.³ Since the majority of requests will be read operations, using SSDs for FTP servers can significantly reduce the data transfer time once the satellite images are ready. However, the vast volume of EROS satellite data and the high cost of SSDs also limit the size of the FTP servers. To trade-off performance and cost, only a small portion of satellite data should reside in the FTP and the majority of data must be stored in the low cost tapes. From this perspective, the FTP can be considered as the cache layer of the EROS hybrid massive storage system. There are two strategies to reduce the FTP miss rate (i.e. improve performance). The first strategy is to expand (e.g. doubling) the FTP server size, which is easy to implement with minimum system changes. The second strategy is to intelligently prefetch popular images to the FTP servers before the actual requests arrive, which requires a comprehensive understanding of global user behavior but can reduce cost and power consumption.

The remainder of the paper is organized as follows. Section 2 contains a brief analysis of the EROS workload characterization. Section 3 will discuss two prefetching algorithms that can be utilized to improve EROS system performance. In Section 4, we present the detailed experimental results in terms of performance and power consumption by exploring various hardware and software configurations available to the EROS system. Section 5 provides other related work and Section 6 concludes our study.

SECTION II. EROS WORKLOAD CHARACTERIZATION

In order to design intelligent prefetching algorithms that are capable of predicting user requests with a high degree of accuracy, we must first understand the critical trends exhibited by historical user requests. By conducting a series of comprehensive studies using over 5 million real world requests provided by USGS, we have identified several distinct characteristics of EROS workload,^{4,5} among which the following two patterns are most important:

• While the majority of locations on Earth have a low number of requests, few locations of the globe are requested with a high frequency in the same time period. We also found that these "hot spots" change over time. Therefore, the proposed prefetching algorithms should focus on images of these popular areas and should be able to automatically handle evolving popularity of satellite images.



Fig. 3 The three dimensions of an EROS satellite image.

• A small group of users is responsible for a disproportionately high volume of requests. In fact, the top 100 aggressive users requested almost half million images, which clearly indicates that targeting on the trends of these specific users would be of great benefit for effective prefetching.

SECTION III. PREFETCHING TECHNIQUES

Our previous work has shown that using conventional caching algorithms (LRU or LFU) alone cannot reach a very high hit ratio on the FTP servers of the EROS system.⁶ Prefetching as a method of offsetting latency caused by accessing low-speed storage components is a technique that has been well explored in several areas of computing.^{7,8,9,10,11,12,13} Unfortunately, existing prefetching algorithms cannot be applied directly to improve the EROS system performance because of its unique workload and user behavior. In this section, we propose two unique prefetching algorithms that are designed based on the EROS workload characterization results stated in Section II.

A. Prefetching based on popularity

Images cataloged by EROS are indexed using a version of the Worldwide Reference System (WRS) known as WRS-2. WRS-2 uses a grid that divides Landsat 5 and Landsat 7's orbital paths into a system of 248 latitudinal segments known as rows (North-South) and 233 longitudinal segments called paths(East-West). Each scene is targeted for imaging by the satellites once every orbit (approximately every 16 days), providing a third dimension of time, as shown in Figure 3. A unique combination of row, path, and acquisition date is referred to as an image.

As Landsat 5 and 7 orbit the Earth and take new satellite images, we must determine which new images should be automatically processed and prefetched into the FTP servers. Although proper prefetching can reduce FTP miss rate, overaggressive prefetching (e.g. prefetching all new satellite images) would not work or even further

degrade performance. It has been previously observed that few of the 17,000 available locations on Earth are "hot spots". Therefore, it is critical to only prefetch the images that are likely to be popular and requested soon.

A satellite image can become popular for a variety of reasons. For example, an image may owe its popularity to historically significant events that occurred in the past. The images that correspond to Chernobyl and Hurricane Katrina belong to this category. For such "hot spots", prefetching newly captured images will not be beneficial. Instead, keeping the old images in the FTP for a longer time would be more helpful. For other locations, users may be interested in the newest images available. For example, researches monitoring the effects of global warming may request the newest images of Greenland as soon as they become available. In this case, it would be beneficial to prefetch new satellite images of Greenland as soon as they are available. An additional consideration is that the popularity of satellite images evolves over time. For example, a group of researchers may be interested in the images of the Mount Everest now for a research project but they may lose their interests after their research is complete.

Determining significant shift in popularity can be accomplished by splitting the trace evenly in two and identifying the most popular 100 images for both the first and second half of the data. An image is popular in the first half but not the second half is claimed to have short term popularity. Conversely, if the image is popular in both the first and second half of the data, then the image has long term popularity. Our solution to the evolving popularity issue is to update the prefetching list at the beginning of every month, which ensures that the most recent popularity is reflected in our prefetching algorithm.

B. Prefetching based on users

The previous popularity based prefetching attempts to capture trends observed by the general audience. However, users are likely to behave differently in terms of areas of geographical interest. Therefore, prefetching rules generated based on one user's pattern may not be suitable for another user with a different research interest. With this in mind, we present the second prefetching algorithm generating prefetching rules that reflect a user's historical requests. The detailed user-specific prefetching algorithm is explained below.

Every unique request for an image consists of six attributes - satellite number, satellite sensor, row, path, acquisition year and acquisition day-of-year. Users can vary their download pattern in any of these six dimensions between two requests, which contributes difficulty to establishing a pattern for each user. Our solution is to represent each important attribute of a request as an integer. Then, we concatenate all six attribute integers to create a single long integer that represents a unique satellite image.

Making use of an integer representation of an image where each set of digits correspond to a different attribute allows us to capture the shift in a user's requests by simple subtraction. We can treat these movements as prefetching rules. For example, if a user requests two images taken by Landsat 7 and Landsat 5 using sensor 1 on the same location (row 30, path 40) at the same time (105th day of year 2000). These two satellite requests will be represented by the integers 7-1-030-040- 2000-105 and 5-1-030-040-2000-105, we evaluate the difference of the two and observe the following movement:

710300402000105–510300402000105=20000000000000. We can use the value 20000000000000 to represent the movement of the same location on earth, taken through a different satellite (in this case, Landast 7 and 5). We record a set of all movements made by each user as that user's personal pattern.



Fig. 4 Impact of User Population on FTP hit rate.

Even more challenging, some user patterns may not come in the expected order. For example, if a user requests a set of images with column increment by 1 as the pattern, the exact request order may show as $\{col + 1, col + 3, col + 2\}$ ' rather than $\{col + 1, col + 2, col + 3\}$. To catch such irregular user patterns, we generate rules by subtracting the numeric representation of each image from a vector of that user's history. However, this will result in an exponential growth in the number of rules generated. To solve this problem, we introduce the following three threshold values to reduce the number of derived rules.

- User Popularity (UP) Threshold UP represents the number of requests that a user has made. We have observed in Section II that aggressive users play an important role in determining the workload of EROS system. Therefore, we want to pay special attention to them. Our algorithm uses UP to filter out users who did not frequently use the EROS system. Increasing UP will require a larger number of downloads by a given user before prefetching rules is applied to that user. Figure 4demonstrates the impact of increasing UP on FTP hit rate. Although a larger UP will lead to lower hit rate, it can significantly reduce the number of prefetched images thereby saving energy.
- Window Size (WS) Threshold WS is a measure of the length of the system global request history that is examined when generating rules for prefetching. Using WS we can ensure that we are able to utilize enough history to generate intelligent prefetches while also limiting the request history to relatively recent trends. Figure 5 demonstrates how shifting window size (25 days, 50 days and 75 days) affects FTP hit rate. The larger the window size, more rules are likely to be generated, which leads to more prefetches and higher hit rate. But at the same time, more energy needs to be consumed to process the prefetched images.



Fig. 5 Impact of Window Size on FTP hit rate.



Fig. 6 Impact of Confidence Threshold on FTP hit rate.

• **Confidence Threshold** - CT indicates the estimated likelihood that a rule will be followed by a user. It is a value between 0 and 1 representing the percentage of times a rule is exhibited by a user divided by the total number of rules generated by that user in the same window (specified by WS). Using CT, we can prune rules that occur very rarely, and the impact of an increasing CT value is shown in Figure 6. A higher CT means a smaller number of applied rules, reduced prefetching requests, and less power consumption.

SECTION IV. EXPERIMENTAL RESULTS

We evaluate the effectiveness of our proposed prefetching algorithms using forty-two months of real world download requests provided by EROS. To accurately evaluate the proposed algorithms with respect to performance and power consumption, we develop an emulator of EROS satellite image distribution system. This emulator is able to emulate the EROS system with a variety of hardware and software configurations, which are shown in Figure 7. In each configuration, we collect the FTP hit rates of the utilized algorithms on a monthly and overall basis in order to determine performance improvement from the current configuration. We also evaluate the impact of different software and hardware configurations on power consumption and cost of the EROS system.

Size	Replacement	Prefetching
100tb LRU no prefetching		
(current configuration)	LRU	No
100tb	LRU	Yes
100tb	LFU	Yes
200tb	LRU	No
200tb	LRU	Yes

Fig. 7 Summary of hardware and software configurations.

The Least Recently Used (LRU) and Least Frequently Used (LFU) cache replacement algorithms are implemented in our emulator to determine which image to be evicted from the FTP server. Additionally, our emulator models a unique aspect: EROS fills the FTP to a certain capacity threshold (e.g. 90% capacity) before triggering a "clean-up" procedure, instead of replacing individual images one by one. Once triggered, the "clean-up" procedure removes a large chunk of images according to the selected replacement policy until the FTP satisfies a certain availability threshold (e.g 45% capacity). Moreover, we implement the EROS "One Week" policy in our emulator, which prevents an image from being removed if it resides in the FTP for less than a week. With the "One Week" policy, users are given at least a week to download their requested images.

In our experiments, the emulator does not prefetch any images if the prefetching option is "No". If the prefetching option is set as "Yes", the emulator applies both popularity-based prefetching and user-based prefetching algorithms.

USGS has supplied a log file containing over 5,000,000 lines of data (cross 42 months), with each line representing one user request for one image to be processed and made available for download. This trace file serves as the input of our emulator. From each request, the emulator extracts the ID of the user generating the request, the date on which the request was made, and an agglomerate variable which represents the location on the globe and the acquisition date of the image being requested.

While processing each request, the emulator records the monthly FTP hit rates and calculates the monthly power consumption of the FTP server and Image Processing Subsystem (Please refer to Figure 2). The EROS system uses a ninenode cluster to process workorders. Each node is capable of processing up to four images simultaneously. A new workorder (either issued by FTP missing or prefetching) will be inserted to the processing queue and assigned to one of the nine compute nodes in a round-robin fashion. A workorder is considered to be utilizing 100% power when processing four or more images, with linear power decline for a partially empty queue. The EROS system FTP server farm is built with Sun F5100 flash storage arrays.³ Each 1920GB array consumes 386W of power when active.¹⁶ Due to the unpredictable user behavior, the nature of SSDs to provide peak performance as well as the EROS "One Week" policy, it is hard to dynamically turn on/off some SSDs in the FTP server to save power. Therefore, in our emulator, all SSDs in the FTP server farm are always powered on.



Fig. 8 Comparison of monthly hit rates for 100TB LRU caches with and without prefetching.



Fig. 9 Average of monthly hit rates for 100TB LRU caches with and without prefetching.

The possible addition of 100TB in storage costs approximately \$8 million.¹⁷ The monthly power cost is calculated using the commercial utility rates for Sioux Falls, South Dakota (utility provider for EROS) which, at the time of study, is \$0.081 per kilowatt.¹⁸

A. Impact on Performance

1) Prefetching Applied to Existing Configuration

Currently, the EROS system utilizes LRU replacement algorithm only, achieving a 65.86% hit rate, as shown in the summary at the end of this section in Table II. We have applied the prefetching algorithms described in Section III to the standard 100TB cache used in EROS currently. 2TB of disk space in the cache is reserved exclusively for images that have been processed as a prefetch request, and no prefetched images will be stored on the remaining 98TB. Figure 8 depicts the monthly hit rates for EROS standard 100TB LRU cache alongside the same 100TB LRU cache with the addition of prefetching. We observe an immediate and consistent increase in hit rate over time from the proposed prefetching algorithms. A software-only change would increase average monthly hit rate by 4.43%, evidenced by Figure 9. Meanwhile, we observe that the system power drops in almost all cases with the proposed prefetching algorithms implemented.



Fig. 10 Comparison of monthly hit rates for 100TB LRU and LFU caches with prefetching.

2) Selecting Another Cache Replacement Algorithm

The current configuration of the EROS system uses LRU replacement policy to determine which images to be removed from the cache. This is a simple policy as it only considers the amount of time elapsed from the last request of an image. A more complex policy might yield better results. As the focus of our study is on the effects of prefetching, we examine the use of the LFU replacement policy in addition to prefetching with a 100TB cache. Both systems will use the same *98TB/2TB* split as described previously.

We demonstrate the monthly hit rates of the LFU and LRU policies in Figure 10 with the current configuration (100TB LRU without prefetching) for reference. We can see that the addition of prefetching still supplies an increase in hit rate over the current configuration, but the change to LFU does result in a lower hit rate than LRU in most months. On average, a 100TB LFU cache with prefetching attains a 68.41% hit rate, which is 1.8% lower than the results achieved with a 100TB LRU configuration in Figure 11.

3) Expansion of Storage Capacity

Up to this point, we have explored the option of using intelligent prefetching to improve the performance of EROS satellite image distribution system. The alternative solution for EROS is to expand the capacity of its existing hybrid storage system, especially the capacity of the FTP server. In this experiment, we evaluate the impact of doubling the FTP server size to 200TB on performance. All other configuration facets such as LRU replacement policy and the lack of prefetching found in the current system remains the same for this experiment.



Fig. 11 Average of monthly hit rates for 100TB LRU and LFU caches with prefetching.



Fig. 12 Comparison of monthly hit rates for 100TB and 200TB LRU caches without prefetching.







Fig. 14 Comparison of monthly hit rates for 200TB LRU caches with and without prefetching.



Fig. 15 Average of monthly hit rates for 200TB LRU caches with and without prefetching.

The performance of the 200TB LRU cache deviates frequently from that of the 100TB current configuration with both positive and negative shifts, as expressed by Figure 12. On average, the monthly hit rate of a larger 200TB cache is 69.58%, which is 3.71% higher than the current 100TB configuration, as evidenced in Figure 13. As expected, a larger FTP server (cache) size provides more storage for processed images and allows for a lower availability threshold for the LRU replacement policy.

4) Prefetching and Expansion of Storage Capacity

As the EROS system continues to expand with more users and requests, the size of the FTP server (cache) may be expanded out of necessity in spite of the higher associated cost. Our final experiments evaluate the effectiveness of the prefetching algorithms described in section III when they are applied to the 200TB LRU cache. Will prefetching continue to improve the performance of EROS system with a larger cache? If so, how much can we offset the additional cost of 100TB of extra SSDs by using prefetching?

Figure 14 plots the monthly hit rates for the two cache configurations of 200TB LRU with and without prefetching. We can observe the configuration enhanced with prefetching consistently outperforms the configuration without prefetching. The monthly results are summarized in Figure 15 and we observe that the 200TB LRU cache enhanced with prefetching has an average monthly hit rate of 71.83%, a marked 2.25% improvement over the 69.58% hit rate attained with a 200TB cache without prefetching.



Fig. 16 Monthly power consumption for 100TB LRU caches with and without prefetching.



Fig. 17 Monthly power consumption for 100TB and 200TB LRU caches without prefetching.

B. Impact on Power Consumption

1) Prefetching Applied to Existing Configuration

On a monthly basis, the addition of prefetching algorithms conserves an average of 159,963 watts of power per month, visible in Figure 16. As both configurations utilize the same number of SSDs, the differences in performance will come from the combination of overhead required to process the additional images and the same overhead not required for hits on prefetched images. It appears that the penalty of additional workload caused by prefetching is much less than the benefit of additional hits that prefetching results in. Prefetching clearly improves both the hit rate and the power consumption, making it a very appealing solution.

2) Expansion of Storage Capacity

Figure 17 plots the monthly power consumption of the 200TB LRU cache against the current configuration (100TB LRU) and we can easily see the extra power consumed by the addition of 100TB is quite significant.



Fig. 18 Monthly power consumption for 200TB LRU caches with and without prefetching.

Cache Configuration	Avg. Monthly Hit Rate	Avg. Monthly Power (Watts)	Avg. Monthly Cost (\$)
100TB LRU No Prefetching (current)	65.8628%	13,908,620	1,127
100TB LRU with Prefetching	70.2583%	13,748,657	1,114
200TB LRU No Prefetching	69.5770%	20,152,915	1,632
200TB LRU with Prefetching	71.8283%	17,052,688	1,381

TABLE II Summary of Average Monthly FTP Hit Rate and Monthly Power Consumption

3) Prefetching and Expansion of Storage Capacity

We now examine power consumption of the same two cache configurations in figure 18. We observe that the inclusion of prefetching results in an average monthly power consumption of 17,052,688 watts, a 15.38% reduction from the 20,152,915 watt average monthly power projection of a 200TB LRU cache without prefetching.

C. Summary of Results

In this section, we provide several possible software and hardware configurations that EROS may choose to improve the performance of its satellite image distribution system, presented previously in Figure 7. EROS currently operates with a cache capacity of 100TB and can achieve a 3.71% increase in hit rate through the use of our proposed prefetching algorithms. In addition to an increase in hit rate, EROS is projected to consume 159,963 less kilowatts of power per month.

If the system were expanded to a cache capacity of 200TB, we predict that an additional 6,244,295 watts of power would be consumed monthly, at the benefit of 3.71% increase in monthly hit rate. If EROS does expand to a larger FTP server, our proposed prefetching algorithms have the potential to reduce the power consumption

by 3,100,227 watts per month (15.38%). Furthermore, prefetching on a 200TB LRU cache would increase the hit rate by 2.25%. The results of our experiments are summarized in Table II.

SECTION V. RELATED WORK

Massive data systems are characterized by their focus on the warehousing of vast amounts of data and the necessity to serve a multitude of clients simultaneously. The current trends in research in the improvement of massive storage systems have focused on the application of solid state disks to take advantage of the myriad of technical benefits they provide. Lee et al. has presented a thorough evaluation of the benefits of using solid state drives as a cache for a larger sized hard disk array to create a tiered storage architecture in order to improve system throughput as well as to reduce power consumption.¹⁴ Other researchers have also explored using SSDs as a faster tier of storage for caching and suggest a similar approach.¹⁵ Schall et al. have explored using SSDs in high-performance massive databases and measured power consumption under a series of differing loads and access patterns, recommending that SSDs not being used as large storage devices but instead as storage for the most highly accessed pieces of data in order to take advantage of the potential power savings provided by SSDs.¹⁹ The EROS system utilizes a tiered hybrid storage system where raw data is stored on magnetic tapes before being requested for processing and sent to the higher layer - the FTP servers made of SSDs, which can take advantage of SSDs and tradeoff performance and cost when appropriate configures are selected.

Prefetching has been widely used to increase cache performance.^{7,8,9,10,11,12} It was reported that an effective prefetching algorithm can improve the cache hit ratio by as much as 50 percent.¹³ It also has been shown that performance can be improved without a higher cost via intelligently prefetching.²⁰ Prefetching techniques can be classified into informed prefetching and predictive prefetching. Informed prefetching relies on user-provided hints while predictive prefetching is based on the analysis results of historical user behavior. For the EROS system, it is almost impossible to obtain a list of a user's future requests before they are submitted to the system. Without prior knowledge of a user's choices, predictive prefetching (NSP)^{21,22,23} are also used to explore spatial locality to improve cache performance. However, the structure of the EROS data prohibits the forming of direct spacial relationships among two images because each request can shift in six possible dimensions (Please refer to Section III-B).

SECTION VI. CONCLUSIONS

In this paper, we have presented a variety of configurations that can be applied to the FTP servers of a publicservicing massive storage system maintained by USGS EROS. We have explored the changes in two key metrics (performance and power consumption) that can be obtained by the adoption of various software and hardware configurations, leading us to the recommendation that prefetching algorithms be applied to any of the proposed cache configurations in order to increase the hit rate and decrease the system power consumption.

In addition, we have examined the effects of a likely future expansion of FTP server size on the EROS system and found a slight increase in hit rate in addition to the expected substantial increase in power consumption. Although the benefit of prefetching diminishes as cache sizes become larger, we continue to suggest the use of prefetching as a mechanism for offsetting the cost of powering additional hardware. The cost of expanding the FTP server farm to 200TB would contain not only a significant up-front investment (about 8 million dollars), but an increasing power draw. Our experimental results (Table II) demonstrate that the EROS system can maintain the same level of performance with 31.78% of power savings by using our proposed popularity-based and user-specific prefetching algorithms, compared to doubling the size of its FTP server farm. Therefore, we suggest administrators of large-scale massive storage systems fully explore all possible software optimizations before physically expanding the existing system.

ACKNOWLEDGMENT

The work reported in this paper is supported by the U.S. National Science Foundation under Grants No. CNS-0915762 and CNS-1212535, and the Texas State University Research Enhancement Program. We also gratefully acknowledge the support from the U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) Center.

REFERENCES

1. http://earthexplorer.usgs.gov/

2. http://glovis.usgs.gov/

3. Faundeen, J.," Archiving strategy for USGS EROS center and our future direction" in Proceedings of 2010 Roadmap for Digital Preservation Interoperability Framework Workshop, No. 5, 2010.

4. Romoser, B., Fares R., Janovics, P., Ruan, X.J., Qin, X., Zong, Z.L."Global Workload Characterization of A Large Scale Satellite Image Distribution System", In Proceedings of the 2012 IEEE International Performance Computing and Communications Conference (IPCCC), Dec. 2012.

5. Z. L. Zong, J. Job, X. S. Zhang, M. Nijim, and X. Qin," Case study of visualizing global user download patterns using Google Earth and NASA World Wind, Journal of Applied Remote Sensing 6 (1), 061703(October 09, 2012); doi: 10.1117/1.JRS.6.061703.

6. Fares, R., Romoser, B., Qin, X., Nijim, M., Zong, Z.," Performance Evaluation of Traditional Caching Policies on A Large System with Petabytes of Data", in Proceedings of the 7th IEEE International Conference on Networking, Architecture, and Storage, 2012.

7. Butt, A.R., Gniady, C., Hu, Y.C.," The Performance Impact of Kernel Prefetching on Buffer Cache Replacement Algorithms", IEEE Transactions on Computers, vol. 56, no. 7., pp. 889-908, Jul. 2007.

8. Grimsrud, K.S., Archibald, J.K., Nelson, B.E.," Multiple Prefetch Adaptive Disk Caching," IEEE Transactions on Knowledge and Data Engineering, vol. 5, no. 1, pp. 88-103, Feb.1993.

9. Jeon, H.S.," Practical Buffer Cache Management Scheme Based onSimple Prefetching", IEEE Transactions on Consumer Electronics, vol.52, no. 3, Aug. 2006.

10. Jeon, J., Lee, G., Cho, H., Ahn, B.," A Prefetching Web Caching Method Using Adaptive Search Patterns", in Proceedings of the IEEE Pacific Rim Conference On Communications, Computers, And Signal Processing,vol. 1, pp. 37-40, Aug. 2003.

11. Cao, P., Felten, E.W., Karlin, A.R., Li, K.," A Study of Integrated Prefetching and Caching Strategies", in Proceedings of the 1995 ACMSIGMETRICS Joint International Conference on Measurement andModeling of Computer Systems, vol. 23, no. 1, pp. 188-197, May 1995.

12. Lan, B., Bressan, S., Ooi, B., Tan, K.," Rule-Assisted Prefetching in Web-Server Caching", in Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 504-511,2000.

13. Nanopoulos, A., Katsaros, D., Manolopoulos, Y. "A Data Mining Algorithm for Generalized Web Prefetching", IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 5, pp. 1155-1169, September 2003.

14. Lee, H., Lee, K., Noh, S.," Augmenting RAID with an SSD for energy relief", In Proceedings of the 2008 conference on Power aware computing and systems. pp. 12-12. 2008.

15. "Hybrid Storage Solutions", Powerfile Technical Report. Retrieved from http://www.imconsulting.ca/files/IMConsultingPowerfileHybridStorageBrochure.pdf in September2012.

16. http://storage-news.com/2009/10/12/sun-f5100-finally-appears/

17. http://www.qdpma.com/storage/SSDother.html

18. http://www.eia.gov/beta/state/print.cfm-sid=SD

19. Schall, D. Hudlet, V., Hrder, T.," Enhancing energy efficiency of database applications using SSDs", In

Proceedings of the Third C* Conference on Computer Science and Software Engineering. pp. 1-9. 2010.

20. Griffoen, J., Appleton, R.," Reducing File System Latency Using a Predictive Approach, in Proceedings of the Summer USENIX Technical Conference, vol. 1, pp. 13-13, 1995.

21. Gindele, J.D.," Buffer Block Prefetching Method" IBM Technical Disclosure Bulletin, vol. 20, no. 2, pp. 696-697, July 1977.

22. Smith, A. "Cache Memories" ACM Computing Surveys, pp. 473- 530, Sept. 1982.

23. Srinivasan, V., Davidson, E., Tyson, G.," A Prefetch Taxonomy", IEEE Transactions on Computers, vol. 53, no. 2, 2004.