International Journal of Scientific Engineering and Technology Research

ISSN 2319-8885
Vol.06,Issue.23
July-2017,
Pages:4490-4495

www.ijsetr.com

Performance-Oriented Data Deduplication to Extend the Performance of Storage Systems in Cloud Environment

NAHIDAH RAHEEM AESA[1], T. RAMDAS NAIK[2]

[1]M.Sc Scholar, Dept of Computer Science, Nizam College, Osmania University, Hyderabad, TS, India.
[2]Assistant Professor, Dept of Computer Science, Nizam College, Osmania University, Hyderabad, TS, India.

Abstract: In large scale storage system data deduplication has gain more popularity and attention. Deduplication is one such storage optimization technique that avoids storing duplicate Copies of data and only one occurrence of the data are stored on storage media .It is essentially a compression method for removing redundant data. As a Space efficient method data de-duplication is used, in storage system to data backup. Storage space is saved by removing redundant data and also in network storage system the transmission of duplicate data is minimize. Scalability of fingerprint-index search for centralized data deduplication is main challenge. For high throughput and performance, removing duplicate contents and balancing load by low RAM overhead SiLo scalable deduplication system is used. Similarity and locality exploit both the similarity and locality approach which are complementarily. In SiLo deduplication system, small files which are related are grouped into a segment and segmentation of large file is done. In Silo RAM usage is reduced for index lookup.

Keywords: I/O Deduplication, Data Redundancy, Primary Storage, I/O Performance, Storage Capacity.

I. INTRODUCTION

Data deduplication has been demonstrated to be an effective technique in Cloud backup and archiving applications to reduce the backup window, improve the storage-space efficiency and network bandwidth utilization. Recent studies reveal that moderate to high data redundancy clearly exists in VM (Virtual Machine) enterprise and High-Performance Computing (HPC) storage systems. These studies have shown that by applying the data deduplication technology to large-scale data sets, an average space saving of 30%, with up to 90% in VM and 70% in HPC storage systems can be achieved For example, the time for the live VM migration in the Cloud can be significantly reduced by adopting the data deduplication technology. The existing data deduplication schemes for primary storage, n such as iDedup and Offline-Dedupe are capacity oriented in that they focus on storage capacity savings and only select the large requests to deduplication and bypass all the small requests (e.g., 4KB, 8KB or less). The rationale is that the small I/O requests only account for a tiny fraction of the storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved. However, previous workload studies have revealed that small files dominate in primary storage systems (more than 50%) and are at the root of the system performance bottleneck. Furthermore, due to the buffer effect, primary storage workloads exhibit obvious I/O burrstones.

From a performance perspective, the existing data deduplication schemes fail to consider these workload characteristics in primary storage systems, missing the opportunity to address one of the most important issues in primary storage, that of performance. With the explosive growth in data volume, the I/O bottleneck has become an increasingly daunting challenge for big data analytics in terms of both performance and capacity. Recent International Data Corporation (IDC) studies indicate that in past five years the volume of data has increased by almost 9 times to 7ZB per year and a more than 44-fold growth to 35ZB is expected in the next ten years Managing the data deluge on storage to support (near) real-time data analytics becomes an increasingly critical challenge for Big Data analytics in the Cloud, especially for VM platforms where the sheer number and dominance of small files overwhelm the I/O data path in the Cloud Moreover, our workload analysis, shows that data redundancy on the critical I/O path is much more pronounced than on the storage devices, largely due to the high temporal locality of small I/O requests. This suggests that, if such redundant I/Os can be removed from the critical I/O path, the performance bottleneck of a primary storage system may be significantly alleviated, if not removed. Thus, we argue that, for primary storage systems in the Cloud, it may be at least as important, if not more so, to de-duplicate the redundant I/Os on the critical I/O path for the sake of performance as to de-duplicating redundant data on storage devices for the sake of capacity savings.

On the other hand, our experimental studies suggest that directly applying data deduplication to primary storage systems will likely cause space contention in the main memory and data fragmentation on disks. This is in part because data deduplication introduces significant index-memory overhead to the existing system and in part because a file or block is split into multiple small data chunks that are often located in non-sequential locations on disks after deduplication. This fragmentation of data can cause a subsequent read request to invoke many, often random, disk I/O operations, leading to performance degradation. Our preliminary evaluations on the VM disk images reveal that the restore times with deduplication are much higher than those without deduplication, by an average of 2.9_ and up to 4.2. These two problems will be particularly acute with the deployment of the data deduplication technology into the primary storage systems for big data analytics in the Cloud. To address the important performance issue of primary storage in the Cloud, and the above deduplication-induced problems, we propose a Performance-Oriented data Deduplication scheme, called POD, rather than a capacity-oriented one (e.g., iDedup), to improve the I/O performance of primary storage systems in the Cloud by considering the workload characteristics.

POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic. More specifically, Select-Dedupe take the workload characteristics of small-I/O-request domination into the design considerations. It de-duplicates all the write requests if their write data is already stored sequentially on disks, including the small write requests that would otherwise be bypassed from by the capacity-oriented deduplication schemes. For other write requests, Select-Dedupe do not de-duplicate their redundant write data to maintain the performance of the subsequent read requests to these data iCache dynamically adjusts the cache space partition between the index cache and the read cache according to the workload characteristics, and swaps these data between memory and back-end storage devices accordingly. During the write-intensive bursty periods, iCache enlarges the index cache size and shrinks the read cache size to detect much more redundant write requests, thus improving the write performance. During the read-intensive bursty periods, on the other hand, the read cache size is enlarged to cache more hot read data to improve the read performance.

Thus, the memory efficiency is maximized. The prototype of the POD scheme is implemented as an embedded module at the block-device level and a sub file deduplication approach is used. To examine the net effect of the POD scheme, in our trace-driven evaluation we use the block level traces that were collected beneath the memory buffer cache so that the caching/buffering effect of the storage stack is already fully captured by the traces. In other words, all the small I/O requests in our evaluation are issued from the buffer cache to the block devices after the former has processed the file system-issued requests. The extensive trace-driven experiments conducted on our lightweight prototype implementation of POD show that POD significantly outperforms iDedup in the I/O performance measure of primary storage systems without sacrificing the space savings of the latter. Moreover, as an application of the POD technology to a background I/O task in primary cloud storage, it is shown to significantly improve the online RAID reconstruction performance by reducing the user I/O intensity during recovery.

II. RELATED WORK

We briefly review the published research most relevant to our POD approach from the viewpoints of data deduplication in primary storage systems and small-write optimization, respectively.

A. Data deduplication

Data deduplication as a space-efficient technique has received a great deal of attention from both industry and academia. It has been demonstrated to be effective in shortening the backup window and saving the network bandwidth and storage space in backup and archiving applications. For example, D2DRR scheduling scheme is proposed to improve the bandwidth efficiency of Avionics Full Duplex (AFDX) networks with the benefits of low complexity and easy implementation Most existing studies focus on how to improve the deduplication efficiency by solving the index-lookup-disk-bottleneck problem and how to find the redundant data as much as possible Recent studies have shown that moderate to high data redundancy also exists in primary storage systems Among the recent studies that leverage the data deduplication technique to improve the I/O performance of HDD-based primary storage systems,the I/O Deduplication scheme focuses on improving the read performance by exploiting and creating multiple duplications on disks to reduce the disk seek delay, but does not optimize the write requests. That is, it uses the data deduplication technique to detect the redundant content on disks but does not eliminate them on the I/O path.

This allows the disk head to service the read requests by prefetching the nearest blocks from all the redundant data blocks on disk to reduce the seek latency. The write requests are still issued to disks even if their data has already been stored on disks. Sequence-based deduplication and iDedup are two capacity-oriented data deduplication schemes that target at primary storage systems by exploiting spatial locality to only selectively de-duplicate the consecutive file data blocks. They only select the large requests to de-duplicate and ignore all small requests (e.g., 4KB, 8KB or less) because the latter only occupy a tiny fraction of the storage capacity. However, previous studies and our workload analysis reveal that the large I/O requests only account for a small portion of all requests while the small

I/O redundancy on the I/O path is significant (Figure 1). It implies that it is the performance on the I/O path, rather than capacity efficiency on the storage devices, that stands to potentially gain more from deduplication for primary storage systems. Different from the above schemes targeted at saving storage space, the request-based Select-Dedupe in POD exploits the I/O redundancy to intelligently and selectively de-duplicate the write requests, thus improving the small-write performance of HDD-based primary storage systems while avoiding the data fragmentation problem. Moreover, none of the existing studies has considered the problem of space allocation between the read cache and the index cache.

Most of them only use an index cache to keep the hot index in memory leaving the memory contention problem unsolved the iCache in POD are designed to address the memory contention and the read amplification problem, although the idea of dynamic cache allocation is not original. Patterson et al. proposed a dynamic cache partition between perfected data and cached data to maximum the cache efficiency. ARC keeps track of frequently used and recently used pages and a recent eviction history for both of them. It uses ghost hits to adapt to the recent change in the resource usage for better performance. In SSD/HDD hybrid storage systems, Yongseok et al. proposed a dynamic scheme to divide the flash memory cache to cache space and over-provisioned space, thus providing better cache performance and GC efficiency inside SSDs. SAR and Nitro combines the flash-based SSD and data deduplication to improve the performance of primary storage systems. The iCache is inspired by these previous studies and designed to collaborate with Select-Dedupe to further eliminate the redundant write requests and address the read amplification problem in primary storage systems. It is orthogonal to and can be incorporated with the existing data deduplication schemes in primary storage systems to further boost the system performance.

B. Optimizations for small writes

Generally speaking, most existing small write optimizations for HDD-based primary storage systems utilize the logging technique that buffers the write data in temporal locations, such as disks, non-volatile RAM and flash, and efficiently flushes them to their original locations eventually. Parity Logging is introduced to overcome the small write problem of RAID5 by using the logging technique. For a write request, the new data block is written in place, while the XOR result of the old data block and new data block is recorded sequentially in a log disk. Log-Structured Array (LSA) and DCD also log the updated data on the new disk instead of writing it in place to improve the write performance. AFRAID updates the write data immediately, but delays the parity update operations to the next idle period between the bursts of the client activities. It essentially sacrifices some reliability for performance improvement. Eager-writing writes data to the free block that is closest to the current disk head position. However, how to track the free blocks and disk head position is a complicated issue in practice. In contrast, Range Writes improves the write performance by changing both the file system and the disk to make the fine-grained data placement. Write-back buffer is effective in improving the write performance. STOW is proposed to effectively capture the temporal locality and spatial locality of write requests, and control the destage rate so that the write-back buffer is neither under-utilized nor over-committed. However, the write back buffer that uses non-volatile RAM is very expensive in cost and small in capacity. Proximal I/O temporarily aggregates random updates in the flash memory and destages them to disk in a single disk revolution. Most of the aforementioned schemes aim to alleviate the small-write problem of disks or parity-based disk arrays by delaying the write and parity update operations to the system idle period. POD, different from them, improves the small write performance by eliminating the write requests when their write data is already stored on disks. Thus, with POD, many small-write requests can be processed in memory without disk I/O operations, which significantly improves the system performance.

III. SYSTEM ANALYSIS

A. Existing System

The existing data deduplication schemes for primary storage, such as iDedup and Offline-Dedupe, are capacity oriented in that they focus on storage capacity savings and only select the large requests to de-duplicate and bypass all the small requests (e.g., 4 KB, 8 KB or less). The rationale is that the small I/O requests only account for a tiny fraction of the storage capacity requirement, making deduplication on them unprofitable and potentially counterproductive considering the substantial deduplication overhead involved. However, previous workload studies have revealed that small files dominate in primary storage systems (more than 50 percent) and are at the root of the system performance bottleneck. Furthermore, due to the buffer effect, primary storage workloads exhibit obvious I/O burrstones.

B. Proposed System

To address the important performance issue of primary storage in the Cloud, and the above deduplication-induced problems, we propose a Performance-Oriented data Deduplication scheme, called POD, rather than a capacity-oriented one (e.g., iDedup), to improve the I/O performance of primary storage systems in the Cloud by considering the workload characteristics. POD takes a two-pronged approach to improving the performance of primary storage systems and minimizing performance overhead of deduplication, namely, a request-based selective deduplication technique, called Select-Dedupe, to alleviate the data fragmentation and an adaptive memory management scheme, called iCache, to ease the memory contention between the bursty read traffic and the bursty write traffic.

Advantages of Proposed System:

1. POD significantly improves the performance and saves capacity of primary storage systems in the Cloud

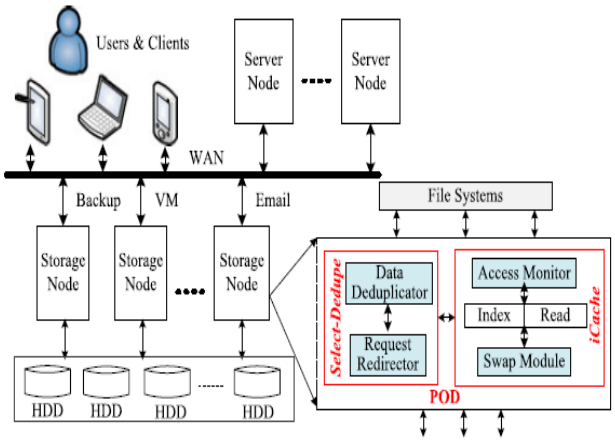

Fig 1. System Architecture.

IV. RESULTS

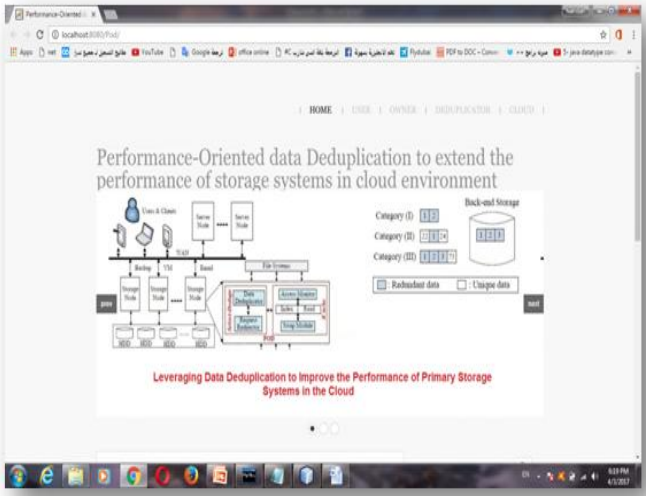

Fig 2. Home page.

A. Home Page

It contain the main menus of the program from them we can do the project work eg. (HOME, USER, OWNER, DEDUPLICATOR And CLOUD)

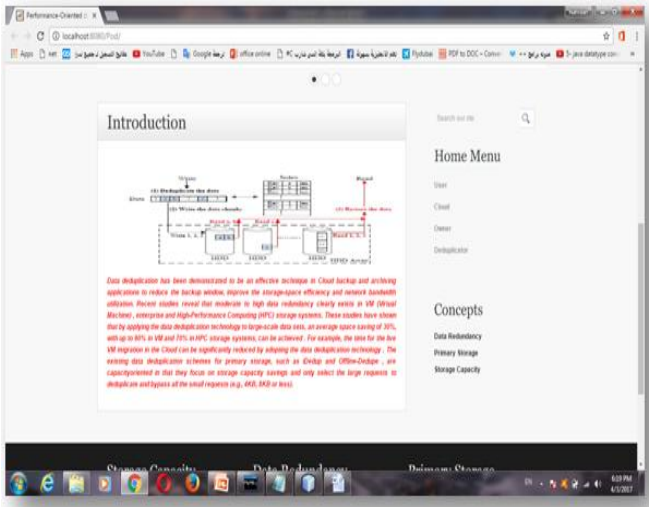

Fig 3. Home Menu.

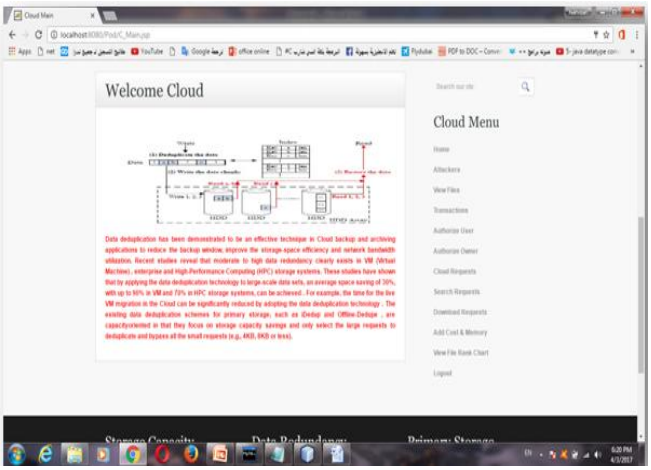

Fig 4. Cloud Menu.

B. Cloud menu: from this menu user can authorize the accounts and redirect requests for different orders e.g.(buy storage area, downloads ,search etc…)

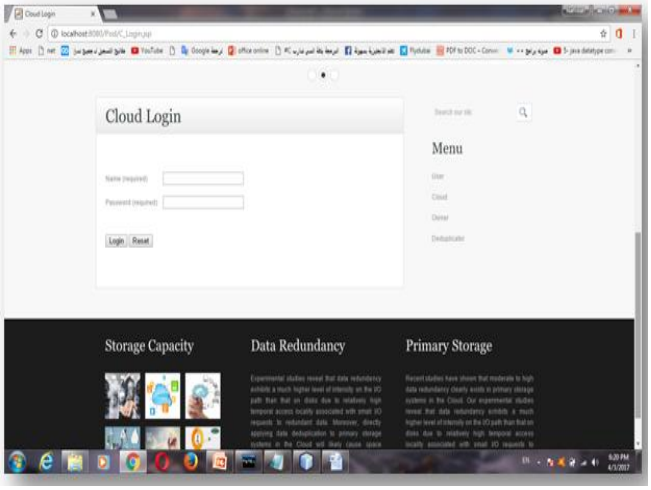

Fig 5. Cloud login page.

C. Cloud login required a User Name and Password to be entered to login to the cloud page

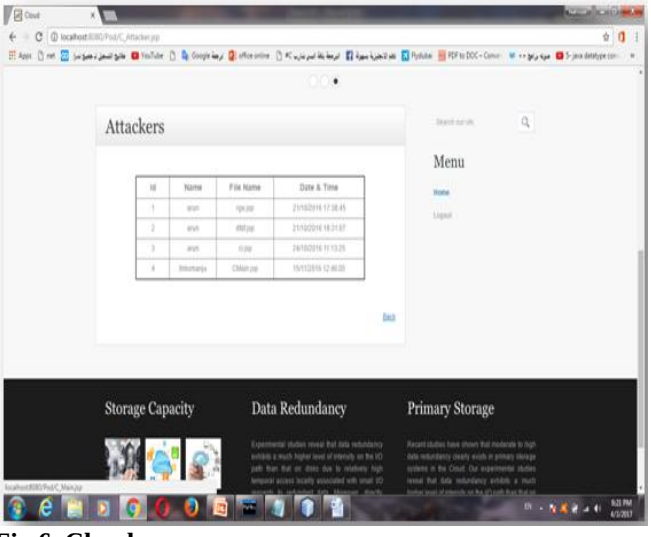

Fig 6. Cloud.

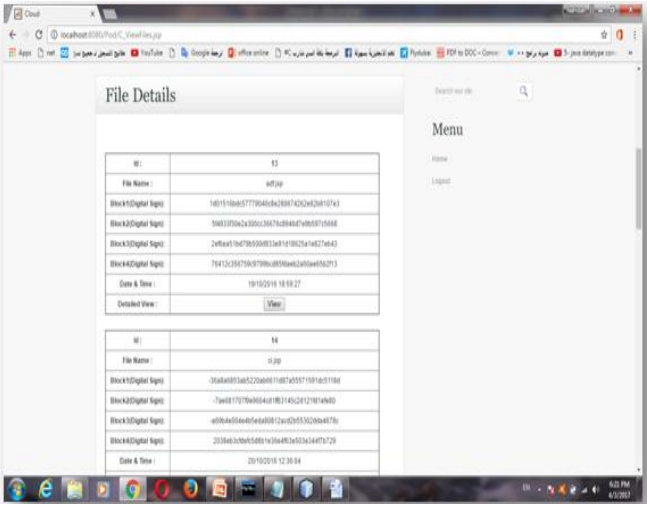

Fig 7. Files which uploads to the cloud.

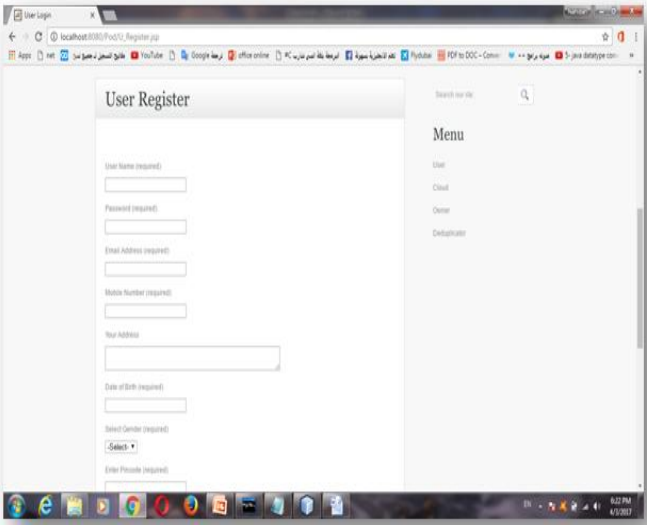

Fig 8. New user registration page.

This page used for register of new user on the cloud system which required entering the (NAME, PASS, DATE OF BEARTH, ZIP ODE, EMAIL, PHONE, GENDER, ADDRESS) all previous information should be entered by user to make his account.

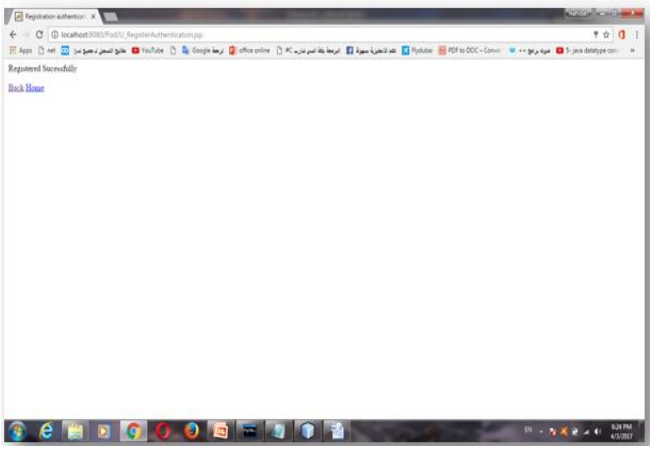

Fig 9. New account registration confirmation page.

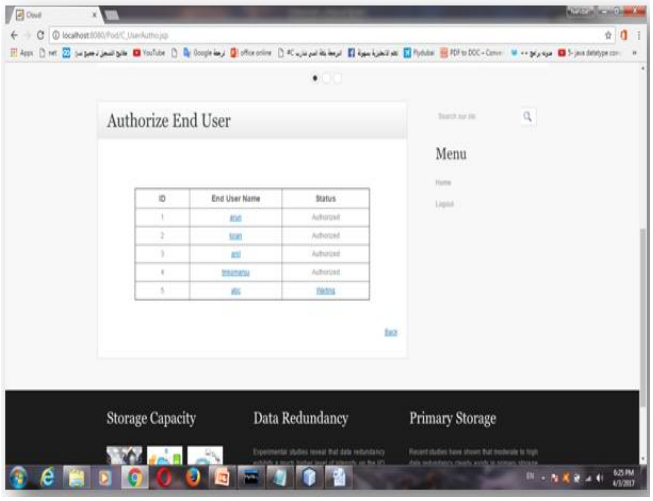

Fig10. New registered account Authorizing page (Cloud).

The new account which already being registered should be authorize in the cloud by the owner to allow the new user account to upload and download files from cloud.

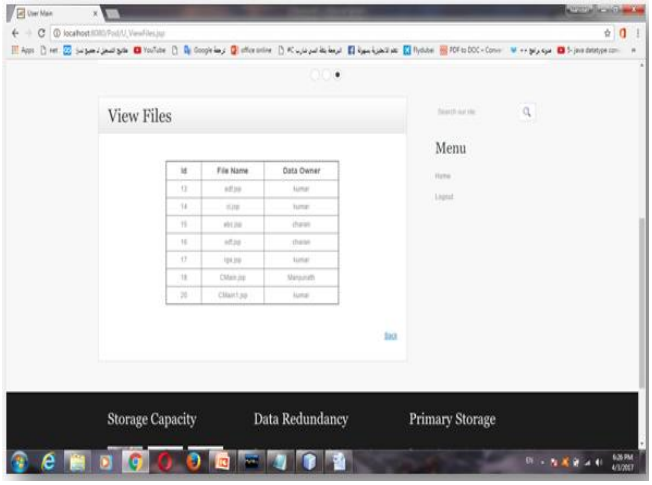

Fig 11. Files which exist in the cloud.

This page has shown the no. of files on cloud system along with owner name and file name.

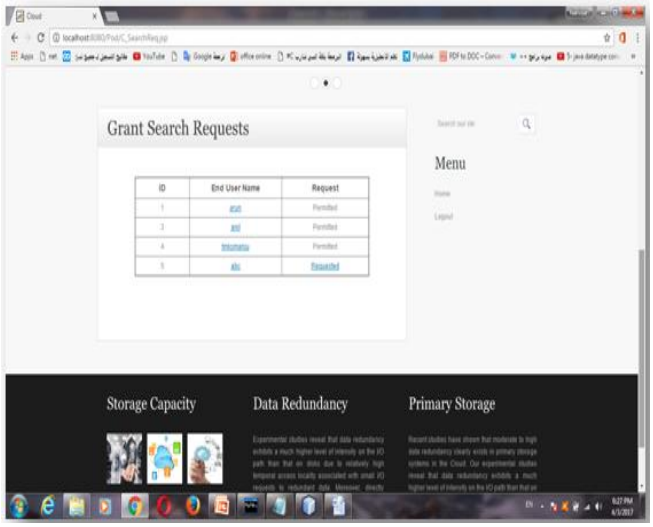

Fig 12. Users search requests.

Requests which done by the user are appears in this page to allow the owner to approve on them or dismiss them.

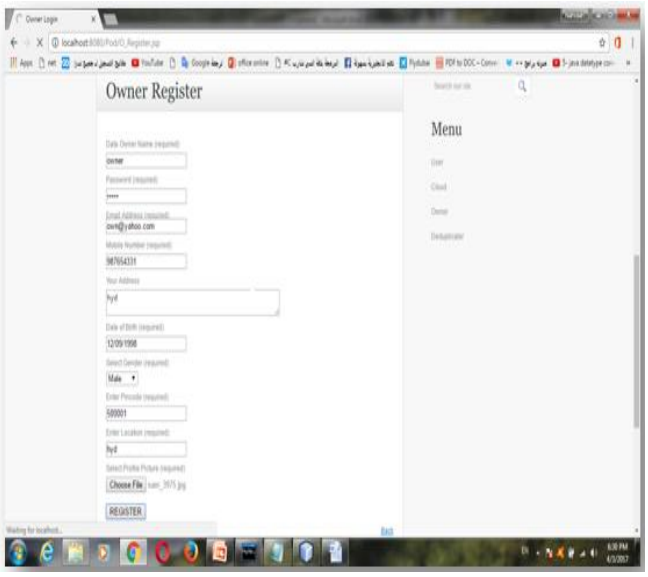

Fig 13. New cloud owner Registration page.

For the registration of new owner all information's (NAME, PASSWORD, GENDER, EMAIL, PHONE, ADDRESS, ZIPCODE and PROFILE PIC.) should be entered by user

V. CONCLUSION

In this paper, we propose POD, a performance-oriented deduplication scheme, to improve the performance of primary storage systems in the Cloud by leveraging data deduplication on the I/O path to remove redundant write requests while also saving storage space. It takes a request-based selective deduplication approach (Select-Dedupe) to de-duplicating the I/O redundancy on the critical I/O path in such a way that it minimizes the data fragmentation problem. In the meanwhile, an intelligent cache management (iCache) is employed in POD to further improve read performance and increase space saving, by adapting to I/O burrstones. Our extensive trace driven evaluations show that POD significantly improves the performance and saves capacity of primary storage systems in the Cloud. POD is an ongoing research project and we are currently exploring several directions for the future research. First, we will incorporate iCache into other deduplication schemes, such as iDedup, to investigate how much benefit iCache can bring to saving extra storage capacity and improving read performance. Second, we will build a power measurement module to evaluate the energy efficiency of POD. By reducing write traffic and saving storage space, POD has the potential to save the power that disks consume. We will compare the extra power that CPU consumes for computing fingerprints with the power that the storage saves, thus systematically investigating the energy efficiency of POD.

VI. REFERENCES

[1] Bo Mao, Member, IEEE, Hong Jiang, IEEE Fellow, Suzhen Wu, Member, IEEE and Lei Tian, Senior Member, IEEE, "Performance-Oriented data Deduplication to extend the performance of storage systems in cloud environment", IEEE 2016.

[2] N. Agrawal, William J. Bolosky, John R. Douceur, and Jacob R. Lorch. A Five-Year Study of File-System metadata. In FAST'07, Feb. 2007.

[3] A. Anand, S. Sen, A. Krioukov, F. Popovici, A. Akella, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau, and S. Banerjee. Avoiding File System Micromanagement with Range Writes. In OSDI'08, Dec. 2008.

[4] A. Batsakis, R. Burns, A. Kanevsky, J. Lentini, and T. Talpey. AWOL: An Adaptive Write Optimizations Layer. In FAST'08, Feb. 2008.

[5] P. Carns, K. Harms, W. Allcock, C. Bacon, S. Lang, R. Latham, and R. Ross. Understanding and Improving Computational Science Storage Access through Continuous Characterization. ACM Transactions on Storage, 7(3):1–26, 2011.

[6] F. Chen, T. Luo, and X. Zhang. CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory based Solid State Drives. In FAST'11, pages 77–90, Feb. 2011.

[7] A. T. Clements, I. Ahmad, M. Vilayannur, and J. Li. Decentralized Deduplication in SAN Cluster File Systems. In USENIX ATC'09, Jun. 2009.

[8] L. Costa, S. Al-Kiswany, R. Lopes, and M. Ripeanu. Assessing Data Deduplication trade-offs from an Energy Perspective. In ERSS'11, Jul. 2011.

[9] A. El-Shimi, R. Kalach, A. Kumar, A. Oltean, J. Li, and S. Sengupta. Primary Data Deduplication - Large Scale Study and System Design. In USENIX ATC'12, Jun. 2012.

[10] FIU traces. http://iotta.snia.org/traces/390.

[11] D. Frey, A. Kermarrec, and K. Kloudas. Probabilistic Deduplication for Cluster-Based Storage Systems. In SOCC'12, Nov. 2012.

[12] M. Fu, D. Feng, Y. Hua, X. He, Z. Chen, W. Xia, F. Huang, and Q. Liu. Accelerating Restore and Garbage Collection in Deduplication-based Backup Systems via Exploiting Historical Information. In USENIX'14, Jun. 2014.