AN EFFECTIVE IMPLEMENTATION OF DATA PREFETCHING TO ALLEVIATE THE STORAGE ACCESS LATENCY

V.Geetha,
Assistant Professor,
Department of Computer Science Engineering,
SCSVMV University,
India.

G.Vamsi,
UG Scholar,
Department of Computer Science Engineering,
SCSVMV University,
India.

Dr.C.K.Gomathy,
Assistant Professor,
Department of Computer Science Engineering,
SCSVMV University,
India.

N.P.Ganesh,
UG Scholar,
Department of Computer Science Engineering,
SCSVMV University,
India.

T.Jayanthi,
Assistant Professor,
Department of Computer Science Engineering,
SCSVMV University,
India.

G.Raheshwara Rao,
UG Scholar,
Department of Computer Science Engineering,
SCSVMV University,
India.

ABSTRACT : Data prefetching in distributed file systems is a well-known optimization technique which can mask both network and disk latency and consequently boost I/O performance. Traditionally, data prefetching is initiated by the client file systems, however, conventional prefetching schemes are not well suited for client machines that have limited memory and computing capacity. To offer an efficient prefetching approach for resource-limited client machines, this paper proposes a novel server side prefetching mechanism. Specifically, we propose to piggyback client identification to I/O requests so that server side block access history can be put into context. On the server side, we utilize the horizontal visibility graph technique to transform per-client time series of block access sequences into a connected graph for which we employ Tarjan's algorithm to disclose cut points in the connected graph. We express these patterns with feature tuples and we propose the X-step pattern matching algorithm to find a matching access pattern (i.e., a feature tuple) for a given block access history. Experimental results indicate that our newly proposed prefetching mechanism can ease client machines and their applications from the process of data prefetching, boosting Performance accordingly, and that it yields an attractive increase in data throughput as well.

I. INTRODUCTION

Technological innovations in distributed systems have been unfolding at an accelerated pace, resulting in the prevalence of cloud computing, which provides a novel pathway for utility computing with unlimited resource. flexibility, agility, scalability;widespread use of mobile devices equipped with limited computing facilities and the increasing speed of wireless networks. The trend is, effectively, to keep client devices simple and push complexity to the cloud. In turn, distributed file systems have been generally adopted as back-end storage systems to offer I/O services for parallel /distributed applications that need to process large amounts of data. To satisfy the ever growing demands on I/O services, it is therefore crucial to optimize distributed file system's performances. In this work using block storage i can store file name in block storage1, file key in block storage 2 and file in block storage 3. Because of data security as can split files like file name, file key and file and stored to block storage 1, block storage 2 and block storage 3. Suppose hacker can hack any block means he will not get the full file. Because of security purpose to split these files and stored in block storage. storage1, block storage2 and block storage3. The file name can be stored in block storage1, file key can be stored in block storage2 and file in block storage3. Because of security purpose we split these files and stored in block storage.

II. RELATED SYSTEM

The Existing System, can store files in single storage. The full file can be stored in single storage. If any want to grab your file means he can get the full file. The single storage gets corrupted. In this existing system there is so much of security issue.

Drawbacks

- Data Stored in single storage
- Data loss
- Low security

III. PROPOSED SYSTEM

File split and store files into three blocks .One block contains of file name, other contains file key and file in the third block .Because of security purpose we split these files and store in block storage. Suppose hacker wants to corrupt and get any of files from storage , then it will not get the full file .

Advantages

- Data Security.
- Data stored in block storage.
- Split files.

Authentication

- The user is first registered his/her own details like user name, email, password and confirm password.
- After registration he has to login his own user name and password.
- If an user name is correct, he has to go to his/her own account, if not it will give you an alert message as "Please Check Your User Name and Password."

File Upload & File Split

- After log in the user upload files to the multiple server.
- The file name will be stored in block storage1, file key will be stored in block storage2 and file will be stored in block storage3.
- In the existing the user upload file and it will be stored in single server but in proposed system the file will be stored in multiple servers because of security purpose.
- The file will be split like file .
- Split files that are uploaded by user, the file will be split-ted in file uploading time.
- At the file uploading time the file will be split ted like file name, file key and file. The split files can be stored in three different servers like block storage1, block storage2 and block storage3.

File Stored in Block Storage & File search

- The file will be stored in block storage.
- The block storage is nothing but small storage, it is a distributed storage, In this storage we can store limited amount of files because it is limited storage . The Single storage can be distributed to block storage1, block storage2 and block storage3 .
- It can search files that are uploaded by user. The file searching can be done by three stages because the file can be stored in three servers like block storage1, block storage2 and block storage3.
- The file searching can be done block-wise. In block storage 1 the storage ask filename, type your filename , If filename matches with the server file you are redirected to next block storage otherwise you will not redirected to next block storage.
- In block storage2 it will ask file key, If file key matches you will redirected to block storage3 otherwise block storage2. Finally the file will be retrieved in block storage 3.

Client File System

- The Client File System is nothing but single storage system, it is an user personal storage, If the user block storage3 filled you would move your files to client file system.
- If you want access client file system you need permission, For that you can request admin for accessing client file system.
- The admin accept request of user and give access permission to user for client file system.

Download File

- User can download files from client file system.
- The user can move their files from block storage 3 to client file system if block storage 3 is overflowed .
- Then the files are stored in client file system. Using client file system we can download files.

IV.SYSTEM ARCHITECTURE

Distributed systems are groups of networked computers, which have the same goal for their work. The terms " concurrent computing","parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel"and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing parallel computing.

In this System Played has client file systems, block storages, user ,admin. of Architecture prefetching on storage servers using blocks are given

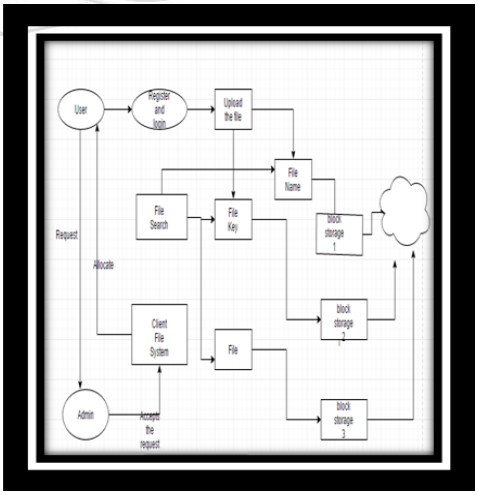


Figure1: Prefetching Architecture

V. ALGORITHM TECHNIQUE

X-STEP ALGORITHM

Data Prefetching is widely used technique for hiding data access latency by referring to identified access patterns. However, the effectiveness of Prefetching is primarily dependent on the prediction accuracy of future access

requests. Furthermore, the speed of predictions on future requests is also critical to the effectiveness of the Prefetching mechanism. Therefore, in order to obtain attractive performance improvements brought by server-side data Prefetching, user have introduced a novel pattern matching algorithm, which user call the X-step matching algorithm. X-step compares the feature tuple of an observed block access with the feature tuples of previously identified access patterns.

VI. RESULTS

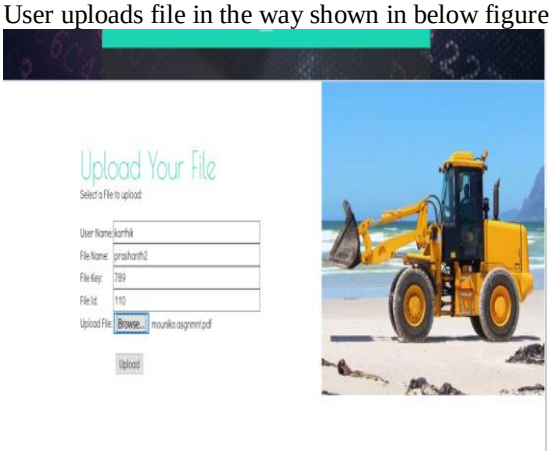
In this project using block storage user can store file name in block storage1, file key in block storage 2 and file in block storage 3. Because of data security user can split files like file name, file key and file and stored to block storage 1, block storage 2 and block storage 3. Suppose hacker can hack any block means he will not get the full file. Because of security purpose user split these files and stored in block storage. While user search file it will file name in block storage1. If user give correct file name it will search in server whether the file name is there or not, if file name matched in server means it will goes to block storage2, In block storage 2 it will ask file key, if user give correct key it will goes to block storage 3 otherwise user will not allowed to block storage 3 because of security purpose.

Home Page

User gets register and Login to his/her own account

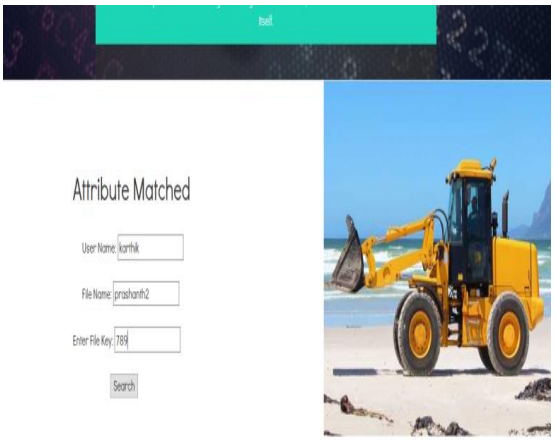
File Upload

User uploads file in the way shown in below figure



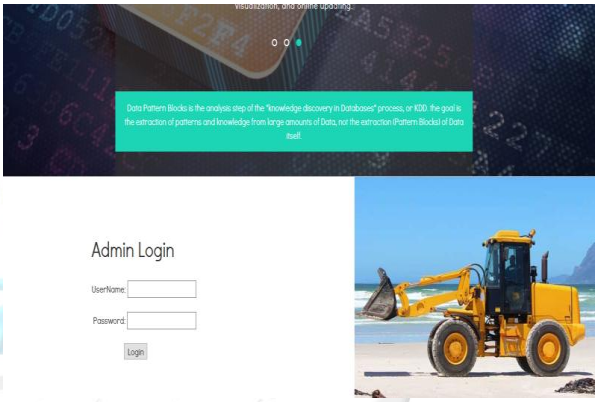
File Search

User search for his/her file to view or download as shown below



Admin Login

Admin logins and accepts the user requests on client file systems



VII. CONCLUSION

Client identification is piggybacked to I/O requests and on the server side the offset sequence of block access events is transformed into a horizontal visibility graph. It translate a time series of block access events into a connected graph and classify block acess patterns by employing the Tarjan algorithm, a graph theory technique to find cut vertices in a connected graph. Furthermore, we represent access patterns in feature tuples in order to save memory space and to conduct pattern matching quickly and accurately. The back experimental results with micro-benchmarks and realsystem block traces have shown that our server-side prefetching mechanism can reduce I/O response time and improve read data throughput compared to prefetching, Read-ahead prefetching and Signature-based prefetching.

VIII. REFERENCES

[1]. Wu S, Wang F, Shi X, and Jin H et al. "Network I/O load basedvirtual machine placement algorithm in HPC Cloud". Science China Information Sciences (Chinese), 2012, vol 42:290-302

[2]. H. Lei and D. Duchamp. "An analytical approach to file prefetching". USENIX Annual Technique Conference, Anaheim, California, USA: USENIX Association, 1997.

[3]. Gomathy, C.K. and S. Rajalakshmi, 2014. Software architecture design using service oriented on quality metrics. Aust. J. Comput. Sci., 1: 9-16
[4]. Liao J, Li L, Chen H et al. "Adaptive Replica Synchronization for Distributed File Systems". IEEE Systems Journal, 9(3): 865–877, 2015.
[5]. Denning P J." The locality principle". Communications of the ACM, 2005,vol 48: 19–24.