International Journal of Scientific Research and Engineering Development-– Volume 6 Issue 4, July- Aug 2023

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Distributed File System To Data Prefetching For Cloud

G.Bharathikannan*, G.Gayathri**

*(Asst ProfessorCSE, SembodaiRukmaniVaratharajan Engineering College, and Sembodai
Email:bharathi348@gmail.com)

** (PG Scholar CSE, SembodaiRukmaniVaratharajan Engineering College, and Sembodai
Email: gayathrigsg2000@gmail.com)

----------**********----------

Abstract:

The history of disk I/O access events, and then send the prefetched data to the relevant user machines proactively. To put this technique to work, the data about client nodes is piggybacked onto the real client This paper presents an initiative information prefetching scheme on the room servers in distributed file systems for cloud computing. In this prefetching technique, the client machines are not substantially involved in the process of information prefetching, but the storage servers can directly prefetch the data after analyzing I/O requests, and then forwarded to the relevant room server. Next, two prediction algorithms have been proposed to forecast past block access operations for directing what data should be fetched on storage servers in advance. Finally, the prefaced data can be pushed to the relevant client machine from the storage server. Through a series of evaluation research with a collection of application benchmarks, we have demonstrated that our presented initiative prefetching technique can advantage distributed file systems for cloud environments to achieve better I/O performance. In particular, configuration-limited client machines in the cloud are not culpable for predicting I/O access operations, which can definitely contribute to preferable scheme performance on them.

Keywords **—client machine, i/o Access, prefetching, cloud computing.**

----------**********----------

I. INTRODUCTION

The assimilation of spread computing for search engines, multimedia websites, and data-intensive applications has brought about the generation of data at unprecedented speed. For instance, the amount of data created, replicated, and consumed in United States may double every three years through the end of this decade, according to the EMC-IDCN umerical Universe 2020 study. In general, the file system deployed in a distributed computing environment is called a distributed file system, which is always used to be a backend packing method to provide I/O services for various sorts of data intensive applications in cloud computing environments.

In fact, the distributed file organism employs multiple distributed I/O devices by striping file data across the I/O nodes, and uses high aggregate bandwidth to meet the growing I/O requirements of distributed and equivalent scientific applications .However, because distributed file systems scale both numerically and geographically, the network delay is becoming the dominant factor in remote file organism access. With regard to this issue, numerous information prefetching mechanisms have been proposed to hide the dormancy in spread file systems caused by network communication and disk operations. In these conventional prefetching mechanisms, the client file organism (which is a part of the file system and runs on theclient machine) is supposed to predict future access by analyzing the history of occurred I/O contact without any application interference. After that, the client file organism may send relevant I/O requests to packing servers for reading

 ©IJSRED:All Rights are Reserved

the relevant data in advance . Consequently, the applications that have intensive read workloads can automatically yield not only superior use of available bandwidth, but also less file operations via batched I/O requests through prefetching . On the other hand, mobile procedures generally have limited processing power, battery life and storage, but cloud computing offers an illusion of infinite computing resources. For combining the mobile devices and cloud computing to create a new frame, the mobile cloud computing research field emerged . Namely, mobile cloud computing provides mobile applications with data storage and processing services in clouds, obviating the condition to equip a powerful hardware configuration, because all resource-intensive computing can be completed in the cloud .

Thus, conventional prefetching schemes are not the best-suited optimization strategies for distributed file systems to boost I/O performance in mobile clouds, since these schemes require the client file systems running on client machines to proactively issue prefetching requests after analyzing the occurred access events recorded by themselves, which must place negative effects to the client nodes. Furthermore, considering only disk I/O events can reveal the disk tracks that can offer critical information to perform I/O optimization tactics,certain prefetching techniques have been proposed in succession to read the data on the disk in advance after analyzing disk I/O traces . But, this kind of prefetching only works for local file systems, and the prefetched data is cached on the local machine to fulfill the application's I/O requests passively. In brief, although block access history reveals the behavior of disk tracks, there are no prefetching schemes on storage servers in a distributed file system for yielding better system performance. And the reason for this situation is because of the difficulties in modeling the block access history to generate block access patterns and deciding the destination client machine for driving the prefetched data from storage servers.

To yield attractive I/O performance in the distributed file system deployed in a mobile cloud environment or a cloud environment that has many resource-limited client machines, this paper presents an initiative data prefetching mechanism. The proposed mechanism first analyzes disk I/O tracks to predict the future disk I/O access so that the storage servers can fetch data in advance, and then forward the prefetched data to relevant client file systems for future potential usages.

2. LITRATURE REVIEW

2.1 The Digital Universe in 2020: Big Data, Bigger Digital Shadows, Biggest Growth in the Far East-United States

J. Gantz , D. Reinsel discussed with— a measure of all the digital information created, replicated, and consumed in a single year. It's also a projection of the size of that cosmos to the end of the decade. The digital cosmos is made up of images and videos on mobile phones uploaded to YouTube, digital movies populating the pixels of our high-definition TVs, banking information swiped in an ATM, security footage at airports and major events such as the Olympic Games, subatomic collisions record by, transponders recording public road tolls, voice calls zip through digital phone and texting as a widespread means of transportation. With the rise of Big information alertness and analytics technology, the digital universe in 2012 has taken on the feel of a tangible geography — a vast, barely charted place full of promise and hazard.

The digital cosmos lives increasingly in a computing cloud, over terra firma of vast hardware datacenters linked to billions of distributed devices, all governed and defined by increasingly clever software. In this context, at the midpoint of a longitudinal study first with information collected in 20051 and extending to 2020,our analysis shows a continuously expanding, increasingly complex, and ever extra interesting digital universe. This is IDC's sixth annual study of the digital cosmos, and it's chock-full of fresh findings: From 2005 to 2020, the digital cosmos will grow by a aspect of 300, from 130 exabytes tov 40,000 exabytes, or 40 Billion gigabytes (more than 5,200 gigabytes for every man, woman, and child in 2020). From at this time until 2020, the digital cosmos will about double every two years.

 ©IJSRED: All Rights are Reserved

The asset in spending on IT hardware, software, services, telecommunications and staffν that could be considered the "infrastructure" of the digital cosmos and telecommunications will grow by 40% between 2012 and 2020. As a result, the investment per gigabyte (GB) during that equal period will drop from $2.00 to $0.20. Of course, investment in targeted areas like cargo space management, security, big data, and cloud computing will grow considerably faster. 1 The first Digital Universe learn was published in 2007. At that time, IDC's forecast for the digital cosmos in 2010 was 988 exabytes. Based on actuals, it was later amend to 1,227 exabytes. 2 ©2012 IDC Between 2012 and 2020, emerging markets' share of the expanding digital cosmos will growν from 36% to 62%. A majority of the information in the digital cosmos, 68% in 2012, is created and consumed by customers — watching digital TV, interacting with social media, sending camera phone images and videos between policy and around the Internet, and so on. Yet enterprises have liability or duty for nearly 80% of the information in the digital universe. They contract with issues of copyright, privacy, and compliance with regulations even when the information zipping through their networks and server farms is created and consumed by consumers.

Only a tiny fraction of the digital cosmos has been explored for analytic value. By 2020, nearly 40% of the information in the digital cosmos will be "touched" by cloud computing provider — meaning that a byte will be stored or processed in a cloud somewhere in its trip from originator to removal. The proportion of information in the digital universe that requires protection is growing faster than theν digital cosmos itself, from less than a third in 2010 to more than 40% in 2020. The amount of information individuals make themselves — writing documents, taking pictures,ν downloading music, etc. — is far less than the amount of information being created about them in the digital cosmos Much of the digital cosmos is transient — phone calls that are not recorded, digital TV imagesν that are watched (or "consumed") that are not saved, packets temporarily stored in routers, digital surveillance images purged from memory when fresh images come in, and so on. Unused storage bits installed throughout the digital cosmos will grow by a factor of 8 between 2012 and 2020 but will still be less than a quarter of the total digital cosmos in 2020.

2.2 Performance Evaluation of the PVFS2 Architecture

Julian M. Kunkel, Thomas Ludwig discussed with the trend to high performance computing leads to supercomputers consisting of a high number of nodes. While the aggregated computational performance rises it is difficult for the I/O subsystem to keep pace. Parallel file systems like e.g. PVFS2 [6, 5], Lustre [2, 17], and GPFS [16] are designed to provide I/O performance that scales well with the number of nodes [15]. Physical I/O devices of an arbitrary number of server nodes can be combined by a parallel file system into one logical file system to increase its size and performance. In the best case a parallel file system provides the aggregated performance of all I/O devices to other client applications at a high abstraction level. However, the achievable aggregated performance of a parallel file system depends on various factors.

Considering a single server the capabilities of CPU, network and I/O devices limit the contribution to the file system's overall performance. On the other hand the access pattern of an application and the distribution of data and metadata across the servers defines the level of parallel servers accesses. Moreover the file system itself is a complex parallel program which has its own bottlenecks. Due to these issues we usually see only a certain percentage of the expected throughput as sustained performance. In order to improve the implementation of a parallel file system developers determine the performance with I/O benchmarks. However, due to the interplay of the components it is hard to identify the reasons for unsatisfactory performance.

It might be induced, for example, by the behavior of the servers' I/O subsystem, by the network or by slow CPUs. Also, as a result of the data distribution the requests might utilize only a few servers while

 ©IJSRED: All Rights are Reserved

others are idle. This paper introduces a systematic approach for performance analysis of a parallel file system's architecture and shows results for various request types. The idea is to replace the parallel file system's methods accessing the physical I/O system with an efficient stub pretending to be real physical storage. This stub represents the necessary data of the file system in the servers' memory and never triggers real I/O operations. Thus, benchmarking of such a modified file system is not influenced by the slow underlying I/O subsystem.

This paper is structured as follows: At first an overview about the state-of-the-art in performance analysis and related work is given. Then, PVFS2 is introduced in brief. Some simple considerations help to estimate the performance of a parallel file system in section 4. At next, details of our benchmarking concept are described. Then, the evaluation environment and test programs are introduced. A detailed summary of practical results using this methods with PVFS2 is given in section 6. These results are discussed in the last section.

3. SYSTEM ANALYSIS

System analysis is the overall analysis of the organism before implementation and for incoming at a precise solution. Careful analysis of a organism before implementation prevents post implementation problems that might arise due to bad scrutiny of the problem statement.

Thus the necessity for systems scrutiny is justified. Analysis is the initial crucial step, detailed learn of the various operations performed by a organism and their relationships within and exterior of the organism. Analysis is defining the boundaries of the organism that will be followed by drawing and discharge.

3.1 EXISTING SYSTEM

- The file organism deployed in a dispersed computing environment is called a dispersed file organism, which is always used to be a backend storage organism to provide I/O services for various sorts of information demanding applications in cloud computing environments.
- In fact, the distributed file organism employs multiple distributed I/O devices by striping file information across the I/O nodes, and uses lofty aggregate bandwidth to meet the growing I/O supplies of distributed and similar scientific applications

3.1.1 Disadvantages

- system delay in numerically and geographically remote folder organism access.
- Mobile strategy normally have limited processing influence, battery life and storage

3.2 PROPOSED SYSTEM

- With regard to this concern, numerous information prefetching mechanisms have been planned to hide the latency in distributed file systems caused by network communication and disk operations.
- In these conventional prefetching mechanisms, the client file system is supposed to predict future access by analyzing the history of occurred I/O access without any application intervention.
- After that, the client file system may send relevant I/O requests to storage servers for reading the relevant data in advance.

3.2.1 Advantages

- The applications that have intensive read workloads can automatically yield not only better use of available bandwidth.
- Less file operations via batched I/O requests through prefetching

4. ARCHITECTURE DIAGRAM

 ©IJSRED: All Rights are Reserved

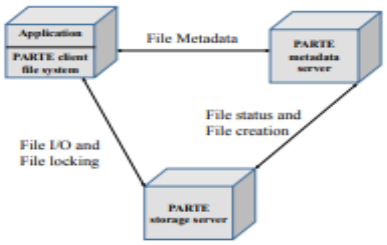

Fig 4.1 Architecture diagram

5. SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical drawing is turned into a working method. The discharge phase constructs, installs and operates the freshorganismThe most centralphase in achieving a fresh successful organism is that it will job efficiently and effectively.

There are more than a few activities involved while implementing a fresh project.

- End user Training
- End user Education
- Training on the function software

5.1 MODULES

1. Initialization of user
2. Location proof generation
3. File uploading(Crypt analysis)
4. Analysis Machine and get I/O
5. Performance analysis

5.2 Module Description

5.1.1 Initialization of User

- In Initialization of User user enter the personal information to register into application.
- After register the information ,user can login into the process through the username and password.
- In this module , we given the location of the user as a proof and validate the information which used for security purpose.

5.2.2 Location Proof Generation

- These approaches require the location proof server to have access to at least the majority of the concurrent (within a short delay) location proofs at the same location (within a small region).
- This needs users to submit their location proofs right after generating them, which is infeasible when there is no Internet connection on-the-spot.

6 CONCLUSION

We have planned, implemented and evaluated an initiative information prefetching come close to on the storage servers for distributed folder systems, which can be in a job as a backend storage system in a dark enenvironment that may have certain resource-limited customer machines. To be specific, the storage servers are capable of predicting future disk I/O access to guide fetching information in advance later than analyzing the accessible logs, and then they proactively push the prefetched information to relevant client folder systems for satisfying future applications' needs For the purpose of effectively modeling disk I/O entry patterns and accurately forwarding the prefetchedinformation the information about client folder systems is piggybacked onto relevant I/O needs, then transferred from customer nodes to corresponding storage server nodes.

Therefore, the client folder systems running on the customer nodes neither log I/O events nor conduct I/O entry prediction; consequently, the thin customer nodes can focus on performing necessary tasks with limited computing capacity and power endurance. Besides, the prefetched information will be proactively forwarded to the relevant customerfolder system, and the latter does not need to problem a prefetching request. So that both system traffics and system latency can be reduced to a certain extent, which have been demonstrated in our estimate experiments. To sum up, although

 ©IJSRED: All Rights are Reserved

the initiative information prefetching approach may place extra overhead on storage servers as they are supposed to predict the future I/Os by analyzing the history of disk I/Os, it is a nice option to build a storage organism for improving I/O performance while the customer nodes that have limited hardware and software configuration.

For instance, this initiative prefetching scheme can be applied in the distributed folder system for a mobile phone cloud computing environment, in which there are several tablet computers as well as smart terminals. The current implementation of our proposed scheme prefetching method can classify only two entry patterns and support two corresponding prediction algorithms for predicting future disk I/O entry After understanding of the fact about block entry events generally follow the allusion of locality, we are planning to job on classifying patterns for a wider range of function benchmarks in the prospect by utilizing the horizontal visibility chart technique.

Because the proposed plan prefetching method cannot job well when there is a mix of different workloads happening in the organism, classifying block entry patterns from the block I/O trace resulted by some workloads is one direction of our future job.

References

[1] J. Gantz and D. Reinsel. The Digital Universe in 2020: Big Data, Bigger Digital Shadows, Biggest Growth in the Far East-United States. http://www.emc.com/collateral/analystreports/idc-digital-universe-united-states.pdf [Accessed on Oct. 2013], 2013.

[2] J. Kunkel and T. Ludwig, *Performance Evaluation of the PVFS2 Architecture*, In Proceedings of 15th EUROMICRO International Conference on Parallel, Distributed and Network-Based Processing, PDP '07, 2007

[3] S. Ghemawat, H. Gobioff, S. Leung, *The Google file system*, In Proceedings of the nineteenth ACM symposium on Operatingsystems principles (SOSP '03), pp. 29–43, 2003.

[4] IO Signature Plus (IOSIG+) Software Suite[Accessed on Nov. 2012]. https://code.google.com/p/iosig/.

[5] UMass Trace Repository: OLTP Application I/O[Accessed on Jan.2014]. http://traces.cs.umass.edu/index.php/Storage/Storage.

[6] MobiDFS: Mobile Distributed File System[Accessed on Nov. 2014]. https://code.google.com/p/mobidfs/.

[7] E. E. Marinelli. Hyrax: Cloud computing on mobile devices using mapreduce. CMU, Tech. Rep., 2009.

[8] P. Sehgal, V. Tarasov, E. Zadok. Evaluating Performance and Energy in File System Server Workloads. the 8th USENIX Conferenceon File and Storage Technologies (FAST '10), pp.253-266, 2010.

[9] V. Tarasov, S. Bhanage, E. Zadok, et al. Benchmarking file system benchmarking: It* is* rocket science. In Proceedings of HotOS XIII,2011.

[10] N. Nieuwejaar and D. Kotz. The galley parallel file system. Parallel Computing, 23(4-5):447–476, 1997.

[11] IOzoneFilesystem Benchmark. http://www.iozone.org.

[12] Filebench Ver. 1.4.9.1 [Accessed on Apr. 2012], http://filebench.sourceforge.net.

[13] SysBench benchmark. http://sysbench.sourceforge.net.

[14] M. Cencini, F. Cecconi and N. Vulpiani. Chaos From Simple models to complex systems. World Scientific, ISBN:9814277657,2010.

[15] J. Liao, Y. Ishikawa. Partial Replication of Metadata to Achieve High Metadata Availability in Parallel File Systems. In the Proceedings of 41st International Conference on Parallel Processing (ICPP '12), pp. 168–177, 2012.

[16] I. Y. Zhang. Efficient File Distribution in a Flexible, Wide-area File System. Master thesis, Massachusetts Institute of Technology, 2009.

[17] S. Akyurek and K. Salem. Adaptive block rearrangement. ACM Transactions on Computer Systems, Vol.13(2):89-121, 1995.

[18] B. Bakke, F. Huss, D. Moertl and B. Walk. Method and apparatus for adaptive localization of frequently accessed, randomly accesseddata. US Patent No. 5765204, Filed June, 1998.

[19] V. Padmanabhan and J. Mogul. Using

 ©IJSRED: All Rights are Reserved

predictive prefetching to improve World Wide Web latency. ACM SIGCOMM ComputerCommunication Review, Vol. 26(3): 22–36, 1996.
[20] H. Lei and D. Duchamp. An analytical approach to file prefetching. In Proceedings of USENIX Annual Technical Conference (ATC'97), Anaheim, California, USENIX Association, 1997.
[21] E. Shriver, C. Small, and K. A. Smith. Why does file system prefetching work? In Proceedings of the USENIX Annual TechnicalConference (ATC '99), USENIX Association, 1999.
[22] T. M. Kroeger and D. Long. Predicting file system actions from prior events. In Proceedings of the USENIX Annual TechnicalConference. USENIX Association, 1996.
[23] T. M. Madhyastha and D. A. Reed. Exploiting global input/output access pattern classification. In Proceedings of the 1997 ACM/IEEEconference on Supercomputing (SC '97), ACM, 1997.

 ©IJSRED: All Rights are Reserved