International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 03 | Mar -2020 ISSN: 2582-3930

Performing Initiative Data Prefetching in Distributed File System for Cloud Computing

1G.Meenatchi,2M.Sabari Ramachandran,3N.Balasubramanion,4M.Mohamed Rafi

1Final year MCA, Mohamed Sathak Engineering College, Kilakarai.

2AssistantProfessor, Dept of MCA, Mohamed Sathak Engineering College,Kilakarai.

3Associate Professor, Dept of MCA, Mohamed Sathak Engineering College,Kilakarai.

4Professor, Dept of MCA, Mohamed Sathak Engineering College,Kilakarai.

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

Keywords—Mobile Cloud Computing, Distributed File Systems, Time Series, Serverside Prediction, Initiative Data Prefetching.

1. INTRODUCTION

The assimilation of distributed 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-IDC Digital 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 storage system to provide I/O services for various sorts of data- intensive applications in cloud computing environments. In fact, the distributed file system employs multiple in 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 parallel scientific applications. The distributed file systems scale both numerically and geographically, the network delay is becoming the dominant factor in remote file system access. With regard to this issue, numerous data prefetching mechanisms have been proposed to hide the latency in distributed file systems caused by network communication and disk operations. In these conventional prefetching mechanisms, the client file system (which is a part of the file system and runs on the client machine) 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. Consequently, the applications that have intensive read workloads can automatically yield not only better use of available bandwidth, but also less file operations via batched I/O requests through prefetching. On the other hand, mobile devices 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 infrastructure, the mobile cloud computing research field emerged. Namely, mobile cloud computing provides mobile applications with data storage and processing services in clouds, obviating the requirement to equip a powerful hardware configuration, because all resourceintensive 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 per- formance 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 ac- cess 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 prefetch ing 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. In short, this paper makes the following two contributions Chaotic time series prediction and linear regression prediction to forecast disk I/O access. We have modeled the disk I/O access operations, and classified them into two kinds of access patterns, i.e. the random access pattern and the sequential access pattern. Therefore, in order to predict the future I/O access that belongs to the different access patterns as accurately as possible (note that the future I/O access indicates what data will be requested in the near future), two prediction algorithms including the chaotic time series predic- tion algorithm and the linear regression prediction algorithm have been proposed respectively. Initiative data prefetching on storage servers. Without any intervention from client file systems except for piggybacking their information onto relevant I/O requests to the storage servers. The storage servers are supposed to log disk I/O access and classify access patterns after modeling disk I/O events. Next, by properly using two proposed prediction algorithms, the storage servers can pre- dict the future disk I/O access to guide prefetching data.

2.RELATED WORK

the purpose of improving I/O system performance for distributed/parallel file systems, many sophisticated techniques of data prefetching have been proposed and developed consecutively. These techniques indeed con- tribute to reduce the overhead brought about by network communication or disk operations when accessing data, and they are generally implemented in either the I/O library layer on client file systems or the file server layer. As a matter of fact, the data prefetching approach has been proven to be an effective approach to hide latency resulted by network communication or disk operations.proposed a technique to reduce the latency perceived by users through predicting and prefetching files that are likely to be requested shortly in the World Wide Web (WWW) servers. There are also several data prefetching tactics for distributed/parallel file systems. For instance, informed prefetching that leverages hints from the application to determine what data to be fetched beforehand, since it assumes that better file system performance can be yielded with information from the running application, However, this kind of prefetching mechanisms cannot make accurate decisions when there are no appropriate hints (I/O access patterns) from the running applications, but inaccurate predictions may result in negative effects on system performance. Moreover, the client file systems are obliged to trace logical I/O event and conduct I/O access prediction, which must place overhead on client nodes. Prefetching in shipped distributed file systems. Regarding the prefetching schemes used in real world distributed file systems, the Ceph file system is able to prefetch and cache files' metadata to reduce the communication between clients and the metadata server for better system performance. Although the Google file system does not support predictive prefetching, it handles I/O reads with quite large block size to prefetch some data that might be used by the following read requests, and then achieves better I/O data throughput. The technique of read cache prefetching has been em- ployed by the Lustre file system using Dell PowerVault MD Storage for processing sequential I/O patterns. Prefetching on file servers. first

investigated the block correlation in the storage servers by employing data mining techniques, to benefit I/O optimization on file servers S. researched disk I/O traffics under different workloads, as well as different file systems, and they declared the modeling information about physical I/O operations can contribute to I/O optimization tactics for better system performance . In an automatic locality-improving storage has been presented, which automatically reorganizes selected disk blocks based on the dynamic reference stream to effectively boost storage performance. After that, DiskSeen has been presented to support prefetching block data directly at the level of disk layout have proposed a server-side I/O collection mechanism to coordinate file servers for serving one application at a time to decrease the completion time. have explored and classified patterns of I/O within applications, thereby allowing powerful I/O optimization strategies including pattern-aware prefetching. which is a block reor-ganisation technique, and intends to reorganize hot data blocks sequentially for smaller disk I/O time, which can also contribute to prefetching on the file server. However, the prefetching schemes used by local file systems aim to reduce disk latency, they fail to hide the latency caused by network communication, as the prefetched data is still buffered on the local storage server side. Distributed file systems for mobile clouds. Moreover, many studies about the storage systems for cloud environments that enable mobile client devices have been published. A new mobile distributed file system called mobiDFS has been proposed and implemented in which aims to reduce computing in mobile devices by transferring computing requirements to servers. Hyrax, which is a infrastructure derived from Hadoop to sup- port cloud computing on mobile devices But Hadoop is designed for general distributed computing, and the client machines are assumed to be traditional computers. In short, neither of related work targets at the clouds that have certain resource-limited client machines, for yielding attractive performance enhancements. Namely, there are no prefetching schemes for distributed file systems deployed in the clouds, which offer computing and storage services for mobile client machine.

3.ARCHITECTURE AND IMPLEMENTATION

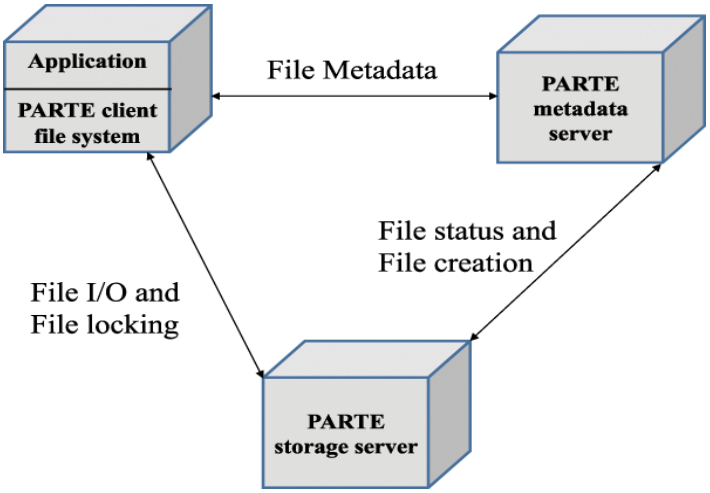

Figure 1:Architecture diagram

This paper intends to propose a novel prefetching scheme for distributed file systems in cloud computing environments to yield better I/O performance. In this section, we first introduce the assumed application con-texts to use the proposed prefetching mechanism; then the architecture and related prediction algorithms of the prefetching mechanism are discussed specifically; finally, we briefly present the implementation details of the file system used in evaluation experiments, which enables the proposed prefetching scheme. Assumptions in Application Contexts This newly presented prefetching mechanism cannot work well for all workloads in the real world, and its target application contexts must meet two assumptions resource-limited client machines. This newly proposed prefetching mechanism can be used primarily for the clouds that have many resource- limited client machines, not for generic cloud environments. This is a reasonable assumption given that mobile cloud computing, which employs powerful cloud infrastructures to offer computing and storage services on demand, for alleviating resource utilization in mobile devices Assumption Online Transaction Processing (OLTP) applications. It is true that all prefetching schemes in distributed file systems make sense for a limited number of readintensive applications such as database-related OLTP and server-like applications. That is because these long-time running applications may have a limited number of access patterns, and the patterns may occur repetitively during the life-time of execution, which can definitely contribute to boosting the effectiveness of prefetching.Piggybacking Client Information Most of the I/O tracing approaches proposed by other researchers focus on the logical I/O

access events occurred on the client file systems, which might be useful for affirming application's I/O access patterns .Nevertheless, without relevant information about physical I/O access, it is difficult to build the connection between the applications and the distributed file system for improving the I/O performance to a great extend. In this newly presented initiative prefetching approach, the data is prefetched by storage servers after analyzing disk I/O traces, and the data is then proactively pushed to the relevant client file system for satisfying potential application's requests. Thus, for the storage servers, it is necessary to understand the information about client file systems and applications. To this end, we leverage a piggybacking mechanism, which is transfer related information from the client node to storage servers for contributing to modeling disk I/O access patterns and forwarding the prefetched data. As clearly described in sending a logical I/O request to the storage server, the client file system piggybacks information about the client file systems and the application. In this way, the storage servers are able to record disk I/O events with associated client information, which plays a critical role for classifying access patterns and determining the destination client file system for the prefetched data. On the other side, the client information is piggybacked to the storage servers, so that the storage servers are possible to record the disk I/O operations accompanying with the information about relevant logical I/O events. demonstrates the structure of each piece of logged information stored on the relevant storage server. The information about logical access includes inode information, file descriptor, offset and requested size. And the information about the relevant physical access contains storage server ID, stripe ID, block ID and requested size.I/O Access Prediction Many heuristic algorithms have been proposed to she- herd distributing file data on disk storage, as a result, data stripes that are expected to be used together will be located close to one another. discovered that the spatial patterns of I/O requests in scientific codes could be represented with Markov models, so that future access can be also predicted by Markov models with proper state definitions have presented an automatic.

4. EXPERIMENTS AND EVALUATION:

We first introduce the experimental platform, the used comparison counterparts and the selected benchmarks in evaluation experiments. evaluation methodologies and relevant experimental results are discussed in details. Then, the analysis on prediction accuracy of two proposed prediction algorithms is presented. At last, we conduct a case study with real block traces.

4.1 Experimental Setup

The cluster and two LANs are used for conducting the experiments, the active metadata server and 4 storage servers are deployed on the 5 nodes of the cluster, and all client file systems are located on the 12 nodes of the LANs. specification of nodes on them. Moreover, for emulating a distributed computing environment, 6 client file systems are installed on the nodes of the LAN that is connected with the cluster by a 1 GigE Ethernet; another 6 client file systems are installed on the nodes of the LAN, which is connected with the cluster by a 100M Ethernet, and both LANs are equipped with MPICH2-demonstrates the topology of our experimental platform Evaluation Counterparts To illustrate the effectiveness of the initiative prefetching scheme, we have also employed a non-prefetchingarly.

5. CONCLUSION

We proposed, implemented and evaluated an initiative data prefetching approach on the storage servers for distributed file systems, which can be employed as a backend storage system in a cloud environment that may have certain resource-limited client machines. To be specific, the storage servers are capable of predicting future disk I/O access to guide fetching data in advance after analyzing the existing logs, and then they proactively push the prefetched data to relevant client file systems for satisfying futureapplications' requests. For the purpose of effectively modeling disk I/O access patterns and accurately forwarding the prefetched data, the information about client file systems is piggybacked onto relevant I/O requests, then transferred from client nodes to corresponding storage server nodes. Therefore, the client file systems running on the client nodes neither log I/O events nor conduct I/O access prediction; consequently, the thin client nodes can focus on performing necessary tasks with limited computing capacity and energy endurance. Besides, the prefetched data will be proactively forwarded to the relevant client file system, and the latter does not need to issue a prefetching re- quest. So that both network traffics and network latency can be reduced to a certain extent, which have been demonstrated in our evaluation experiments. To sum up, although the initiative data 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 system for improving I/O performance while the client nodes that have limited

hardware and software configuration. For instance, this initiative prefetching scheme can be applied in the distributed file system for a mobile cloud computing environment, in which there are many tablet computers and smart terminals. The current implementation of our proposed initiative prefetching scheme can classify only two access patterns and support two corresponding prediction algorithms for predicting future disk I/O access. After understanding of the fact about block access events generally follow the reference of locality, we are planning to work on classifying patterns for a wider range of application benchmarks in the future by utilizing the horizontal vis- ibility graph technique. Because the proposed initiative prefetching scheme cannot work well when there is a mix of different workloads happening in the system, classifying block access patterns from the block I/O trace resulted by several workloads is one direction of our future work. Besides, applying network delay aware replica selection techniques for reducing network transfer time when prefetching data among several replicas is another task in our future work.

ACKNOWLEDGMENTS:

This work was partially supported by ”National Nat- ural Science Foundation of China (No. 61303038)” and ”Natural Science Foundation Project of CQ CSTC (No. CSTC2013JCYJA40050)”. We would like to thank Dr.Yanlong Yin for providing specifications about IOSig+.

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/analyst- reports/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 Process- ing, PDP ’07, 2007.

[3] S. Ghemawat, H. Gobioff, S. Leung, The Google file system, In Proceedings of the nineteenth ACM symposium on Operating systems 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 Conference on 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(45):447–476, 1997.