A backend storage system in a cloud environment

A Ramadevi (M.Tech student), P. Ashwini Goud (Asst. Professor)
Computer Science and Engineering, Ganapathy Engineering College. Warangal, Telangana, India.

Abstract: 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, which can definitely contribute to preferable system performance on them.

Key Words: piggybacked, prediction, prefetched, future block access operations.

1. INTRODUCTION:

We have 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 future applications' requests. For the purpose of effectively modeling disk I/O access patterns and accurately forwarding the prefetched data.

2. METHODOLOGY:

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 resourcelimited 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. On-Line Transaction Processing (OLTP) applications. It is true that all prefetching schemes in distributed file systems make sense for a limited number of read-intensive 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 lifetime of execution, which can definitely contribute to boosting the effectiveness of perfecting. 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 illustrated in proposed system to 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 Figure 1, when 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.

3. AN OVERVIEW OF PROPOSED SYSTEM:

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 mobile.

DFS 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 support cloud computing on mobile devices. But Hadoopis 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 peperformance enhancements. we first introduce the assumed application contexts 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. The storage servers proactively forward the prefetched data to the relevant client file system for satisfying future application's requests.

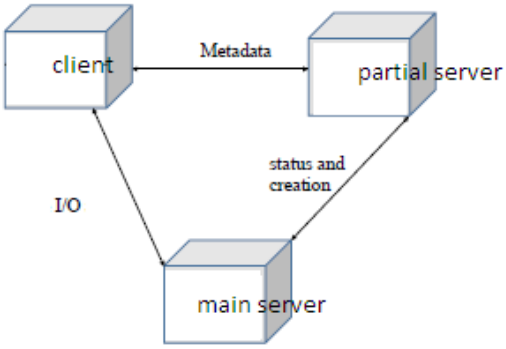

Fig1: Proposed System

4. CONCLUSION:

, 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 request. So that both network traffics and network latency can be reduced to a certain extent, which have been demonstrated in our evaluation experiments.

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. Oct.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 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.
6. MobiDFS: Mobile Distributed File System[Accessed on Nov. 2014].
7. E. E. Marinelli. Hyrax: Cloud computing on mobile devices using mapreduce. CMU, Tech. Rep., 2009.
8. Jianwei Liao, Francois Trahay, Guoqiang Xiao, Li Li, Yutaka Ishikawa Member, Performing Initiative Data Prefetching in Distributed File Systems for Cloud Computing IEEE TRANSACTIONS ON CLOUD COMPUTING , Volume: PP,Year: 2015.