First Name
affiliation

Second Name
affiliation

1 Problem and Motivations

Modern distributed file systems, such as GFS[**?**], HDFS[**?**], Ceph[**?**], which consist of three components, metadata servers(MDS), object-based storage devices(OSD) and clients as Fig. 1 shows. Clients perform metadata operations(*open, stat*) directly with MDS and file I/O directly with data server. This architecture decouples metadata I/O from data IO and increases overall scalability. For example, client requests data with pathname /var/log/client.log. By multiple interactions to MDS, client lookups the directory of `root`, `var` and `log` to locate the inode of `client.log` and checks the permission. Once client obtain the inode, it must get the capability to access to data. Then, client will request a lease on this file to MDS. Finnaly, client interact with OSD to operator file I/O. Due to the iterative lookup and checking permission operations, the IOs between clients and MDS are more than OSD. Because overall 50% of IOs are to metadata[**?**], metadata performance is crucial.

Especially in web content system, loading one web file leads to multiple scripts and images requesting. We analyse the access log[**?**] of web server and find that there are 23% of requests are correlated. These too frequently correlated metadata fetching increases the traffic to MDS and network. If there is no optimization on metadata fetching, too more traffic will overwhelm MDS and becomes a bottleneck in the system.

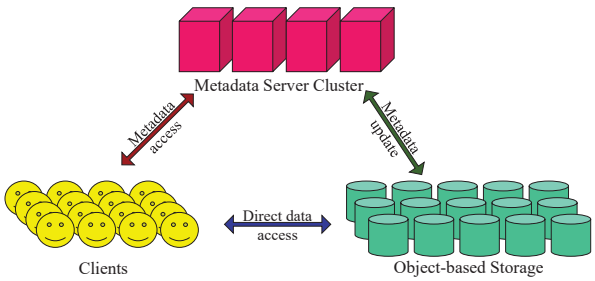

Figure 1: Architecture of distributed file systems.

2 Background and Related Work

Previous works focus on mining relationships among files according to their access pattern. If two files are frequent accessed closely in the past, they will be accessed together with higher probability near the future. Based on this definition, C-Miner[**?**] adopts a frequent sequence mining method to explore block correlations. DiskSeen[**?**] exploits disk layout and based on analysis tof temporal and spatial relationships to improve the sequentiality of disk access.

To reduce the IOs to file system, Kroeger[**?**] introduces a extended partitioned context modeling(EPCM) to explore correlations by building access trie according to access pattern and prefetches the files with the highest weight. Gu[**?**] proposes a the weighted-graph-based grouping method to predict and prefetch the following access sequence.

These above two methods require extra space to store the correlations and the algorithm complexity is heavy. Meanwhile, based on the access pattern to exploring correlations is offline, but the correlations among files will evolve as time goes. And they only consider the access pattern and forget the intrinsic relations among files.

3 Our Approach and Novelty

4 Conclusion and Future Work