INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 9, Number 2, Pages 289–303

ADAPTIVE TRADEOFF IN METADATA-BASED SMALL FILE OPTIMIZATIONS FOR A CLUSTER FILE SYSTEM

XIUQIAO LI, BIN DONG, LIMIN XIAO, AND LI RUAN

Abstract. Metadata-based optimizations are the common methods to improve small files performance in local file systems. However, serval problems will be introduced when applying the similar optimizations for small files in cluster file systems. In this paper, we study the tradeoffs between the performance of metadata and small files in metadata-based optimizations for a cluster file system. Our method aims to guarantee the metadata performance by adaptively migrating small files among file system nodes. We establish a theory model to analyze the small files load need to be migrated. To compute the migrated load in advance, a novel forecasting method is devised to accurately predict the one-step-ahead load of metadata and small files on a MDS. Then we propose a adaptive small file threshold model to decide the small files to be migrated. In the model, we consider the long-term and short-term tradeoffs respectively. To reduce the migration overhead, we discuss the migration tradeoffs for small files and present methods and schemes to eliminate unnecessary overheads. Finally, experiments are performed on a cluster file system and the results show the efficiency of our method in terms of promoting the load forecasting accuracy, trading off the performance of metadata and small files, and reducing migration overhead.

Key words. metadata-based small files optimization, adaptive tradeoff, load forecasting, cluster file systems

1. Introduction

Recently, the small files problem in cluster file systems has aroused wide concern [1, 2, 3] in the high performance computing area. Modern cluster file systems such as PVFS2 [4] and Lustre [5] exploit a similar client/server architecture, which divides file system nodes into three roles: client, metadata server (MDS) and I/O server (IOS). Current design of cluster file systems mainly focus on optimizing large file I/O, which improve performance by distributing files among multiple IOSs to increase parallelism. However, network overheads are introduced as clients require to connect a MDS to retrieve the file layout information before transferring file data. Compared with large file accesses, small files accesses cannot benefit from the parallel I/O due to the small amount of data. According to one study [6] on access patterns in scientific computing, small file requests account for more than 90% of total requests while only contributing to less than 10% of total I/O data. Therefore, the performance of small files becomes one of the bottlenecks for cluster file systems.

In local file systems, metadata-based optimizations [2, 7] are common techniques to reduce disk accesses and improve small file I/O performance. This type of optimizations store a small file with its file metadata. Thus the file data can be fetched in a single disk access. Cluster file systems can also apply the similar

Received by the editors December 6, 2009 and, in revised form, August 7, 2010.
2000 *Mathematics Subject Classification.* 62M10.
This research was supported by the National Natural Science Foundation of China under Grant No. 60973007, the Fundamental Research Funds for the Central Universities under Grant No. YWF-10-02-05, the Doctoral Fund of Ministry of Education of China under Grant No. 20101102110018, and the fund of the State Key Laboratory of Software Development Environment under Grant No. SKLSDE-2009ZX-01.

ideas to eliminate the bottleneck of small files. Compared with local file systems, however, serval problems will be introduced when small files are stored with their file metadata on MDSs in cluster file systems.

1) MDS overload

When large amount of small files are placed on MDSs, the small files will definitely increase server load and degrade the performance of metadata. According to the studies on file system traces, metadata requests account for up to 83 percent of the total number of I/O requests in many large scale file systems [6]. Therefore, the performance of metadata cannot be guaranteed when the small files accesses overload the MDS in file system.

2) Migration overhead

Another problem introduced by metadata-based optimizations is that small files need to be migrated to IOSs as the file size is increasing. Clients need to wait for the completion of migration before performing subsequent I/O requests. Moreover, small files can be concurrently accessed by multiple clients in many workloads, such as Web applications and scientific applications. In this case, the application performance can be significantly affected by the migration overhead. To the best of our knowledge, no substantial research is conducted on this problem at this time.

In this paper, we study the adaptive tradeoff between the performance of metadata and small files in metadata-based optimizations for a cluster file system. Our method can guarantee the metadata performance when the load of small files on MDSs are heavy. The small files are dynamically migrated among MDSs and IOSs.

First, we model the load of MDSs when enabling metadata-based optimizations for small files and analyze the theoretical tradeoffs for the cases of multiple MDSs and a single MDS in a cluster file system. Second, we present a novel forecasting method to predict the one-step-ahead load of metadata and small files on a MDS. Then we propose a adaptive small file threshold model to decide the files stored on a MDS dynamically. The model considers serval factors, such as the spare storage capacity, and the load of a MDS. Third, we present the methods of selecting small files to migrate in order to reduce the small file load to guarantee the metadata performance on the MDS. Moreover, we also propose serval methods and schemes to reduce the migration overhead.

The main feature of our method is that file migration can be performed adaptively and dynamically. Therefore, the shortcomings of metadata-based optimizations can be overcome with our method. The performance of small files can be traded off without degrading the metadata performance. We evaluate our method in a well-known cluster file system PVFS2 [4] to show the merits.

The rest of this paper is presented as follows. Section 2 describes the overview and design objectives of our method. Section 3 gives the theory model analysis of tradeoffs in metadata-based optimization for small files. Section 4 presents the details of our method. Section 5 reports the results of our method with serval experiments. Related work and Concluding remarks are provided in Sections 6 and 7, respectively.

2. Method overview and design objectives

The design of PVFS2 emphasizes on improving large file I/O while little consideration is made for the performance of small files. The file metadata in PVFS2 contains two types of attributes: common attributes, file objects related attributes and extended attributes. Common attributes contains Unix-like file attributes, such as create time, file type, and credentials. The second one includes special attributes