FastDFS 下载效率的优化

张寅，林昭文，徐明昆
（北京邮电大学信息网络中心，北京 100876）

摘要：本文针对 FastDFS 系统架构可能带来的资源浪费问题进行了研究，并从分布式文件系统的下载效率方面提出了优化方法——通过多线程的的方法从每台存储数据的服务器上下载文件。实验结果表明：本文所提出的方法不仅大大提高了文件的下载效率，也使得 FastDFS 中的存储服务器的利用率大大提升，避免了服务器资源的浪费。

关键词：分布式存储；FastDFS；多线程
中图分类号：TP393

Improving downloading performance of Fast Distributed File System

ZHANG Yin, LIN Zhaowen, XU Mingkun
(Infomation and Network Center of BUPT, Beijing 100876)

Abstract: This paper analyses resource wasting which the architecture of FastDFS brings, and proposes an optimization improving FastDFS downloading perfrmance by multi-threads. The result shows not only the downloading performance is improved, but also the storage servers' use ratio is improved which avoids resource wasting.

Keywords: Distributed File System; FastDFS; Multi-threads

0 引言

FastDFS 全称为 Fast Distributed File System（Fast 分布式存储系统），是一个轻量级的开源分布式文件系统[1]。FastDFS 通过软件方式的 RAID[2]，使用廉价的 IDE 硬盘，解决了大容量的文件存储和高并发访问的问题，实现了文件存取的负载均衡，并支持存储服务器在线扩容，特别适合以文件为载体的在线服务，如相册网站、视频网站等等。

FastDFS 服务器端有 2 个角色：跟踪器（tracker）和存储节点（storage）。跟踪器主要做调度工作，在访问上起负载均衡的作用。存储节点存储文件，完成文件管理的所有功能：存储、同步和提供存取接口，同时对文件的 metadata 进行管理（所谓 metadata，就是文件的相关属性，以键值对（key-value pair）方式表示，文件 metadata 是文件属性列表，可包含多个键值对）[3]。FastDFS 不需要存储文件索引信息，所有服务器都是对等的，不存在 Master-Slave[4]关系。存储服务器采用分组方式，同组服务器上的文件完全相同（RAID 1），不同组的存储服务器之间不会相互通信，存储服务器总是向跟踪服务器报告状态信息。

如果存储集群中的某组有 2 台或者 2 台以上的服务器，由于它们存储的文件完全相同，下载文件的时候就可以从这几台服务器上分别下载文件的某一部分，这样可以大大增加 FastDFS 文件下载的效率。本文就此提出并实现一个可行的方案。

1 分析 FastDFS 文件下载的过程

1.1 FastDFS 文件上传过程简介

FastDFS 是一种基于本地文件系统的分布式文件系统，存储在 FastDFS 中的文件与 OS

作者简介：张寅，（1986-），男，硕士研究生，计算机网络。 E-mail: harry_zy@126.com
通信联系人：徐明昆，（1963-），男，高级工程师，操作系统和组件的研发。E-mail: xumk@bupt.edu.cn

文件系统中的文件一一对应，并且支持主从文件（文件 ID 有关联的文件）和保存文件属性（meta-data）。

在分析 FastDFS 文件下载过程之前，有必要对 FastDFS 文件的上传过程进行简单的介绍。图 1 展示了 FastDFS 上传文件的流程。首先，client 询问 tracker server 文件要上传到的 storage server，tracker server 会返回一个可用的 storage server 的地址，此后，client 直接和 storage server 通信，完成文件上传，并且由 storage server 返回文件 ID（文件 ID 包含了组名和文件名，如图 2 所示）。

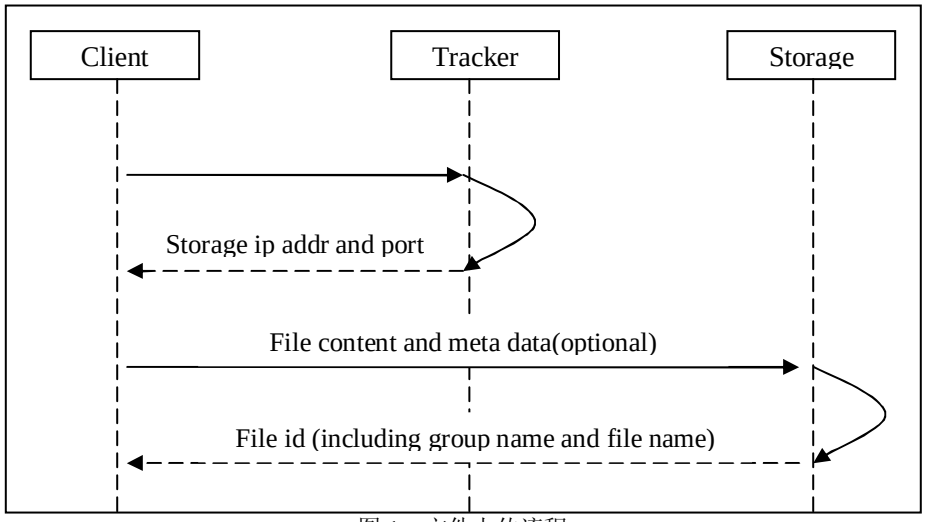

图 1　文件上传流程

Fig. 1　File Uploading Flow Chart

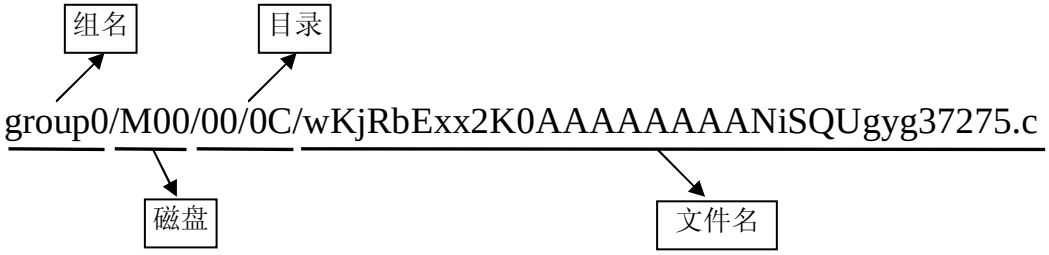

图 2　文件 ID 示例

Fig. 2　File ID Sample

1.2　FastDFS 文件下载过程分析

client 向 FastDFS 请求文件下载时，首先询问 tracker server 可以使用的 storage server，参数为文件 ID（包括组名和文件名），tracker 返回满足如下四个条件之一的 storage[5]：

1) (当前时间 -文件创建时间戳) > 同步延迟阀值（如一天）

2) 文件创建时间戳 < Storage 被同步到的时间戳

3) 文件创建时间戳 == Storage 被同步到的时间戳 且(当前时间 - 文件创建时间戳) > 文件同步最大时间（如 5 分钟）

4) 该文件上传到的源头 storage

tracker server 返回一台可用的 storage server，此后 client 直接和 storage server 通信，完成文件的下载。如图 3 所示。

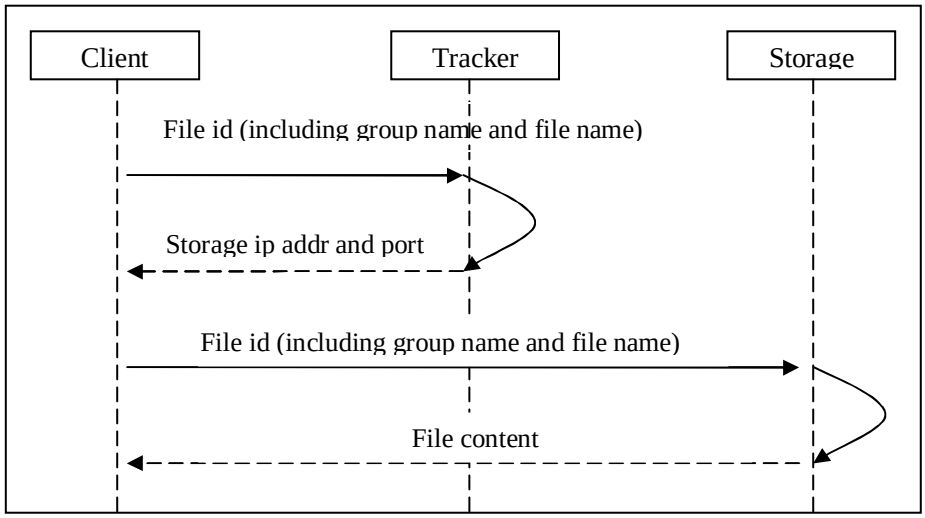

图 3　文件下载流程
Fig. 3　File Downloading Flow Chart

2　FastDFS 下载过程的优化

FastDFS 中 storage server 由若干 group 组成，每个 group 又包括至少一台的存储服务器，然而从集群中下载文件的时候是 client 直接从 group 中的一台服务器上下载。当 group 有大于 1 台服务器的时候，通过多线程下载的方式从同组的其他存储服务器存储来加快下载的速度，同时避免同组存在多台服务器所造成的资源浪费[6]。

假设要请求的文件 ID 为

group1/M00/00/02/0hmJ-U5aBkmC105tAJxAACLDpCU8206233。

1）将文件 ID 作为参数，向 tracker server 请求存储文件的 server 个数 *k*，并将存储服务器信息保存在 *server_info[]*数组中，

2）使用动态数组，创建 *k* 个文件下载线程[7]，

3）将文件名作为参数，向 storage server 请求文件的大小 *file_size*，

4）建立本地文件描述符 *fd*，大小为 *file_size*，等待写入下载文件，

5）计算文件分块 *block=file_size / k*，

6）*k* 个线程分别去 *k* 个 storage 上下载文件，第 *n* 个线程下载的文件偏移量为 *n*block*，下载的大小为 *block*，

7）每个线程下载的数据按照相同的文件偏移量和数据块大小写入描述符 *fd*，

8）线程全部结束时，程序结束。

3　实验结果与分析

3.1　实验环境及参数配置

网络拓扑结构如图 4 所示，tracker server 通过 100Mbps 的网络与 3 台 storage server 相连，这 3 台 storage server 同属一组（由配置文件设置）。

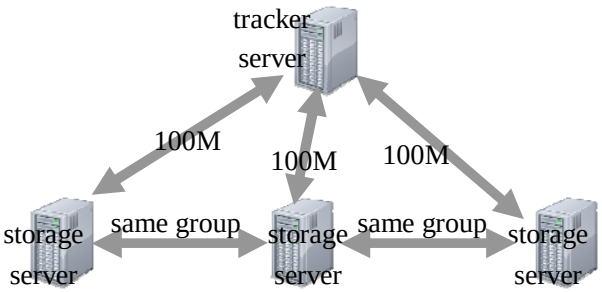

图 4　实验环境网络拓扑
Fig. 4　Network Topology of Experiment Enviroment

3.2　实验数据分析

为了验证本文对 FastDFS 下载效率的优化程度，实验分 3 种情况进行：(1)group 中只有 1 台服务器， (2)group 中有 2 台服务器，(3)group 中有 3 台服务器，分别测试 10M，30M，50M，100M，300M，500M，800M，1G 文件的下载时间，得到的结果如表 1 及图 5 所示。

表 1　测试结果
Tab. 1　Test Result

服务器个数 / 文件大小	1 台	2 台	3 台
1M	0.58s	0.81s	0.93s
5M	1.32s	1.41s	1.46s
10M	2.3s	2.1s	1.6s
30M	6.1s	5.1s	4.7s
50M	7.6s	7.2s	6.9s
100M	13.5s	12.9s	11.1s
300M	51.9s	47.7s	42.1s
500M	65.4s	60.1s	52.7s
800M	87.8s	80.1s	72.3s
1G	112.0s	99.7s	85.3s

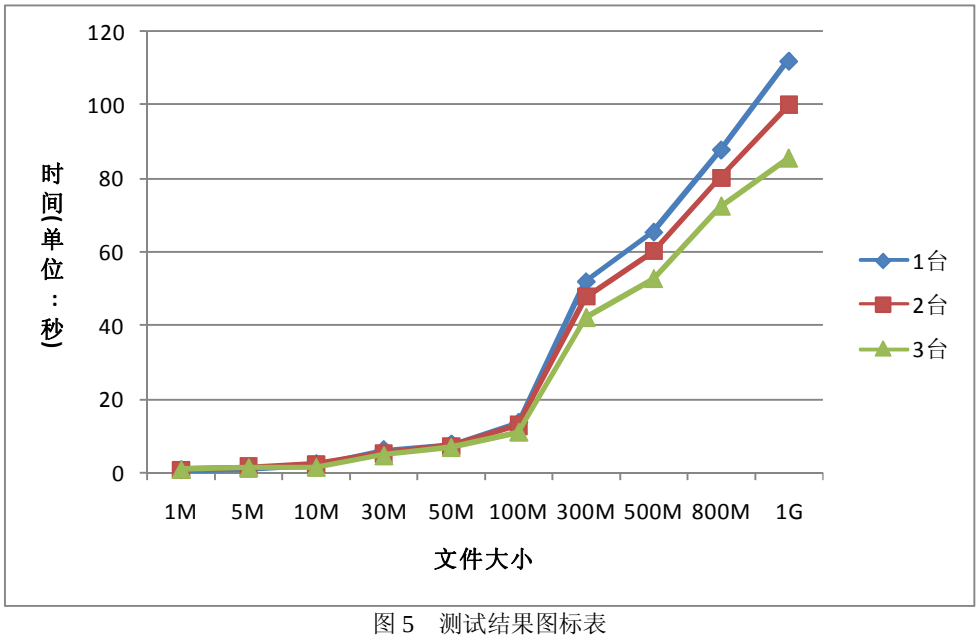

图 5　测试结果图标表
Fig. 5　Test Result Chart

通过测试结果可以看出，当文件大小在 10M 以下时，服务器个数越多下载速度反而越慢，这是因为程序创建下载线程所用的时间与下载文件所用的时间相差不多，创建下载线程拖慢了整个程序。文件大小在 10M 以上，100M 以下时，单线程与多线程下载的速度区别并不明显。当文件大小大于 100M 时，创建线程的时间与下载文件的时间相比可以忽略不计了，体现出多线程下载的明显优势。

4 结论

本文对 FastDFS 的下载过程进行了简要分析，并对同组多服务器情形下的下载效率进行了优化，利用 FastDFS 的文件存储策略，使得同组中的每台服务器都能够得到充分的利用。通过实验数据及分析，可以得出一个结论，经过本文提出的算法优化的 FastDFS 下载效率得到了明显提高。

[参考文献] (References)

[1] 余庆. FastDFS 介绍 PPT[OL]. [2009-9-13]. http://bbs.chinaunix.net/thread-1958475-1-1.html
[2] Charles M. Kozierok. RAID Level 7[OL]. [2001-4-7]. http://www.pcguide.com/ref/hdd/perf/raid/levels/singleLevel7-c.html
[3] 余庆. FastDFS 一个高效的分布式文件系统[OL]. [2009-2-18]. http://bbs.chinaunix.net/thread-2001101-1-1.html
[4] BORTHAKUR D. HDFS Architecture[OL]. [2009-9-2]. http://hadoop.apache.org/common/docs/current/hdfs_design.html.
[5] 余庆. FastDFS HOWTO — 同步机制[OL]. [2009-2-19]. http://bbs.chinaunix.net/thread-2001008-1-1.html
[6] 祖研，帅仁俊，陈平. 基于分布式文件系统的图片存储服务的研究[J]. 通信技术，2011,44（4）：138-139.
[7] 毛光喜. 多线程下载工具的开发与应用[J]. 计算机应用与软件，2006,23（7）：136-138.