文章编号: 1001 - 9081(2011) S2 - 0161 - 03

一种优化分布式文件系统的文件合并策略

陈　剑,龚发根
(广东科学职业技术学院 计算机工程技术学院,广东 珠海 519090)
(zhuhaicjf@163. com)

摘　要: 分布式文件系统的性能对整个分布式系统的性能有着重要的影响,以 Hadoop 分布式文件系统(HDFS) 为研究目标,针对 HDFS 处理小文件数据性能差的问题,分析存在的问题,提出一种新的文件合并策略,优化系统 I/O 性能。实现结果表明这种合并策略能有效提高分布式文件系统的性能。

关键词: 分布式文件系统; Hadoop 分布式文件系统; 性能优化
中图分类号: TP393　　**文献标志码:** A

File merging strategy for optimization of distributed file system

CHEN Jian, GONG Fa-gen
(*Computer Engineering Technical College*, *Guangdong Institute of Science and Technology*, *Zhuhai Guangdong* 519090, *China*)

Abstract: The performance of distributed file system has an important effect on the performance of the whole distributed system. This paper took Hadoop Distributed File System (HDFS) as the research object. In order to solve the poor performance problem of HDFS in term of processing small files, this paper analyzed the existing problem and presented a novel file merging strategy to optimize system I/O performance. The experiment results show this merging strategy can effectively improve the performance of distributed file system.

Key words: distributed file system; HDFS file system; performance optimization

0　引言

随着互联网技术的飞速发展,存储技术成为制约信息技术发展和分布式应用推广的关键问题。分布式文件系统具有较好的海量数据存储能力、较高的 I/O 吞吐量、可靠性和可扩展性而得到了学术界和工业界的关注[1]。20 世纪 80 年代出现的 NFS (Network File System) 使得分布式应用快速地应用到各个领域。目前分布文件系统在体系结构、系统规模、性能和可靠性等方面都经历了较大的变化。主流的分布式文件系统如 PVFS(Parallel Virtual File System) , Lustre, GFS(Google File System) , HDFS (Hadoop Distributed File System) 等应用到高性能计算、搜索引擎、云计算等相关研究领域。

目前分布式文件系统的研究主要集中在两个方面: 1) 文件系统性能分析和优化。这方面的研究侧重于用一些实验的方法对文件系统的性能进行测试, 分析影响性能的因素,通过量化这些性能因素来优化系统配置。如文献 [2] 提出了一种二级元数据管理策略,优化元数据的存储能力和查询效率; 文献 [3] 通过设计一系列测试方案,引用队列模型对存储系统的性能进行参数分析和优化。2) 文件系统性能建模和预测。这方面的研究主要通过分析和研究分布式文件系统中性能因素,抽象出一些性能模型,对分布式的性能进行预测,并指导系统的优化。如文献 [4] 分析了分布式文件系统潜在的性能因素,对分布式文件系统进行了系统的性能评估,挖掘出分布式/并行文件系统的性能特点; 文献 [5] 根据分布式文件系统性能的特点, 基于 Lustre 和灰色系统理论进行性能预测; 文献 [6] 根据分布式文件系统的性能评估结果,建立了一个针对分布式/并行文件系统的相对预测模型,取得了较好的预测效果。

在分布式文件系统中, HDFS 因其良好的扩展性、容错性和开源性得到业界的广泛关注。由于 HDFS 主是为搜索引擎应用而设计,因此如何有效地将 HDFS 应用到其他应用领域是一个值得研究问题。本文以 HDFS 为研究对象,针对 HDFS 在小文件数据读写方面的不足,提出了一种基于文件合并策略的优化方案,较好地解决了 HDFS 处理小文件数据时性能较差的问题。

1　HDFS 基本架构及问题分析

1.1　HDFS 基本架构

HDFS 来源于 Hadoop 项目,是一个运行在普通的硬件之上的分布式文件系统,具有高容错性。HDFS 提供高吞吐量的对应用程序数据访问,它适合大数据集的应用程序。HDFS 的设计从许多方面仿照了 GFS 的设计理念。HDFS 的基本架构如图 1 所示。

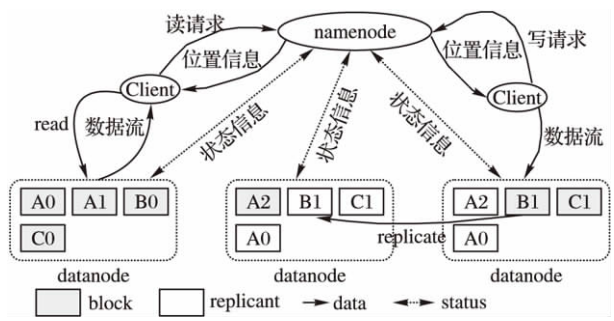

图 1　HDFS 基本架构

HDFS 采用了 Master/Slave 架构。Client 和 Namenode, Datanode 之间的通信都是建立在 TCP/IP 的基础之上的,一个 HDFS 集群一个 Namenode 和一定数目的 Datanode 和 Client 组

收稿日期: 2011 - 04 - 02; 修回日期: 2011 - 10 - 11。

作者简介: 陈剑(1965 -) ,男,江西吉安人,高级工程师,硕士,主要研究方向: 数据通信、网络安全;　龚发根(1971 -) ,男,江西吉安人,讲师,硕士,主要研究方向: 计算机网络。

成。Namenode 是一个中心服务器,负责管理文件系统的名字空间以及客户端对文件的访问。Datanode 一般是一个节点一个,负责管理它所在节点上的存储。关于 HDFS 架构更多的介绍可参考文献[7-8]。

1.2 存在问题及分析

HDFS 作为 Hadoop 项目的子项目,其设计是仿照 GFS 的设计理念。HDFS 和 GFS 的设计从一开始就是为了满足搜索引擎应用的高效而诞生的。HDFS 的设计在充分满足搜索引擎应用的同时也不可避免地造成了对其他某些应用的低效。

由于 HDFS 是为搜索引擎设计的,在搜索引擎中,通常都是涉及到数量级非常大的文件(GB 到 TB 级别),因此 HDFS 能够很好地支持大数据集的读写,提供很高的聚合带宽,但它的设计架构使得 HDFS 对于小文件的支持非常低效。主要原因有以下两方面:

1) HDFS 把整个名字空间都持久化在一个叫作 FsImage 的文件中,每次 HDFS 启动时就会将这个文件加载到内存当中。当文件数量较多时,会使名字空间迅速膨胀,引起 Namenode 内存吃紧,从而影响 Namenode 中元数据的检索效率,造成系统性能降低。

2) HDFS 中将文件划分成多个 block,存储到 Datanode 中。每一个 block 的大小都是相同的,默认为 64 MB。当存在大量小文件,尤其是存在大量的小于 64 MB 的小文件时,会形成大量的 block,即每个 block 中存储的实际文件内容很少。因此在文件读写过程中会形成大量的 I/O 操作,消耗大量数据传输时间,引起整个系统性能下降。

从上面的分析可知,由于 HDFS 本身设计的问题,形成了它在处理小文件数据方面的性能不足,而在处理大文件数据时性能较好的特点。因此,笔者通过设计一种基于文件合并策略的新方法,较好地解决了 HDFS 在处理小文件方面的不足。

2 基于文件合并策略的优化方案

2.1 设计的基本思想

基于文件合并策略的主要思想是:通过将大量的小文件输入到一个 SequenceFile 文件中,从而把大量的小文件数据变成大文件数据,减少了 Namenode 中的元数据数量,提高了元数据的检索和查询效率,降低了文件读写的 I/O 操作,节省了大量的数据传输时间。其中,SequenceFile 是 HDFS 中自带的一种文件格式,如图 2 所示。关于 SequenceFile 更多的介绍可参考文献[9]。

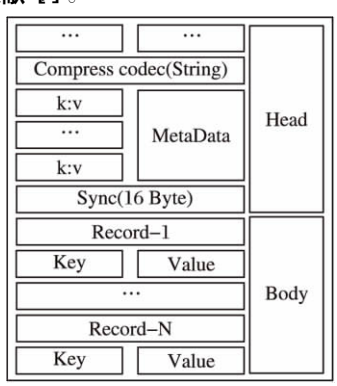

图 2 SequenceFile 格式

2.2 文件合并策略

文件合并策略通过合并大量小文件到一个大 SequenceFile 文件中,每个小文件对应 SequenceFile 中的一个记录(Record),其中将小文件的路径对应 Record 中的 Key,小文件的实际内容对应 Record 中的 Value;然后,再以 SequenceFile 为单位进行文件系统 I/O 操作。详细的文件合并流程如图 3 所示。

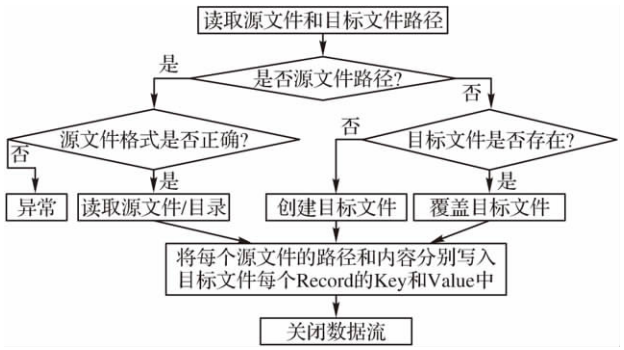

图 3 文件合并策略流程

从图 3 可以发现,文件合并策略的详细处理流程如下:

1) 获取源文件(大量小文件)和目标文件(SequenceFile 文件)的路径参数。

2) 检查源文件和目标文件路径是否正确,检查源文件是否存在,如果不存在,就抛出异常;检查目标文件是否存在,如果不存在则创建新文件,如果存在则覆盖该文件。

3) 读取文件数据,如果源文件路径是文件,则直接读取;如果源文件路径是个目录,则将源文件递归的读取。

4) 写入文件数据,调用 SequenceFile 文件的 write() 函数,将每个文件分别写入 SequenceFile 文件中的一个 Record 中,文件的路径写入 Key 中,文件内容写入 Value 中。

5) 当全部数据写完后关闭数据流。

将小文件写入 SequenceFile 大文件中后,便可以以 SequenceFile 为单位进行文件 I/O 操作,从而提高元数据检索效率,节省大量 I/O 操作和数据传输时间。

3 实验及结果分析

3.1 实验环境

为了验证优化方案,设计了一系列实验方案,详细的实验环境如表 1 所示。

表 1 实验环境配置

组件	详细描述
Hadoop 版本	0.19.1
NameNode 节点	1x Cluster Node
DataNode 节点	8x Cluster Node
操作系统	CentOS 5.3
Linux 内核	kernel 2.6.18
CPU&	双核 Intel 处理器(2.8 GHz)
内存	8 GB
网络	千兆 TCP/IP Ethernet

为了验证文件合并策略的可行性,设计一系列测试数据集,如表 2 所示。

表 2 数据集设计

数据集	小文件描述	生成方式
Dataset1	100 个大小为 30 MB 的小文件	由 dd 命令生成
Dataset2	10 000 个大小为 30 MB 的小文件	由 dd 命令生成
Dataset3	100 000 个大小为 30 MB 的小文件	由 dd 命令生成
Dataset4	1 000 000 个大小为 30 MB 的小文件	由 dd 命令生成

在表 2 中,分别设计了 Dataset1、Dataset2、Dataset3 和 Dataset4 共 4 个数据集,每个数据集分别由不同数量级的小文件组成,这些小的文本文件由 dd 命令生成,每个小文件的大小为 30 MB。

3.2　实验结果及分析

根据上述的实验数据集,对原 HDFS(简称 Ori_HDFS) 和改进后的 HDFS(简称 Imp_HDFS) 分别进行了读和写两方面的性能测试。针对读和写操作,分别在各数据集上进行了 3 次测试,取 3 次测试的平均运行时间(单位: s) 作为性能评估标准,其中读操作的性能表现如图 4 所示,写操作的性能表现如图 5 所示。

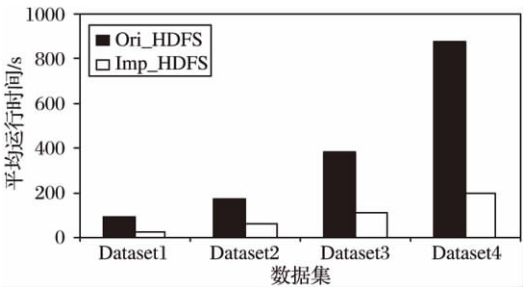

图 4　读操作的性能

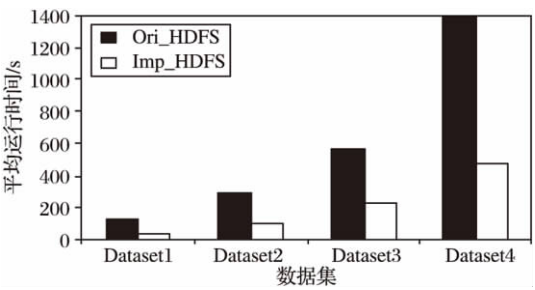

图 5　写操作的性能

由图 4 和图 5 所示,不难发现以下结论:

1) 在处理小文件数据时,改进后的 HDFS(Imp_HDFS) 在读和写操作方面的性能都要比原 HDFS(Ori_HDFS) 要好,主要原因在于,Imp_HDFS 通过采用文件合并策略,将小文件合并成大文件再进行 I/O 操作,提高了元数据的检索效率,减少了大量的 I/O 操作和数据传输时间。

2) Imp_HDFS 和 Ori_HDFS 上的读/写的平均运行时间随着小文件数据集的数据量增加(从 Dataset1 到 Dataset4,数据量失效呈数量级增长) 而呈超线性增长,主要是因为: 随着小文件数据量的快速增长,文件系统中的元数据操作和检索性能呈非线性变化,故引起读/写性能的非线性变化。

3) HDFS(包括 Ori_HDFS 和 Imp_HDFS) 的读性能均要优于写的性能,这主要和HDFS“一次写多次读”的设计模型有关; 另外,在写操作时,HDFS 要维护数据的一致性,这也会引起写操作性能较读操作性能要差。

4　结语

随着分布式应用和云计算的迅速推广,分布式文件系统的研究越来越影响互联网的发展。本文以典型的 HDFS 为研究目标,分析了 HDFS 的系统架构,搜索了 HDFS 处理小文件数据方面性能较低的原因。通过设计一个基于文件合并策略的优化方案,较好地解决了 HDFS 在处理小文件时面临的问题。实验结果表明: 在处理小文件时,改进后的 HDFS(Imp_HDFS) 读/写性能都得到了明显的提高; 同时发现 HDFS 的读/写性能随着小文件数据量的增加呈非线性变化,读的性能比写的性能要好,为其他分布式文件系统研究者提供参考。

参考文献:

[1] BOKHARI S, RUTT B, WYCKOFF P, *et al*. Experimental analysis of a mass storage system [J]. Concurrency and Computation: Practice and Experience, 2006, 18(4): 1929 – 1950.

[2] WANG FANG, YUE YINLIANG, FENF DAN, *et al*. High availability storage system based on two-level metadata management [C]// FCST 2007: Proceedings of the 2007 Japan-China Joint Workshop on Frontier of Computer Science and Technology. Piscataway, NJ: IEEE, 2007: 41 – 48.

[3] LI HUAIYANG, LIU YAN, CAO QIANG. Approximate parameters analysis of a closed fork-join queue model in an object-based storage system [C]// Proceedings of the Eighth International Symposium on Optical Storage and 2008 International Workshop on Information Data Storage, SPIE 7125. [S. l.]: SPIE, 2008: 1 – 6.

[4] ZHAO TIEZHU, VERDI M, DONG SHOUBIN, *et al*. Evaluation of a performance model of Lustre file system [C]// Proceedings of the fifth Annual ChinaGrid Conference. Piscataway, NJ: IEEE, 2010: 191 – 196.

[5] ZHAO TIEZHU, HU JINLONG. Performance evaluation of parallel file system based on Lustre and grey theory [C]// Proceedings of the 2010 Ninth International Conference on Grid and Cloud Computing. Washington, DC: IEEE Computer Society, 2010: 118 – 122.

[6] 赵铁柱,董守斌,MARCH V, 等. 面向并行文件系统的性能评估及相对预测模型[J]. 软件学报,2011,22(9): 2206 – 2221.

[7] KONSTANTIN S, HAIRONG K, SANJAY R, *et al*. The Hadoop distributed file system [C]// Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies. Piscataway, NJ: IEEE, 2010: 1 – 10.

[8] 栾亚建,黄骅民,龚高晟,等. Hadoop 平台的性能优化研究[J]. 计算机工程,2010,36(14): 262 – 266.

[9] Apache Hadoop Project. SequenceFile Class [EB/OL]. [2011 – 02 – 17]. http://hadoop.apache.org/common/docs/current/api/org/apache/hadoop/io/SequenceFile.html.

(上接第 158 页)

4　结语

本文设计并实现了 ARM-Linux 下的旋转编码器接口电路和驱动程序,以 E6B2-CWZ6C 编码器为例,实现了在 ARM-Linux 平台下对旋转编码器加减计数、清零、预置计数值功能,在实际应用中该计数器运行良好,能满足实际需要,达到了预期目的,并已应用于材料冲击试验机的摆锤角度采集中,希望此方案能给其他同类系统设计提供有益的参考。

参考文献:

[1] 石奋苏. 自动冲击试验机计算机测控系统设计[J]. 宁夏工程技术,2005,4(2): 182 – 184.

[2] 徐海,胡荣贵,张东. 基于单片机的旋转编码器鉴相方法研究[J]. 微型机与应用,2010(13): 20 – 29.

[3] Atmel Corporation. AT91RM9200 Datasheet[EB/OL]. [2010 – 09 – 30]. http://www.atmel.com.

[4] 欧姆龙(中国) 有限公司. E6B2 旋转编码器使用说明书[EB/OL]. [2010 – 09 – 01]. www.omron.com.cn.

[5] 潘明东. 光电编码器输出脉冲的几种计数方法[J]. 电子工程师,2004,30(8): 69 – 71.

[6] 孙祥明,齐明侠,沈蓉. 编码器换向误码输出原因探讨及鉴相电路改进[J]. 石油大学学报: 自然科学版,2005,29(4): 91 – 94.

[7] 魏永明,耿 岳,钟书毅. Linux 设备驱动程序[M]. 3 版. 北京: 中国电力出版社,2005.

[8] 宋宝华. Linux 设备驱动开发详解[M]. 北京: 人民邮电出版社,2008.