分布式文件系统

1.概念

相对于本机端的文件系统而言，分布式文件系统（英语：Distributed file system, DFS），或是网络文件系统（英语：Network File System），是一种允许文件透过网络在多台主机上分享的文件系统，可让多机器上的多用户分享文件和存储空间。

在这样的文件系统中，客户端并非直接访问底层的数据存储区块，而是透过网络，以特定的通信协议和服务器沟通。借由通信协议的设计，可以让客户端和服务端都能根据访问控制清单或是授权，来限制对于文件系统的访问。

相对地，在一个分享的磁盘文件系统中，所有节点对数据存储区块都有相同的访问权，在这样的系统中，访问权限就必须由客户端程序来控制。

分布式文件系统可能包含的功能有：透通的数据复制与容错。也就是说，即使系统中有一小部分的节点离线，整体来说系统仍然可以持续运作而不会有数据损失。

2.原因

由Computer World报道，从1986年到2007年，人均数据量以每年23%的速度在不断上升。实际上，我们每天都有大量的数据在不断产生，使得这个数据在不断的增加。

然而，无论是从经济角度还是技术角度而言，仅仅通过传统的存储技术是难以存储下如此庞大的数据——传统的技术不够灵活，不够迅捷，且造价昂贵。所以我们需要新的概念来解决这个问题。这个概念是：

“将数据存储在不同的服务器上，但它们却仿佛在一个相同存储系统之中。”

于是，我们提出了分布式文件系统的概念。

3.特点

分布式系统有如下特点：

灵活：从用户角度而言，其访问服务器的过程和访问本地磁盘并无大的差异。但由于数据存储于服务器中，而本机是通过网络去访问服务器从而取得数据，所以用户无需自己增加硬盘，而仅仅添加更多的服务器和节点就可以增加存储容量。

迅速：在分布式系统中，由于每个服务器拥有其自身的CPU，RAM，驱动和网络接口。所以在增加服务器时，势必可以增加整个用户系统的速度和存储能力。

费用低：据调查，使用了分布式文件系统的公司或组织，其基础设施的搭建费用少了仅90%。其原因在于公司仅仅是使用服务器，驱动和网络，使得整体的费用变得便宜。实际上，每在服务器上花费1元，其效果等同于在传统存储上花费5元。

基于上述特点，有报告称，截至2018年，有63%的公司和组织已经决定使用分布式文件系统。

4.发展

1.GFS

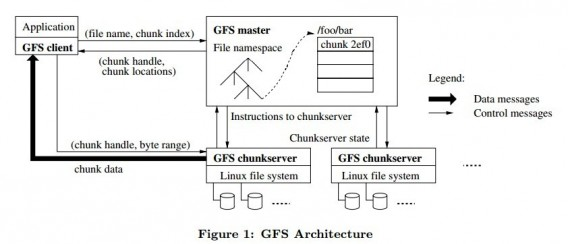
GFS系统由三部分组成：GFS master、GFS Client、GFS chunkserver。其中，GFS master任意时刻只有一个，而chunkserver和gfs client可能有多个。

一份文件被分为多个固定大小的chunk（默认64M），每个chunk有全局唯一的文件句柄——一个64位的chunk ID，每一份chunk会被复制到多个chunkserver，以此保证可用性与可靠性。chunkserver将chunk当做普通的Linux文件存储在本地磁盘上。

　　GFS master是系统的元数据服务器，维护的元数据包括：命令空间（GFS按层级目录管理文件）、文件到chunk的映射，chunk的位置。其中，前两者是会持久化的，而chunk的位置信息来自于Chunkserver的汇报。

　　GFS master还负责分布式系统的集中调度：chunk lease管理，垃圾回收，chunk迁移等重要的系统控制。master与chunkserver保持常规的心跳，以确定chunkserver的状态。

GFS client是给应用使用的API，这些API接口与POSIX API类似。GFS Client会缓存从GFS master读取的chunk信息（即元数据），尽量减少与GFS master的交互。

 2.HDFS

HDFS 与其他分布式文件系统有许多相似点，但也有几个不同点。一个明显的区别是 HDFS 的 “一次写入、多次读取（write-once-read-many）” 模型，该模型降低了并发性控制要求，简化了数据聚合性，支持高吞吐量访问。

HDFS 的另一个独特的特性是下面这个观点：将处理逻辑放置到数据附近通常比将数据移向应用程序空间更好。

HDFS 将数据写入严格限制为一次一个写入程序。字节总是被附加到一个流的末尾，字节流总是以写入顺序存储。

HDFS 有许多目标，下面是一些最明显的目标：

通过检测故障和应用快速、自动的恢复实现容错性

通过 MapReduce 流进行数据访问

简单可靠的聚合模型

处理逻辑接近数据，而不是数据接近处理逻辑

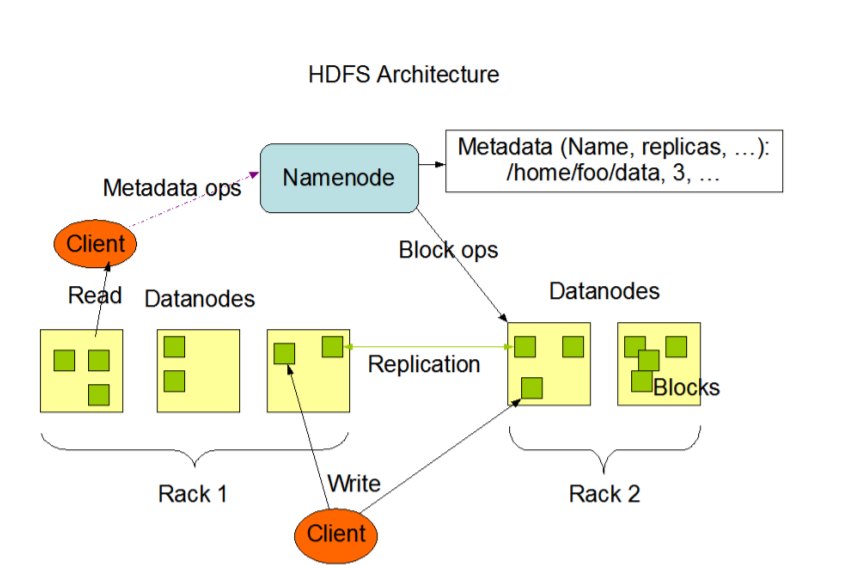
跨异构普通硬件和操作系统的可移植性

可靠存储和处理大量数据的可伸缩性

通过跨多个普通个人计算机集群分布数据和处理来节约成本

通过分布数据和逻辑到数据所在的多个节点上进行平行处理来提高效率

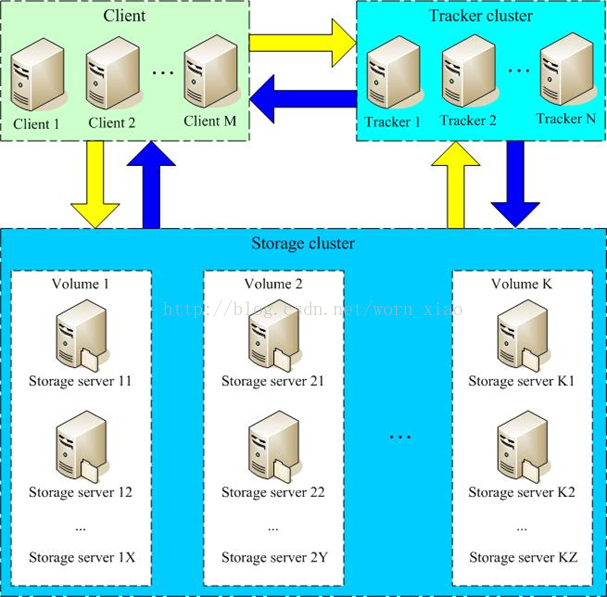
通过自动维护多个数据副本和在故障发生时自动重新部署处理逻辑来实现可靠性



3.FastDFS

FastDFS是一个开源的轻量级分布式文件系统，它对文件进行管理，功能包括：文件存储、文件同步、文件访问（文件上传、文件下载）等，解决了大容量存储和负载均衡的问题。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是一个开源的轻量级分布式文件系统,由跟踪服务器（tracker server）存储服务器（storage server）客户端（client）



问题分析：

    从FastDFS的整个设计看，基本上都已简单为原则。比如以机器为单位备份数据，简化了tracker的管理工作；storage直接借助本地文件系统原样存储文件，简化了storage的管理工作；文件写单份到storage即为成功、然后后台同步，简化了写文件流程。但简单的方案能解决的问题通常也有限，FastDFS目前尚存在如下问题:

数据安全性：

    写一份即成功：从源storage写完文件至同步到组内其他storage的时间窗口内，一旦源storage出现故障，就可能导致用户数据丢失，而数据的丢失对存储系统来说通常是不可接受的。

   缺乏自动化恢复机制：当storage的某块磁盘故障时，只能换存磁盘，然后手动恢复数据；由于按机器备份，似乎也不可能有自动化恢复机制，除非有预先准备好的热备磁盘，缺乏自动化恢复机制会增加系统运维工作。

   数据恢复效率低：恢复数据时，只能从group内其他的storage读取，同时由于小文件的访问效率本身较低，按文件恢复的效率也会很低，低的恢复效率也就意味着数据处于不安全状态的时间更长。

   缺乏多机房容灾支持：目前要做多机房容灾，只能额外使用工具来将数据同步到备份的集群，无自动化机制。

存储空间利用率：

    单机存储的文件数受限于inode数量

   每个文件对应一个storage本地文件系统的文件，平均每个文件会存在block\_size/2的存储空间浪费。

文件合并存储能有效解决上述两个问题，但由于合并存储没有空间回收机制，删除文件的空间不保证一定能复用，也存在空间浪费的问题

负载均衡：

    group机制本身可用来做负载均衡，但这只是一种静态的负载均衡机制，需要预先知道应用的访问特性；同时group机制也导致不可能在group之间迁移数据来做动态负载均衡

5.不足

低延时访问

分布式文件系统不太适合于那些要求低延时（数十毫秒）访问的应用程序，因为分布式文件系统是设计用于海量数据处理的，这是以一定延时为代价的。对于低延时访问，经典和传统的办法就是数据库。

频繁修改的文件的应用

       目前常用的分布式文件系统，基本都是“一次写多次读”的模式，如果涉及到大量数据的频繁修改，那么这个问题就相对比较麻烦；

海量小文件

分布式文件系统把文件系统的元数据放置在内存中，所以文件系统所能容纳的文件数目是有限的。一般来说，每一个文件、文件夹和Block需要占据150字节左右的空间，所以，如果你有100万个文件，每一个占据一个Block，你就至少需要300MB内存。当前来说，数百万的文件还是可行的，当扩展到数十亿时，对于当前的硬件水平来说就很痛苦了。因此由于海量元数据的因素，分布式文件系统对待海量小文件相对比较乏力。

6.立项依据（如果要做这方面，但建议不要，因为有别的组做ceph的分布式）

1.用FastDFS搭建分布式文件系统，做到能在不同机器之间传输数据（google学术上有相关运用FastDFS搭建分布式的例子）

2.对FastDFS经行优化或改造

FastDFS源码由C语言编写，同时FastDFS有如上所说的问题。我们可以对其进行优化，比如提升文件存储率，或者对其负载算法经行优化，解决其负载不均衡的问题。

参考文献

Ghemawat S, Gobioff H, Leung S T. The Google file system[J]. 2003.

Borthakur D. HDFS architecture guide[J]. Hadoop Apache Project, 2008, 53: 1-13.