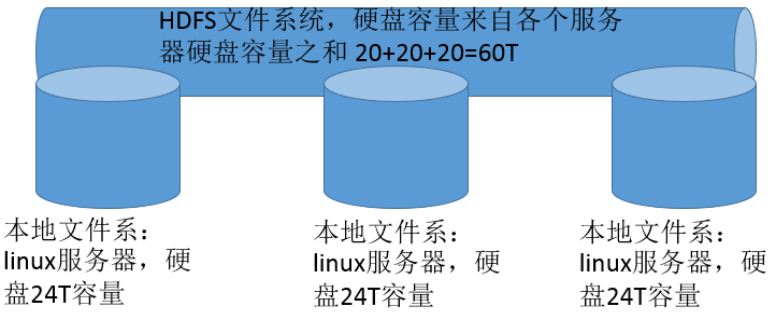
1、分布式文件系统详细介绍

在 hadoop 当中，分布式文件系统（HDFS），对文件系统有一个抽象，HDFS 属于当中的一个实现类，也就是说分布式文件系统类似于一个接口，定义了标准，下面有很多的实现类，其中 HDFS 是一个子实现类而已，但是现在很多人都只知道一种就是 HDFS 的实现，并没有了解过其他的实现类，其实分布式文件系统的实现有很多种，

具体详细参见 hadoop 权威指南第三版第 59 页



3.4 Hadoop 文件系统

Hadoop 有一个抽象的文件系统概念，HDFS 只是其中的一个实现。Java 抽象类 `org.apache.hadoop.fs.FileSystem` 定义了 Hadoop 中的一个文件系统接口，并且该抽象类有几个具体实现，如表 3-1 所示。

表 3-1. Hadoop 文件系统

文件系统	URI 方案	Java 实现(均包含在 org.apache.hadoop 包中)	描述
Local	file	fs.LocalFileSystem	使用了客户端校验和的本地磁盘文件系统。没有使用校验和的本地磁盘文件系统 RawLocalFileSystem。详情参见 4.1.2 节
HDFS	hdfs	hdfs.DistributedFileSystem	Hadoop 的分布式文件系统。将 HDFS 设计成与 MapReduce 结合使用，可以实现高性能
HFTP	Hftp	hdfs.hftpFileSystem	一个在 HTTP 上提供对 HDFS 只读访问的文件系统(尽管名称为 HFTP，但与 FTP 无关)。通常与 distcp 结合使用(参见 3.8 节)，以实现在运行不同版本的 HDFS 的集群之间复制数据
HSFTP	hsftp	hdfs.HsftpFileSyste	在 HTTPS 上提供对 HDFS 只读访问的文件系统(同上，与 FTP 无关)
WebHDFS	Webhdfs	Hdfs.web.WebHdfsFileSystem	基于 HTTP，对 HDFS 提供安全读写访问的文件系统。WebHDFS 是为了替代 HFTP 和 HSFTP 而构建的
HAR	har	fs.HarFileSystem	一个构建在其他文件系统之上用于文件存档的文件系统。Hadoop 存档文件系统通常用于需要将 HDFS 中的文件进行存档时，以减少 namenode 内存的使用。参见 3.9 节
hfs (云存储)	kfs	fs.kfs.kosmosFileSystem	CloudStore(其前身为 Kosmos 文件系统)是类似于 HDFS 或是谷歌的 GFS 的文件系统，用 C++写。详情参见 *http://kosmosfs.sourceforge.net/*
FTP	ftp	fs.ftp.FTPFileSystem	由 FTP 服务器支持的文件系统
S3 (原生)	S3n	fs.s3native.NativeS3FileSystem	由 Amazon S3 支持的文件系统。参见 *http://wiki.apache.org/hadoop/AmazonS3*
S3 (基于块)	S3	fs.sa.S3FileSystem	由 Amazon S3 支持的文件系统，以块格式存储文件(与 HDFS 很相似)以解决 S3 的 5 GB 文件大小限制

续表

文件系统	URI 方案	Java 实现(均包含在 org.apache.hadoop 包中)	描述
分布式 RAID	hdfs	hdfs.DistributedRaidFileSystem	RAID 版本的 HDFS 是为了存档而设计的。针对 HDFS 中的每个文件，创建一个(更小的)校验文件，并允许 HDFS 中的数据副本由 3 降为 2，由此可以减少 25%～30% 的存储空间，但是数据丢失的概率保持不变。分布式 RAID 模式需要在集群中运行一个 RaidNode 后台进程
View	viewfs	viewfs.ViewFileSystem	针对其他 Hadoop 文件系统挂载的客户端表。通常用于联邦 namenode 创建挂载点。详情参见 3.2.3 节。

Hadoop 对文件系统提供了许多接口，它一般使用 URI 方案来选取合适的文件系统实例进行交互。举例来说，我们在前一小节中遇到的文件系统命令行解释器可以操作所有的 Hadoop 文件系统命令。要想列出本地文件系统根目录下的文件，可以输入以下命令：

```
% hadoop fs -ls file:///
```

其中我们课程当中重点突出讲解 HDFS 这种文件系统