|  |  |  |  |
| --- | --- | --- | --- |
| 技术实现 | 概述 | 长处、适用 | 短处、不适用 |
| Lustre | * Lustre是一个面向大规模集群的[存储](http://storage.chinabyte.com/" \t "_blank)架构，基于[Linux](http://www.chinabyte.com/keyword/Linux/)平台的开源集群并行文件系统，符合GNU GPL v2协议，其最大特征是[高扩展](http://www.chinabyte.com/keyword/%E9%AB%98%E6%89%A9%E5%B1%95/" \t "_blank)性和高性能，能够支持数万客户端系统、PB级存储容量、GB/s级别的I/O吞吐量。 * **全球顶级超级计算机系统中有大量都使用了Lustre，百强中份额超过40％，前三十名中更是占据了一半** * **通用分布式文件系统** | * Lustre的集群和并行架构，非常适合众多客户端并发进行大文件读写的场合， * 整合磁盘空间，提供高的I/O等 * 应用端可以mount使用 | * 目前对于小文件应用非常不适用，尤其是海量小文件应用LOSF（Lots Of Small Files）。 * (看lustre 700多页的pdf文档，让人头昏)。 * **安装Lustre需要内核的配合，需要重新编译内核，工作量巨大。** * Lustre并没有对存储设备进行多个拷贝，Failover也仅仅实现的是服务器节点的冗余，如果磁盘发生损坏，整个Lustre空间将损坏，官方manual也声明，Lustre的存储必须基于Raid阵列等硬件冗余才能保证数据的完整。 |
| MogileFS | * MogileFS 是一个开源的分布式文件系统，用于组建分布式文件集群 | * 无单点失败 — MogileFS分布式文件存储系统安装的三个组件（存储节点、跟踪器、跟踪用的数据库），均可运行在多个 机器上，因此没有单点失败。 * 自动的文件复制 | * 不适合做通用文件系统，适合存储静态只读小文件，比如图片 * 文件上传后不能修改。不能mount使用，需要使用专有API对文件进行访问 |
| MooseFS | * mooseFS是一款网络分布式文件系统。它把数据分散在多台服务器上，但对于用户来讲，看到的只是一个源。MFS也像其他类unix文件系统一样，包含了层级结构（目录树），存储着文件属性（权限，最后访问和修改时间），可以创建特殊的文件（块设备，字符设备，管道，套接字），符号链接，硬链接。 * **通用分布式文件系统** | * 通用文件系统，不需要修改上层应用就可以使用（那些需要专门api的dfs很麻烦！）应用端可以mount使用。 * 部署简单。 * 高可用，可设置任意的文件冗余程度（提供比raid1+0更高的冗余级别，而绝对不会影响读或者写的性能，只会加速！） | * - master server 存在单点故障 * - master server 很耗内存(短期对策：按业务切分) * MooseFS本地不会缓存Chunk信息， 每次读写操作都会访问Master， Master的压力较大。 * **用perl编写，性能相对较差** |
| FastDFS | * FastDFS是一个开源的轻量级分布式文件系统。它解决了大数据量存储和负载均衡等问题。特别适合以中小文件（建议范围：4KB < file\_size <500MB）为载体的在线服务，如相册网站、视频网站等等。 * **FastDFS设计时借鉴了MogileFS的一些思路。**FastDFS是一个完善的分布式文件存储系统，通过客户端API对文件进行读写。可以说，MogileFS的所有功能特性FastDFS都具备 * GoogleFS以及FastDFS、mogileFS、HDFS、TFS等类GoogleFS都不是系统级的分布式文件系统，而是应用级的分布式文件存储服务。 * fastDFS：国人在mogileFS的基础上进行改进的key-value型文件系统，同样不支持FUSE，提供比mogileFS更好的性能。 | * 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。 * 在UC基于FastDFS开发向用户提供了：网盘，社区，广告和应用下载等业务的存储服务。 * FastDFS不会对文件进行分块存储，客户端上传的文件和Storage server上的文件一一对应。 * **FastDFS最大的亮点就是对小文件的存储性能较好，这主要来自于其文件名策略。** * FastDFS只有三个角色；且跟踪服务器和存储服务器均不存在单点。 | * - 只能通过 API 使用，不支持 fuse（用户空间文件系统Filesystem in Userspace，简称FUSE） * FastDFS没有对文件做分块存储，因此**不太适合分布式计算场景。** |

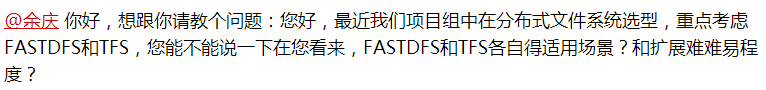
|  |  |  |
| --- | --- | --- |
| **指标** | **通用分布式**  **文件系统** | **专用分布式**  **文件系统** |
| 开发者友好性 | 较好 | 较差 |
| 系统复杂性 | 较高 | 较低 |
| 系统性能 | 一般 | 较高 |

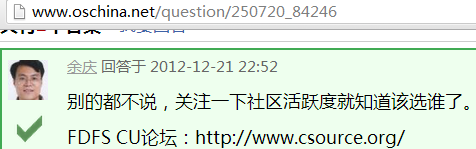
FastDFS

众所周知，大多数网站都需要存储用户上传的文件，如图片、视频、电子文档等。出于降低带宽和存储成本的考虑，网站通常都会限制用户上传的文件大小，例如图片文件不能超过5MB、视频文件不能超过100MB等。我认为，对于互联网应用，文件分块存储没有多大的必要。它既没有带来多大的好处，又增加了系统的复杂性。FastDFS不对文件进行分块存储，与支持文件分块存储的DFS相比，更加简洁高效，并且完全能满足绝大多数互联网应用的实际需要。

已知使用FastDFS的用户

* 某大型网盘（公司名对方要求保密）
* UC（<http://www.uc.cn/>）
* 支付宝（<http://www.alipay.com/>）
* 京东商城（<http://www.360buy.com/>）
* 淘淘搜（<http://www.taotaosou.com/>）
* 飞信（<http://feixin.10086.cn/>）
* 赶集网（<http://www.ganji.com/>）
* 淘米网（<http://www.61.com/>）
* 迅雷（<http://www.xunlei.com/>）
* 蚂蜂窝（<http://www.mafengwo.cn/>）
* 丫丫网（<http://www.iyaya.com/>）
* 虹网（<http://3g.ahong.com/>）
* 5173（<http://www.5173.com/>）
* 华夏原创网（http://www.yuanchuang.com/）
* 华师京城教育云平台（http://www.hsjdy.com.cn/）
* 视友网（<http://www.cuctv.com/>）
* 。。。





<http://code.taobao.org/p/tfs/wiki/index/>

Q: 使用TFS一定需要64bit Liunx？

A: 是的，否则整个项目不能正常编译通过。

TFS开源用户大都只使用TFS的基本功能：nameserver，dataserver，client和tool

metaserver是我们在2.0版本引进的一个服务. 用来存储一些元数据信息, 这样原本不支持自定义文件名的 TFS 就可以在 metaserver 的帮助下, 支持自定义文件名了.

客户端在做自定义文件名的读操作的时候, 会先从 rootserver 得到关于 metaserver 的信息, 并缓存在自己的内存中. 然后用自定义文件名去 metaserver 中查找 TFS 文件名信息, 再去 TFS 中访问该文件. 客户端在做自定义文件名的写操作的时候, 会先写入到 TFS 中, 再把 TFS 文件名和自定义文件的对应关系写入metaserver中.

TFS 在2.0版本增加了一个server, 叫做 rcserver. 这个 server 主要是为了淘宝内部管理使用 TFS 的各个应用. 我们给每个应用分配一个唯一的 AppKey. TFS 客户端使用这个 AppKey 登录到 rcserver, 取得自己应该访问的 TFS 集群信息. 客户端还会定期把自己的一些统计值发送给 rcserver. 具体信息可以参看源码中 doc 目录下的关于 rcserve 的文档.