FastDFS 可行性报告

可行性分析：理论依据、技术依据、创新点

理论依据：

1. FastDFS ： FastDFS是一个开源的高性能分布式文件系统（DFS）。 它的主要功能包括：文件存储，文件同步和文件访问，以及高容量和负载平衡。它由跟踪服务器(Tracker Server)、存储服务器(Storage Server)、客户端(Client)组成。其中，Track Server主要做调度工作。起到均衡作用，负责管理所有的storage server和group，每个storage在启动后会连接Track Server并告知自己所属group等存储信息。Track Server是FastDFS的核心组成。由于FastDFS 开源性质，我们可以通过修改其源代码来改善FastDFS的功能。
2. 去中心化：去中心化是将核心结点的功能分散到不同部位来实现分担核心结点的负担并增强系统的稳定性。

由于Tracker Server集中处理了用户的请求和调度工作，若Tracker Server受到攻击或者出现故障，系统的稳定性将大大降低，因此可以采用去中心化的方法来修改Tracker。

在一般情况下，分布式存储系统是将分组消息形成的向量集中进行编码处 理的。为了实现去中心化，思想是使用分布式的纠删码原理实现分布式编码。将文 件信息通过分布式编码分散到不同服务器处理。

1. 动态负载：为了使分布式文件系统的多个结点很好的共同完成任务，消除或避免现有网络负载分布不均、数据流量拥挤、反应时间长的平静，人们提出了多种负载聚恒算法，主要分为两大类：静态负载均衡算法和动态均衡算法。其中，FastDFS采用了静态均衡算法中的优先权：当最高优先级中所有服务器出现故障才将请求发送给次优先级的服务器组。实际上它为用户提供了一种热备份的方式。

虽然FastDFS的静态均衡算法实现简单，但是没有考虑到各节点的实际负载情况，从而不能很好地实现负载均衡。而动态均衡算法引进了一种动态反馈机制，各服务器节点向负载均衡节点周期地反馈其实时负载情况，负载均衡节点根据此信息分配客户请求。

技术依据：

1. FastDFS：FastDFS源代码可在其开发者余庆<https://github.com/happyfish100/fastdfs>

的github里下载查看。 FastDFS的安装和常见问题可以在oschina.net以及csdn、github、stackoverflow等论坛得到解决。

2．去中心化：实现去中心化可以采用DHT(Distributed Hash Table)模型。它是分布式计 算系统的一类，用来将一个关键值(key)的集合分散到所有在分散式系统中的节点， 并且可以有效地将消息转送到唯一一个拥有查询这提供的关键值的节点(Peers)。

DHT强调以下特性：

， 离散性：构成系统的节点并没有任何中央式的协调机制。

伸缩性：即使有成千上万个节点，系统仍应该十分有效率。

容错性：即使节点不断加入、离开或停止工作，系统仍然必须达到一 定的可靠度。

DHT主要利用一致性hash，把节点和资源都表示成一个hash值，放入到 大hash环中，每个节点负责路由靠近它的资源。

具体的DHT实现有余庆先生的FastDHT。

典型的DHT实现算法：Chord算法、KAD算法等。

对于实现去中心化的分布式文件系统来说，我们可以参考一下Leviathans 所采用的Gossip算法。（详见：<https://github.com/abbshr/Leviathan>）

3．动态负载：动态负载均衡算法包括: 最少连接数,最快响应速度，观察方法，预测法， 动态性能分配，动态服务器补充，服务质量，服务类型，规则模式。

1.最少连接方式(Least Connection)：传递新的连接给那些进行最少连接处理的服 务器。当其中某个服务器发生第二到第7 层的故障，BIG-IP 就把其从服务器 队列中拿出，不参加下一次的用户请求的分配, 直到其恢复正常。

2.最快模式（Fastest）：传递连接给那些响应最快的服务器。当其中某个服务器发 生第二到第7 层的故障，BIG-IP 就把其从服务器队列中拿出，不参加下一次的用 户请求的分配，直到其恢复正常。

3.观察模式（Observed）：连接数目和响应时间以这两项的最佳平衡为依据为新的 请求选择服务器。当其中某个服务器发生第二到第7 层的故障，BIG-IP就把其从 服务器队列中拿出，不参加下一次的用户请求的分配，直到其恢复正常。

4.预测模式（Predictive）：BIG-IP利用收集到的服务器当前的性能指标，进行预测 分析，选择一台服务器在下一个时间片内，其性能将达到最佳的服务器相应用户的 请求。(被BIG-IP 进行检测)

5.动态性能分配(Dynamic Ratio-APM):BIG-IP 收集到的应用程序和应用服务器的 各项性能参数，动态调整流量分配。

6.动态服务器补充(Dynamic Server Act.):当主服务器群中因故障导致数量减少时， 动态地将备份服务器补充至主服务器群。

7.服务质量(QoS）:按不同的优先级对数据流进行分配。

8.服务类型(ToS): 按不同的服务类型（在Type of Field中标识）负载均衡对数据流 进行分配。

9.规则模式：针对不同的数据流设置导向规则，用户可自行。

动态负载均衡的相关工作：

1.《Fast DFS 分布式文件系统负载均衡算法的改进研究》 熊建波， 武汉理工大学

2.《基于 FastDFS 的目录文件系统的研究与实现》 周子涵， 电子科技大学

3.《基于 FastDFS 架构的小文件存储系统的设计与实现》 李长平 哈尔滨工业大学

4.《FastDFS 负载均衡算法的改进及其在水土保持网站系统的应用》 曾致远 华中科 技大学

5.《分布式文件系统的动态负载均衡算法》 张聪萍，尹建伟 浙江大学

创新点：

选用FastDFS一方面原因是很多企业都采用或改用了FastDFS，它对于小文件的存储管理尤其适用，适合存储4KB~500MB的小文件，如图片网站、短视频网站、文档、app下载站等，UC、京东、支付宝、迅雷、酷狗等都有使用；另一方面是它是由C语言实现的开源轻量级系统，支持Linux、FreeBSD、AID等Unix系统，便于扩展功能和移植。

从增强安全性和系统稳定性方面来说，我们选择将之前的Tracker Server的中心化处理改为去中心化；从增强性能来说，我们将采用动态负载的算法来优化FastDFS 的负载均衡。这样可以在现有的基础上“看得见地”提高FastDFS的效率并应用到存储中小文件的系统中。