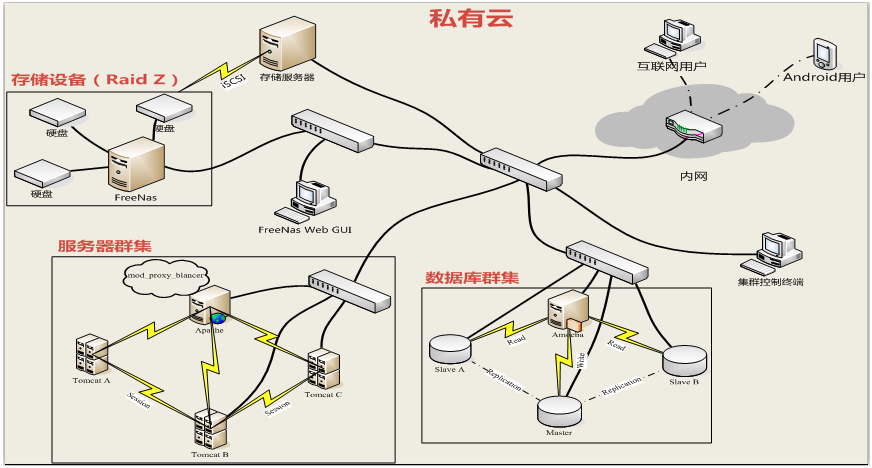
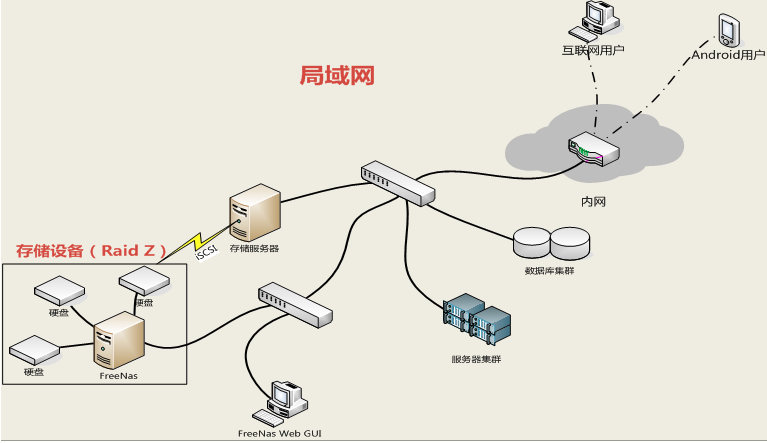
**"云盘"设计方案**

总方案



主要组成部分：   
1、存储(IP SAN+RAID-Z)   
2、数据库集群(主从复制+读写分离)   
3、服务器集群(Apache+Tomcat)   
4、多客户端(Web端+Android端)

存储



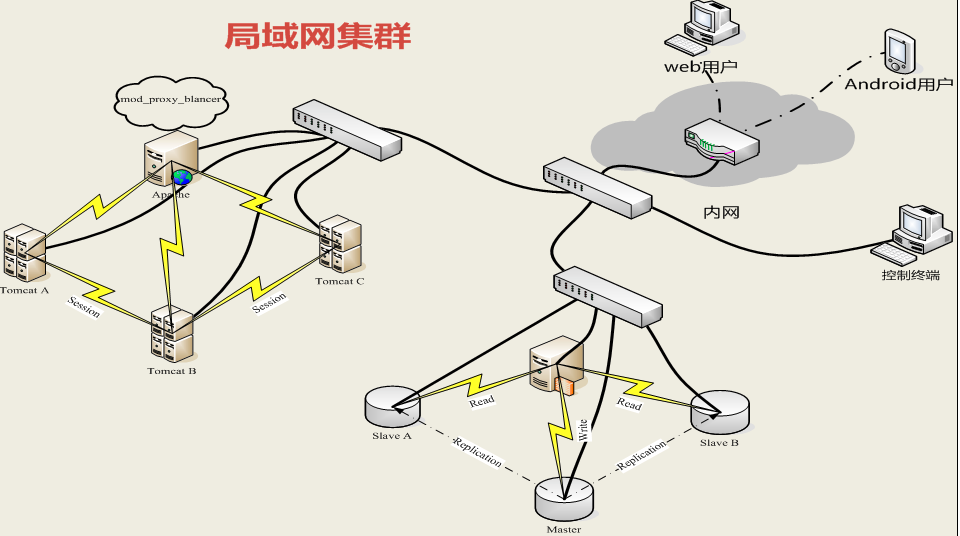
利用FreeNAS及PC实现IP SAN

FreeNAS是一套免费的NAS服务器，它能将一部普通PC变成网络存储服务器。该软件基于FreeBSD，Samba 及PHP，支持CIFS (samba), FTP, NFS protocols, Software RAID (0,1,5) 及 web 界面的设定工具。用户可通过Windows、Macs、FTP、SSH 及网络文件系统 (NFS) 来访问存储服务器；FreeNAS可被安装于硬盘或移动介质USB Flash Disk上，所占空间不足 16MB。



在FreeNas中将三块硬盘组成RAID-Z后，在服务选项中激活iSCSI服务，即可在 Windows中使用iSCSI发起程序挂载磁盘阵列。

集群



数据库集群

一台MySQL服务器是很难做到高可用（HA），如果没有及时做数据备份，一旦遇到故障宕机后将无法继续提供后端数据业务。单一的数据库节点对于高并发、高负载场景呈现出各种瓶颈问题也越来越突出。

数据库集群技术通过水平扩展的方式将负载分散到多个节点上以解决一个数据库节点的不足。

本方案基于MySQL数据库主从复制及基于Amoeba For MySQL读写分离，利用Amoeba将请求分发到不同的MySql服务器，以达到使我们项目在原有的基础上具有更快的读写速度，给用户更快的响应，更高的负载。

服务器集群

Tomcat集群负载均衡有三种方式:mod\_proxy、mod\_proxy\_blancer、mod\_jk

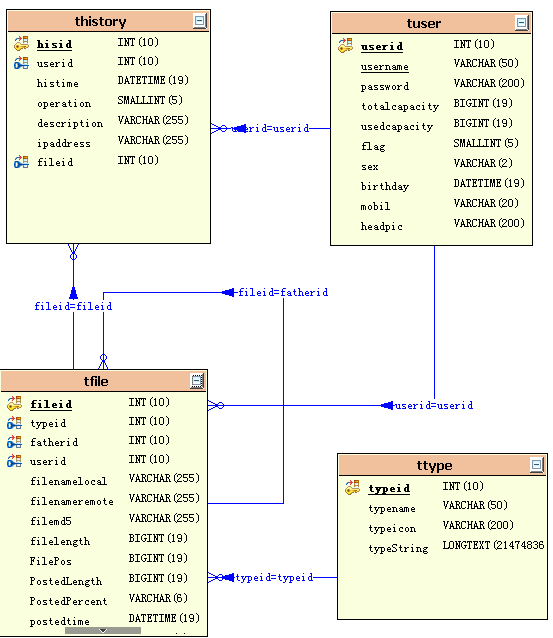
本方案采用mod\_proxy\_balancer，它是mod\_proxy的扩展，提供负载平衡支持，通过mod\_proxy\_balancer.so包实现负载平衡。

采用此方式，当用户请求发布的web项目时，apache服务器会根据我们的配置，将请求分发给集群中的不同的Tomcat服务器，实现服务器的负载均衡，相对于一个服务器来说，这种集群可以响应更多的用户请求而一直保持活跃状态。

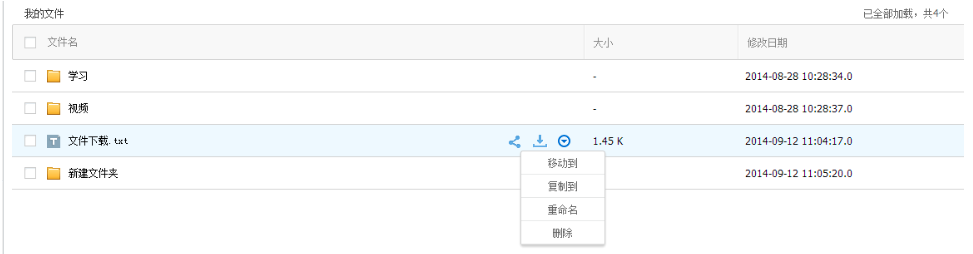
同时，当我们集群中的某台Tomcat服务器意外停止，这时，集群中的其它服务器会照常工作，并不会影响用户对我们项目的访问。当意外停止的服务器恢复工作时，apache会实时把它加入集群当中，实现请求分配均衡。

对于多个服务器session同步问题，采用session复制模式实现。这样即使用户的请求被分发到不同的服务器，session域里面的数据是实时共享的。

数据库设计

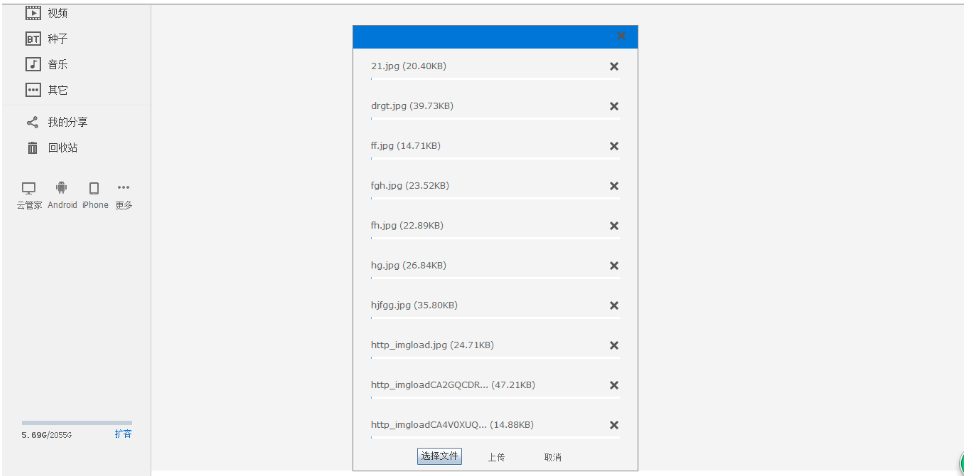


Web客户端



上传文件   
上传之前先扫描文件，利用MD5算法生成文件报文摘要，和已经上传的文件摘要对比，如果相等则放弃上传文件，直接记录该文件的新信息，实现秒传功能。

新的文件提交到存储设备上的接受程序，Tomcat实现跨域文件表单提交时，在要被跨域提交tomcat服务器的WebApps下的root文件夹添加一个，否则会出现安全错误不可以提交的问题。



注意：使用ajax的时候，一定要在url上添加时间戳，不然会导致ajax的作用只能使用一次，因为如果浏览器在解析地址的时候如果发现某次的地址跟之前的完全一样，就会直接从浏览器的缓存中读取数据，而不会发送请求，以提高速度。所以我们在使用ajax的时候一定要加上时间戳。

Android客户端

利用Apache开发的HttpClient模仿浏览器的表单提交，采用的数据格式是JSON数据格式，得到了数据之后再由Android客户端解析JSON数据，并将其封装到Java Bean中传输到Activity中进行使用。

