**搜狐云存储应用分析**

**2012-2-11**

**Leon**

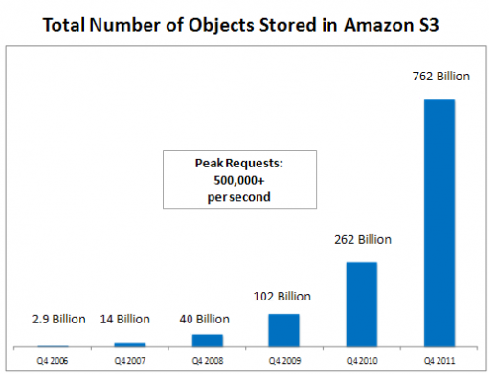
1. **BladeFS 与 s3 设计实现的差异**
   1. 一致性： BFS采用强一致性，Master与2个Slave之间，满足NWR规则为 N=W=3，R=1; 而s3则是弱一致性，N=R=3, W=1。
   2. 版本：由于不同的一致性规则，导致s3需要使用版本处理发生不一致之后，如何进行取舍。
   3. 容量大小：BFS设计意图是给予相册系统支持，因而最大单个对象的大小是16M;而s3采用Bitstore系统来保存文件，单个对象的大小最大可到5TB。
   4. 区域存储：s3可以提供Bucket区域选择的功能，这来源于其全球数据中心的优势；而BFS暂时还没有相关接口，当然这里不是指的全球范围区域的选择，而是国内电信、网通、教育网之间的指定。
   5. Shrink维护暂停写功能：BFS系统设计采用内部单文件的方式将多个小文件（16M以下，平均数百K）保存在一起，这样避免了ext X文件系统内部多次I/O索引的消耗；同时，也产生了一个空洞的问题。在培训中，咨询过关于是否产生空洞的问题，刘总给出的解答方案是会在评测某些结点空洞到达一定阈值后，会进行重组，我称之为数据结构的Shrink操作。那么这个操作一定会导致当前结点的暂时不可用，于是系统应该判定该Group处于不完整状态，会停止写入操作，但是读取仍旧没有问题。（由于不知道s3的内部处理方式，暂且将这点设置为差异，也许内部机制大相径庭）
2. **基于BladeFS 的可行存储方案**
   1. 架构

详细描述：

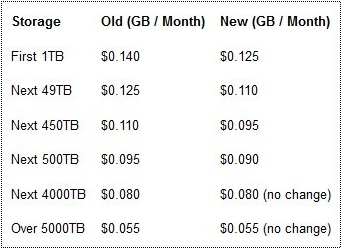
1. 核心数据库采用Redis主备份模式，硬盘无需再做Raid5：
   1. 有效利用刀片服务器的大内存，提供超高I/O性能。
   2. Redis主备模式提供相对可用性。
   3. Redis自带恢复功能，能迅速对损毁数据进行恢复。
2. Nginx服务器可以满足高并发网络访问，同时也可以将云对象的创建、访问与删除功能以Nginx模块的形式编译到服务器内核中，达到以下几个优点：
3. 云存储功能模块划分更精确：其一，提供带有安全认证的网络存取与删除，需要直接和BFS进行交互；其二，提供功能可扩展的复杂的上层应用，诸如账号注册、注销、授权、对象与Bucket生命周期的管理，甚至于未来可能的订阅功能等。
4. Nginx服务器模块提供方便和高效的扩展（类似于责任链模式），将网络请求在各个模块中传递，非常有利于提供第一点中需要的功能。同时，能保证未来在海量用户访问时，也能提供低延迟，高并发的网络服务。
5. BFS原生的C接口，刚好可以满足Nginx模块的调用，保证了Nginx与BFS之间的无缝连接。
6. JBoss/Tomcat容器，利用J2EE相关的组件和架构提供更多复杂应用功能：
   1. Web系统提供账号注册、注销、授权与文件目录管理相关功能。
   2. Web Service服务，可以再未来提供除restful接口以外，更多选择。
   3. 网页式服务器内部管理的相关功能。
   4. 功能

S3系统是目前世界知名的产品，它的成功也是集成了amazon公司技术专家多年的努力才取得的。目前，搜狐想走s3类似的道路，建立符合中国网络拓扑结构的简单云存储系统。首要的事情，也确实是带着产品上路。以产品为导向，平台化，打响自己的云系统的名声，比如目前正在研发的搜狐随身读。暂时以考虑该类产品，以及目前市面流行的云端产品，提出以下亟须实现的功能：

1. 账号相关：注册、通知、注销、密码/密钥管理、授权等。
2. Bucket/File相关：Bucket网络线路调整，Bucket权限分级（需要分开Bucket遍历功能），对象生命周期管理，Bucket重要通知，Bucket相关日志访问等。
3. 服务器端加密功能。
4. 静态网页：设定与取消默认访问文件，url别名等。
5. 计费系统：访问次数统计，容量占用统计。
6. 审计系统：系统日志与BFS日志审计，多日志核对。
7. S3走势
   1. S3存储对象个数走势



* 1. 近期降价策略



从上述图表中可以看出，亚马逊云计算服务由于更先进的技术和更优化的存储方案，一方面吸引了更多的付费用户，另外一方面有效降低了存储单价。依靠一定的小存储量的降价策略，s3应该在2012年会吸引更多小型公司投入怀抱；而对于超过500TB的大公司，数据迁移的成本明显较大，s3则没有做出过多的调整。

搜狐云存储系统应该早日启动，并由自身项目作为驱动，一方面打响品牌名声，一方面争取内部更多项目使用，降低存储单价。相信将来的云门槛也会随着单价下降逐步提高，后进入的公司需要一次性投入更多资源，才能参与分享云的蛋糕；同时，也将面临更高的风险！