Scalability

对于文件共享系统而言，Scalability主要是在体现在系统是否仍能保证效率的同时，支持日益增多的文件数量和文件大小，与日益增多(大量)的实时用户访问量。

1 关于文件量这个方面

①我们没有专门的文件存储系统(file server)，所有文件将存储在内存上，而随着存入的resources数量的增加，占用的内存资源也增加，最后可能🡪out of memory。

②我们的系统是将资源信息存在了 HashMap里，同理，随着存入resources数量的增多，Haspmap也会出现out of memory 的情况。

这使得我们的系统不能继续正常工作(down)，提供服务.

Revisions:

1>建一个专门的File server 将所有文件存入进去.

2>using of database instead for storing all resources information.

2 文件大小方面----然后就没有然后了哈哈 其他两个方面字数也挺够的 所以可以不用说这个啦

3 日益增多(大量)的实时用户访问量方面

①我们服务端开启线程采用的是thread per connection的原则，

随着客户端连入数量增加，开启的线程数量也就越多，对内存的消耗也越来越大，巨大的client连接数量有可能🡪out of memory🡪cannot receive more connections from clients （Under heavy load, a multi-threaded web server consumes large amounts of memory due to one thread stack for one connection)

②当许多用户同时访问修改shared resource的时候会产生大量的线程锁等待（incur many waiting times），从而影响到系统的性能。

Revisions:使用 event-driven model(这个我想再查下资料)

(An event-driven model can break up request processing into a series of callbacks, each thread can service multiple requests, it allows a pool of N threads to handle M connections.) Thus, it saves the cost of creating threads and reduces the number of threads creation.

Reference:上面括号内的内容是copy下来滴

<https://www.quora.com/What-is-the-ideal-design-for-server-process-in-Linux-that-handles-concurrent-socket-I-O>

<http://berb.github.io/diploma-thesis/original/042_serverarch.html>