

关于cse lab3的说明

前两个lab中的理想网络

在前两个lab中，为了将逻辑分离，extent service和lock service都是采用C/S的架构，通过网络（即RPC）来实现extent和lock需要的逻辑，并且考虑的都是不会丢包的理想网络。

网络丢包

在lab3中，我们开始考虑网络丢包的情况，是通过RPC_LOSSY这个环境变量进行模拟的。即client向server发送请求和server给client回复请求的时候他们之间会建立连接（Connection），RPC_LOSSY的实现机制即这个连接会有概率性的中断，中断的概率有RPC_LOSSY的数值决定，为0时表示理想网络，1~5数值越大，中断的概率越大。

RPC提供的保证

假如网络存在丢包的现象，应当由RPC提供exactly once的保证，即在client给server发包的时候假如连接断掉，client会不断重新与server建立连接，再次发送相同的包，直到这个连接返回reply，以此来保证at least once；在server收到client的请求的时候会记录下这个请求的一些信息，以此来判断之前是否已经收到过此请求，如已经收到过，直接从记录下来的之前的执行结果返回，没收到过则会去执行这个请求并返回，这样在server端保证at most once，server回复client的请求也可能会丢失，但server回复之后不管回复是成功了还是丢失了server都会退出。

谁来实现exactly once

exactly once的功能应当由RPC来实现，因为在RPC内部才能有request id，client id等信息来确定某个rpc包是不是相同的。其次如果RPC不提供exactly once，而是由业务层提供的话，那么针对每一个业务都得去实现exactly once的逻辑，即这个lab的lock service和extent service都得去实现exactly once的逻辑才行。

关于测试

这个lab会测试三项

1. Your caching lock server passes lock_tester with RPC_LOSSY=0 and RPC_LOSSY=5. 测试方法见step2和step3，测试正确会输出passed all tests successfully
2. Your file system using the caching lock client passes all the Lab 3 tests (a, b) with RPC_LOSSY=0 and RPC_LOSSY=5. 测试方法·

```

//test without lossy
% ./start.sh
% ./test-lab-3-a ./yfs1 ./yfs2
% ./stop.sh

% ./start.sh
% ./test-lab-3-b ./yfs1 ./yfs2
% ./stop.sh
//test with lossy
% ./start.sh 5
% ./test-lab-3-a ./yfs1 ./yfs2
% ./stop.sh

% ./start.sh 5
% ./test-lab-3-b ./yfs1 ./yfs2
% ./stop.sh

```

1. Your lab 2 code generates about a tenth as many acquire RPCs as your lab 3 code on test-lab-3-b, 测试方法

```

//first in your lab2 context
% export RPC_COUNT=20 //you can set the value smaller to get more accurate counting
% ./start.sh
% ./test-lab-3-b ./yfs1 ./yfs2
% ./stop.sh

//then in your lab3 context
% export RPC_COUNT=20 //you can set the value smaller to get more accurate counting
% ./start.sh
% ./test-lab-3-b ./yfs1 ./yfs2
% ./stop.sh

```

综上，在前面lab以及lab3中提供给大家的rpc库没有实现exactly once的功能，是助教的考虑不周，这里向大家抱歉哈。实现了exactly once功能的rpc库已经更新在git repo上了，大家可以单独获取下来，替换自己的rpc库，具体操作方法在lab3的说明页已经更新，并且本次lab延迟到11.19号截止。