1. 代码结构

从编译得到的结果来看，dmfs是主程序。



从编译文件（build/ link.txt）来看，链接中用到的文件有

|  |  |
| --- | --- |
| 文件 | 说明 |
| src/fs/dmfs.cpp |  |
| src/fs/filesystem.cpp |  |
| src/fs/storage.cpp |  |
| src/fs/lock.cpp |  |
| src/fs/bitmap.cpp |  |
| src/fs/TxManager.cpp |  |
| src/fs/hashtable.cpp |  |
| src/fs/sha256.c |  |
| src/net/Configuration.cpp |  |
| src/net/RPCServer.cpp |  |
| src/net/RdmaSocket.cpp |  |
| src/net/mempool.cpp |  |
| src/net/RPCClient.cpp |  |
| src/tools/debug.cpp |  |

从这里看，dmfs.cpp是入口。

由于文件不太多，可以选择按文件来写简要描述。

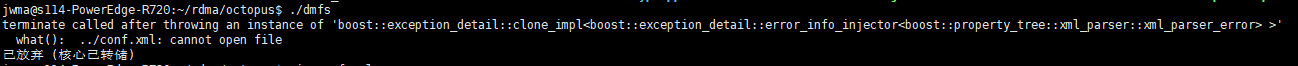
TxManager-Transaction Manager

1. dmfs.cpp

就两个函数，一个函数用来做正常退出时的提示。主函数无限循环。

主函数里在接收到一个字符后输出一个12位的地址。

2. 运行流程



由此可以看到首先读取配置文件conf.xml。Configuration类里面读取配置文件。

但配置文件里面仅仅描述了节点的信息。

<?xml version="1.0" encoding="utf-8"?>

<address>

<node>

<id>1</id>

<ip>192.168.0.114</ip>

</node>

</address>

有了配置文件之后就是共享内存错误。



shmat是共享内存函数。

在mempool里类初始化时被调用。

在110上会出现shmget error和shmat error，但114上只有shmat error。

shmid = shmget(SHARE\_MEMORY\_KEY, DMFSTotalSize + LOCALLOGSIZE + DISTRIBUTEDLOGSIZE, IPC\_CREAT);

if (shmid == -1) {

Debug::notifyError("shmget error");

}

shmptr = shmat(shmid, 0, 0);

if (shmptr == (void \*)(-1)) {

Debug::notifyError("shmat error");

}

3. 代码细节

面对多QP时的轮换，由上层应用进行选择

/\* Server interconnect, use cq at 0. \*/

server互联的时候也也是使用qp[0]