搭建runConsensus 遇到的错误

1 sudo vi /etc/resolv.comf 要把dns服务器提到第一位

2 出现各种连接错误的情况：

403forbidden bitcoind没有开启网段接收

Value 是json数据的解析内容类 是value\_\_type是value\_impl

CAmount 是聪位单位的整形类 COutput包含交易指针cwallettx、深度、是否可用

Pair包含String\_type和Value\_type两个值可用get\_name(pair) get\_value(value)

Object 是键值对Pair的vector实现

ValuefromAmount 输入CAmount聪返回Value比特币值

Find\_value 输入Object和name 查值，再用get类型可以获取响应类型的参数。而原生接口json不是键值对，不需要这种方法

CTransaction包含版本，输入输出，锁时间。扩展出 CMerkleTx扩展出 CWalletTx 是钱包交易 CMutableTransaction是可变更版的CTransaction

CWallet扩展自 CCryptotoKeyStore和CValidationInterface，全局变量钱包，声明在init.cpp

CKeyStore 扩展出CBasicKeyStore 扩展出CCryptoKeyStore

CCoinsView一个抽象工具类，提供根据txid获取交易输出的信息，如状态，币数量等的方法。

扩展出CCoinsViewBacked再扩展出CCoinsViewCache 添加一个缓存类存放上述获取的信息

CCoins 是修剪版的CTransaction 而CCoinsCacheEntry缓存可以

CCoinsMap无序映射 可用txid查CCoinsCacheEntry CCoinsModifier

CTxMemPool 储存即将可能加入区块链的交易。交易池。

COutput高度封装的交易类，含txid，cwallettx，输出的索引i



CRecipent 是CTxOut 稍稍扩大版

平均一条CTXIN大小增加41B 一条CTXOUT大小增加34B，通常比特币交易平均500B， 。交易的GetSerializeSize获取大小，GetMinimumFee获取可能需要的过路费，原生的函数。当交易池交易数量太多，会显示优先级不足的错误。增加交易费用可以解决。因为免费额度有限，0.12版大约每分钟处理免费的15KB交易。超过免费额度就必须有交易费，交易费至少要大于一个交易的大小限制MAX\_STANDARD\_TX\_SIZE=100000。

::minRelayTxFee.GetFee(nSize) 根据交易大小判断当前交易池内加入此交易所需的广播交易费

AddToWallet 函数用来检测是否插入交易，fInsertedNew 判断是否插入到mapWallet缓存 fUpdated判断是否更新钱包到 wallet.dat，只有发送/接收金币方调用此函数

Maptxspends 记录已花费的交易，避免双重支付

Process Message ip:....处理来自的消息

if (mapBlockIndex.count(hash) == 0)

throw JSONRPCError(RPC\_INVALID\_ADDRESS\_OR\_KEY, "Block not found");

CBlock block;

CBlockIndex\* pblockindex = mapBlockIndex[hash];

if (fHavePruned && !(pblockindex->nStatus & BLOCK\_HAVE\_DATA) && pblockindex->nTx > 0)

throw JSONRPCError(RPC\_INTERNAL\_ERROR, "Block not available (pruned data)");

if(!ReadBlockFromDisk(block, pblockindex))

throw JSONRPCError(RPC\_INTERNAL\_ERROR, "Can't read block from disk");

if (!fVerbose)

{

CDataStream ssBlock(SER\_NETWORK, PROTOCOL\_VERSION);

ssBlock << block;

std::string strHex = HexStr(ssBlock.begin(), ssBlock.end());

return strHex;

}

return blockToJSON(block, pblockindex);自。。。。。。的消息

Push Message ip:.....发送消息到。。。。。。

Rpc:接（默认：125）

-addnode= 添加一个节点以供连接，并尝试保持与该节点的连接

-connect= 仅连接到这里指定的节点

-irc 使用 IRC（因特网中继聊天）查找节点（默认：0）

-listen 接受来自外部的连接（默认：1）

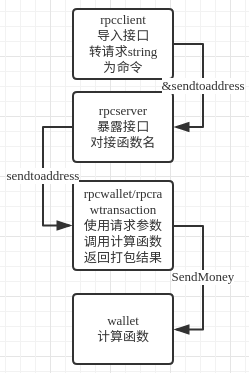
-dnsseed 使用 DNS 查找节点（默认：1）

-banscore= 与行为异常节点断开连接的临界值（默认：100）

-bantime= 重新允许行为异常节点连接所间隔的秒数（默认：86400）

-maxreceivebuffer= 最大每连接接收缓存，\*1000 字节（默认：10000）

-maxsendbuffer= 最大发送缓存



内存

所有交易的集合 mayWallet

未花费的可用交易集合mapUTXOCache

索引存储在unordered map mapblockindex

索引储存在vector cchain

帮助->控制台的ui修改 rpcconsole.cpp

chainparams.cpp

checkpointdata 检查点数据，抽查历史节点的哈希值，判断是否在主链上