

EOS 节点分类浅析

EOS 编译、测试指南（五）



扫一扫关注欧链小秘书

EOS 团队于 2017 年 9 月 14 日推出了 Dawn 1.0 release 版本，熟悉 Linux 和 C++ 的开发人员可以在此基础上尝试自己的合约开发了。OracleChain 团队一直以来都在对 EOS 代码进行了编译和测试，陆续发布了四份《EOS 编译、测试指南》，详细介绍了 EOS 的入门知识、基本结构，帮助开发人员学习、研究 EOS。

基于 EOS 刚刚推出的 Dawn 1.0，OracleChain 团队进行了测试，本次将以节点功能为例，详细介绍 EOS 上不同节点服务之间的区别，基于此开发人员可以根据自己的业务需求，在 EOS 网络上运行自定义节点，从而参与、利用 EOS 网络为外部提供服务。

该指南主要依据 EOS 的官方文档，以及欧链团队的实践经验编写，开发者有任何技术问题可以关注欧链小秘书、在欧链科技社区里向 OracleChain 团队提问或者发送邮件到 contact@oraclechain.io。

EOS 文档：<https://eosio.github.io/eos/>

EOS 代码：<https://github.com/EOSIO/eos>

备注：OracleChain 团队使用 Mac 环境进行开发。

上一章中，我们通过源码研究了 EOS 合约执行机制。合约执行内容主要包括三方面：合约发布和存储，合约执行环境的数据存取接口，以及状态数据持久化等。依靠 web assemble 沙盒技术，让合约拥有与 native code 一样的执行速度。

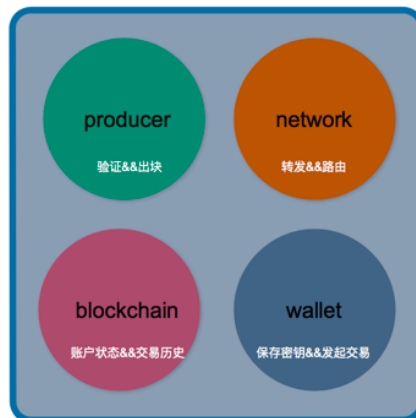
与 bitcoin 相似，EOS 各节点通过 P2P 网络互联，相互传递 block 信息，任何一个节点都可以参与验证和转发 block 信息，从而达成全网共识。而与 bitcoin 不同的是，EOS 内部采用 DPOS 的出块机制。每个节点可以选出代理，或者直接投票，选出其信任的 21 个出块节点。这样做，既避免了出块权过于集中，又

保证了共识机制不至于分叉（投票表决）。同时，这种投票保证的信任机制避免了计算资源的浪费，将资源真正运用到 DAPP 的服务中。因此，我们可以把 EOS 合约，当作一个运行在“云”上的服务

一、EOS 节点的功能和分类

EOS 网络中，所有节点都可以参与验证/转发交易和区块，发现和维护一个点对点网络。

EOS 网络节点可根据函数功能分为四个集合。开发者可以组合不同的集合，发布符合自己需求的节点，加入到 EOS 网络中，再由这个节点为最终用户提供更多差异化服务。



二、network（转发路由）

所有节点都包含转发路由模块。通过这个模块，节点可以加入到 EOS 网络中，参加对交易和区块的验证，发现和共享 p2p 连接池。另外，如果要投票或选择代理人选出出块节点，就可以通过组合其他模块来实现。

要启动只带有转发和验证功能的节点，我们只需要在启动 eosd 的时候注释

掉所有 plugin。config.ini 如下：

```
genesis-json = ./genesis.json
block-log-dir = blocks
readonly = 0
shared-file-dir = blockchain
shared-file-size = 8192
http-server-endpoint = 127.0.0.1:8891
listen-endpoint = 0.0.0.0:9879
public-endpoint = localhost:9879
#plugin = eos::producer_plugin
#plugin = eos::chain_api_plugin
#plugin = eos::wallet_api_plugin
#plugin = eos::account_history_plugin
#plugin = eos::account_history_api_plugin
```

三、producer（验证出块）

只有出块节点包含有效的 **producer function**。出块节点可以根据自己的资源提供交易和存储服务。普通节点根据节点的价值偏好（可以是资源对应 **token** 的单价），投票选出出块节点。选出的出块节点将组成一个网络，任意一个出块时间段内，只有一个出块节点被全网授权去验证交易，并将交易打包成块。最后，**block** 被转发到其他节点验证和转发，全网达成共识。

启动一个 **producer** 节点时，需要在 **config.ini** 文件中启动 **producer plugin**，并且配置出块节点相关数据：

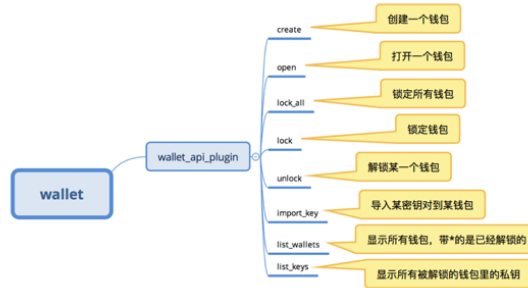
```
genesis-json = ./genesis.json
block-log-dir = blocks
readonly = 0
shared-file-dir = blockchain
shared-file-size = 8192
http-server-endpoint = 127.0.0.1:8889
listen-endpoint = 0.0.0.0:9877
public-endpoint = localhost:9877
remote-endpoint = localhost:9878
remote-endpoint = localhost:9879
enable-stale-production = true
required-participation = true
private-key = ["EOS6MRyAjQq8ud7hVNYcfnVPJqcVpscN5So8BhtHuGYqET5GDW5CV", "5KQwrPbwdL"]
plugin = eos::producer_plugin
#plugin = eos::chain_api_plugin
#plugin = eos::wallet_api_plugin
#plugin = eos::account_history_plugin
#plugin = eos::account_history_api_plugin
producer-name = initb
```

四、wallet（保存私钥和发起交易）

钱包用户都需要有 **wallet api** 来维护自己的 **EOS** 智能资产，例如桌面端的 **EOS** 客户端、钱包 **app** 等。对于资源吃紧的手机终端的钱包 **app** 来讲，由于不能保存所有的账户信息，所以只保留用户相关的数据，并使用类似比特币的 **SPV** 技术，

远程拉取交易所需信息并发出交易。

在 EOS 源码中, wallet 以 plugin 的形式提供对外服务, 我们可以使用 eoscli 对这些接口进行简单调用。



在配置中, 也需要指定加载 wallet plugin:

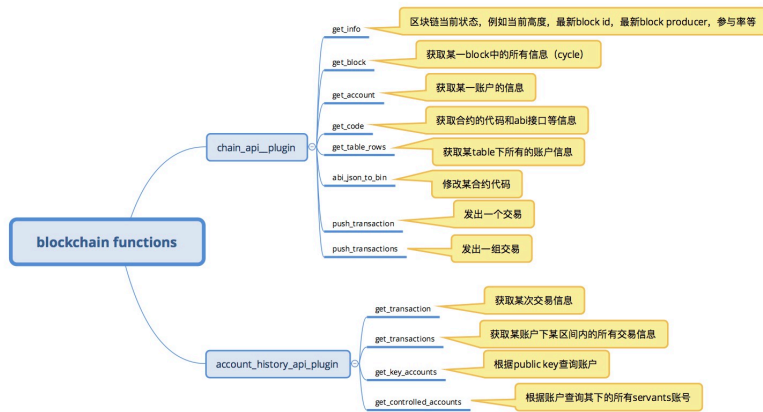
```
genesis-json = ./genesis.json
block-log-dir = blocks
readonly = 0
shared-file-dir = blockchain
shared-file-size = 8192
http-server-endpoint = 127.0.0.1:8891
listen-endpoint = 0.0.0.0:9879
public-endpoint = localhost:9879
#plugin = eos::producer_plugin
#plugin = eos::chain_api_plugin
plugin = eos::wallet_api_plugin
#plugin = eos::account_history_plugin
#plugin = eos::account_history_api_plugin
```

五、blockchain functions (账号状态和交易历史)

通常只有全节点才会使用 blockchain functions。blockchain 模块实际包含两部分数据, 一部分是保存着所有账号的状态机, 另一部分是保存着所有历史交易的 block 数据。全节点依靠自身保存的完整的账号状态和交易历史, 可以在本地独立地验证所有交易。

举个例子, 对于出块节点来说, 不一定会带有 wallet 模块, 但一定会有带有 blockchain 模块。blockchain 模块是出块节点产生共识的基础数据, 出块节点只有遵守共识, 产生的 block 数据才会被区块链所接受。

blockchain 模块也通过 http api 对外提供查询接口, 我们可以使用 eoscli 对这些接口进行调用:



在配置一个全节点时，我们的配置文件如下：

```

genesis-json = ./genesis.json
block-log-dir = blocks
readonly = 0
shared-file-dir = blockchain
shared-file-size = 8192
http-server-endpoint = 127.0.0.1:8891
listen-endpoint = 0.0.0.0:9879
public-endpoint = localhost:9879
#plugin = eos::producer_plugin
plugin = eos::chain_api_plugin
plugin = eos::wallet_api_plugin
plugin = eos::account_history_plugin
plugin = eos::account_history_api_plugin
  
```

六、总结

在 EOS 目前的版本之上，我们已经可以初步的搭建自己的测试网络，完成区块链基本功能测试。我们相信再不久的将来，基于 EOS，将会开发出丰富的区块链应用。

OracleChain 团队
2017 年 11 月 1 日