Clone of the filecoin testnet (based on go-filecoin and lotus)

FileCoin (https://learnblockchain.cn/tags/FileCoin) IPFS (https://learnblockchain.cn/tags/IPFS)

> I got this in July last year. At that time, I was thinking about mining with a mining machine to dig by myself. Later I found that the hardware configuration requirements were too high, and there were no orders, so I temporarily sealed it, and took out the previous notes today For your reference. Here is a compilation test based on go-filecoin and lotus respectively.

I got this in July last year. At that time, I wanted to sell it as a mining machine, but it was temporarily sealed because I didn't receive the order. It’s been a while, so let’s find it out for your reference.
Let me talk about the difference between go-filecoin and lotus first. The former is a complete filecoin solution, but because it is more troublesome to implement, a lotus was later released. lotus can be simply understood as a simplified test version of filecoin. Of course, it has been more than half a year now, and it may have changed a lot.

Clone of the go-filecoin testnet

First of all, because a large amount of content needs to overturn the wall, the compiler should try to use Hong Kong server as much as possible.

One environment compilation

```
apt update
apt install mesa-opencl-icd ocl-icd-opencl-dev gcc git bzr jq pkg-config curl
apt upgrade
apt install software-properties-common
add-apt-repository ppa:longsleep/golang-backports
apt update
apt install golang-go gcc git bzr jq pkg-config mesa-opencl-icd ocl-icd-opencl-dev
mkdir -p /path/to/filecoin-project
git clone https://github.com/filecoin-project/go-filecoin.git /path/to/filecoin-project/go-fileco
git checkout v0.6.0//(从0.6.1到0.6.4都测试过了，都在go-filecoin daemon报错)
git submodule update --init --recursive
make deps
```

ps1: Here is the link setup

Second compilation

```
make
go run ./build test//（期间会有fail出现，但没搞明白）
go run ./build best//（期间会有fail出现，但没搞明白）
go-filecoin init --genesisfile=https://ipfs.io/ipfs/QmXZQeezX1x8uRQX9EUaYxnyivUpTfJqQTvszk3c8SnFP
go-filecoin daemon
```

ps2: The obtained id is QmdMXaWc5Ww2PWWHGQvwqSPmv2rSnEPzxYwjBjLBVVZGJg, not the address remember.
At this point, the basic environment is started, and then the connection is made.

Three connection

First connect with ssh,

```
go-filecoin config bootstrap.addresses
go-filecoin swarm connect <any-filecoin-node-mulitaddr>
```

ps: This part failed. But it is very strange that it does not affect the subsequent acquisition of coins, and the balance can also be seen.
Then enter the second root user in the web window, enter:
"t1e5xumuipwdputhrdm2sygngaynmhpnmhvt5mozq"

A simple summary of the first stage of four

The process is clearer, but there are a lot of problems.
We are based on filecoin, and the process is simply to download and then compile.
There is a problem here, but it is not obvious.
Then it came time to run the go-filecoin daemon, because it was 0.6.0, there was no problem. But when it comes to downloading the blockchain through the address, I can't live or die. It is always an error. I feel that there is a problem with the version.
go-filecoin config bootstrap.addresses
go-filecoin swarm connect
There is nothing wrong with getting coins and block explorer display further down.
go-filecoin address ls
https://faucet.testnet.filecoin.io (https://faucet.testnet.filecoin.io)
https://filscout.io/en (https://filscout.io/en)
But when it comes to checking the wallet locally, there is another problem, showing

that the address does not exist.
go-filecoin wallet balance ${WALLET_ADDR}
logically can operate normally in this place, but it is not
logical , is there any way, so the direct operation based on filecoin ends. We will compare filecoin and lotu in the next section and use lotu to solve it.

lotu compilation process

A compilation

```
apt update
apt install mesa-opencl-icd ocl-icd-opencl-dev gcc git bzr jq pkg-config curl
apt upgrade
apt install software-properties-common
add-apt-repository ppa:longsleep/golang-backports
apt update
apt install golang-go gcc git bzr jq pkg-config mesa-opencl-icd ocl-icd-opencl-dev
git clone https://github.com/filecoin-project/lotus.git
cd lotus/
make clean && make all
make install
IPFS_GATEWAY=https://proof-parameters.s3.cn-south-1.jdcloud-oss.com/ipfs/   //（如果在国内就增加上面这
```

Two start running

lotus daemon
(the message will appear and keep going)
and then open the second root, enter
lotus net peers | wc -l
(I show 32 here, indicating that 32 peers are connected)

Three open the third root

Input
lotus sync wait
will display the synchronization status, this step will take several hours to several days.
Observe the block explorer:
https://stats.testnet.filecoin.io/ (https://stats.testnet.filecoin.io/)

Three enable putty

Enter:
lotus wallet new
t1djagnv6wjmfowmf5damd3zoeeicg66ik4pir3ai
https://faucet.testnet.filecoin.io/ (https://faucet.testnet.filecoin.io/)
Enter the address and get coins: (about 50 at a time)
https://filscout.io/en (https://filscout.io/en)
enter to check whether
lotus wallet is in place in the network balance <YOUR_NEW_ADDRESS>
(If your blockchain is being updated, the above command will not see the coins)
lotus send
(Transfer, also the blockchain is being updated, this is abnormal)

Four additional functions

rm -rf ~/.lotus ~/.lotusstorage
clear data
ulimit -n
open file limit, must be greater than 10000 (the default is generally 60,000, enough, if not enough, modify: ulimit -n 10000)
Five the following is the establishment of miners
if For domestic use, you need to add this
IPFS_GATEWAY=" https://proof-parameters.s3.cn-south-1.jdcloud-oss.com/ipfs/ (https://proof-parameters.s3.cn-south-1.jdcloud-oss.com/ipfs/) "
Create a bls address (you need putty, then copy the address)
lotus wallet new bls
copy the address
https://faucet.testnet.filecoin.io/ that (https://faucet.testnet.filecoin.io/) appears
and click Create Miner
Don't refresh the page. This operation may take some time. The task will be completed when you see the following:
New storage miners address is: <YOUR_NEW_MINING_ADDRESS>
In the CLI window, use the following command to initialize your miner:
lotus-storage-miner init --actor=ACTOR_VALUE_RECEIVED --owner=OWNER_VALUE_RECEIVED
Example lotus-storage-miner init --actor=t01424 --owner=t3spmep2xxsl33o4gxk7yjxcobyohzgj3vejzerug25iinbznpzob6a6kexcbeix73th6vjtzfq7boakfdtd6a
Ps: This command may take a while to complete. It may take a while to complete this operation.

Five formal mining

lotus-storage-miner run to
get miner information
lotus-storage-miner info # example: miner id `t0111`
sealed random number starts to generate post
lotus-storage-miner sectors pledge
sealed random number starts to generate post
(note that the written $TMPDIR may not be the largest Partition, you may need to specify, of course, for me, the formal operating interval must be specified, at least not less than 512gb)
The following three lines are for obtaining computing power, obtaining the computing power of a certain miner, and obtaining the sectors of a certain miner
lotus state power # returns total power
lotus state power
lotus state sectors
Some parameter adjustments (I don't know where these parameters are set)
FIL_PROOFS_MAXIMIZE_CACHING=1 environment variable
At the cost of some memory (1x sector size) to make the precommit1 step faster.
FIL_PROOFS_USE_GPU_COLUMN_BUILDER=1 environment variable
enable experiment precommit2 GPU acceleration lotus-storage-minerlotus-seal-workerlotus-bench

Some suggestions about official mining machines

Since half a year has passed, it is estimated that the changes will be great, so this part of the content is for reference only.

Master Miner node:

Use a large-capacity machine with GPU as the miner master node. The main work of this machine is to add pieces and submit PoSt proving (the focus here is that you need to bring gpu to encapsulate the post, 2T hard disk, 8g memory is enough)

Seal Worker node:

Multi-core CPUs are equipped with high-speed enterprise hard disks to increase the sealing speed. If the CPU is not strong, you can use GPU instead for better results. (The focus here is that the cpu is better, the high-speed ssd, the memory must be large, the cpu is not good, gpu can be replaced)
The official team of the protocol laboratory released the basic configuration of the tested hardware on the official Lotus document, which clearly stated the minimum size of the mining machine RAM is not less than 128GB. Because it consumes a lot of memory when calculating the copy proof, it is about two to three times the sector size. Corresponding to a 32GB sector, it is conservatively estimated that 96GB of RAM will be used for copy proof. In addition to the RAM required by other processes in the system, it is reasonable for the laboratory to recommend increasing the RAM to 128GB.

Some simple conclusions:

The CPU of the Miner node is less than 10% (high graphics card plus low cpu, with the support of the graphics card, 4 core cpu is sufficient), and the memory used is less than 5% (12.8g). The Miner node configuration can be reduced a lot, and I personally feel that 32GB of RAM is enough.
With the current machine configuration, we can run one more Worker process on the Miner node, which can increase the number of Workers. In fact, we have tested it this way, and it is very easy to implement on GammaOS, just a few clicks of the mouse. The speed of Sealing has indeed increased, but it was discovered that it caused an unexpected side effect: the block burst rate of the Miner node was reduced. Because a large amount of resources of the Miner were occupied, the block was not produced within the specified time, resulting in the block being used by other miners. Robbed. So we finally uninstalled the lotus-seal-worker application from this node.
We use GammaOS data to calculate the real-time RAM occupancy of the lotus-seal-worker process since the start of the miner, and found that the highest peak value is 30.4%, then the actual RAM occupancy = 256 x 30.4% ~= 78GB (64gb most of the time) . That is just in line with the official sector size (sector size) of 2-3 times.
Miner cannot deploy workers, otherwise it will affect the burst speed.

Two easily overlooked factors affecting block production:

The storage system is deployed on the mining network, which directly affects our mining efficiency. Moreover, according to the increase of Worker machines, there is a lot of room for adjustment and optimization of network configuration. Proper network optimization can bring more than 50% efficiency improvement.