Filecoin 项目 Lotus 安装和维护

（Bugx 编写于 2021 年 1 月，苏州）

目录：

一、LOTUS，MINER，WOKER 安装

1、硬件要求

miner/lotus 服务器要求：Nvidia 显卡，RAM 128G 以上，磁盘：需要有一个存储 lotus 日志的盘，每星期增加 12GB 的数据。

PC1 阶段要求：AMD cpus，每个进程使用 1 个 cpu 内核，需开启 FIL_PROOFS_USE_MULTICORE_SDR，RMA128G 以上，越大越好。磁盘：nvme ，pc1 缓存 32g 一块。

PC2 阶段要求：显卡：有显卡用 gpus，没有则用所有的 cpu 内核。磁盘：一个进程需要缓存 600 多 G 数据，需要 nvme 磁盘 2T

Commit1 阶段要求：所有 CPU 内核，和 Commit2 合在一起。这个阶段比较快。

Commit2 阶段要求：有 GPU 用 GPU，没有则用全部 cpu 核。GPU 缓存需要 9G 以上才能完整发挥。内存：173G 以上一个进程。

Unseal 阶段要求：内存：128G 以上。每个进程使用 1 个 cpu 内核。

Proving WindowPoSt：使用 GPU，无 GPU 使用所有 cpus。需要 30 秒内完成。

Proving WinningPoSt：使用 cpu，25 秒内完成。

下文安装都以 Ubuntu 系统为例：

2、安装 lotus、miner、worker

filecoin 的挖矿由三部分组成，第一部分是 LOTUS 节点，仅支持 linux 或 mac os。需要 AMD 芯片的 SHA 支持，至少 8 核 cpu，32GB 内存。足够的 SSD 空间容纳每周 12GB 的链同步数据。

为了以后减少麻烦，可以禁止更新内核。

sudo apt-mark hold linux-image-generic linux-headers-generic

1）安装必要组件

sudo apt install mesa-opencl-icd ocl-icd-opencl-dev gcc git bzr jq pkg-config curl clang build-essential hwloc libhwloc-dev wget -y && sudo apt upgrade -y

2）安装 Rustup

curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh

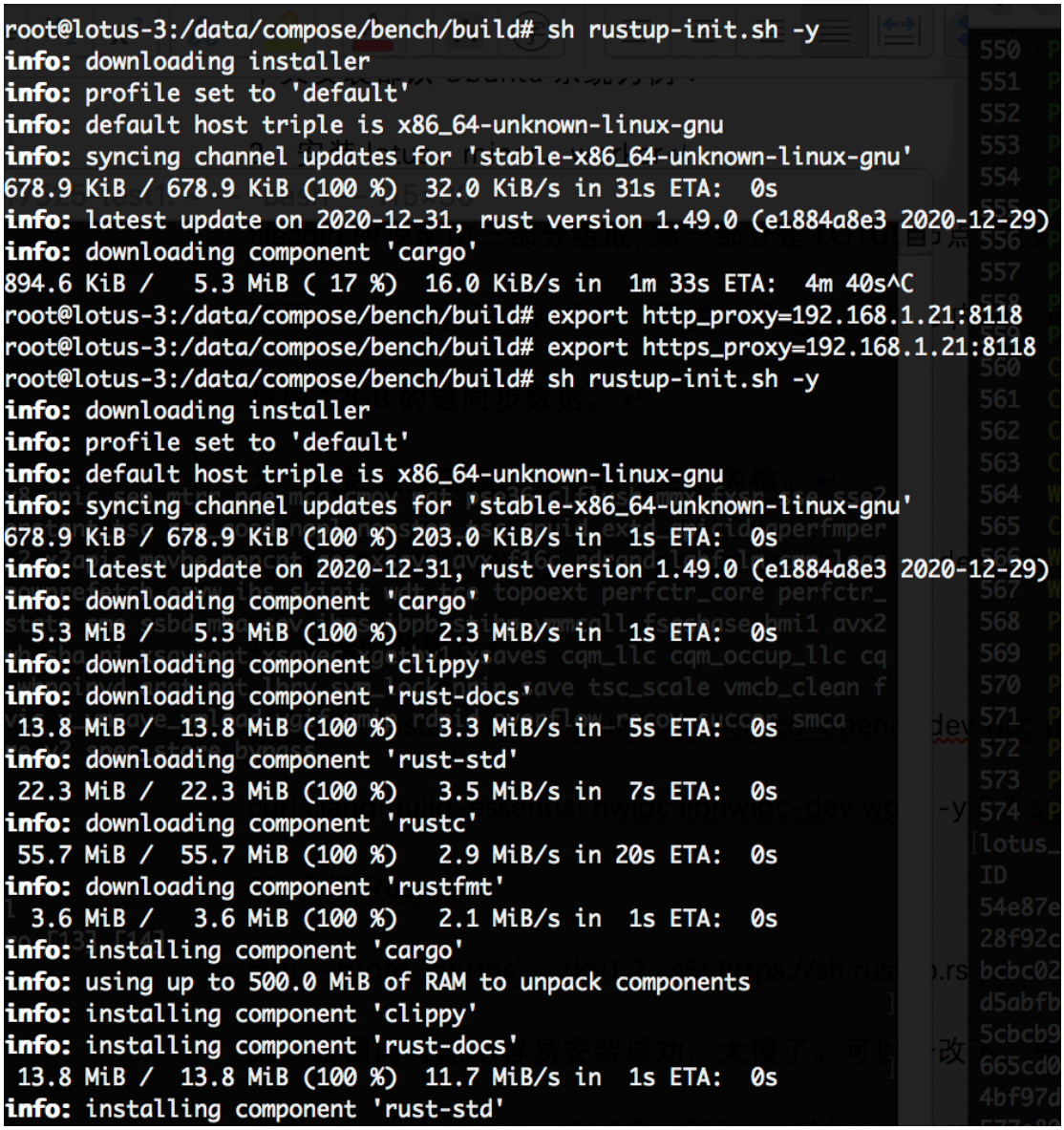
这一步国内可能不容易安装成功，太慢了。可以修改下，先保存 sh 文件下来

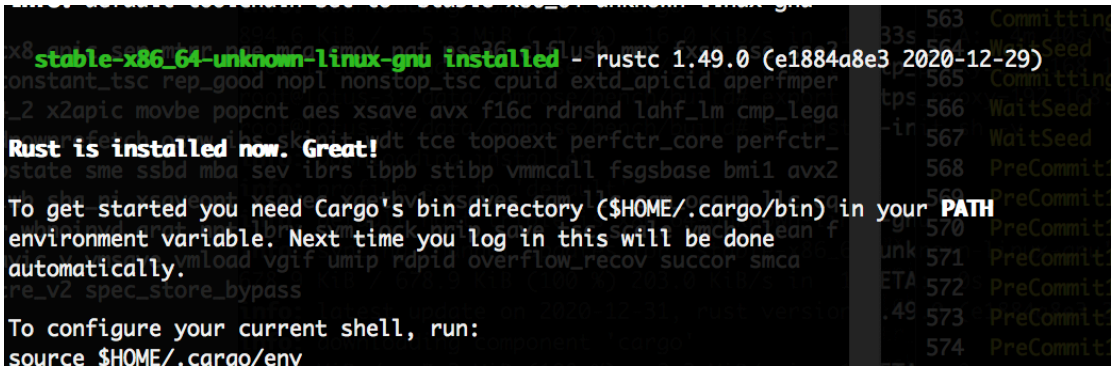
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs > rustup-init.sh

设置环境变量

export RUSTUP_DIST_SERVER=https://mirrors.tuna.tsinghua.edu.cn/rustup

这就很快了（如果国内源安装出错，用梯子切换回原来的。）。





安装完一定要执行

source $HOME/.cargo/env

3）安装 go

wget -c https://golang.org/dl/go1.15.5.linux-amd64.tar.gz -O - | sudo tar -xz -C /usr/local

go 的运行路径写到 home

export PATH=/usr/local/go/bin:$PATH

export GOPATH="/usr/local/go"

source $HOME/.profile

4）安装 lotus

AMD 芯片的需要设置环境变量

export RUSTFLAGS="-C target-cpu=native -g"

export FFI_BUILD_FROM_SOURCE=1

一些老的机器没有 ADX 指令的增加

export CGO_CFLAGS_ALLOW="-D__BLST_PORTABLE__"

export CGO_CFLAGS="-D__BLST_PORTABLE__"

git clone https://github.com/filecoin-project/lotus.git

这一步很慢，前面加一句

git config --global url."https://github.com.cnpmjs.org/".insteadOf

https://github.com/ 用来加速

cd lotus/

git checkout master

make clean all

sudo make install

lotus，Miner，worker 的安装方法都一样，如果在 intel 的机器上安装，记住把 AMD 特性的那几个删除。环境一致的话，只要安装依赖包在服务器上，然后将二进制拷贝到其他机器上就可以使用了。

要装 lotus-bench, lotus-shed，可以单独执行编译，make lotus-shed ，这是一个小工具，建议安装在 miner 机器上。lotus-bench 用来做测试的。

可以在 worker 机器上只安装 lotus-worker, lotus-miner 是具备 worker 程序的功能的，也可以执行部分任务。

Lotus 提供了服务安装，将 lotus 和 miner 设置成服务，服务脚本可以编辑定义路径之类。

make install-daemon-service

make install-miner-service

5）安装完成后，启动

国内运行加上两个环境变量：

export GOPROXY=https://goproxy.cn

export IPFS_GATEWAY=https://proof-parameters.s3.cn-south-1.jdcloud-oss.com/ipfs/

同步快照后启动，不然的话这个同步得花好几天

lotus daemon --import-snapshot https://fil-chain-snapshots-

fallback.s3.amazonaws.com/mainnet/minimal_finality_stateroots_latest.car

这个快照如果慢的话可以下载到本地。

lotus daemon --import-snapshot /home/xxxx.car

同步完成后,可以 lotus sync wait 验证下同步完成没,没完成会一直在那面执行。

按照完成后，可以通过 lotus、louts-miner、lotus-woker 来执行交互命令

这样默认启动后，根目录路径在 ~/.lotus

6）更新版本(更新时使用)：

到 git 目录

git pull

git checkout master

再编译一次

7）安装显卡驱动

安装驱动，在 NVIDIA 官网找到对应驱动，20xx、30xx 系列目前是 460 版本，

下载后安装驱动。

8)、优化下系统参数

ulimit -SHn 65535　调整系统的文件打开数限制，这个在多任务并发的时候需要，不然可能影响速度。这个是临时修改，系统永久修改可以查阅的各自运行系统的做法。

修改 vm.swappiness，增加内存的使用效率，减少 swap 的使用

/etc/sysctl.conf 增加一行 vm.swappiness = 0

临时修改 echo 0 > /proc/sys/vm/swappiness

二、docker 方式安装 lotus

1、安装 docker

```
sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent
software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
```

2、安装 nvdia-docker

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \
   && curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key
add - \
   && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-
docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list

curl -s -L https://nvidia.github.io/nvidia-container-
runtime/experimental/$distribution/nvidia-container-runtime.list | sudo tee
/etc/apt/sources.list.d/nvidia-container-runtime.list
sudo apt-get update
```

sudo apt-get install -y nvidia-docker2

3、为了管理方便，接下来安装一个 docker-compose

sudo curl -L

"https://github.com/docker/compose/releases/download/1.28.0/docker-

compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

装好 nvdia-docker 之后，这个 docker 就可以支持共享显卡了

可以看到/etc/docker/daemon.json 中有

```
"runtimes": {
        "nvidia": {
                "path": "/usr/bin/nvidia-container-runtime",
                "runtimeArgs": []
                }
       }
```

这里可以增加 docker 的国内源

```
"registry-mirrors" : [
     "https://mirror.ccs.tencentyun.com",
     "http://registry.docker-cn.com",
     "http://docker.mirrors.ustc.edu.cn",
     "http://hub-mirror.c.163.com"
  ],
  "insecure-registries" : [
```

"registry.docker-cn.com",

"docker.mirrors.ustc.edu.cn"

],

4、编写 Dockerfile，这个不在教程里列出，参考单独的文件。也可以根据裸机安装的方法，编写一个。

编译镜像 docker build ./

编译好后，给镜像设置一个 tag，方便使用。docker image tag xxxxxid lotus-worker:1.0

5、新建一个目录比如 miner-docker，在目录内编写 docker-compose.yml

下面是一份 version3 的版本，yml 的格式要求严格，自行编写对应的 yml。

Worker，miner，lotus 都要一一对应一份。

主要注意点是 runtime: nvidia，有了这个才能共享显卡，这个 runtime 对应前面的配置文件中的。

```
version: '3'
services:
  lotus-miner:                （服务名字不能重复）
    container_name: miner （容器名字不能重复）
    image: lotus-miner:1.0 （这里根据自己的镜像改名字）
    ports:
      - "24001:24001"
      - "2345:2345"
    runtime: nvidia
```

environment:　　（以下设置全局变量，路径是容器内路径）

NVIDIA_VISIBLE_DEVICES: all

LOTUS_MINER_PATH: /home/lotus_user/.lotusminer

FIL_PROOFS_PARAMETER_CACHE: /filecoin-proofs-parameters

FIL_PROOFS_PARENT_CACHE: /filecoin_proofs_parent_cache

TMPDIR: /home/temp

BELLMAN_CPU_UTILIZATION: 0.875　　（cpu 和 gpu 计算量，跑 c2 设置 0）

FIL_PROOFS_MAXIMIZE_CACHING: 1

FIL_PROOFS_USE_GPU_COLUMN_BUILDER: 1

FIL_PROOFS_USE_GPU_TREE_BUILDER: 1

FIL_PROOFS_USE_MULTICORE_SDR: 1

FULLNODE_API_INFO: "xxxxxxxxxxxxx "

ulimits:

nproc: 65535

nofile:

soft: 65535

hard: 65535

volumes:　　（映射文件夹，需要重启不丢文件的部分都要映射）

- /data/lotusminer:/home/lotus_user/.lotusminer

- ./config.toml:/home/lotus_user/.lotusminer/config.toml

- /filecoin/filecoin-proof-parameters:/filecoin-proofs-parameters

```
      - /filecoin/filecoin_proofs_parent_cache:/filecoin_proofs_parent_cache
      - /filecoin/sealstorge:/sealstorge
      - /storge/lotus_storge:/storge
      - /filecoin/temp:/home/temp
    logging:
      driver: "json-file"
      options:
        max-size: "20g"
```

6、启动 docker。在目录内，执行 docker-compose up –d 启动容器。docker-compose restart 重启服务 docker-compose down 卸载容器。需要修改映射的都要卸载容器后，再启动一个新容器。

需要查日志的时候，先 docker ps 查看当前需要查的那个容器 id，然后执行 docker logs -f -t --since="2021-01-01" --tail=1000 <id>，可以查最近 1000 条日志，更多参数需要参考 docker 官方文档。

三、初始化项目

1、初始化 lotus

按照安装步骤中，启动 lotus，同步链。

配置 lotus 的配置文件~/.lotus/config.toml

```
[API]
  # Lotus API 的绑定地址
  ListenAddress = "/ip4/127.0.0.1/tcp/1234/http"
  # lotus 无需配置
  RemoteListenAddress = ""
```

```
    # 网络超时时间
    Timeout = "30s"
  # Libp2p 功能设置，filecoin 的节点配置
  [Libp2p]
    # libp2p 绑定地址 — 0 代表随机端口。这里设置公网 ip 地址，如果是路由器映
射的绑定一个端口
    ListenAddresses = ["/ip4/0.0.0.0/tcp/0", "/ip6/::/tcp/0"]
    # 这里可以指定一个宣告地址，留空让系统选择
    AnnounceAddresses = []
    # 这里可以屏蔽宣告地址.
    NoAnnounceAddresses = []
    # 连接管理设置，具体作用不明。
    ConnMgrLow = 150
    ConnMgrHigh = 180
    ConnMgrGrace = "20s"
# Pubsub is used to broadcast information in the network
#这个是广播地址，具体做什么用的，还不清楚
[Pubsub]
  Bootstrapper = false
  RemoteTracer = "/dns4/pubsub-
tracer.filecoin.io/tcp/4001/p2p/QmTd6UvR47vUidRNZ1ZKXHrAFhqTJAD27
rKL9XYghEKgKX"

#这里是检索功能设置
[Client]
    UseIpfs = false
    IpfsMAddr = ""
    IpfsUseForRetrieval = false
    #同时数据传输和存储的最大数量
    SimultaneousTransfers = 20
  #名字设置
[Metrics]
    Nickname = ""
    HeadNotifs = false
# 钱包设置
[Wallet]
    EnableLedger = false

# 费用配置
[Fees]
    DefaultMaxFee = 0.007
```

配置一个 toekn，用来让 miner 连接

lotus auth create-token --perm admin

生成 token，用在 miner 上，来连接这个 lotus

生成一个钱包地址，这个用来做矿工的钱包

lotus wallet new bls

多签名钱包

lotus msig create address1 address2

lotus 的主目录$LOTUS_PATH/keystore 记得备份，非常重要，

lotus wallet set-default <address> 用生成的钱包地址来设置默认的 ower 地址。这个地址先打 1fil 进去，

检查设置是否正确 lotus wallet balance

可以重复再创建一个钱包地址，作为 miner 的钱包，用下面命令可以将钱包的钱打给另外一个地址

lotus send <target address> <FIL amount>

也可以指定钱包

lotus send --from=<sender address> <target address> <FIL amount>

导出钱包私钥，保存好。不要放在互联网上。

lotus wallet export <address> > <address>.key

2、初始化 miner

等 lotus 启动后，先配置 miner，miner 需要配置一些全局环境变量

export BELLMAN_CPU_UTILIZATION=0.875

export FIL_PROOFS_MAXIMIZE_CACHING=1

export FIL_PROOFS_USE_GPU_COLUMN_BUILDER=1

export FIL_PROOFS_USE_GPU_TREE_BUILDER=1

export FIL_PROOFS_USE_MULTICORE_SDR=1

export

FULLNODE_API_INFO=<api_token>:/ip4/<lotus_daemon_ip>/tcp/<lotus_daemo

n_port>/http

export FIL_PROOFS_PARAMETER_CACHE=/path/to/folder/in/fast/disk

export FIL_PROOFS_PARENT_CACHE=/path/to/folder/in/fast/disk2

下载证明文件

lotus-miner fetch-params 32GiB

lotus-miner fetch-params 64GiB(除非确定你要运行 64G 的封装，否则使用

32G)

初始化 miner，注意两个地址都要充币至少 1 个。开始初始化可以不配置存储目

录，在后面配置

lotus-miner init --owner=<address> --worker=<address> --no-local-storage

配置 miner 的 config.toml

```
[API]
  # 绑定 miner API
  ListenAddress = "/ip4/127.0.0.1/tcp/2345/http"
  # 设置内网可访问的地址
  RemoteListenAddress = "127.0.0.1:2345"
  # General network timeout value
  Timeout = "30s"
[Libp2p]
  ListenAddresses = ["/ip4/0.0.0.0/tcp/0", "/ip6/::/tcp/0"]
  AnnounceAddresses = []
```

```
    NoAnnounceAddresses = []
    ConnMgrLow = 150
    ConnMgrHigh = 180
    ConnMgrGrace = "20s"
  [Pubsub]
    Bootstrapper = false
    # FIXME
    RemoteTracer = ""
   #检索和存储服务，filter 可以先关了 RetrievalFilter = "/bin/false"
  [Dealmaking]
    ConsiderOnlineStorageDeals = true
    ConsiderOfflineStorageDeals = true
    ConsiderOnlineRetrievalDeals = true
    ConsiderOfflineRetrievalDeals = true
    PieceCidBlocklist = []
    ExpectedSealDuration = "12h0m0s"
    Filter = "/absolute/path/to/storage_filter_program"

    RetrievalFilter = "/absolute/path/to/retrieval_filter_program"
   [Sealing]

     #最大交易封装等待数量

   MaxWaitDealsSectors = 2

     # 最大多少个扇区等待封装

     MaxSealingSectors = 0

     # 交易封装最大等待数量

     MaxSealingSectorsForDeals = 0

     #新创建的扇区在封装前等待交易的时间

     WaitDealsDelay = "1h0m0s"
  [Storage]

    #存储部分设置，miner 和 worker 分离的做法，就全部设置 false，把
工作都交给 worker 做

     ParallelFetchLimit = 10
     AllowPreCommit1 = true
     AllowPreCommit2 = true
     AllowCommit = true
     AllowUnseal = true

  [Fees]
```

```
        # 费用部分设置，这里都是最大费用，设置高一些，不然会卡消息。这里并不是实际的消息费用
        MaxPreCommitGasFee = "0.5 FIL"
        MaxCommitGasFee = "0.5 FIL"
        MaxWindowPoStGasFee = "50 FIL"
```

完成配置后，设置参数

先定义 miner 的存储目录。Seal 为封装工作目录（高速 nvme），store 为存储目录（sata 存储集群，或者 nfs 映射目录）

lotus-miner storage attach --init --seal <PATH_FOR_SEALING_STORAGE>

lotus-miner storage attach --init --store <PATH_FOR_LONG_TERM_STORAGE>

将你的公网 ip 发布

lotus-miner actor set-addrs /ip4/<YOUR_PUBLIC_IP_ADDRESS>/tcp/24001

3、启动 lotus-worker

worker 第一次启动会自动初始化，需要设置以下变量，注意环境变量在系统中的作用域。

NVIDIA_VISIBLE_DEVICES=all　　(如果多显卡，可以设置使用特定的显卡)

MINER_API_INFO="这里写 miner 的 API:/ip4/Miner 的 IP 地址/tcp/minerIP 的 API 端口/http"

LOTUS_WORKER_PATH=/home/lotus_user/　（指定 work 的主目录，PC1、PC2 需要快速的 nvme 盘）

设置证明文件缓存位置（多个 worker 可以共用这个目录，指定即可）

FIL_PROOFS_PARAMETER_CACHE: /home/filecoin-proofs-parameters

设置计算证明的缓存目录，不设置默认放在 LOTUS_WORKER_PATH 目录下

FIL_PROOFS_PARENT_CACHE: /home/lotus_user/filecoin_proofs_parent_cache

TMPDIR: /home/lotus_user/cache　这个目录需要创建好，缓存目录，需要高速磁盘。

设置 cpu 和显卡的工作量比，如果单开 c2，优化设置 0

BELLMAN_CPU_UTILIZATION=0.825

FIL_PROOFS_MAXIMIZE_CACHING=1

这两项开启 gpu 计算，PC2，C2 阶段必设置

FIL_PROOFS_USE_GPU_COLUMN_BUILDER= 1

FIL_PROOFS_USE_GPU_TREE_BUILDER=1

开启 sdr 功能，在 PC1 中尤为重要

FIL_PROOFS_USE_MULTICORE_SDR=1

设置 gpu 计算的时候，每次计算的大小，根据显卡自行测试调整

FIL_PROOFS_MAX_GPU_TREE_BATCH_SIZE=800000

FIL_PROOFS_MAX_GPU_COLUMN_BATCH_SIZE=600000

设置显卡型号，GeForce GTX 3080 为显卡型号，8704 为 gpu 数量，可以在 NVIDIA 官网查询。

BELLMAN_CUSTOM_GPU="GeForce GTX 3080:8704"

RUST_LOG=Trace　　（日志等级）

设置完环境变量，就可以启动 worker 了，注意设置 ip 和端口，设置本进程接的任务种类。需要开启的任务设置 true，否则设置 false。

lotus-worker run --listen=本机 IP:2339 --precommit1=true--precommit2=false --commit=false --addpiece=false --unseal=false

双开 worker 的方法：

建议使用 screen 工具，Ubuntu 自带，CentOS 需要安装一下。

输入 screen 打开一个新窗口。按上述方法启动 worker。

按住 CTRL，分别按 A，D，挂起窗口在后台运行。

screen –ls 查询后台运行的窗口，screen –r ID，可以进入指定 ID 的窗口查看。

4、运行测试工具 lotus-bench

lotus-bench 是一个测试工具，需要在编译那一步，加上 make lotus-bench 编译二进制。默认并不会安装。

lotus-bench 用来测当前环境下，每个步骤的花费时间。

和 worker 一样设置环境变量，可以不设置 miner api 部分，lotus-bench 是可以脱离 miner 独立运行的。注意运行的目录要有足够的空间，所以目录必须要设置一下，否则写到/var 和/home 下，可能影响测试结果。

简单讲下启动命令。

lotus-bench sealing --save-commit2-input=/home/lotus_user/c2value --storage-dir=/home/lotus_user --sector-size 32GiB

四、 常用维护指令

lotus 的维护主要是 lotus，miner，worker 三个进程，还有一个 lotus-shed，这个需要安装的时候 make lotus-shed 另外编译。以下命令以 1.4.1 版本为参照，后续版本可能存在一些改动和删减

1、 lotus 命令

lotus daemon　　启动 lotus 节点

lotus daemon stop 停止 lotus 节点

lotus daemon 可用参数

--api value　　设置 api 默认 1234

--genesis value 指定第一次节点运行的 genesi 文件

--bootstrap 引导程序（默认 true）

--import-chain value　　首次运行，导入的链同步文件，可以 url 或者本地文件。这个方式是验证数据的

--import-snapshot value　导入链快照，可以 url 或者本地文件，这个方式不验证数据

--halt-after-import　　导入链文件后不启动同步进程。如：--import-snapshot --halt-after-import <filename>

--pprof value　指定 cpu 配置文件

--profile value　指定节点类型

--api-max-req-size　value　JSON RPC 服务最大接受请求的大小。默认 0

--restore value　　从备份文件中恢复

--restore-config value　　从备份文件中恢复时指定 config 文件

lotus backup [选项] [备份路径]　　备份节点

在线备份：为了安全因素，LOTUS_BACKUP_BASE_PATH 变量必须设置在 lotus 主目录下。

--offline 离线备份，默认 false。离线备份指在节点不运行的情况下备份

lotus 基础命令

lotus send [命令选项][目标地址][金额]　　发送 FIL

--from value　指定从哪个钱包转出

--gas-premium value　　指定 gas 价格，单位 attoFil，默认 0

--gas-feecap value　　指定 gas fee cap,单位 attoFil，默认 0

--gas-limit value　　指定 gas limit 价格 ，默认 0

--nonce value　　指定 nonce

--method value　　指定调用方法

--params-json value　指定调用参数，json 格式

--params-hex value　　指定调用参数，hex 格式

--force　　如果可能是 SysErrInsufficientFunds，必须指定

lotus wallet [命令选项][参数]　管理钱包

lotus new　　创建钱包

lotus wallet new　[bls|secp256k1 (默认 secp256k1)]

lotus wallet list　　列出钱包信息

--addr-only，-a　　只打印地址

--id，-I　　输出 ID 地址

--market，-m　　输出市场账户

lotus client 交易，存储数据，检索数据。这个命令分成 data、retrieval、storage、util 四个部分的功能

data 部分：

lotus client import [命令选项] [输入路径]

--car　　从 car 文件导入，默认 false

--quiet　　静默模式，只输出 CID

lotus client drop　　[import ID...]　　放弃导入

lotus client local　　列出本地导入的数据

lotus client stat <cid>　　打印一个本地存储文件的信息

retrieval 部分：

lotus client find [命令选项] <cid>　　在网络上找数据

--pieceCid value　　从一个指定的 Piece CID 检索数据

lotus client retrieve [命令选项] [dataCid outputPath]　　在网络上检索数据

--from value　　从一个发送交易的地址

--car　　导出一个 car 文件

--miner value　　检索的 miner 地址，如果不指定，将使用本地搜索

-- maxPrice value　　最高价格，默认 1Fil

--pieceCid value　　从一个指定的 Piece CID 检索

storage 部分：

lotus client deal [命令选项] [dataCid miner price duration]　　初始化一个存储交易

--manual-piece-cid value　　手动指定 piece 提交(dataCid 必须是一个 car 文

件）

--manual-piece-size value　　如果手动指定 piece cid，通常指定大小。默认 0

--from value　　指定支付交易的地址

--start-epoch value　　指定交易开始的区块

--fast-retrieval　　指示数据可用于快速检索，默认 true

--verified-deal　　指示交易计入验证客户端总数，默认 true（如果 client verified，否则 false）

--provider-collateral value　指定矿工提供的质押

lotus client query-ask [命令选项] [minerAddress]　　寻找一个矿工询价

--peerid value　　指定 peer ID

-- size value　　数据大小（字节）默认 0

--duration　value　交易期限，默认 0

lotus client get-deal　　打印详细交易信息

lotus client list-asks　　列出询单

--by-ping　　默认 false

lotus client list-deals [命令选项] 列出存储市场的交易

--verbose，-v　　打印详细

--color　　使用颜色区分，默认 true

--show-failed　　显示失败的交易

--watch 观察模式，实时观察

lotus client deal-stats 打印本地存储交易的统计

--newer-than value 时间范围，默认 0s

util 部分：

lotus client commP [命令选项] [inputFile] 计算 CAR 文件的 piece-cid

lotus client generate-car [inputPath outputPath] 生成一个 car 文件

lotus client balances 打印存储交易市场 client 账户信息

lotus client list-transfers 列出正在进行数据传输的交易

--verbose, -v 打印详细

--color 显示颜色

--completed 显示完成数据传输的

--watch 观察模式，实时显示

--show-failed 显示失败/取消的传输

lotus msig 多签名钱包交互命令。

用法：lotus msig command [command options] [arguments...]

lotus msig create [命令选项] [address1 address2 ...] 创建多签名钱包

--required value 设置需要批准的数量(如果省略，使用签名者的数量)

--value value 初始资金用于多签名，默认 0

--duration value 资金解锁的期限，默认 0

--from value　　用于发送创建钱包的消息的地址

Inspect a multisig wallet　[命令选项] [address]　检查多签名钱包

--vesting　　包含归属细节（默认 false）

--decode-params　　交易解码参数，默认 false

lotus msig propose [命令选项] [多签名地址 目标地址 value <methodId methodParams> (optional)]　　发起一个多签名交易

--from value　发送消息的地址

lotus msig propose-remove [命令选项] [multisigAddress signer]　移除一个签名者

--decrease-threshold　是否减少所需签名的数量，默认 false

--from value　发送消息的地址

Approve a multisig message [命令选项] <multisigAddress messageId> [proposerAddress destination value [methodId methodParams]]　批准一条多签名消息

--from value　　发送消息的地址

lotus msig add-propose [命令选项] [multisigAddress signer]　　提出增加一个签名者

--increase-threshold　是否增加所需签名的数量，默认 false

--from value　　发送消息的地址

lotus msig add-approve [命令选项] [multisigAddress proposerAddress txId newAddress increaseThreshold]　　批准添加签名者的消息

--from value　　发送消息的地址

lotus msig add-cancel [命令选项] [multisigAddress txId newAddress increaseThreshold]　　取消一条添加签名者的消息

--from value　　发送消息的地址

lotus msig swap-propose [命令选项] [multisigAddress oldAddress newAddress]　交换签名者

--from value　发送消息的地址

lotus msig swap-approve [命令选项] [multisigAddress proposerAddress txId oldAddress newAddress]　批准一条交换签名者的消息

--from value　发送消息的地址

lotus msig swap-cancel [命令选项] [multisigAddress txId oldAddress newAddress]　　删除一条交换签名者的消息

--from value　　发送消息的地址

lotus msig vested [命令选项] [multisigAddress]　获取两个区块间归属多签名金

额

--start-epoch value　　开始的区块

--end-epoch value　　结束的区块

lotus msig propose-threshold [命令选项] <multisigAddress newM>　　在一个账户上设置一个不同的阈值

--from value　　发送消息的地址

lotus paych　　管理支付渠道

lotus paych command [命令选项]

lotus paych add-funds [命令选项] [fromAddress toAddress amount]　　在fromAddress 地址到 toAddress 地址的付款渠道中添加资金

--restart-retrievals　　在此付款渠道上重启停滞的检索交易

lotus paych list　　列出所有本地注册的支付渠道

lotus paych voucher　　支付渠道凭证交互

lotus paych voucher create [命令选项] [channelAddress amount]　　创建一个签名支付渠道凭证

--lane value　　指定要使用的支付渠道通道

lotus paych voucher check　[channelAddress voucher]　检查付款渠道凭证的有效性

lotus paych voucher add [channelAddress voucher]　　在本地数据库添加一个付款渠道凭证

lotus paych voucher list [命令选项] [channelAddress] 列出一个付款渠道的存储凭证

--export　　以序列化字符串打印凭证

lotus paych voucher best-spendable [command options] [channelAddress]

打印每个通道当前可使用的具有最高价值的凭证

--export　　以序列化字符串打印凭证

lotus paych voucher submit [channelAddress voucher]　　提交凭证上链更新付款渠道状态

lotus paych settle [channelAddress]　　结算一个付款渠道

lotus paych status-by-from-to [fromAddress toAddress]　　通过发送方和接收方显示有效支付渠道的状态

lotus paych collect [channelAddress]　　收集付款渠道资金

lotus 的开发者命令：

lotus auth　　管理 RPC 权限

lotus auth create-token [命令选项] 创建 token

--perm value 允许访问的权限：read, write, sign, admin

lotus auth api-info [命令选项] 查看 token

--perm value 允许访问的权限：read, write, sign, admin

lotus mpool 管理消息池

lotus mpool pending 获取待处理消息

--local 打印本地钱包地址待处理消息

--cids 仅仅打印消息 cids

--to value 返回 to 某个地址的消息

--from value 返回 from 某个地址的消息

lotus mpool clear[命令选项] 清理消息池

--local 清理本地消息

--really-do-it 必须指定这个动作才能执行命令

lotus mpool sub 订阅消息称变动，实时显示

lotus mpool stat 打印消息池状态

--local 仅打印本地消息池

--basefee-lookback value 查看 gas 费最低的块的数量

lotus mpool replace [命令选项] <from nonce> | <message-cid>

替换消息池中的一条消息

--gas-feecap value　　设置 gas feecap（燃烧和付给矿工的费用，attoFIL 单位）

--gas-premium value　设置消息的 gas 价格（付给矿工的费用，attoFIL 单位）

--gas-limit value　　设置小的 gas limit（GasUnit）

--auto　　给指定的消息自动替代价格

--max-fee value　　设置消息的最大价格 X attoFIL，（在 auto 模式中使用）

lotus mpool find [命令选项][参数]　查找消息池中一条消息

--from value　　查找 from 地址的消息

--to value　　查找 to 地址的消息

--method value　　查找给定方法的消息

lotus mpool config [new-config]　　获取/设置当前消息池配置

获取如下：

{"PriorityAddrs":null,"SizeLimitHigh":30000,"SizeLimitLow":20000,"ReplaceByFeeR

atio":1.25,"PruneCooldown":60000000000,"GasLimitOverestimation":1.25}

lotus mpool gas-perf [命令选项]

检查消息池中的 gas 表现

--all　　打印所有消息池中的消息的 gas 的表现（仅仅打印本地的）

lotus state　　与 filecoin 链状态交互和查询命令

lotus state power [<minerAddress> (可选参数)]　　查询矿工算力

lotus state sectors [minerAddress]　　查询某个矿工的扇区

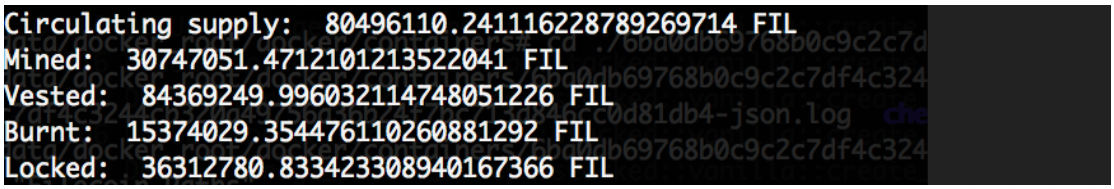
lotus state active-sectors　[minerAddress]　查询某个矿工的激活扇区

lotus state list-actors　　查询链上所有的角色

lotus state list-miner　　查询链上所有的矿工

lotus state circulating-supply[命令选项] 查询目前链上供应量

--vm-supply　　计算 VM 供应的近似值，就是显示各个部分的量



lotus state sector [miner address] [sector number]　　获取矿工扇区信息

lotus state get-actor　[actorrAddress]　　查询某个角色信息

lotus state lookup [命令选项] [address]　　查询对应的 id 地址

--reverse, -r　　反向查询

lotus state replay [命令选项] <messageCid>　　重播一个特定消息

--show-trace　　打印完整执行跟踪

--detailed-gas　　打印给定消息的详细 gas 费支付

lotus state sector-size [minerAddress]　　查询矿工的扇区大小

lotus state read-state [actorAddress]　　查询一个角色的 json 形式的信息

lotus state list-messages [命令选项]　　列出符合条件的链上消息

--to value　　指定 to 的地址

--from value　　指定 from 的地址

--toheight value　　不查询给出高度之前的区块

--cids　　消息 ID

lotus state compute-state [命令选项]　执行状态计算

--vm-height value　设置 vm 高度

--apply-mpool-messages　将消息从内存池应用于计算的状态

--show-trace　打印指定 tipset 的完整执行跟踪

--html　生成 html 报告

--json　生成 json 报告

--compute-state-output value　包含预先存在的计算状态的输出的 json 文件，以无需重新运行状态更改生成 html 报告

--no-timing　不在 html 跟踪中显示时间信息

lotus state call [options] [toAddress methodId <param1 param2 ...> (可选)]

调用一个本地角色的方法

--from value　默认 f00

--value value　指定调用值的字段

--ret value　指定如何处理输出（auto, raw, addr, big）

lotus state get-deal [dealId] 查看上链的交易信息

lotus state wait-msg [command options] [messageCid]　等待一个消息上链

--timeout value　超时时间，默认 10m

lotus state search-msg [messageCid]　查看消息是否上链

lotus state miner-info [minerAddress]　查看 miner 的信息

lotus state market　检查存储市场参与者

lotus state market balance <address> 获取给定帐户的市场余额（锁定和托管）

lotus state exec-trace <messageCid>　获取给定消息的执行跟踪

lotus state network-version　　获取网络版本号

lotus chain　　和 filecoin 区块链交互命令

lotus chain head　　打印链头部

lotus chain getblock [options] [blockCid]　　获取一个区块及详细信息

--raw　　仅打印原始区块的头部

lotus chain read-obj [objectCid]　　读取一个对象的原始字节

lotus chain delete-obj [options] [objectCid]　　从链数据库中删除一个对象（从链数据库删除错误对象可能导致同步问题）

--really-do-it

lotus chain stat-obj [options] [cid]　　收集对象的大小和 ipld 链接数

--base value　　忽略在这个对象中找到的链接

lotus chain getmessage [messageCid]　　通过 cid 获取和打印一条消息

lotus chain sethead [options] [tipsetkey]　　手动设置本地节点头部区块 tipset（用在恢复上）

--genesis　　重置 genesis

--epoch value　　重置指定高度的区块头部

lotus chain list [options]　　查看链的一部分

--height value　　高度

--count value　　默认 30

--format value　　指定输出 tipset 的格式。默认 "<height>: (<time>) <blocks>"

--gas-stats　　查看这条链的 gas 费统计

lotus chain get [options] [path]　　通过路径获取链 DAG 节点

通过制定路径获取 ipld 节点

lotus chain get /ipfs/[cid]/some/path

路径前缀:

- /ipfs/[cid], /ipld/[cid] – 遍历 IPLD 路径
- /pstate – 从 head.ParentStateRoot 便利

注意:

可以使用特殊的路径元素遍历某些数据结构:

- /ipfs/[cid]/@H:elem - get 'elem' from hamt
- /ipfs/[cid]/@Hi:123 - get varint elem 123 from hamt
- /ipfs/[cid]/@Hu:123 - get uvarint elem 123 from hamt
- /ipfs/[cid]/@Ha:t01 - get element under Addr(t01).Bytes
- /ipfs/[cid]/@A:10　- get 10th amt element
- .../@Ha:t01/@state　- get pretty map-based actor state

类型列表 types:

- raw
- block
- message
- smessage, signedmessage
- actor
- amt
- hamt-epoch

- hamt-address

- cronevent

- account-state

--as-type value 指定输出类型

--verbose 详细模式

--tipset value 传递逗号分隔的 CID 数组

lotus chain bisect 事件的二等分链

lotus chain bisect [minHeight maxHeight path shellCommand

<shellCommandArgs (if any)>]

lotus chain export [options] [outputPath] 导出链到一个 car 文件

--tipset value

--recent-stateroots value 指定要包含在导出中的最近状态根数

--skip-old-msgs 跳过旧数据

lotus chain slash-consensus [options] [blockCid1 blockCid2] 报告共识错误

--from value 指定从某个账户报告

--extra value 额外的块 id

lotus chain gas-price 预估 gas 费

lotus chain inspect-usage [options] 检查指定 tipset 的块空间使用情况

--tipset value　　指定 tipset。默认　“@head”

--length value　　检查块空间使用的链长度.默认 1

--num-results value　　每个类别打印的结果数量，默认 10

lotus chain decode　　解码各种类型

lotus chain decode params [options] [toAddr method params]　　解码消息参数

--tipset value

--encoding value　　指定输入编码，默认 base64

lotus chain encode　　编码各种类型

lotus chain encode params [options] [toAddr method params]　　编码消息参数

--tipset value

--encoding value　　指定输入编码，默认 base64

lotus chain disputer　　与 window post disputer 互动

--max-fee　　设置每个 DisputeWindowedPoSt 最高花费 xFil

--from value　　可选指定发送消息的账户

lotus chain disputer start [options] [minerAddress]　开始 window post disputer

--start-epoch value　　仅从这个区块开始 PoSts disputer

lotus chain disputer dispute　[minerAddress index postIndex]　　发送一个 DisputeWindowedPoSt 的消息

lotus log　　日志管理

lotus log list　　列出日志系统

lotus log set-level [options] [level]　设置日志级别

--system value　　限制到日志系统

可用级别 Levels:

debug

info

warn

error

环境变量:

GOLOG_LOG_LEVEL – 对所有日志系统默认级别

GOLOG_LOG_FMT　- 改变输出日志格式 (json, nocolor)

GOLOG_FILE　　- 日志写入文件

GOLOG_OUTPUT　- 指定是否输出到文件, stderr, stdout 或者组合，比如 file+stderr

lotus wait-api　　等待 lotus api 上线

lotus fetch-params [sectorSize]　　抓取证明参数

lotus 网络命令：

lotus net　　管理 P2P 网络

lotus net peers　　打印节点

--agent, -a　　打印代理名字

lotus net connect [peerMultiaddr|minerActorAddress]　　链接到一个节点

lotus net listen　　lotus 监听地址

lotus net id　　lotus 节点 ID

lotus net findpeer [peerId]　　找到给出的 peerid 的地址

lotus net scores　　打印节点的 pubsub 分数

--extended　　以 json 方式打印扩展信息

lotus net reachability　　打印链接到互联网的信息

lotus net bandwidth　　打印宽带使用信息

--by-peer　　安节点打印宽带使用

--by-protocol　　按协议打印宽带使用

lotus net block　　管理网络连接选通规则

lotus net block add[comand]　　添加链接选通规则

lotus net block add peer <Peer>　　阻止一个节点

lotus net block add ip <IP> 阻止一个 IP

lotus net block add subnet <CIDR> 阻止一个 IP 子网

lotus net block remove [comand] 删除一条规则

lotus net block remove peer <Peer>　　解除一个节点

lotus net block remove ip <IP> 解除一个 IP

lotus net block remove subnet <CIDR> 解除一个 IP 子网

lotus sync　　和链同步检查和交互命令

lotus sync status　　检查同步状态

lotus sync wait　　等待同步完成

--watch　　实时显示，在同步后不退出

lotus sync mark-bad　[blockCid]　　将给定的块标记为坏块，将阻止同步到包含该块的链

lotus sync unmark-bad [options] [blockCid]　　取消标记给定的块为坏块，使其可以同步到包含该块的链

--all　　删除整个坏块缓存

lotus sync check-bad　[blockCid]　检查指定的块是否标记为不良，以及什么原因

lotus sync checkpoint [options] [tipsetKey]　将某个 tipset 标记为检查点；节点将永远不会离开这个 tipset

--epoch value　　检查点 tipset 在指定的区块上

六、 常见错误

1、　如何删除已经过期的扇区和错误的扇区

2、　C2 速度 1 小时多，同样配置比标准速度慢。

单独的 c2 进程上，设置参数 BELLMAN_CPU_UTILIZATION=0，可提高 gpu 的使用率。

3、 P1 速度比同等配置标准速度慢好几个小时，是什么原因。

首先，检查缓存盘是否配置再 nvme 区。其次检查 sdr 编译的时候有没有加入 export RUSTFLAGS="-C target-cpu=native -g"

export FFI_BUILD_FROM_SOURCE=1

开启

export FIL_PROOFS_USE_MULTICORE_SDR=1

export FIL_PROOFS_MULTICORE_SDR_PRODUCERS=1 （cpu 生产者，根据 cpu 类型设置）

如果接一个 p1 任务正常，多开了就慢，那么设置 ulimit -SHn 65535

4、 C2 启动后不接任务，重启 C2 也没反应，日志并无错误。

lotus mpool pending –local

检查消息池，看是否消息卡住

5、 错误 beignet-opencl-icd: no supported GPU found, this is probably the wrong opencl-icd package for this hardware

删除 apt-get remove beignet-opencl-icd

6、 都优化了，但是 p1 还是非常慢，是什么原因？

检查服务器的 cpu 支持 SDR 吗？比如 AMD7302 就不能很好地支持。

7、 初始化 miner 卡在 waiting for confirming

检查消息池 lotus mpool pending –local

如果有未发出的消息进行 replace 操作让消息发出，自行替换参数，gas-premium 需要高于原来值的 25%

lotus mpool replace --gas-feecap 10000000000 --gas-premium <premium> --gas-limit <limit> <from> <nonce>

8、 卡消息怎么处理？

同第 7 个问题

9、 内存到 30%就开始用 swap

echo 0 > /proc/sys/vm/swappiness

/etc/sysctl.conf 增加一行 vm.swappiness = 0