Parity Documentation: Miners

Ethcore.io resources can be found here: https://github.com/ethcore/parity

Overview

Parity is designed to be fully compatible with `ethminer`. If you're already using a combination of `geth` and `ethminer` then you'll find mining with Parity very familiar.

First get a Parity node up and running (either build yourself or install one of the packages; the Quick-start (https://github.com/ethcore/parity/wiki/Quick-start) guide can help you).

Next, you'll need to install `ethminer`.

Prepare ATI rigs for mining

Install Ubuntu 14.04.4 on your rig Ubuntu Trusty
Update system and install ATI drivers. Open the terminal and run the following commands:

```
sudo su
apt-get update
apt-get dist-upgrade -y
apt-get install fglrx fglrx-core fglrx-amdcccle -y
apt-get autoremove -y
apt-get clean
nano /etc/defaut/grub
```

Add *nogpumanager* in GRUB_CMDLINE_LINUX_DEFAULT="quiet nosplash nogpumanager"
save Ctrl+O and exit Ctrl+X

```
update-grub
#Initial all GPU adapters
aticonfig --adapter=all --initial -f
aticonfig --adapter=all --od-enable
reboot
```

Everything is ready. You now need to install the miner and configure it.

Rolling your own

There are packages built for various platforms, though if all else fails, you can just build it straight from the main repository (OpenCL only):

```
#Install dependencies
```

```
sudo apt-get update
sudo apt-get -y install software-properties-common
add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install git cmake libcryptopp-dev libleveldb-dev libjsoncpp-dev libjson-rpc-cpp-dev
libboost-all-dev libgmp-dev libreadline-dev libcurl4-gnutls-dev ocl-icd-libopencl1 opencl-headers
mesa-common-dev libmicrohttpd-dev build-essential -y
git clone https://github.com/ethereum/cpp-ethereum
cd cpp-ethereum
mkdir build
cd build
cmake -DBUNDLE=miner ..
make
```

`ethminer` can be found in the `libethereum/ethminer` directory; you'll probably want to copy it to somewhere in your `$PATH`.

Genoil ethminer (source)

Building on Ubuntu.

Ubuntu 14.04+ (16.04 not support) OpenCL only (for AMD cards)

```
sudo apt-get update
sudo apt-get -y install software-properties-common
add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install git cmake libcryptopp-dev libleveldb-dev libjsoncpp-dev libjson-rpc-cpp-dev
libboost-all-dev libgmp-dev libreadline-dev libcurl4-gnutls-dev ocl-icd-libopencl1 opencl-headers
mesa-common-dev libmicrohttpd-dev build-essential -y
git clone https://github.com/Genoil/cpp-ethereum/
cd cpp-ethereum/
mkdir build
cd build
cmake -DBUNDLE=miner ..
make
```

You can then find the executable in the ethminer subfolder

Ubuntu 14.04. OpenCL + CUDA (for NVIDIA cards)

```
wget
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1404/x86_64/cuda-repo-ubuntu
1404_7.5-18_amd64.deb
sudo dpkg -i cuda-repo-ubuntu1404_7.5-18_amd64.deb
sudo apt-get -y install software-properties-common
sudo add-apt-repository -y ppa:ethereum/ethereum
sudo apt-get update
```

```
sudo apt-get install git cmake libcryptopp-dev libleveldb-dev libjsoncpp-dev libjson-rpc-cpp-dev
libboost-all-dev libgmp-dev libreadline-dev libcurl4-gnutls-dev ocl-icd-libopencl1 opencl-headers
mesa-common-dev libmicrohttpd-dev build-essential cuda -y
git clone https://github.com/Genoil/cpp-ethereum/
cd cpp-ethereum/
mkdir build
cd build
cmake -DBUNDLE=miner ..
make
```

You can then find the executable in the ethminer subfolder

Building on Windows

Download or clone this repository
Download and install Visual Studio 12 2013 and CMake
Run getstuff.bat in cpp-ethereum/extdep
Open a command prompt and navigate to cpp-ethereum directory

```
mkdir build
cd build
cmake -DBUNDLE=cudaminer -G "Visual Studio 12 2013 Win64" ..
```

If you don't want/need CUDA support, use "miner" instead of "cudaminer". This will only compile OpenCL support to speed up compilation a bit, you can add -DCOMPUTE=xx , where x is your CUDA GPU Compute version * 10. i.e -DCOMPUTE=52 for a GTX970.
You may disable stratum support by adding -DETH_STRATUM=0
When CMake completes without errors, on ethereum.sln created in the build directory in Visual Studio
Set "ethminer" as startup project by right-clicking on it in the project pane
Build. Run

Note. In order to improve the function, such as stratum and generation of DAG in GPU memory, you need to switch to the branch 110. Run this command after cloning and switching to a folder cpp-ethereum.

```
git checkout 110
git pull
```

Official ethminer

Ubuntu

To install `ethminer` on Ubuntu, you can use the package manager:

```
sudo add-apt-repository ppa:ethereum/ethereum
sudo apt-get update
sudo apt-get install ethminer
```

Windows & OSX

Download and install latest release from GitHub

Starting it

Once you have a synced Parity node running the JSONRPC interface, you'll be able to mine with ethminer. You'll probably want to set the destination address (where the rewards go). If you have an address already, great. If you don't, then you can make one in Parity with:
parity account new

You'll be asked for a password and be given an address. Once done, you should run parity and tell it to mine to that address when required. Supposing your address is `0037a6b811ffeb6e072da21179d11b1406371c63`, then you would run:

```
parity --author 0037a6b811ffeb6e072da21179d11b1406371c63
```

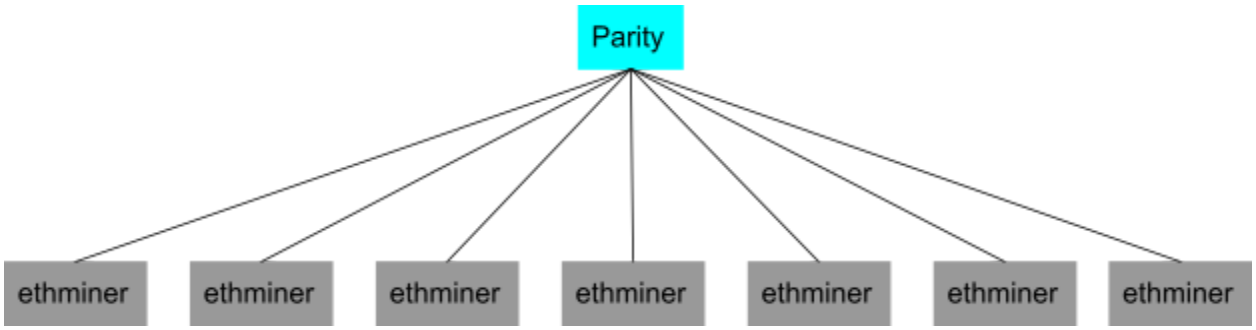
Once Parity is running and synced, running `ethminer` with work without any further configuration. e.g. run `ethminer -G --opencl-device 0` to GPU mine on the first OpenCL device found.

Solo mining with Parity

For solo mining on Parity there are two alternatives:

1. Mining directly on Parity.
2. Mining on Parity via a proxy.

Mining directly on Parity



To mine directly, run the following command line to start Parity (we use the example of a real working configuration):

```
parity --jsonrpc-addr 192.168.1.2 --maxpeers 100 --cache 3062 --extradata
"your mark" --author 0037a6b811ffeb6e072da21179d11b1406371c63 --identity "you
node info" --dapps-interface 192.168.1.2 --dapps-user "admin" --dapps-pass
"password"
```

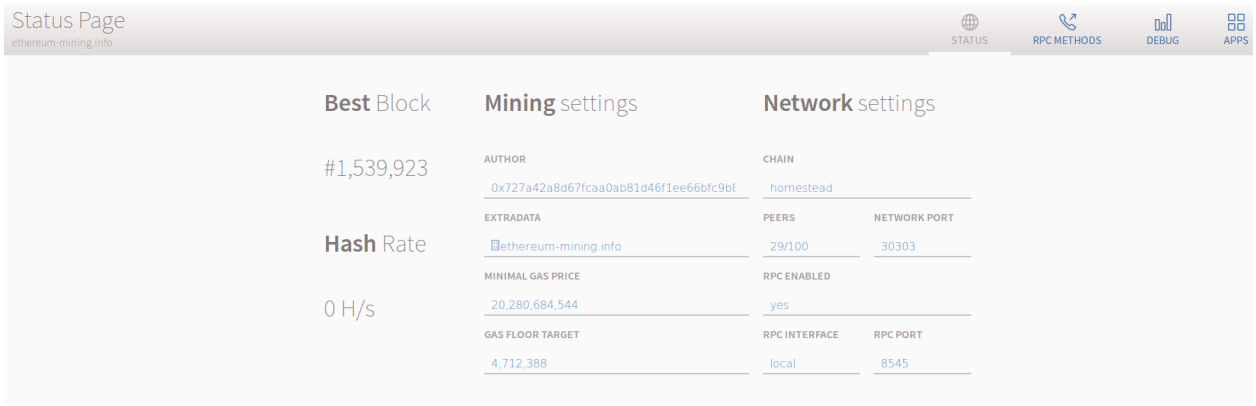
Run parity --help to see all the available options.

For ethminer:

```
ethminer -G -F http://192.168.1.2:8545
```

This assumes that the node is located on network address `192.168.1.2`.

The following status page will be available to you via a web interface and it provides a summary of the hashrate and the debug window on node:



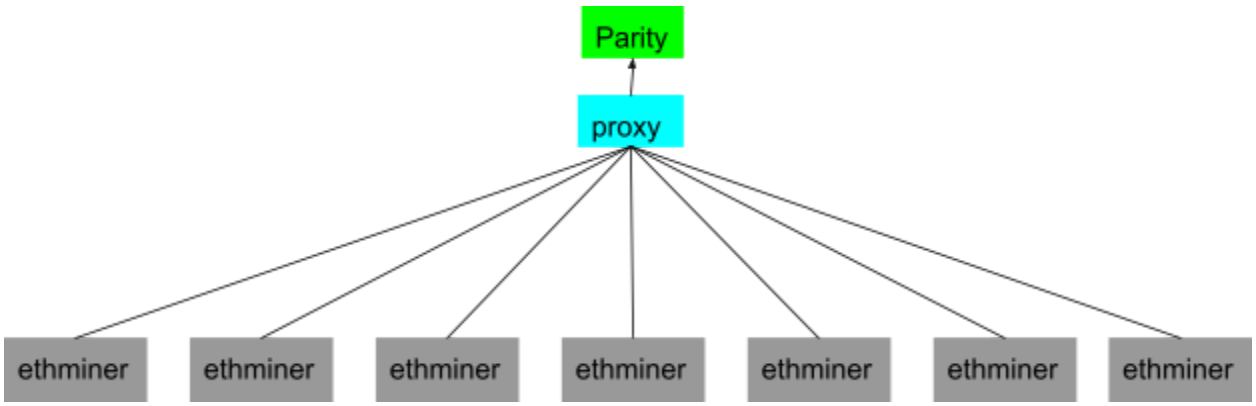
To use it, use a web browser to go to the site located on the host's 8080 port. You can use the admin/password credentials to login (these credentials should be changed on the CLI for security).

Please note:

- There is no need to store the key locally on the node
- This eliminates the possibility of the theft of Ether
- It minimises the ping time between the nodes and as a consequence increase the probability of the mining node finding a reward solution

The disadvantage of this alternative is the difficulty surrounding the mining algorithm. The level of difficulty is approximately equal to the difficulty of mining on the network, which is to say, very high. As such, the nodes will rarely report any solutions and “share accepted” will therefore be highly volatile (1 valid share = 1 valid block).
Also, in the case of a stoppage or failure of the node, the entire mining rig stops working. A possible workaround for this problem is elaborated below.

Mining using proxies



To reduce the level of difficulty for each `ethminer` instance and to get a visual overview over each and every rig, we can utilise [ether-proxy].

The command line to initialise Parity remains the same; there is an additional configuration file to launch a proxy which is as follows:

```
{
    "threads": 4,
    # ^^^ depending on the load the number of threads can be increased

    "proxy": {
       "listen": "192.168.1.2:8546",
       # ^^^ proxy can be run together with node and on the other computer
       "clientTimeout": "3m",
       "blockRefreshInterval": "100ms",
       "hashrateWindow": "15m",
       "submitHashrate": false,
       "luckWindow": "24h",
```

```
        "largeLuckWindow": "72h"
    },

    "frontend": {
        "listen": "192.168.1.2:8081", # the web interface for proxy
        "login": "admin",
        "password": "admin"
    },

    "upstreamCheckInterval": "30s",
    "upstream": [
        {
               "pool": false, # the main node
               "name": "parity",
               "url": "http://192.168.1.2:8545",
               "timeout": "60s"
        },
        {
               "pool": false, # backup node
               "name": "backup",
               "url": "http://192.168.1.3:8545",
               "timeout": "60s"
        },
        {
               "pool": true, # backup pool
               "name": "coinlab",
               "url":
"http://eth.coinlab.com:8888/0x727a42a8d67fcaa0ab81d46f1ee66bfc9b8789ac/proxy"
,
               "timeout":"60s"
        },

        {
               "pool": true, #backup pool
               "name": "suprnova",
               "url": "http://eth-high.suprnova.cc:3000/rig.1/100",
               "timeout": "60s"
        }
    ]
}
```

EtherProxy

Hashrate	1 739 986	**Accepted**	10
Hashrate 24h	1 851 649	**Rejected**	0
Total Miners	31	**Miners Timed Out**	4
Miners Online	27	**Last Submission**	6 hours ago

Block height: 1540076 **Difficulty:** 37 482 607 638 848 **Next Epoch:** in 80 hours **Mode:** SOLO

Blocks 24h: 3 **Shares/Diff 24h:** 125,84 % **Blocks 72h:** 10 **Shares/Diff 72h:** 118,89 %

Run: `ethminer -F http://0.0.0.0:8546/miner/<diff>/<id>`

Upstream

Name	Url	Accepted	Rejected	Fails
parity	http://127.0.0.1:8545	10	**0**	10
coinlab	http://92.255.232.100:8888/0x727a42a8d67fcaa0ab81d46f1ee66bfc9b8789ac/proxy	269	**0**	0
node	http://127.0.0.1:8545	0	**0**	3
suprnova	http://eth-high.suprnova.cc:3000/maniakfall.101a/100	0	**0**	0

Miners

ID	IP	HR	HR 24h	Last Share	Accepted	Rejected	Upstream Accepted	Upstream Rejected
10A	178.95.194.19	36 000	43 791	4 seconds ago	18 747	**1**	8	0
10M	37.52.244.194	46 666	55 752	3 seconds ago	13 570	**2**	7	0
11A	178.95.194.19	50 666	46 875	8 seconds ago	19 931	**0**	15	0
11M	37.52.244.194	48 888	55 219	41 seconds ago	13 378	**2**	9	0
11PC	192.168.1.11	0	0	yesterday	3	**0**	0	0
12A	178.95.194.19	53 333	49 780	1 minute ago	8 362	**0**	2	0
12M	37.52.244.194	60 000	625	28 seconds ago	6 916	**0**	0	0
15M	37.52.244.194	45 555	49 826	11 seconds ago	12 259	**2**	5	0
16M	37.52.244.194	38 888	51 215	2 seconds ago	12 524	**1**	6	0
17M	37.52.244.194	45 555	32 280	2 seconds ago	6 864	**2**	7	0
18M	37.52.244.194	48 888	47 835	14 seconds ago	11 986	**2**	4	0

```
ethminer -G F http://192.168.1.2:8546/miner/DIFF/ID
```

Where:

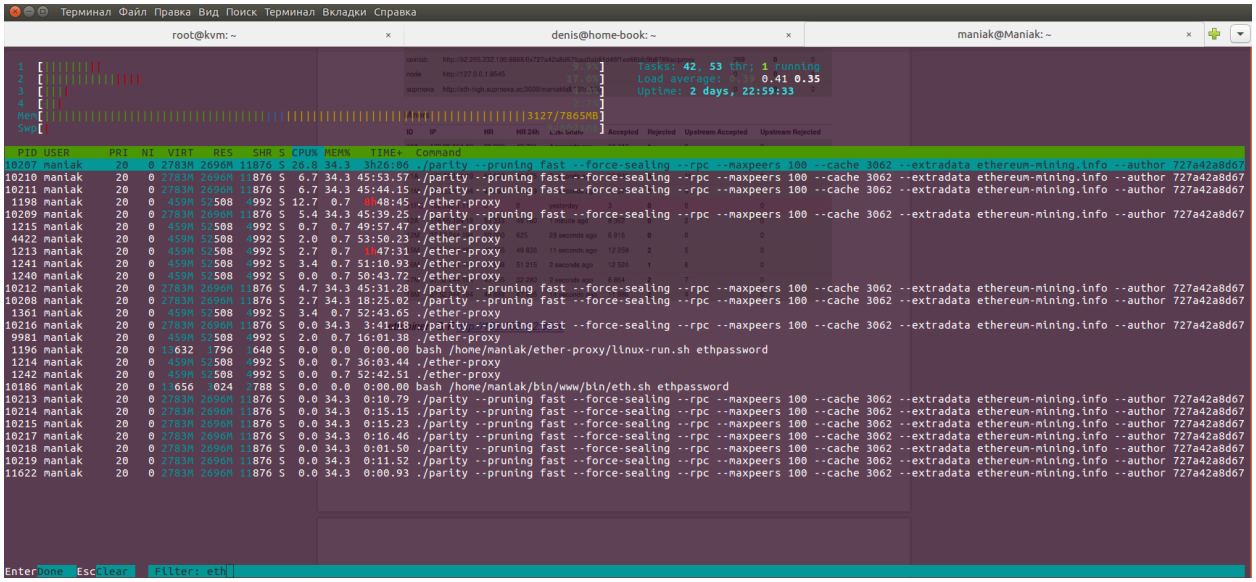
`DIFF`: mining difficulty 10 - 1000000000

`ID`: rig name

A proxy log example would be:

```
Valid share from 10M@37.52.244.194 at difficulty 1000000000
```

Example resource usage for a node: CPU Intel I7; RAM 8GB; SSD 64GB; Ubuntu 14.04.4 x86_64



At a high hashrate, the resource consumption of the rig may increase. It is possible to allocate separate servers for the purposes of backup and proxy nodes. This also increases the probability of finding blocks and creates uniformity of expectations in the rate of solutions. The advantages associated with solo mining are substantial. We minimise losses due to the latency of getting work from the pool (the mining server and the rigs are placed together on same network and ping-time is minimised); furthermore we do not pay fees to a pool. This keeps all fees in the block and of course maintains full control.