PROJECT PAI

Solo Mining

Solo mining is an alternative to mining via a collective pool like PAI Coin Pool. When you solo mine, your device or server locally constructs valid blocks until it finds one with a hash that meets the PAI Blockchain's difficulty target (i.e., a "lucky hash"). Once found, it broadcasts the block to other peers for acceptance across the network.

The main implication of solo mining is that, if and when your device finds a valid block that's accepted by the network, you get to keep the entire portion of the block reward reserved for mining. On the other hand, you will not earn anything from mining if your device never finds a valid block.

Solo mining is in contrast to pool mining, where block rewards are shared proportionally among all contributors to a pool, regardless of which individual contributor actually found the valid block with a lucky hash. Simply put, within certain bounds, solo mining is likely to result in relatively bigger rewards paid out less often, while pool mining usually results in smaller rewards, but paid out more often. For blockchains with a very high overall network hash rate, solo mining will be infeasible for miners will relatively low computational power.

Solo mining is beneficial to the PAI Blockchain because it further decentralizes the network by distributing the sources of mining out from one or a few large mining pools (e.g., PAI Coin Pool), which can pose a central-point-of-failure risk to the network.

Table of Contents

Setting up a full node with a wallet

NOTE: If you wish to store your mining rewards in a command line wallet on the same machine doing the mining, set up a full node with a wallet according to the steps below. If you wish to

store your mining rewards elsewhere (e.g., in a PAI Up mobile or web wallet), you should skip this section and set up a full node according to this guide.

Execute the following commands to build a full node with a wallet.

First, install some necessary dependencies.

```
sudo apt update
sudo apt install autoconf autogen automake make gcc libcurl4-gnutls-dev
build-essential libtool autotools-dev pkg-config libssl-dev libevent-dev
bsdmainutils python3 libboost-all-dev
```

Next, fetch and compile Berkeley DB 4.8, which is necessary for the wallet.

```
wget http://download.oracle.com/berkeley-db/db-4.8.30.NC.tar.gz
echo '12edc0df75bf9abd7f82f821795bcee50f42cb2e5f76a6a281b85732798364ef  db-
4.8.30.NC.tar.gz' | sha256sum -c

tar -xvf db-4.8.30.NC.tar.gz
sed -i 's/__atomic_compare_exchange/__atomic_compare_exchange_db/g' db-
4.8.30.NC/dbinc/atomic.h
cd db-4.8.30.NC/build_unix
mkdir -p build
BDB_PREFIX=$(pwd)/build
../dist/configure --disable-shared --enable-cxx --with-pic --
prefix=$BDB_PREFIX
make install
```

Now, configure and build PAI Coin Core.

```
cd
git clone https://github.com/projectpai/paicoin.git
cd paicoin

./autogen.sh
./configure CPPFLAGS="-I${BDB_PREFIX}/include/ -O2" LDFLAGS="-
L${BDB_PREFIX}/lib/" --disable-tests
```

You should see:

```
Options used to compile and link:
  with wallet   = yes
  with gui / qt = no
  with zmq      = no
  with test     = no
  with bench    = yes
  with upnp     = auto
  use asm       = yes
  debug enabled = no
  werror        = no

  target os     = linux
  build os      =
```

```
  CC            = gcc
  CFLAGS        = -g -O2
  CPPFLAGS      = -I/home/ubuntu/db-4.8.30.NC/build_unix/build/include/ -O2 -
DHAVE_BUILD_INFO -D__STDC_FORMAT_MACROS
  CXX           = g++ -std=c++11
  CXXFLAGS      = -g -O2 -Wall -Wextra -Wformat -Wvla -Wformat-security -Wno-
unused-parameter -Wno-implicit-fallthrough
  LDFLAGS       = -L/home/ubuntu/db-4.8.30.NC/build_unix/build/lib/
  ARFLAGS       = cr
```

Lastly, compile:

```
make
```

Then, you should see:

```
  CXX      stake/libpaicoinconsensus_la-votebits.lo
  CXX      libpaicoinconsensus_la-uint256.lo
  CXX      libpaicoinconsensus_la-utilstrencodings.lo
  CXX      support/libpaicoinconsensus_la-cleanse.lo
  CXX      libpaicoinconsensus_la-sync.lo
  CXX      libpaicoinconsensus_la-fs.lo
  CXX      libpaicoinconsensus_la-random.lo
  CXX      libpaicoinconsensus_la-utiltime.lo
  CXX      libpaicoinconsensus_la-util.lo
  CXXLD    libpaicoinconsensus.la
make[2]: Leaving directory '/home/ubuntu/paicoin/src'
make[1]: Leaving directory '/home/ubuntu/paicoin/src'
Making all in doc/man
make[1]: Entering directory '/home/ubuntu/paicoin/doc/man'
make[1]: Nothing to be done for 'all'.
make[1]: Leaving directory '/home/ubuntu/paicoin/doc/man'
make[1]: Entering directory '/home/ubuntu/paicoin'
make[1]: Nothing to be done for 'all-am'.
make[1]: Leaving directory '/home/ubuntu/paicoin'
```

Once successfully compiled, create the `~/.paicoin` directory, and inside, create a text file called `paicoin.conf` at least containing the following (set the `rpcuser` and `rpcpassword` values to whatever you prefer):

```
daemon=1
rpcuser=user
rpcpassword=password
```

Finally, launch the full node by running `./paicoind` in the `paicoin/src` directory and wait for it to sync. You can see the progress of the sync by running `paicoin-cli getmininginfo` or viewing `~/.paicoin/debug.log` and checking the local block height. The sync is complete when the local block height matches the blockchain height as seen, for example, on https://paichain.info.

You can use your PAI Coin wallet with commands like `paicoin-cli getbalance`, `paicoin-cli sendtoaddress`, and `paicoin-cli getnewaddress`. A full list of possible commands is available by executing `paicoin-cli help`.

Building and installing the solo mining software

Install a dependency:

```
sudo apt install libcurl4-gnutls-dev
```

Clone the cpuminer repository and configure the build.

```
git clone https://github.com/projectpai/cpuminer.git
cd cpuminer
./autogen.sh
```

This should output:

```
configure.ac:9: installing './compile'
configure.ac:4: installing './config.guess'
configure.ac:4: installing './config.sub'
configure.ac:6: installing './install-sh'
configure.ac:6: installing './missing'
Makefile.am: installing './INSTALL'
Makefile.am: installing './depcomp'
```

Then, execute:

```
./nomacro.pl
```

Lastly, run:

```
./configure CFLAGS="-O3"
```

This should output:

```
Compilation.............: make (or gmake)
  CPPFLAGS..............:
  CFLAGS................: -O3
  LDFLAGS...............:  -pthread
  LDADD.................: -lcurl compat/jansson/libjansson.a -lpthread

Installation............: make install (as root if needed, with 'su' or
'sudo')
  prefix................: /usr/local
```

Finally, compile the software.

```
make
```

Running the solo miner

With `paicoind` running, executing the following command in the `cpuminer` directory will launch cpuminer.

```
./minerd -a paicoin -o http://127.0.0.1:8566 -u user -p password --coinbase-addr=paiaddress --no-stratum &> ~/cpu-miner-output.log &
```

In the above command, you should replace `user` and `password` with the `rpcuser` and `rpcpassword` values that you specified in your `paicoin.conf` file. The `paiaddress` value should be set equal to the PAI Coin address you want to use to collect any block rewards you may earn from mining.

You can check the `cpu-miner-output.log` file to assess the performance of the miner.

References

Some tips for installing Berkeley DB 4.8 were found here:
https://gist.github.com/danieldk/5700533