spruned Documentation

Release 0.0.1a7

Guido Dassori

May 05, 2019

Contents

This documentation is a work-in-progress, and your contributions are welcome. Your contributions are welcome at <https://github.com/gdassori/spruned>.

Contents:

CHAPTER 1

Getting Started with spruned

1.1 Installation

To install spruned, run this command in the terminal:

```
$ cd ~/src
$ sudo apt-get install libleveldb-dev python3-dev git virtualenv gcc g++
$ git clone https://github.com/gdassori/spruned.git
$ cd spruned
$ virtualenv -p python3.5 venv
$ . venv/bin/activate
$ pip install -r requirements.txt
$ python setup.py install
```

Then you will have a $USER/src/spruned folder with a dedicated python interpreter. The usage of a virtual environment is warmly advised.

1.2 Usage

As with bitcoind, also with spruned you can hit the ‘–help’ argument, to see how to configure it:

```
$ ~/src/spruned/venv/bin/spruned --help
```

And, as in bitcoind, you can set up a config file with the parameters listed:

```
$ nano ~/.spruned/spruned.conf
```

And you can put into it, for instance, the RPC settings:

```
rpcuser=your_username
rpcpassword=your_password
rpcport=18332
```

(continues on next page)

(continued from previous page)

```
rpcbind=0.0.0.0
network=bitcoin.testnet
```

CHAPTER 2

Overview

2.1 Motivations

In the past I wrote a couple of POC, like ChainDNS, Gangsta, Notarium and others. All of them needed a full node. And so I set up a full node everytime. Then I needed a small explorer, and I set up a full node to back ABE. Finally, Clightning went out, and obviously it needs a full node. And I set up another full node. However, with the growing of the Bitcoin Blockchain and the spread of low end boxes, setting up a full node may be tricky.

2.2 How it works

spruned enable a reduced set of the bitcoind RPC API by querying the blockchain and the electrum network for the needed informations, without sacrificing the cryptographically proof of data.

To do so, at the first run spruned downloads and do POW verification, since the genesis block, of the whole chain of headers, (and this, paradoxically, makes spruned faster to sync on mainnet than testnet). Headers are saved into a sqlite database. The first ~130mb (250mb on testnet) are gone.

At this point (it takes 15~30 minutes on a modern server, ~2 hours on a RPI zero), spruned is ready to serve headers, block and transactions.

Headers are downloaded from the Electrum network.

2.3 Usage of the bitcoind p2p network

At this very moment, spruned usage of the p2p network is limited to the blocks download.

Once the headers are synced to the best one available in the Electrum network, a set of connection to the p2p network is initialized, target is to keep the default value of 8 connections.

Everytime the `` `getblock` `` command is used, a local cache layer is queried to check if the block was previously saved. If the block is found, no network calls are made, otherwise the block is downloaded from a random peer of the network, taken from the established connections pool.

Remote peers, however, already notify new transactions to spruned, and the mempool emulation is WIP. This will be the second usage of the P2P network in this first stage.

2.4 Usage of the electrum network

The electrum network is used for multiple purposes. Transactions, fees, headers, a lot of data is fetched from the electrum servers.

- getrawtransaction
- sendrawtransaction
- getblockheader
- estimatefee

Needs the electrum network to work properly. Also, to crypto-verify a transaction, the tx merkle proof is downloaded from the electrum network.

2.5 Dependencies

For sake of coding speed, right now spruned relay on a set of open source code dependecies. It would be nice in the future to reduce this set of dependencies. Right now dependencies are a necessary evil.

When you install spruned, you will also use:

- async-timeout==2.0.1
- jsonrpcserver==3.5.3
- sqlalchemy==1.2.6
- coverage==4.5.1
- plyvel==0.9.0
- daemonize==2.4.7
- aiohttp==3.0.0b0
- git+https://github.com/gdassori/pybitcointools.git@spruned-support#egg=pybitcointools-1.0.0a1
- git+https://github.com/gdassori/connectrum.git@spruned-support#egg=connectrum-0.5.3a4
- git+https://github.com/gdassori/pycoinnet.git@spruned-support#egg=pycoinnet-0.19a3
- git+https://github.com/gdassori/pycoin.git@spruned-support#egg=pycoin-0.90a4

daemonize is self explaining, it helped me to put spruned in background (can be easily removed).

plyvel is the LevelDB driver, no way this will be removed.

aiohttp and coverage are used only for testing, can be installed separately.

async-timeout is a well known library and cannot be replaced unless recoded.

there’s no large usage of sqlalchemy, and can be removed, but the whole transactional sqlite part must be recoded.

jsonrpc is a well known and trusted dependency, I don’t feel the need to remove it.

A different chapter must be open for the github dependencies.

I personally mantain this pybitcointools fork, and I don't feel the need to remove it.

Connectrum can be removed, but must be recoded, and however Coinkite (the mantainer) is very well reputed in the crypto ecosystem.

pycoin and pycoinnet from Richard Kiss are very large reviewed and respectful libraries of the crypto ecosystem.

No need to remove them but… I do a very small usage of the pycoin library (and a huge usage of pycoinnet!) so I may, in the future, think about removing only the pycoin dependency.

2.6 Todo

A lot must be done, and spruned is just in its early stage. My first goal would be to find fellow developers that contribute to the project :-)

With them I would love to do the mempool emulation, reduce the dependency from the electrum network, write an extensive integration tests suite and many other things. This library can be promising, but without extensive review from different people, me neither would advice its usage in a production environment.

CHAPTER 3

Configuration

Is easy to configure spruned! Some parameters are the same of bitcoind. Something else is part of spruned only. Let's see.

3.1 RPC

spruned try to emulate the bitcoind RPC interface, so can work with bitcoin-cli. To do so, if you use config files, the must be aligned.

Those two files:

:: ~/.bitcoin/bitcoin.conf ~/.spruned/spruned.conf

Must have the same configuration for the following parameters:

:: rpcuser rpcpassword rpcbind rpcport

And bitcoin-cli will be able to communicate with spruned as it would do with bitcoind.

3.2 Cache

3.3 Other

3.4 debug

CHAPTER 4

Commands

Supported bitcoind API in spruned

4.1 Blockchain

- getbestblockhash
- getblock “blockhash” (verbosity)
- getblockchaininfo
- getblockcount
- getblockhash height
- getblockheader “hash” (verbose)
- gettxout “txid” n (include_mempool)

4.2 Rawtransactions

- getrawtransaction “txid” (verbose)
- sendrawtransaction “hexstring” (allowhighfees)

4.3 Util

- estimatefee nblocks
- estimatesmartfee conf_target (“estimate_mode”)

CHAPTER 5

Contribute

5.1 How to contribute to spruned

If you went trough the reading till here… thanks :-)

Right now, even just using spruned is a way to contribute, if you take a minute to report bugs or inconsistencies with bitcoind, so if you're going to set up a new spruned instance: thank you.

If you are a developer, feel free to open a PR. Not sure? Contact me before (mn3monic @ freenode).

And, obviously spruned needs:

- a better documentation :-)
- use cases
- proposals
- fixes
- extensive review of the code and the english (yes, I'm not ashamed and gladly accept english corrections, that is not my first language).

CHAPTER 6

Indices and tables

- genindex
- modindex
- search