Fast-sync in Cuprate jomuel, dllud

Fast-sync in Cuprate

Memory-safe Monero node implementation gets closer to feature parity.

dllud * iomuel

Developed for MoneroKon's Hackathon Anti-fragility Award.

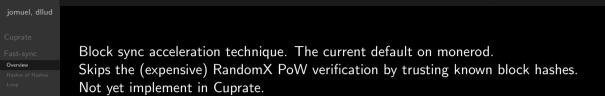
^{*}with great support from boog900

st-sync in Cuprate		Cuprate
nuel, dllud		0
orate		
	Monero full node written from scratch in Rust.	
	Brings in client diversity to Monero:	
	 independent validation of Monero consensus rules 	
	 protects network from implementation bugs in monerod (remember block 202612) 	

Written in a memory-safe language: avoids (most) memory errors and related

security vulnerabilities.

Cupr



Fast-sync Overview

In a VPS with: 3 vCore, 4GB RAM, 200 Mbit/s network, SSD*

"Slow" sync: 15 days.

Fast-svnc in

Cuprate

Fast sync: 4 days.

The catch: you are trusting monerod binary on the old blocks PoW validity.

^{*}https://gist.github.com/DaWe35/aaa0d1a99be4a6fb0977fb7df7ddb702

Fast-sync in Cuprate	Hashes of Hashes
jomuel, dllud	
Cuprate	
Fast-sync Overview	
Hashes of Hashes Loop	
Implementation Overview Hairy details	Storing each block hash in monerod would bloat the binary too much.
	Instead monerod hashes groups of block hashes and stores that.
	Current default is 512 hashes per group, which creates a binary blob with 379 KiB.

Cuprate

Fast-syn

Hashes of Hash

Loop

Overview

TODO

Feedback

Peer sends block hashes.

- 2 Hashes get grouped, hashed together and checked for validity.
- 3 If valid, the full blocks are requested from peers.
- 4 Each individual block hash gets checked.
- **5** If it matches, block is inserted into DB with PoW hash as zeros.

Cuprate

Fast-sync

Loop

Implementa

Overview Hairy details

TODO

Eandbook'

Based on detailed instructions by boog900 (Cuprate's main dev). https://github.com/Cuprate/cuprate/issues/153

Split into 2 pull requests:

- 1 #155 Merged. Implements:
 - FastSyncService, ValidateHashes and tests.
 - A tool to generate the hashes of hashes from a synced blockchain.
- 2 #156 Draft. Will implement ValidateBlock.

Hairy details

Hashes of hashes stored as a text file:

```
hex!("1adffbaf832784406018009e07d3dc3a39da7edb6632523c119ed8acb32eb934").
hex!("ae960265e3398d04f3cd4f949ed13c2689424887c71c1441a03d900a9d3a777f"),
hex!("938c72d267bbd3a17cdecbe02443d00012ee62d6e9f3524f5a914192110b1798"),
```

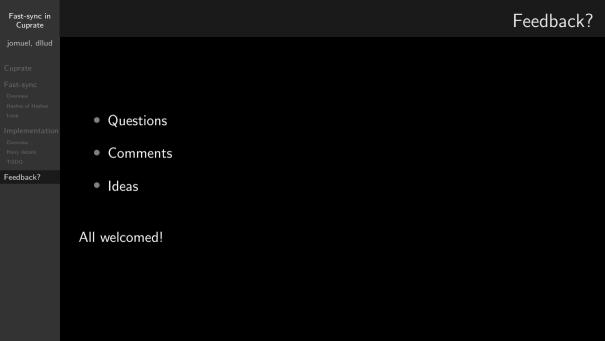
that gets inline included in the source:

```
static HASHES_OF_HASHES: &[HashOfHashes] = &include!("./data/hashes_of_hashes");
```

Text file 2x bigger than binary file (checkpoints.dat) in monerod's repo.

Bloat in the final Cuprate binary is the same, code is simpler.

- Full block validation with PoW skipping.
- Command line option to enable/disable fast-sync.
- Test with full mainnet chain.
- Document code.
- Get it reviewed and merged.



Licenses