Pruning Nodes

Cyrus Malekpour
cm7bv

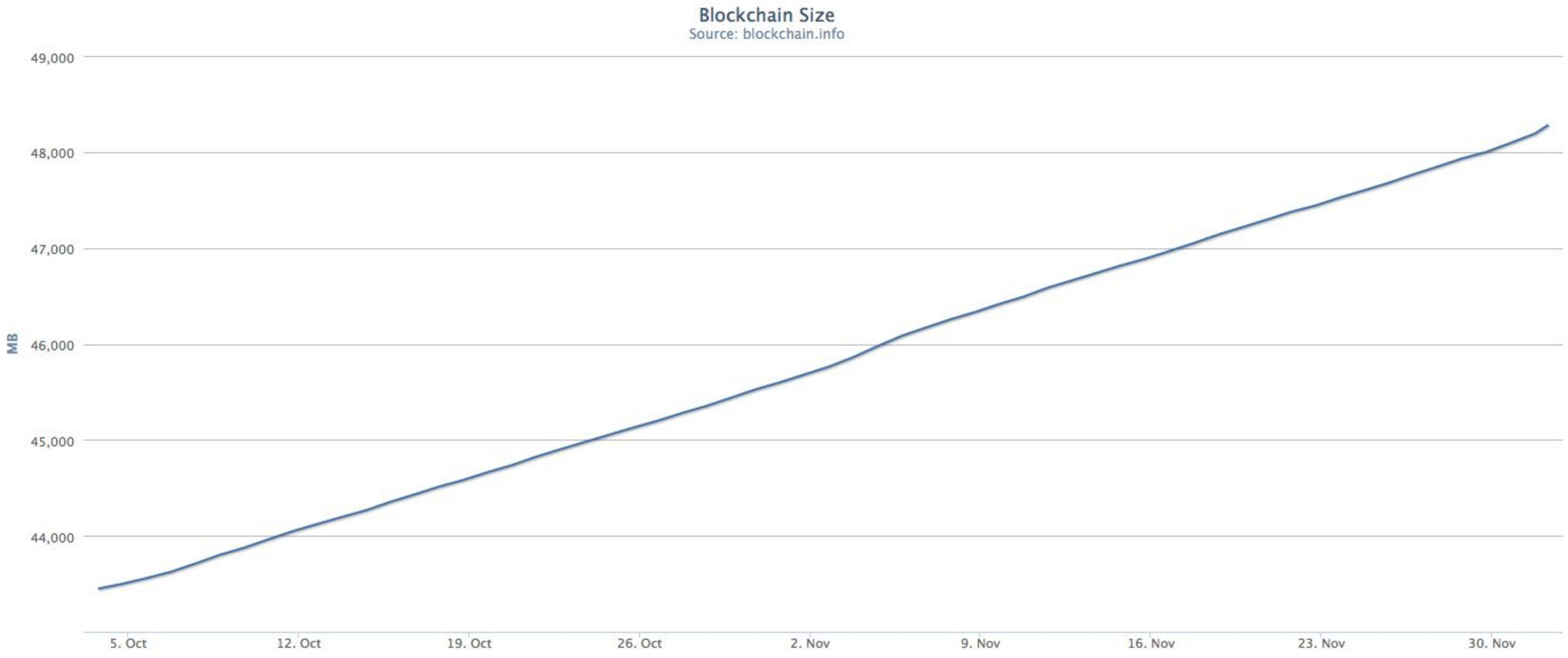
Project Overview and Goals

- **Pruning Node** - a node that stores only the latest *n* blocks in the blockchain
    - Can also optionally perform “fast sync” of blocks when initially syncing blockchain
- Goal: modify a bitcoin full node to operate as a pruning node
    - Determine if there are theoretical or implementation problems with creating a pruning node
    - Run the pruning node to determine if any problems emerge during operation

Motivation

- Full node initial sync of the blockchain is very slow
  - For current blockchain sizes, it can take multiple days on an average connection
- Full node data storage requirements continue increasing (current: 48.2gb)
- Number of full nodes has fallen **19%** in past 90 days (32% over past year)
- SPV requires less space, but can’t really prove anything for itself
- SPV nodes rely on other nodes for confirmations
  - 30 confirmations (!) recommended for SPV

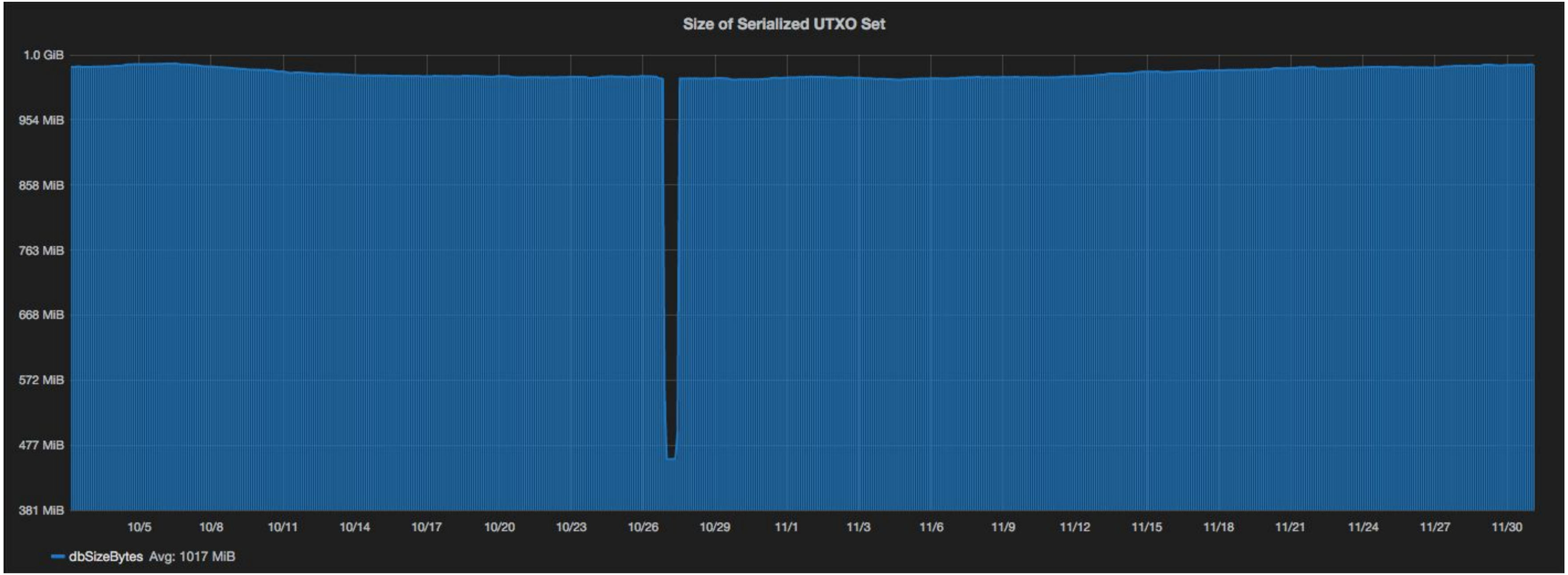
Motivation


Blockchain Size
Source: blockchain.info
MB
49,000
48,000
47,000
46,000
45,000
44,000
5. Oct
12. Oct
19. Oct
26. Oct
2. Nov
9. Nov
16. Nov
23. Nov
30. Nov

How it works

- All (full) nodes need to store UTXO set in order to validate new transactions
- If we don’t care about transaction history, no need to save old blocks
- Instead, store only the *n* latest blocks (to ensure we’re on the longest chain)
    - bitcoin.org recommends 6 confirmations as minimum for safety, but we can save 100 or 500 without significant disk cost
- As a new block is added to the chain, we delete our oldest block
- Ideally, **no less secure** than a full node

How it works


Size of Serialized UTXO Set
1.0 GiB
954 MiB
858 MiB
763 MiB
668 MiB
572 MiB
477 MiB
381 MiB
10/5
10/8
10/11
10/14
10/17
10/20
10/23
10/26
10/29
11/1
11/3
11/6
11/9
11/12
11/15
11/18
11/21
11/24
11/27
11/30
dbSizeBytes Avg: 1017 MiB

How it works

- **Checkpoints** are hardcoded block hashes in the Bitcoin core software
    - When selected, they correspond to blocks in the far past with thousands of confirmations
    - Hardcoded checkpoints include both block hash and block height
- Checkpoints are intended to limit impact of malicious forks (51% attack)
- We can use latest checkpoint as a starting point for downloading blocks
    - At time of writing, only need to download **4012 blocks** from latest checkpoint (height 382320)
- We rely on # of confirmations for unknown UTXOs

Using Pruning Nodes

- Full node operators are essential for network security and operation
  - Yet they have no financial incentive to operate their nodes
  - SPV nodes rely on full nodes to provide confirmations
- Operators of pruning nodes benefit from improved sync time and disk space
  - Selfish benefit for pruning node operator at cost to rest of network
- With full UTXO set, no loss in verification of incoming transactions/blocks
- Bitcoin protocol has no way to penalize people for pruning
  - Full storage would fall on those who only willing to bear additional cost
- Could eventually lead to more nodes that verify blockchain (unlike SPV)

Downsides

- Corruption in UTXO set would require a full sync
    - No saved blockchain to regenerate from
- If using the “fast sync” method, old tx inputs/outputs can’t be verified
    - We don’t have the transactions that these outputs came from!
    - If we encounter an unknown UTXO during fast sync, we check # of confirmations for the transaction (and hope full nodes will handle it)
- Can’t import existing wallet keys/addresses without downloading all blocks
- If full nodes disappear, nobody to store old data or generate checkpoints
- We can’t help SPV nodes

Security Concerns

- Pruning
  - Forked chain with > *n* nodes could destroy validity of our database (we simply regenerate)
  - Without full nodes, this could wipe out all pruned nodes with only *n* stored blocks
- Fast sync
  - Hardcoded checkpoints must be trusted (comes with client code)
  - Unknown UTXOs during fast sync rely on other nodes for verification (no less secure than SPV method)

Current Progress

- Work is being done on btcd at recommendation of Prof. Evans
- Client currently performs fast sync by only requesting blocks after the latest checkpoint
- Currently working on the pruning functionality in the node

Questions?