IBLT and weak block propagation performance

Kalle Rosenbaum (Popeller) &

Rusty Russell (BlockStream)

Invertible Bloom Lookup Tables (IBLT)

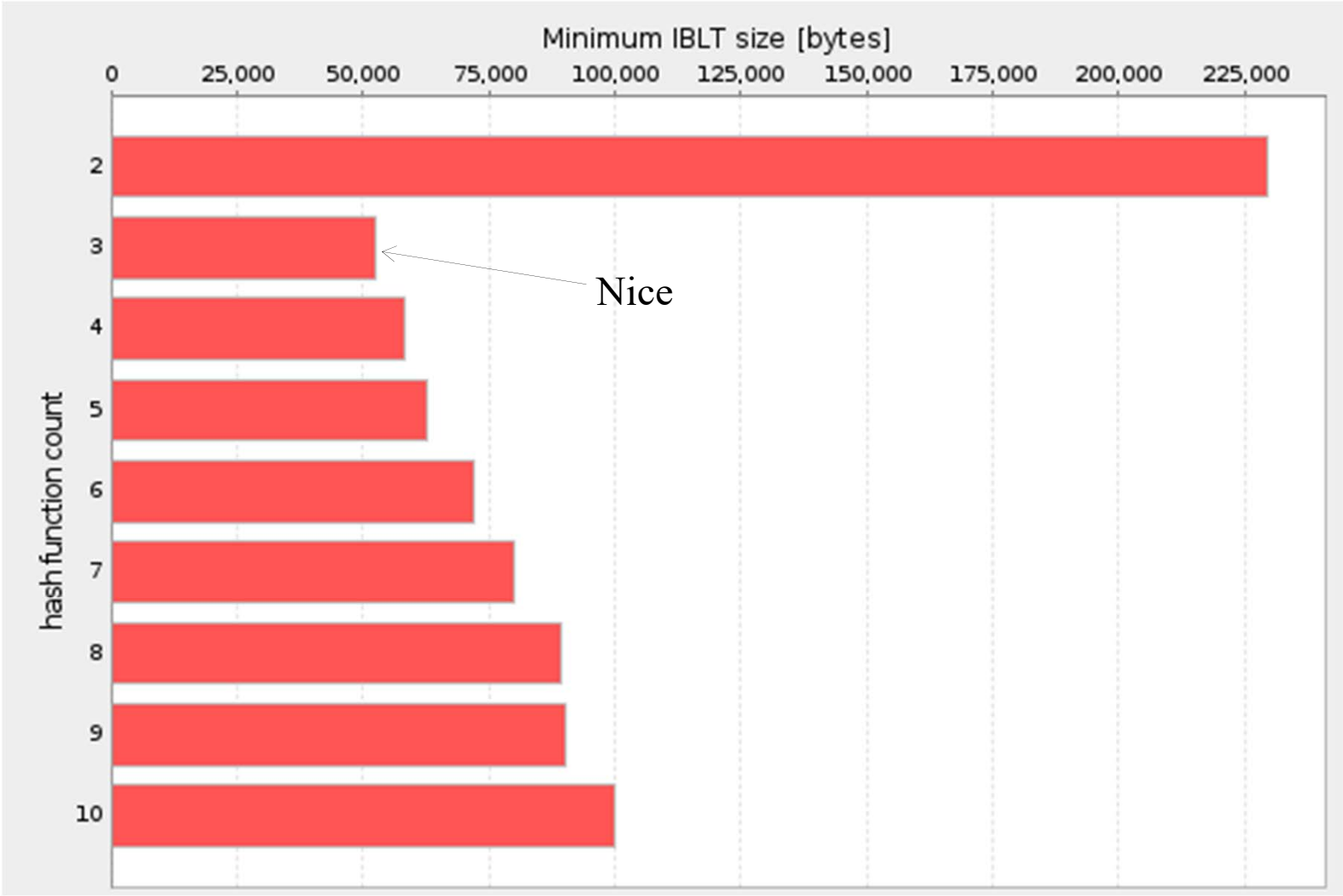
- Credit Gavin Andresen
- Based on the work of Michael T. Goodrich and Michael Mitzenmacher
  - http://arxiv.org/abs/1101.2245
- Allows efficient reconciliation of similar sets
- Great if block is similar to mempool

IBLT

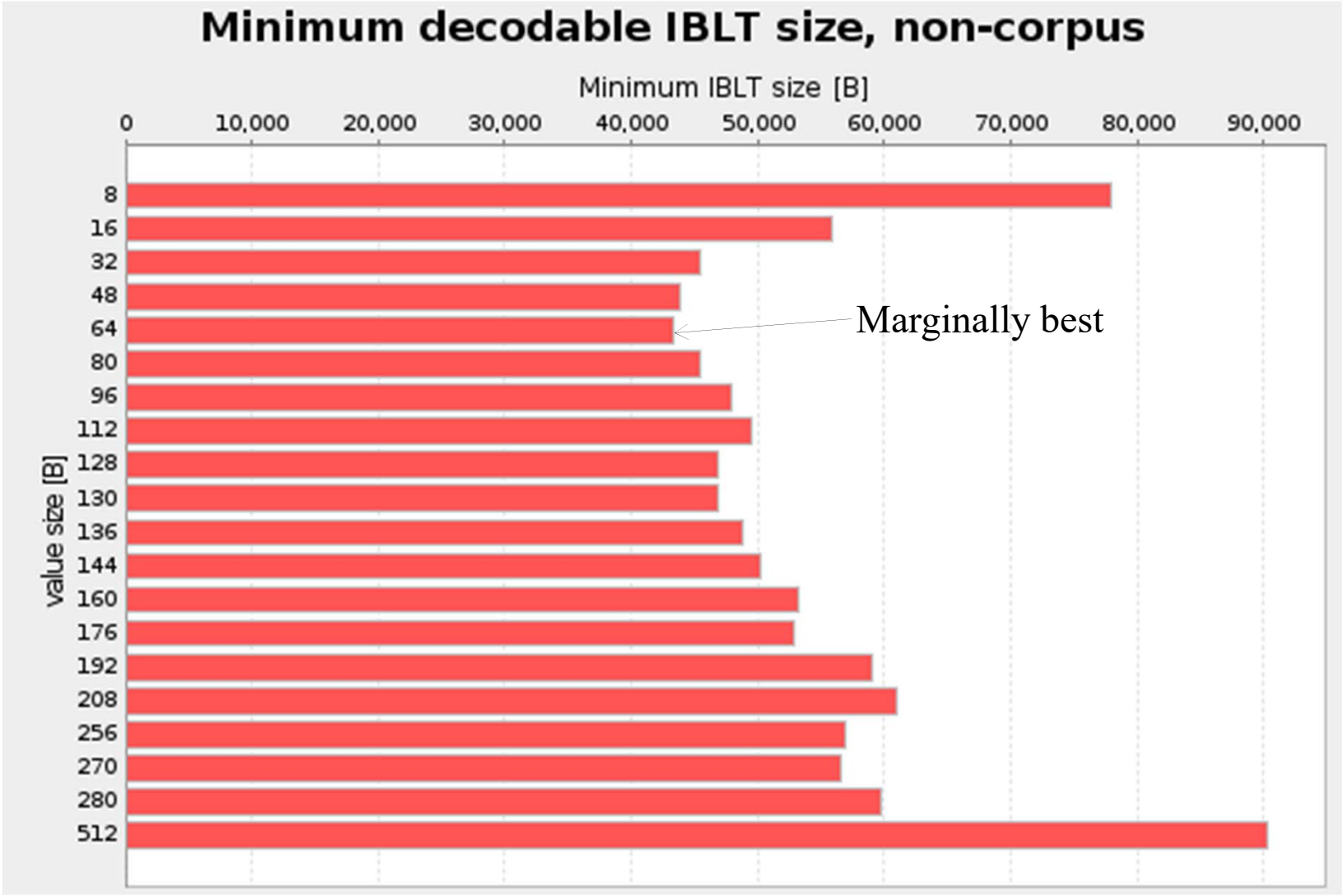
IBLT parameters

- Number of hash functions?
- Value size, 8B? 64B? 128B?
  - Bigger values → More waste
  - Smaller values → More cell overhead

IBLT hash functions



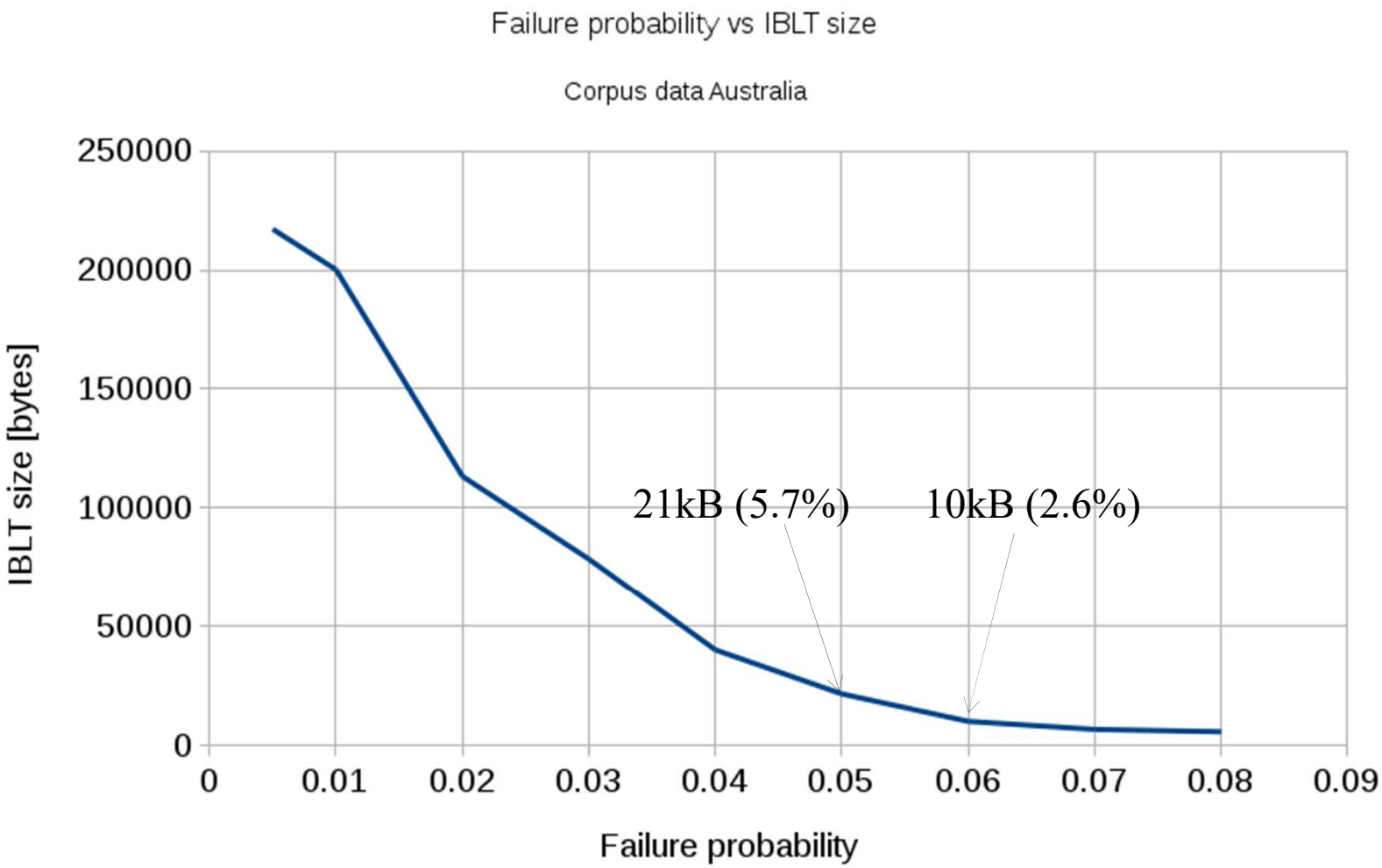
Value size, we like 64 bytes



Real data?

- Bitcoin-corpus covers 721 blocks from 4 nodes
- Average block size 381891
- Focus (randomly) on Australian node
- How small can we make the IBLT?

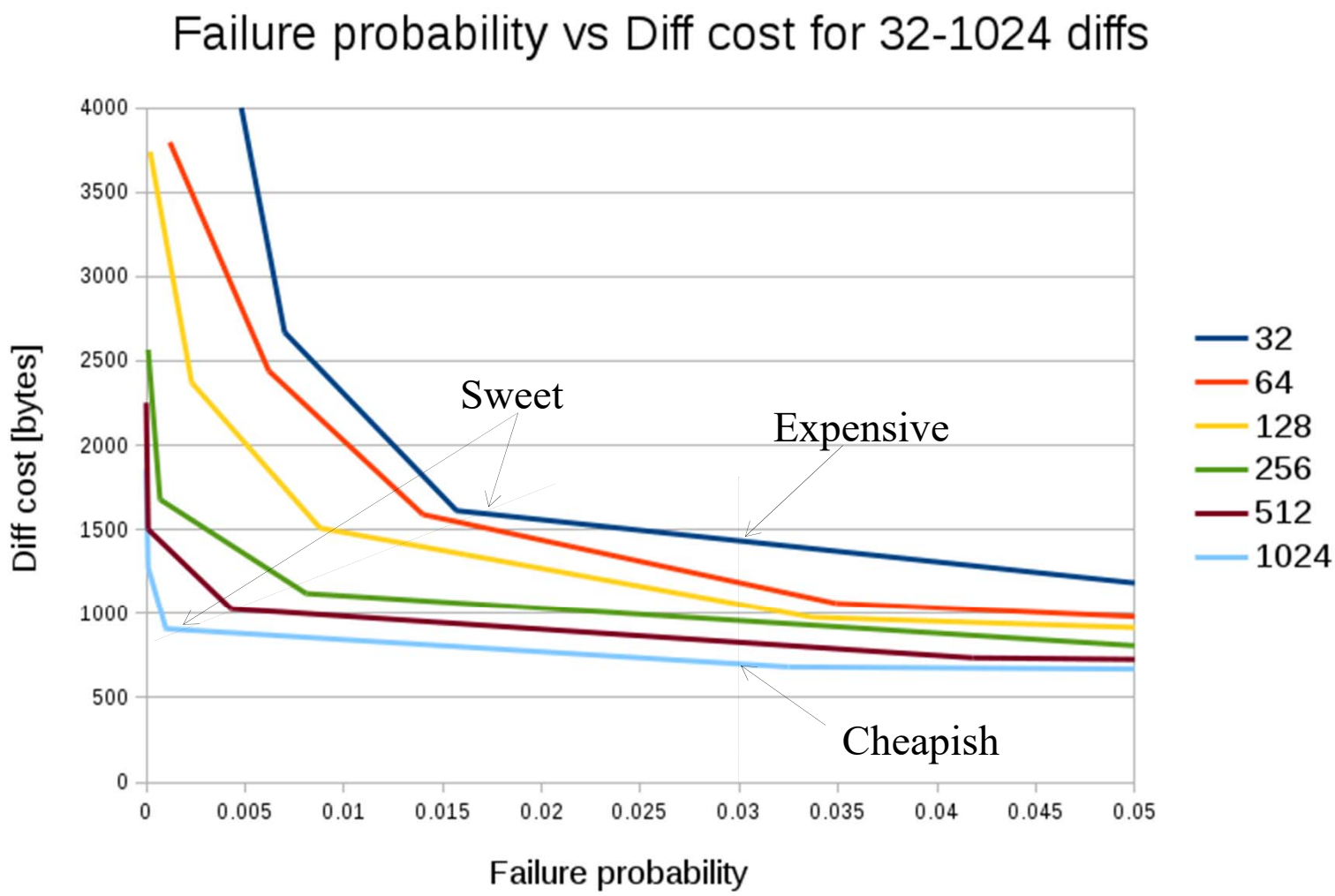
Result for Australia



Scaling

- What happens when differences increase?
- Measure failure probability for increasing differences.
  - Select diffs randomly
  - Select IBLT size
  - Encode/decode many times to measure failure probability

Cost per diff

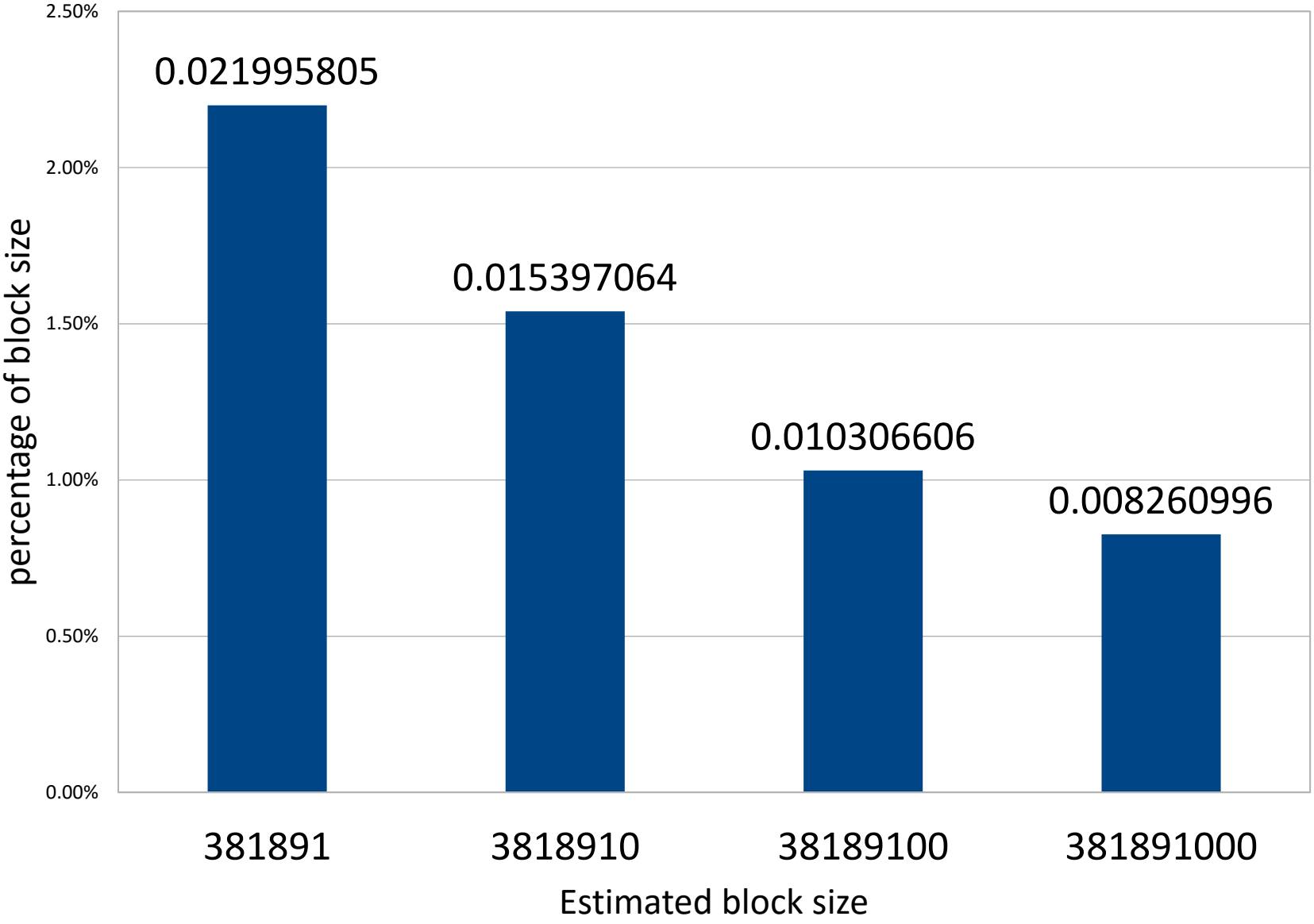


Example

- Assumptions
  - Diffs increase linearly with tx rate
- Open question: How do differences change with transaction rate?
  - Block size increase linearly with tx rate
- Corpus average 6 diffs
- Increase tx rate by factor 10, 100 and 1000
- Target 5% failure probability

Example

5% failure probability target



So far we've...

- tested fixed sized IBLTs on bitcoin-corpus. **21KB (5.7%).**
- examined the scaling properties of IBLTs. **The bigger, the better**.

Bitcoin IBLT Protocol

Bitcoin IBLT Protocol

Transaction which occurs in block

Bitcoin IBLT Protocol

Transaction which occurs in block

Node 1 remembers how different the incoming block was from its mempool (assuming it will be similar for peers)

Bitcoin IBLT Protocol

Bitcoin IBLT Protocol

Bitcoin IBLT Protocol

Bitcoin IBLT Protocol

Bitcoin IBLT Protocol

Bitcoin IBLT Protocol

Bitcoin IBLT Protocol

IBLT Protocol: Dynamic Sizing

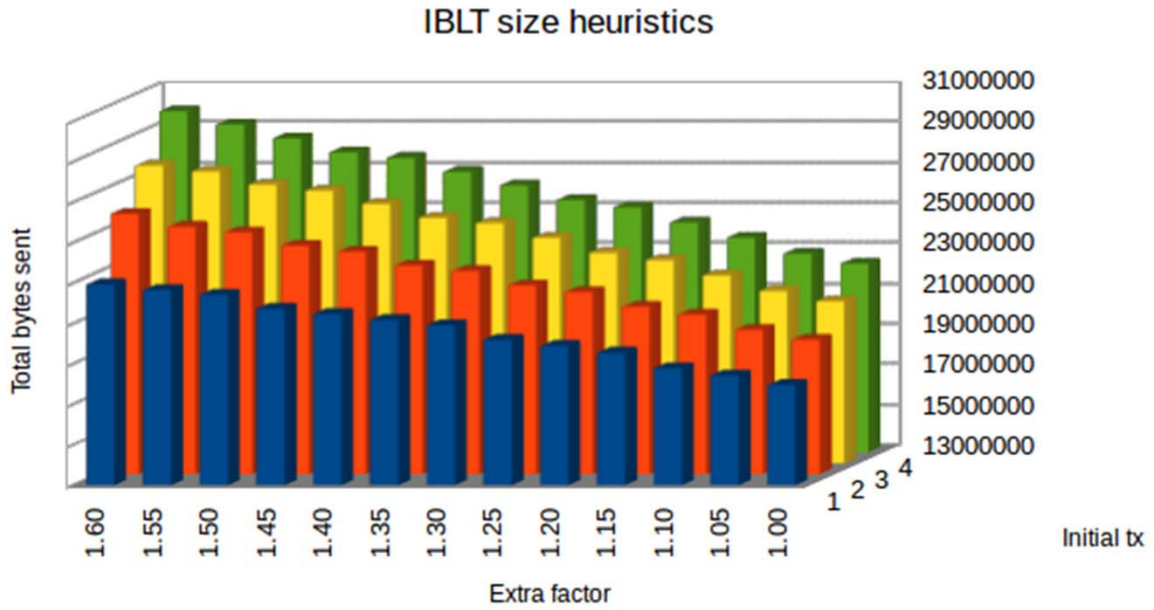
- How do we choose IBLT size?
  - Mempools tend to be very similar.
- Assume receiver's mempool about as different from block as ours was.
  - Add some extra to cover differences...

Dynamic Estimate Extra Factors

- Fixed factor (eg. assume an extra 2 txs to reconstruct)
- Variable factor (multiply total slices)

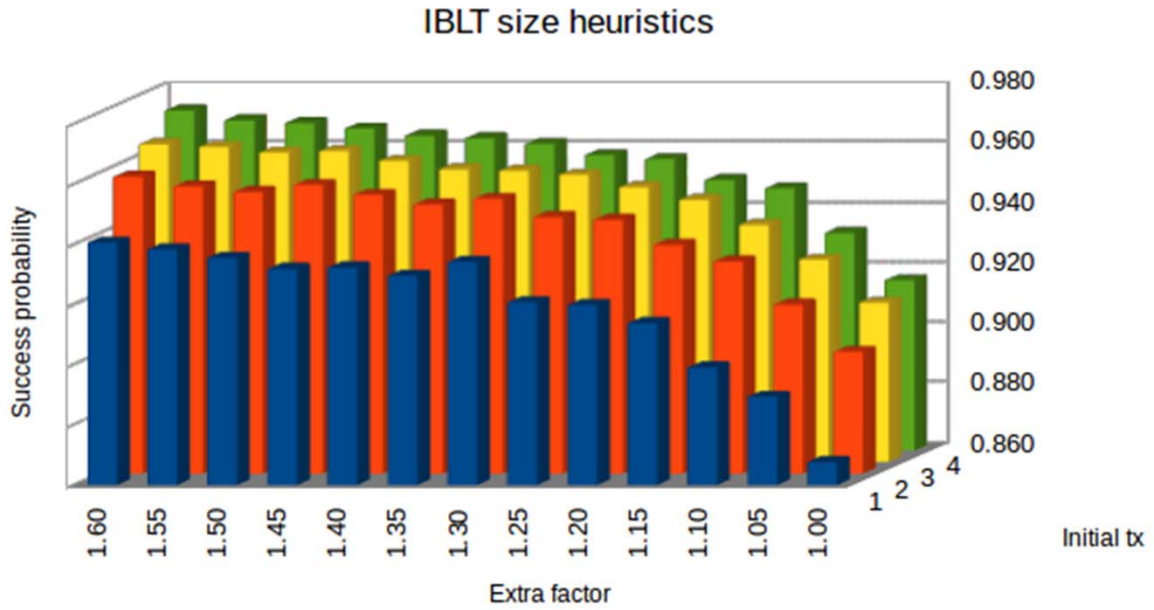
Dynamic Estimate Extra Factors

- Fixed factor (eg. assume an extra 2 txs to reconstruct)
- Variable factor (multiply total slices)



Dynamic Estimate Extra Factors

- Fixed factor (eg. assume an extra 2 txs to reconstruct)
- Variable factor (multiply total slices)



Dynamic Estimate Extra Factors

- Fixed factor (eg. assume an extra 2 txs to reconstruct): **+10 slices**
- Variable factor (multiply total slices): **x1.35**

IBLT Corpus Performance

- Across our 825 MB corpus:
  - 20 MB transmitted (95% reconstructed)
  - 4% of blocks sent “raw”

Weak Blocks

aka. Near Blocks

Weak Blocks

- Miners broadcast “not quite good enough” blocks.
  - eg. within 20x required difficulty.
- Naturally ratelimited
- Offers (provable!) insight into miner mempools
- **All blocks can be simply encoded in terms of previous weak blocks.**

Weak Blocks

Weak Blocks

Simple 2-byte encoding:

18,1,3,8,12,-1,14,0,10,-1,20,11,13,4,19,9,-1,-1,
5,16,-1,2,7,-1,-1,-1,-1
<tx1><tx2><tx3><tx4><tx5>...<tx10>

Weak Blocks Simulation

- Take corpus, randomly generate weak block from (best paying) txs in mempool approximately every 30s.
- Assume these weak blocks instantly transmitted to other nodes.
- First node to see a block calculates encoding to other nodes vs. last known weak block (if any)

Weak Blocks Simulation

- Raw blocks: 825MB
- Strong blocks using 30-second weak blocks:
  - 35MB (+/- 3MB)
  - **Total size increases to 1.51GB though!**

Super Weak Blocks?

- If blocks are full, we want the first weak block as soon as possible.

Super Weak Blocks?

- If blocks are full, we want the first weak block as soon as possible.
- 16x super-weak first* blocks:
  - 27MB (+/- 1.7MB)
  - (Total size increases to 1.53GB)
  -

*Handwave: define first!

Weak Blocks Simulation

- Note that we've seen that real blocks diverge much more than bitcoin-corpus peers!
  - **Expect worse compression in practice.**

IBLT + Weak Blocks?

IBLT + Weak Blocks?

- Raw blocks: 825 MB

IBLT + Weak Blocks?

- Raw blocks: 825 MB
- IBLT: 20 MB (95% recovery)

IBLT + Weak Blocks?

- Raw blocks: 825 MB
- IBLT: 20 MB (95% recovery)
- Weak blocks: 27 MB / 1530 MB total

IBLT + Weak Blocks?

- Raw blocks: 825 MB
- IBLT: 20 MB (95% recovery)
- Weak blocks: 27 MB / 1530 MB total
- Together: 15 MB (98%) / 233 MB (65%) total
  - Or 434MB and 86% (fixed 133 buckets)
  - Or 804MB and 96% (fixed 400 buckets)

Deployment

- New block transmission message:
  - [prev-weak-block][references][rawtxs][ibltseed][fee-hint][added-set][removed-set][iblt-size][iblt-buckets][ordering-info]

Deployment: Weak Blocks

- Start with weak block threshold 1/10000 difficulty
  - Rachet up to 1/20 as we see stronger weak blocks.

Future

- Canonical fee-per-byte ordering?
  - Much better for IBLT and weak block encoding.
- Coinbase encoding
  - Incentive to publish weak blocks (save 500 bytes)
- Block blast
  - Over half encodings give block < 3k.
- IBLT Mempool Sync
  - gmaxwell, may save ~70 bytes per tx per peer.

Questions?