Probabilistic Data Structures

(Wow that's hard to spell)

Mark Stetzer, Proofpoint
@stetzer

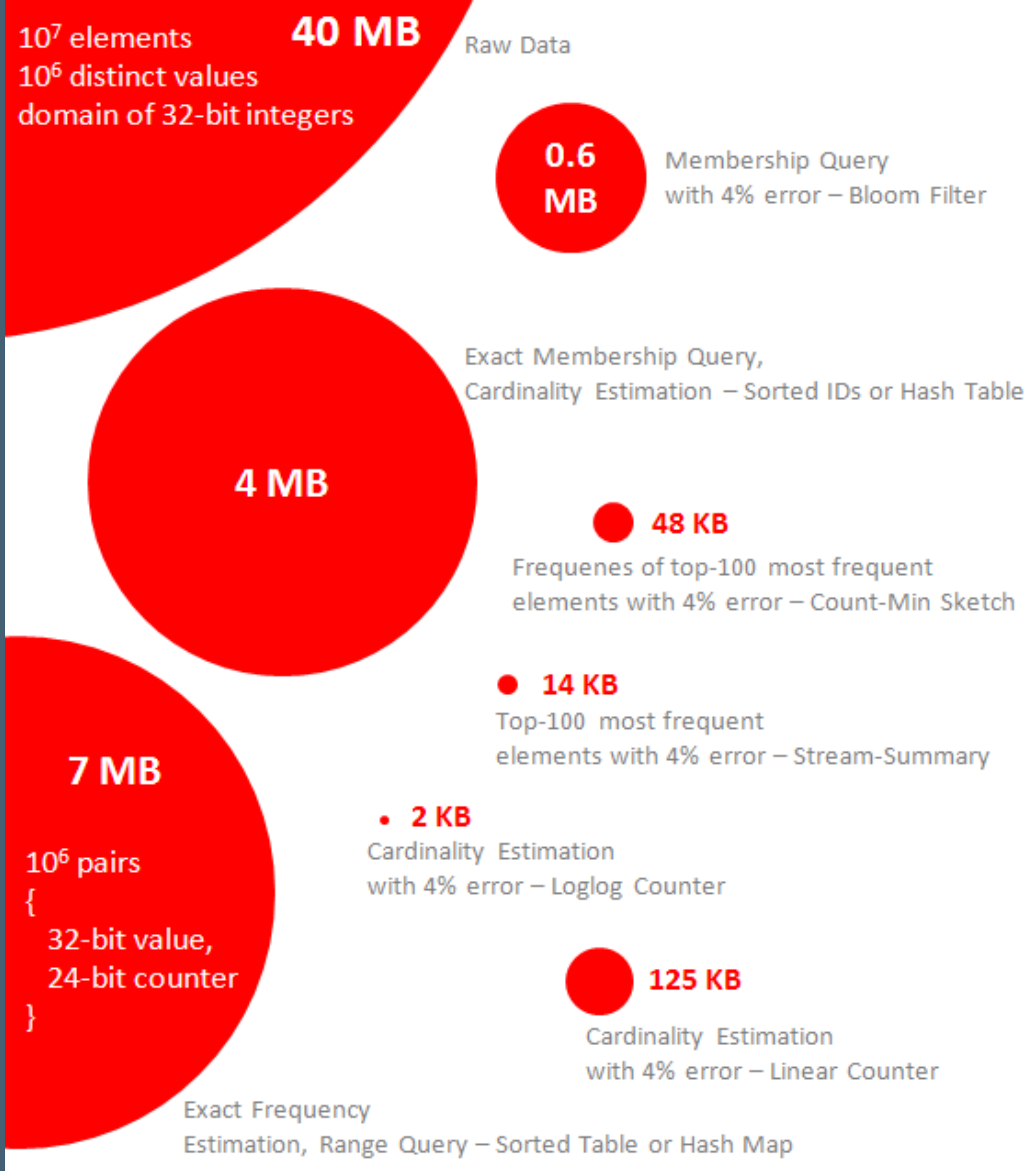
Highlights

- Why this topic?
- Bloom Filter
- Count-min sketch
- HyperLogLog

Why this topic [1]?

- Big data is about taking different approaches to data problems due to your dataset size
- Just because you *can* compute the **exact** number of distinct customers in 1PB of logs doesn't mean you *should*
- Let's use really old, un-hip, proven techniques like sampling and hashing!

[1] Notice how it's *"this topic"* so I don't have to spell probabilistic again?



Size saving appoximations[2]

[2] https://highlyscalable.wordpress.com/2012/05/01/probabilistic-structures-web-analytics-data-mining/

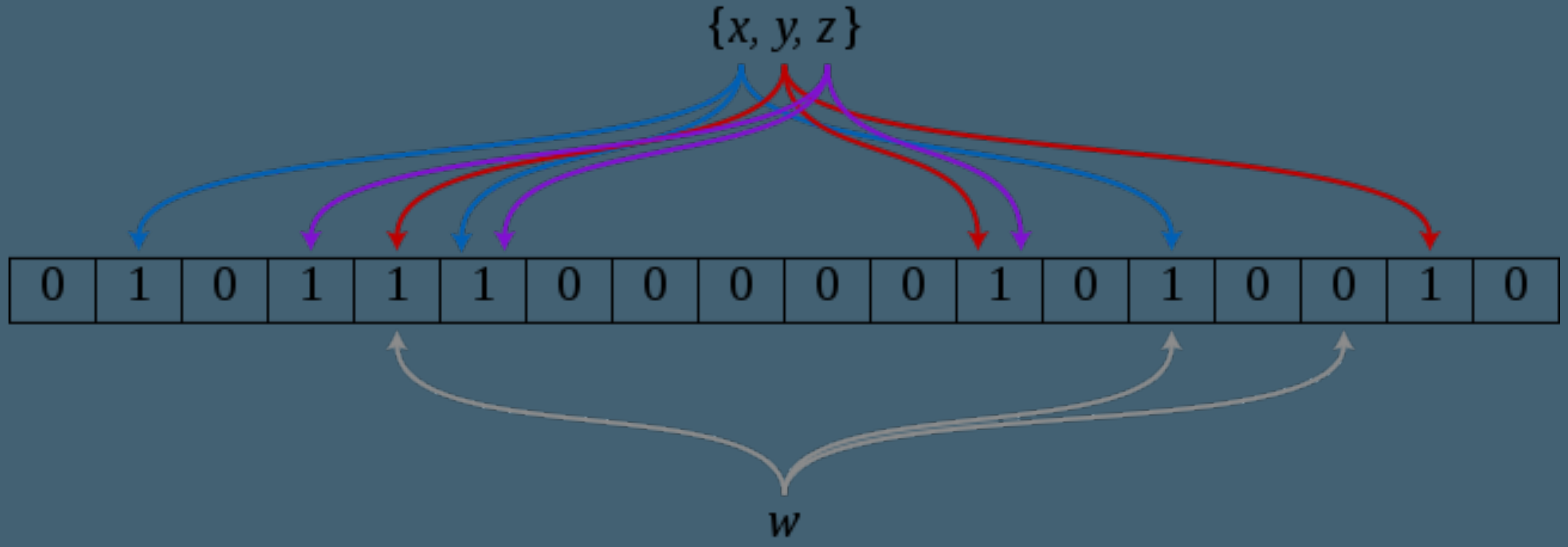
Problem:

Have we seen this item before?

Bloom filter

- Conceived by Burton Howard Bloom back in 1970; probably most well-known probabilistic data structure
- Used to estimate set membership
- False positives are possible (thought we had something we don't), false negatives are not; recall therefore 100%
- Technique involves maintaining a bitset where each bit is mapped to some number of fixed bits by independant hash functions

Bloom filter[3]



[3] https://en.wikipedia.org/wiki/Bloom_filter

Bloom filter

- Lots of work done around tuning for space & error rates
- Able to estimate set membership of **10^9** items, with false positive rate of **1%**, using **1.12g** space[4]
- Typical use cases include pre-checks for expensive API lookups and distributed system coordination
- **LOTS** of OSS implementations exist; Google Guava ships w/ serializable version (no guaranteed backward compatibility)

[4] http://hur.st/bloomfilter

Problem:

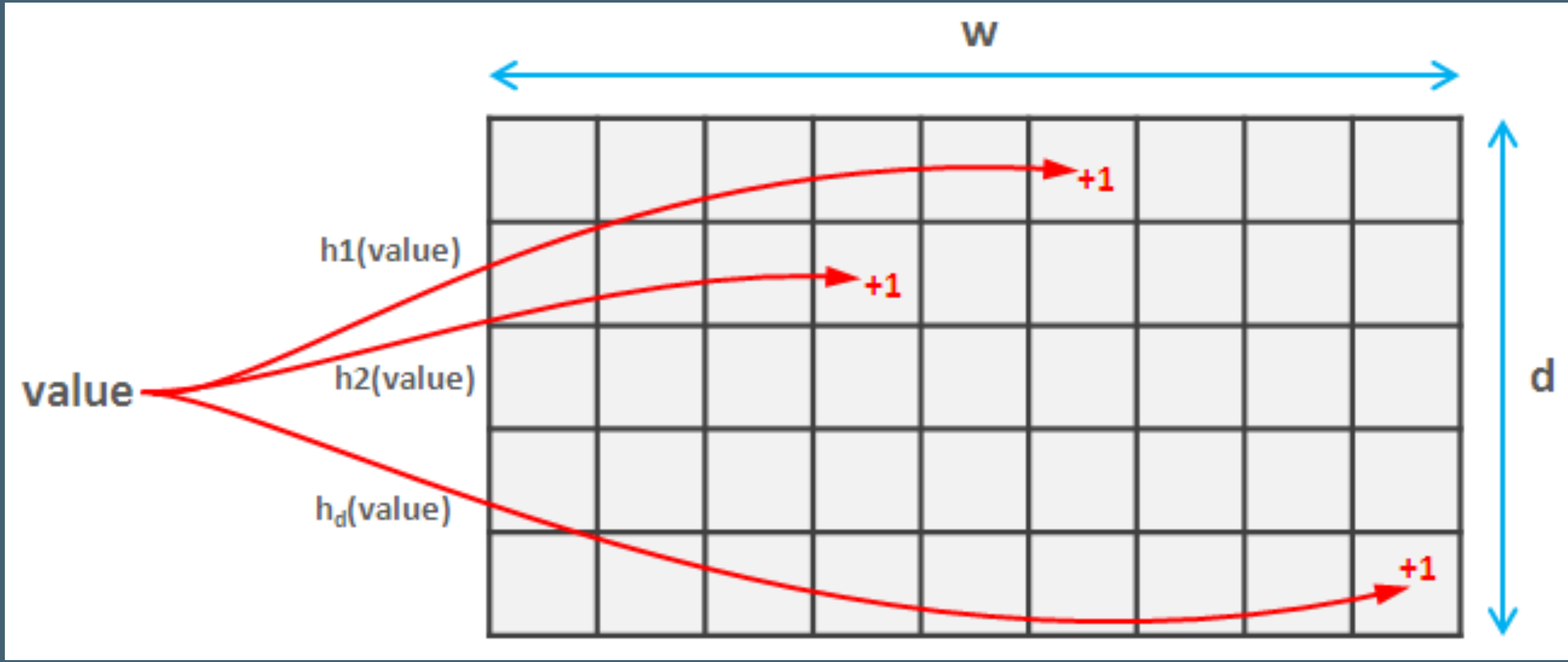
What are the 100 most popular items in our dataset (and what is the measure of their popularity)?

Count-min sketch[5]

- Serves as a frequency table of events from a stream of data
- Technique involves hashing events to frequencies, using multiple hash functions (*recognize a pattern here?*), using minimum hashed value (w/ potential collisions)
- Similar to bloom filters, approximate count will never be more than estimate but could be less

[5] http://sites.google.com/site/countminsketch/cm-latin.pdf

Count-min sketch[2]



[2] https://highlyscalable.wordpress.com/2012/05/01/probabilistic-structures-web-analytics-data-mining/

Count-min sketch

- Hard to generalize results due to sketch size, error rate, & distribution; analysis against different real-world data sets has been done[6]
- First example: Top-100 w/ **4%** error on **10^7** values using **48kb**[2]
- CMS can perform poorly on non-Zipfian distributions; *Count-mean-min sketch* estimates noise for each hash & subtracts

---

[6] http://www.cs.rutgers.edu/~muthu/cmz-sdm.pdf

[2] https://highlyscalable.wordpress.com/2012/05/01/probabilistic-structures-web-analytics-data-mining/

Problem:

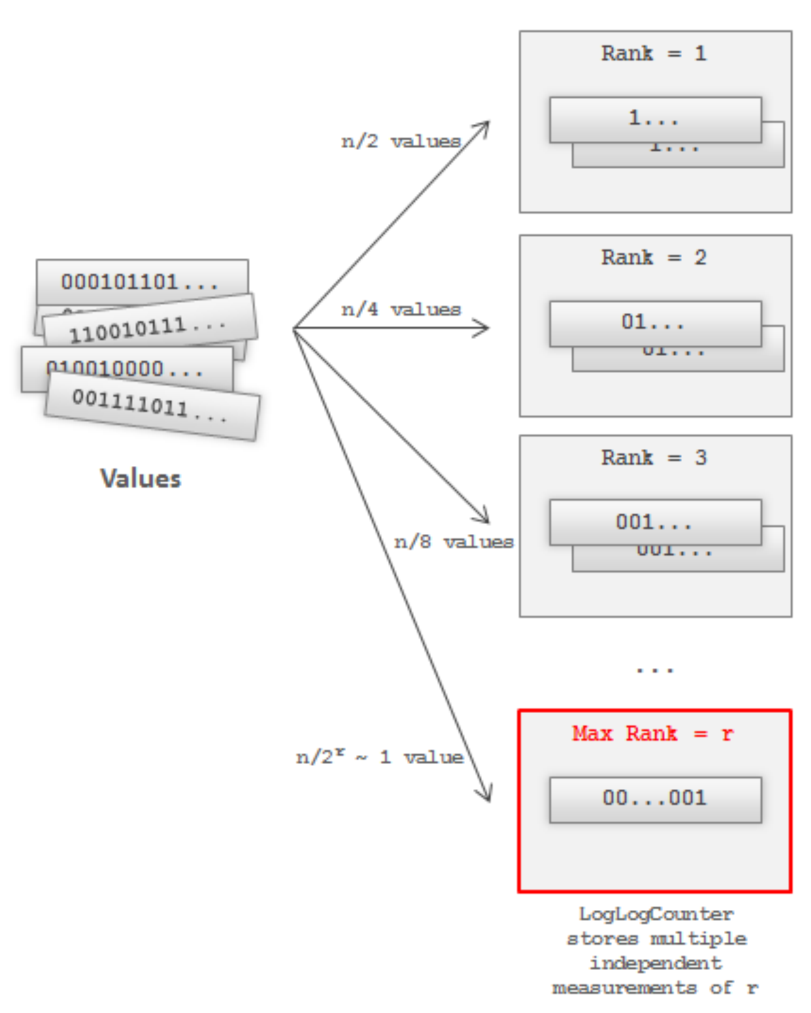
How many distinct items are in this set?

HyperLogLog[7]

- Extension of prior LogLog algorithm
- Approximates number of unique elements in a set using very little memory
- Technique involves hashing original set elements, then estimating cardinality by calculating maximum number of leading zeros in binary string version of hashed elements

[7] http://algo.inria.fr/flajolet/Publications/FlFuGaMe07.pdf

HyperLogLog[2]



[2] https://highlyscalable.wordpress.com/2012/05/01/probabilistic-structures-web-analytics-data-mining/

HyperLogLog

- Able to estimate cardinalities beyond **10^9**, with error rate of **2%**, using only **1.5kb** memory
- See this one a lot in ad serving and analytics contexts
- Many OSS implementations exist; well-known implementation in Redis[8]

[8] http://redis.io/commands/pfcount

Matching Exercise

- Can we track the most-retweeted tweets with the *#walkingdead* hashtag?
- How many people have visited this article today?
- Do we have to include this server in our search list?

Matching Exercise

- Can we track the most-retweeted tweets with the *#walkingdead* hashtag?
  - **Count-min sketch**
- How many people have visited this article today?
  - **HyperLogLog**
- Do we have to include this server in our search list?
  - **Bloom filter**

Conclusions

- Optimizing search space to specific dataset problems should always result in space/performance gains; trade-off is time
- We can split the difference by tuning general-purpose space-optimized structures if we are willing to give up a small amount of accuracy
- *"Big Data"* doesn't have to mean **MOAR**; can also mean making informed compromises on accuracy VS time, money, & complexity

Additional Reading

- https://www.mapr.com/blog/some-important-streaming-algorithms-you-should-know-about
- http://www.infoq.com/presentations/abstract-algebra-analytics
- http://pages.cs.wisc.edu/~cao/papers/summary-cache/node8.html
- http://www.umiacs.umd.edu/~amit/Papers/goyalSketchEMNLP11.pdf

Additional Reading

» https://sites.google.com/site/countminsketch/home/faq

» http://tech.adroll.com/blog/data/2013/07/10/hll-minhash.html

» http://highscalability.com/blog/2012/4/5/big-data-counting-how-to-count-a-billion-distinct-objects-us.html

Questions?