Random Sampling applied to Rapid Disk Analysis

System & Network Engineering — Research Project

Nicolas Canceill

University of Amsterdam

Nederlands Forensisch Instituut
Ministerie van Veiligheid en Justitie

July 4, 2013

1. Rapid Disk Analysis
2. The Math
3. The Aftermath
4. Conclusions

Introduction

Background

Assoc. Prof. S. Garfinkel — *Navy Postgraduate School*

- Advanced Forensics Format
- The Sleuth Kit
- Better analysis for digital evidence

"Searching a 1TB hard drive in 10 minutes" (ACM 2013)

Research

E. van Eijk, Z. Geradts — *Nederlands Forensisch Instituut*

- Stability?
- Scalability?
- Precision?

1 Rapid Disk Analysis

2 The Math

3 The Aftermath

4 Conclusions

Rapid Analysis: Why?

Traditionally: investigation was "leisurely"

- Reading a 1TB hard drive: about 3.5h
- The cost of "seek": $1 \times 36\text{GB} \approx 100,000 \times 64\text{KiB}$

New challenges

- Large installations: computers room, datacenter...
- Forensics control at checkpoints: border crossing, airports...

"The bomb will go off in the next hour!"

Rapid Analysis: What for?

- ~~Profit~~
- Indications

Data analysis

- Determine free/wiped space
- Characterize data based on signatures
- Hash sectors to look for specific data

Rapid Analysis: How?

Data characteristics

- Described (header/trailer)
- Encoded/formatted
- Sectorized and distributed

Analysis strategies

- Simplify: hashing
- Tolerate: extract signature
- Reduce: **random sampling**

Research scope

Research question

How can random sampling help forensically investigate hard disk drives?

- What kind of indications may be provided?
- Which parameters are in play?
- Which degree of certainty may be achieved?

1 Rapid Disk Analysis

2 The Math

3 The Aftermath

4 Conclusions

Analysis process

Built on top of S. Garfinkel's `frag_find` tool

Input

- Image file to search
- Data-set/Signatures-set to look for
- Parameters: hashing, **sampling**, **tolerance**

Process

- Build Bloom filter (hashing)
- **Select sample**
- For each block **in sample**: filter (and compare)

Random sampling: Basic model

Using a random sample of a statistical population to estimate/predict characteristics

Simple scenario

"Is this hard drive empty/wiped?"

- M empty blocks out of N
- n sampled blocks out of N

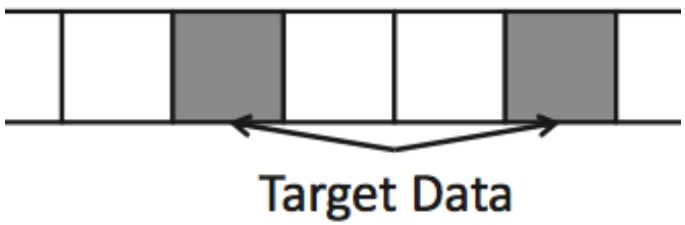
Error rate

The probability to sample only empty blocks:

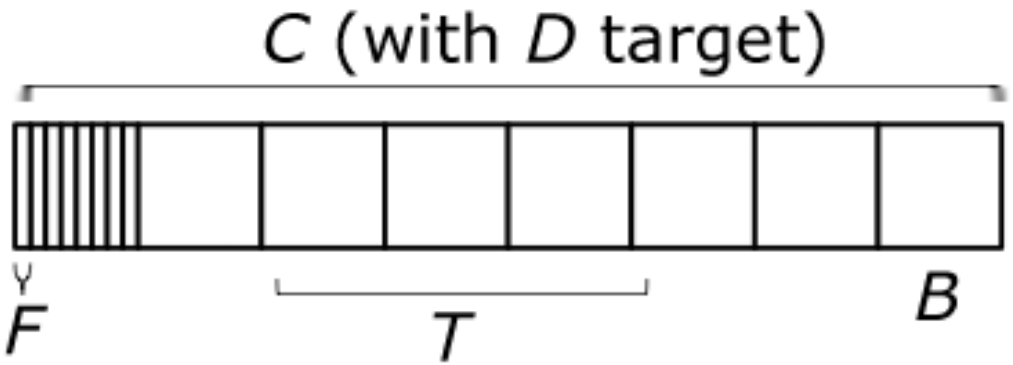
$$E = \prod_{i=1}^{i=n} \frac{N - (i-1) - M}{N - (i-1)}$$

Random sampling: Data layout

Data is sectorized:



Data is not always aligned:



Random sampling: Advanced model

A more realistic scenario

"Does this hard drive contain the target block?"

- All possible offsets: overlap transactions by $B - F$
- All possible transactions: $N = \left\lceil \frac{C}{T-(B-F)} \right\rceil$
- All target transactions: $M = \left\lceil \frac{D}{T} \right\rceil$

Error rate

The probability to miss all target blocks:

$$E = \prod_{i=1}^{i=n} \frac{\left\lceil \frac{C}{T-(B-F)} \right\rceil - (i-1) - \left\lceil \frac{D}{T} \right\rceil}{\left\lceil \frac{C}{T-(B-F)} \right\rceil - (i-1)}$$

Experimental protocol

Experimental image set

Parameters: image size, sector size, % of empty sectors, length of target data, offset size
Input: Random files and NSRL Reference DataSet

Experimental process

Parameters: image size, sector size, transaction size, sampling fraction

- Randomly select a **master file signature**
- Generate **several images** (length of target data, % of empty sectors)
- Successively run **several timed searches**

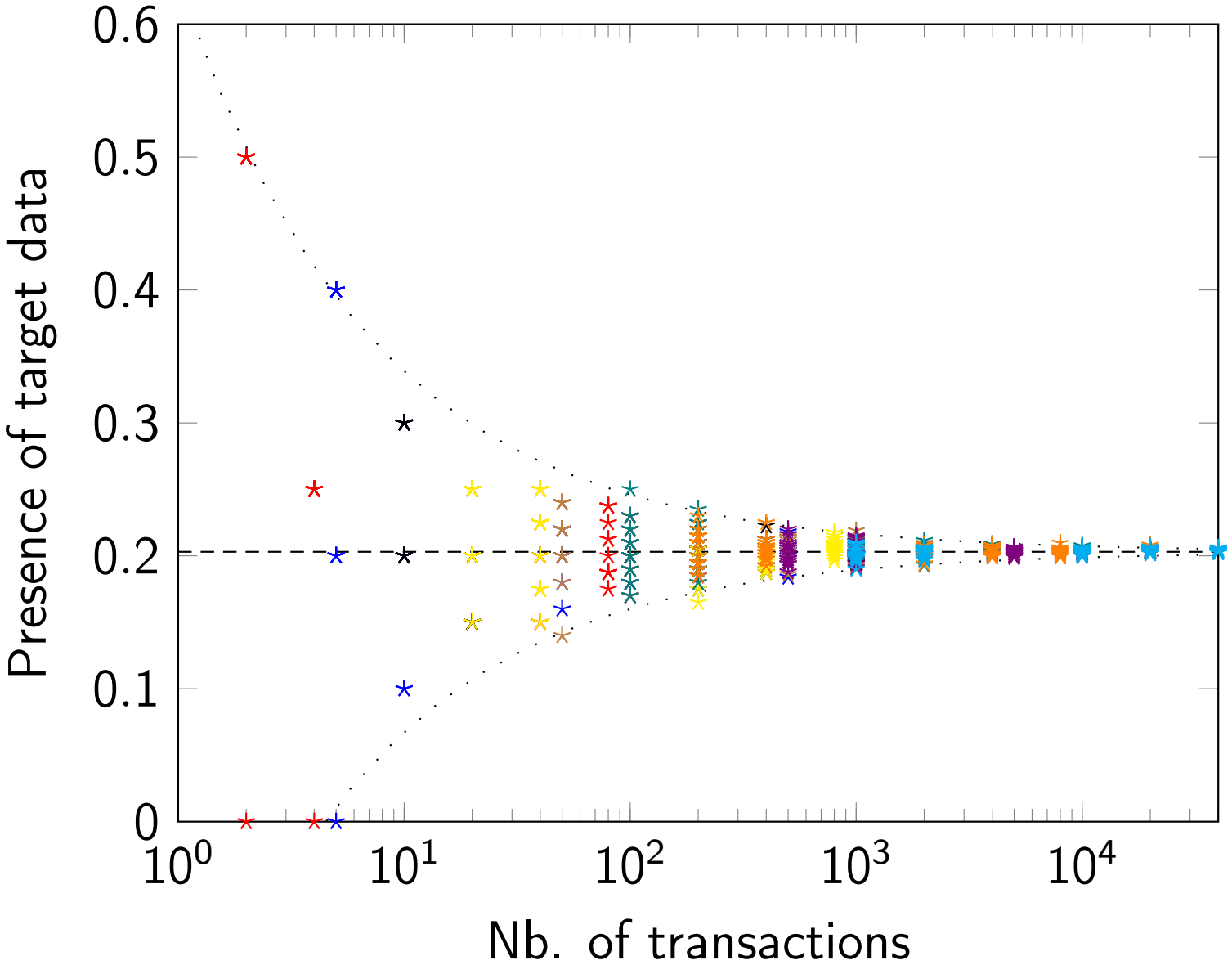
1 Rapid Disk Analysis

2 The Math

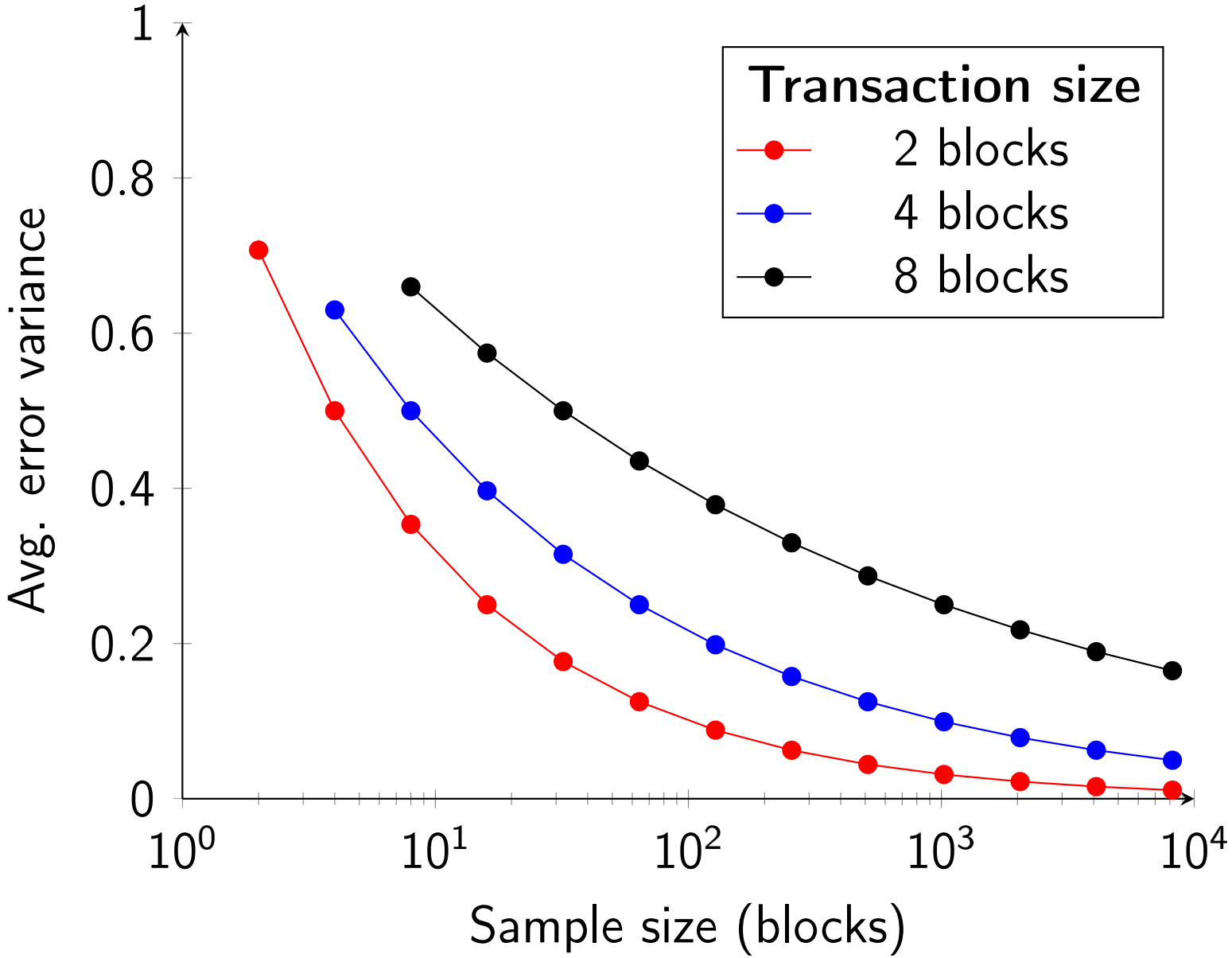
3 The Aftermath

4 Conclusions

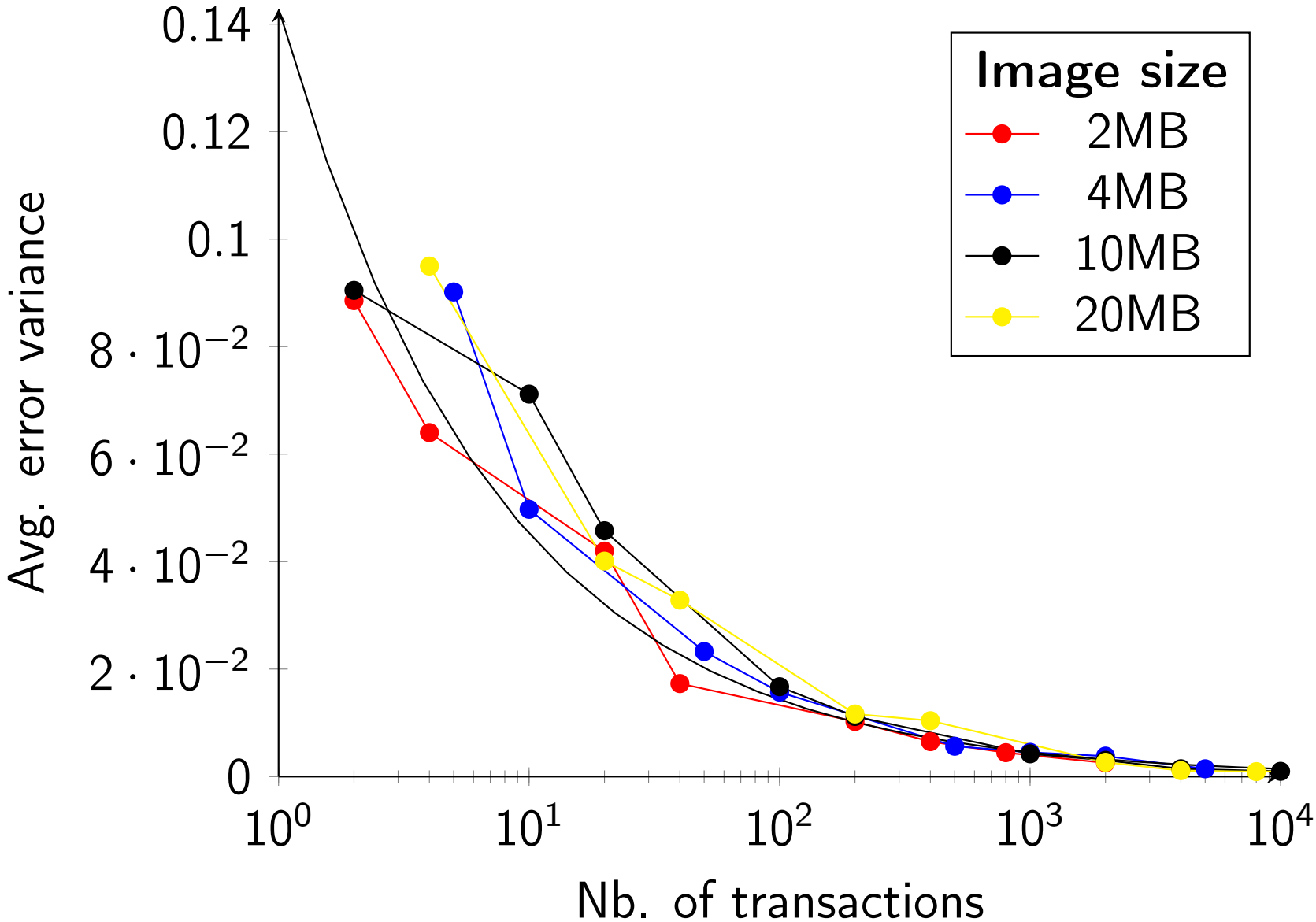
Results: statistical distribution



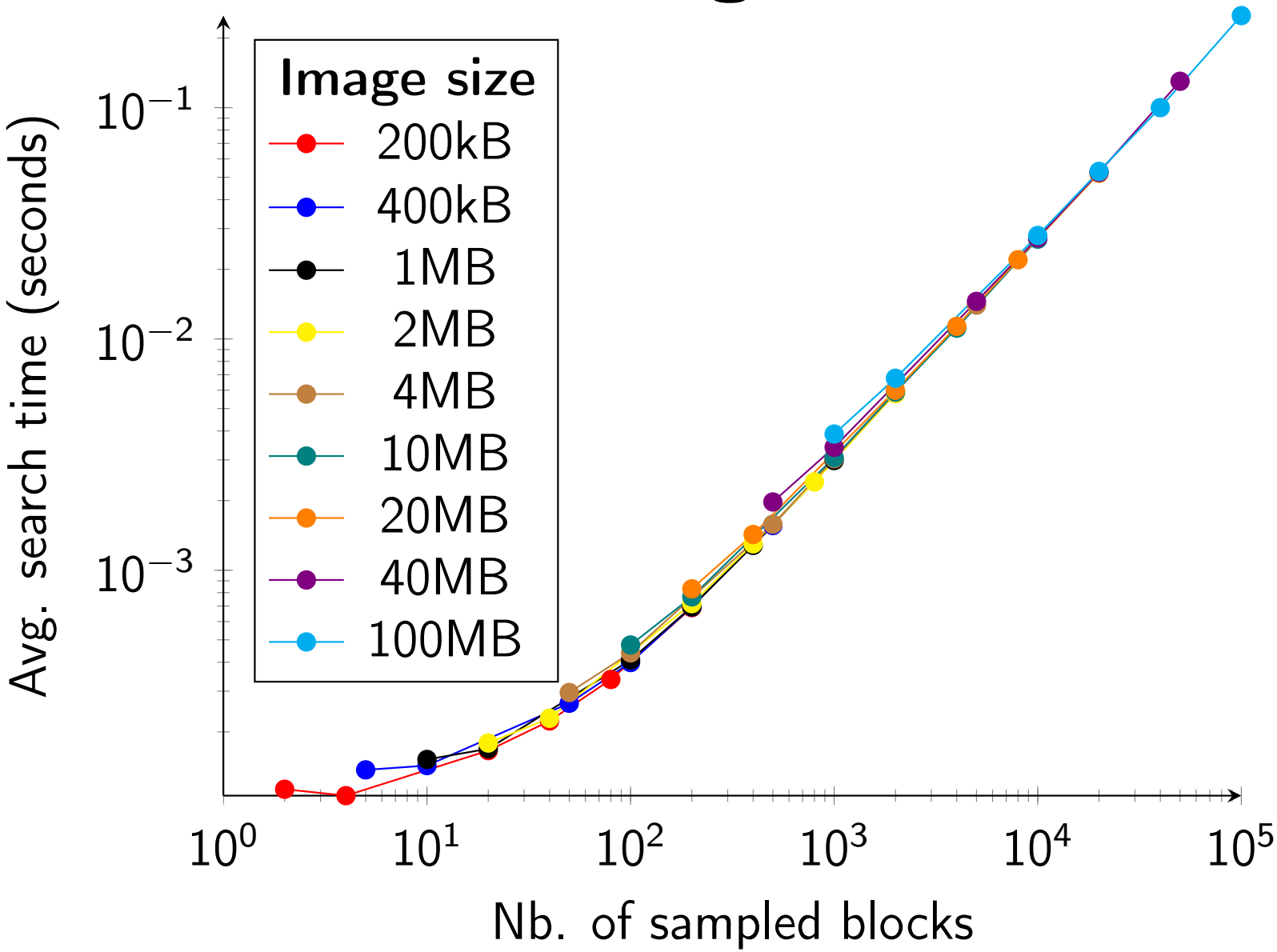
Results: block-to-transaction scaling



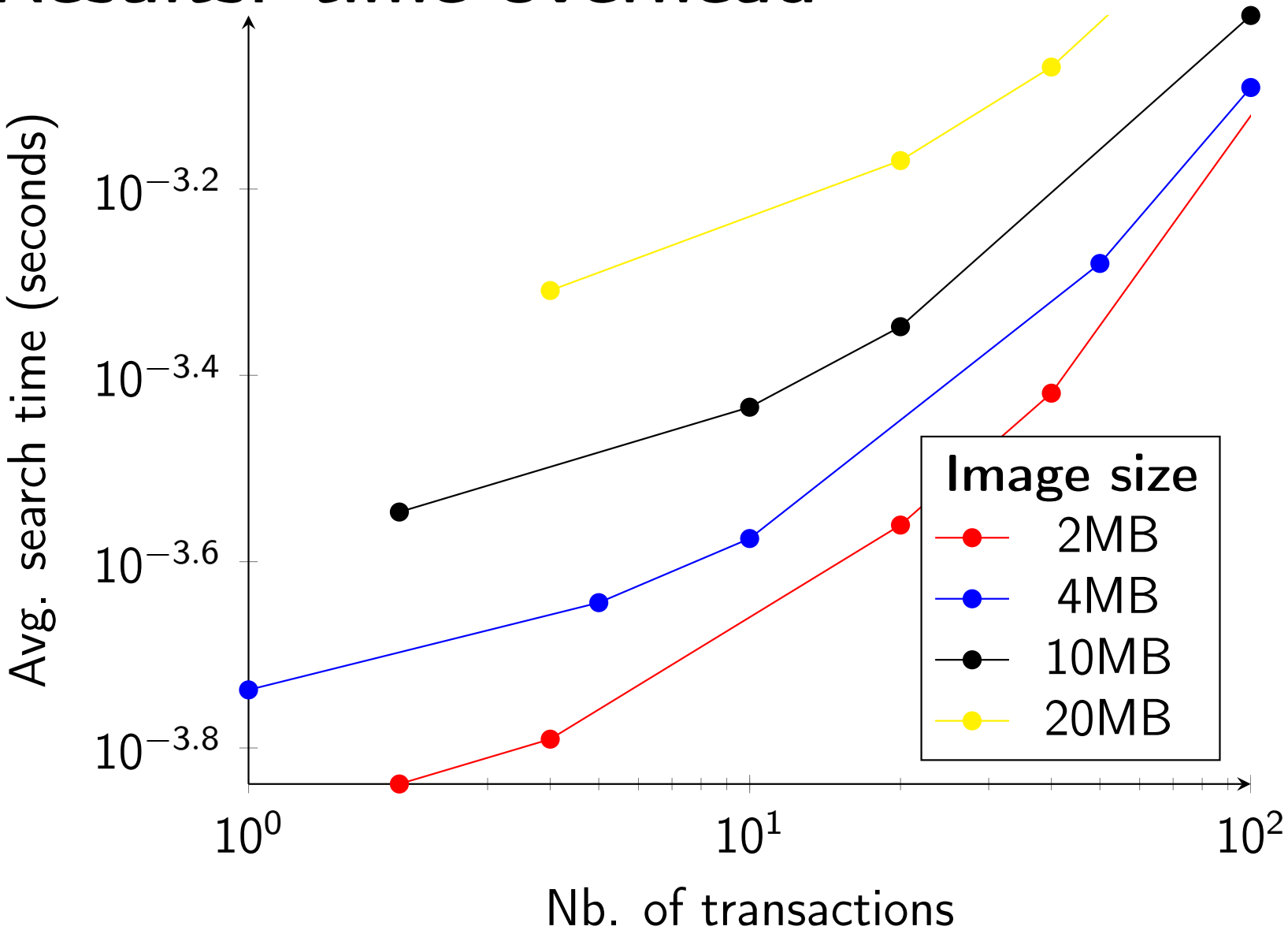
Results: precision scaling



Results: time scaling



Results: time overhead



1 Rapid Disk Analysis

2 The Math

3 The Aftermath

4 Conclusions

Contributions

Main findings

Parameters analyzed:

- Image characteristics: image size, sector size, data alignment, size of target data
- Sampling settings: sample size, transaction size, tolerance

Scalability:

- Sample size scales with time: $S \sim t$
- Error rate scales with time: $E \sim \frac{1}{\sqrt{t}}$

Public material

Fork of S. Garfinkel's tools on GitHub
Most of experimental scripts on Gist

Research answers

- What kind of indications may be provided?
  Presence/absence of target data or signature
- Which parameters are in play?
  Disk and data characteristics
  Sampling parameters
- Which degree of certainty may be achieved?
  Certainty scales well with time
  Insight about target disk will improve certainty

Random sampling is a powerful, scalable, adaptive technique for fast HDD analysis

Efficiency relies on suitable sampling settings, and limited insight on target HDD

Further research

Improving insight of target

- Pre-determine sector size, data alignment
- Look for optimal block-to-transaction ratio
- One step further: pre-sampling

Automate decision process

- Optimal time spending
- Automatic settings balance
- Simple user-side: time or certainty

Appendix 1: Bloom Filter (a)

Hash-based filtering technique

Initialize

An array of n bits set to zero

k different hash functions uniformly mapping to $[0 - n]$

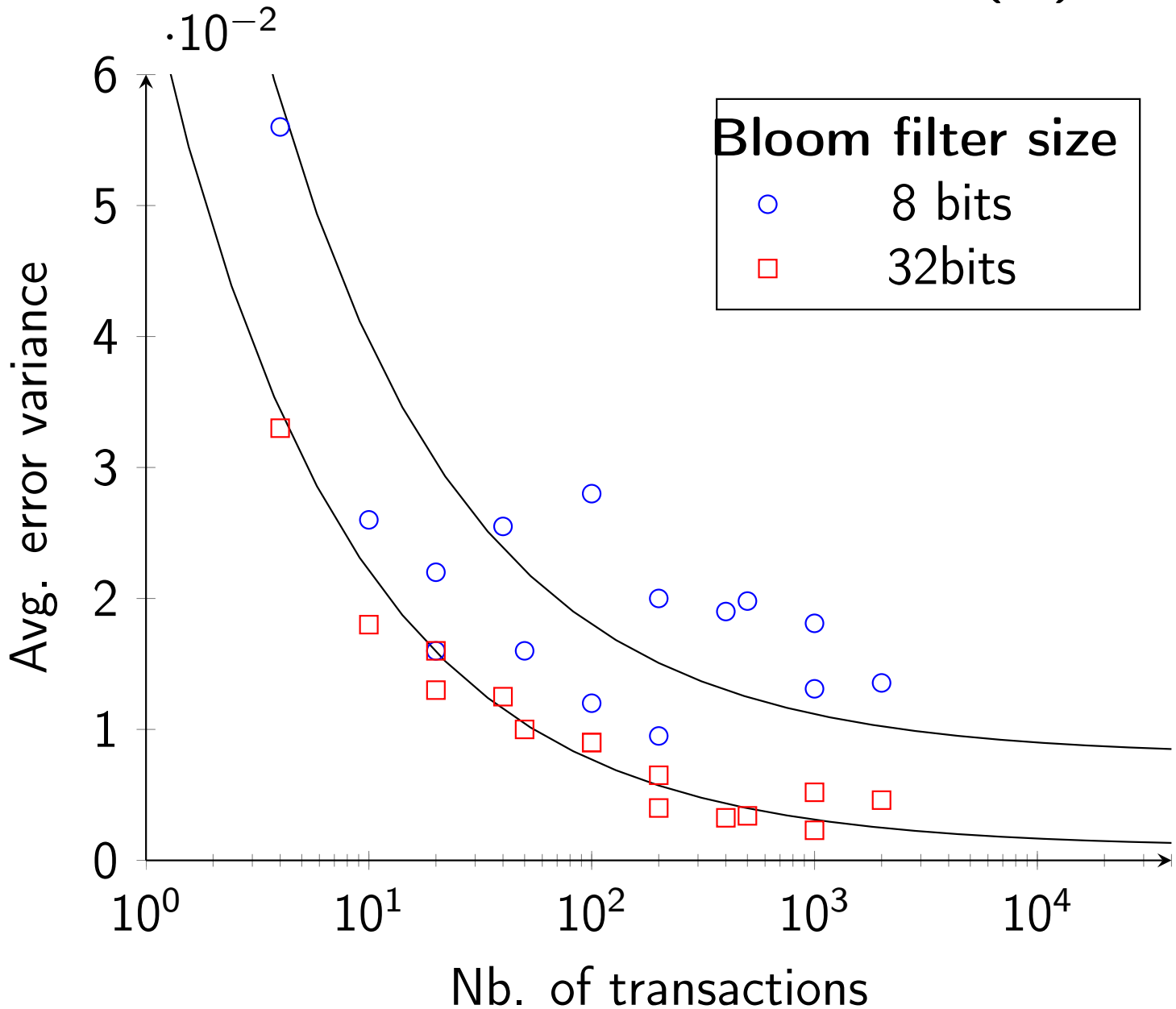
Add an element

- Apply functions to compute k integers in $[0 - n]$
- Set k corresponding bits to 1

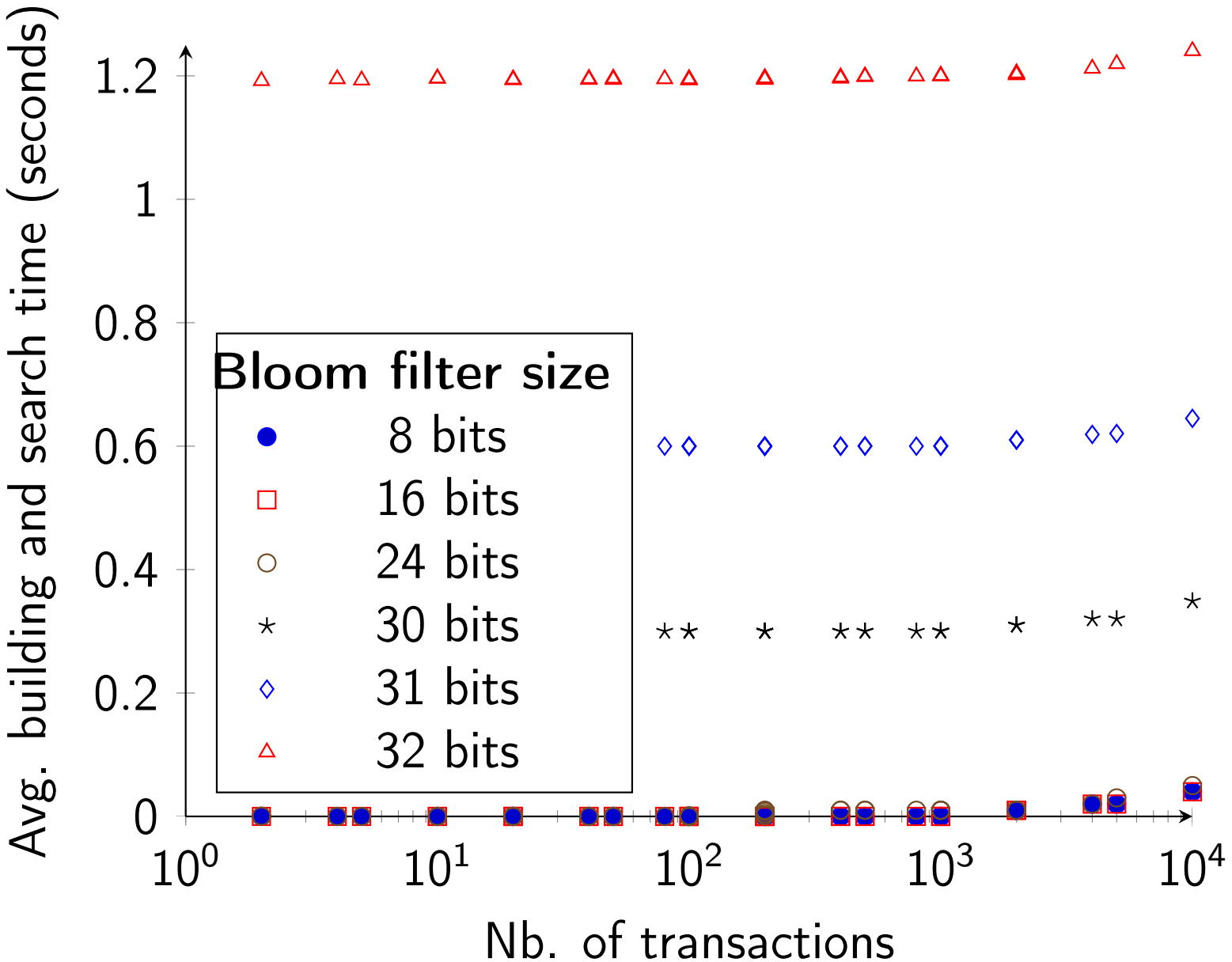
Query an element

- Apply functions to compute k integers in $[0 - n]$
- Check if k corresponding bits are all 1

Appendix 1: Bloom Filter (b)



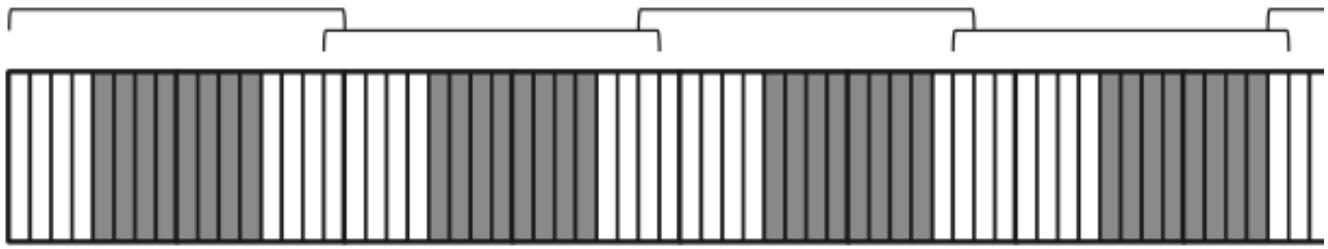
Appendix 1: Bloom Filter (c)



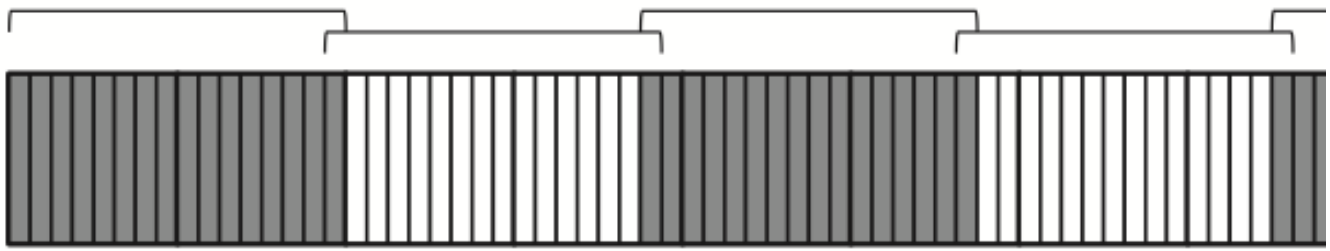
Appendix 2: Data layout (a)

Optimal transaction size depends on sector size

Best case:



Worst case:



Appendix 2: Data layout (b)

Optimal transaction size depends on data layout