SeqCDC: Hashless Content-Defined Chunking for Data Deduplication

Sreeharsha Udayashankar, Abdelrahman Baba and Samer Al-Kiswany

UNIVERSITY OF WATERLOO

ACM / IFIP International Middleware Conference, 2024

Introduction

- Data explosion
  - Global data production expected to exceed 180 ZB by 2025 [1]
- Mechanisms
  - Distributed file systems [2]
  - Storage Architectures [3]
  - Data Deduplication [4]

aws

Google Cloud

Microsoft Azure

Hewlett Packard Enterprise

IBM Cloud

DELL EMC

[1] Arne Holst. *Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2025*. Statista, 2021.

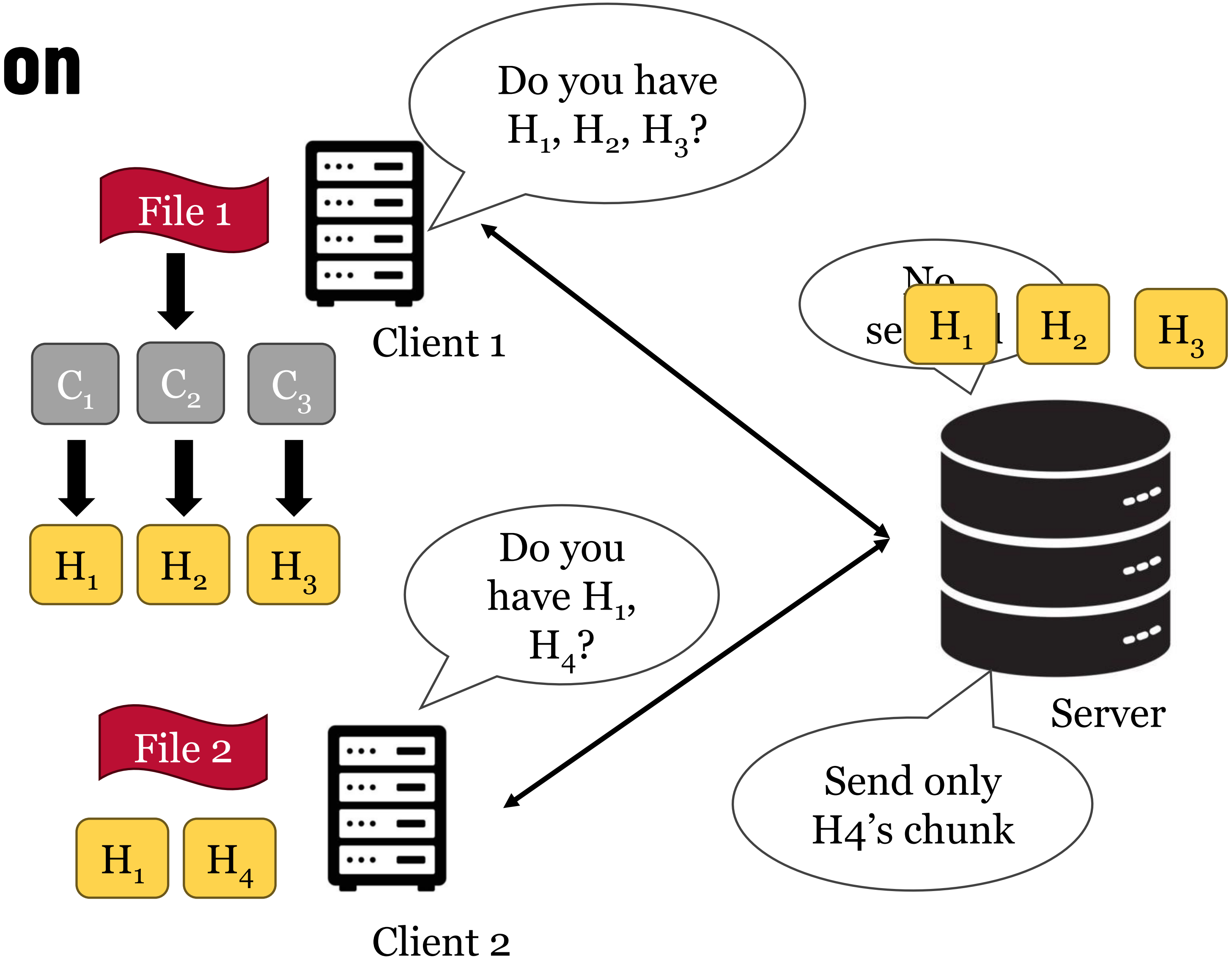
[2] Sanjay Ghemawat et al. *The Google File System*. SIGOPS Oper. Syst, 2003.

[3] Peter M Chen et al. *RAID: High-performance, reliable secondary storage*. ACM Computing Surveys (CSUR), 1994.

[4] Nagapramod Mandagere et al.. *Demystifying data deduplication*. ACM/IFIP/USENIX Middleware'08 Conference, 2008

Introduction: Deduplication

- Data Deduplication [5]
  - Identify and eliminate duplicate data
- Deduplication Overview [6]
  - File Chunking and Hashing
  - Fingerprint Comparison
  - Data Storage



[5] Dutch T Meyer et al. *A study of practical deduplication*. ACM Transactions on Storage (ToS), 2012.

[6] Alan Liu et al. *Dedupbench: A Benchmarking Tool for Data Chunking Techniques*. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2023.

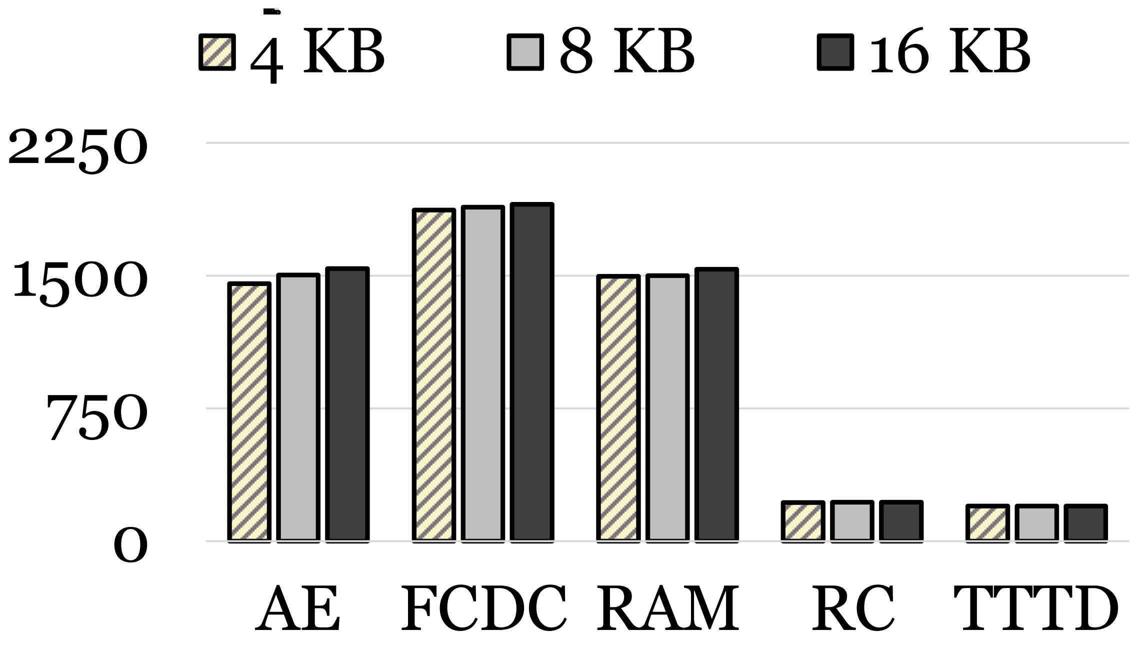
Introduction: File Chunking

- Content-Defined Chunking (CDC) [7, 8, 9]

Existing CDC algorithms are slow!

Existing CDC algorithms designed for small chunks!

- Systems in production favor larger chunks
  - Metadata concerns
  - Storage fragmentation concerns



(a) Chunking Throughput on Random Data

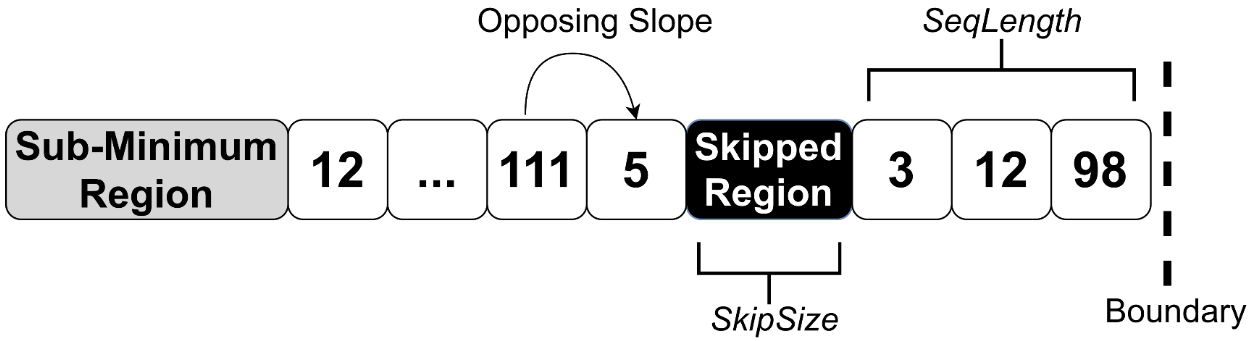
[7] Athicha Muthitacharoen et al. *A low-bandwidth network file system*. SOSP, 2001.

[8] Yucheng Zhang et al. *AE: An asymmetric extremum content defined chunking algorithm for fast and bandwidth-efficient data deduplication*. INFOCOM, 2015.

[9] Kave Eshghi et al. *A framework for analyzing and improving content-based chunking algorithms*. Hewlett-Packard Labs Technical Report, 2005

SeqCDC

- Novel CDC algorithm
  - Lightweight boundary detection to reduce complexity
  - Content-defined skipping to selectively avoid scanning *unfavorable regions*
  - **1.5x – 3x higher throughput** than state-of-the-art



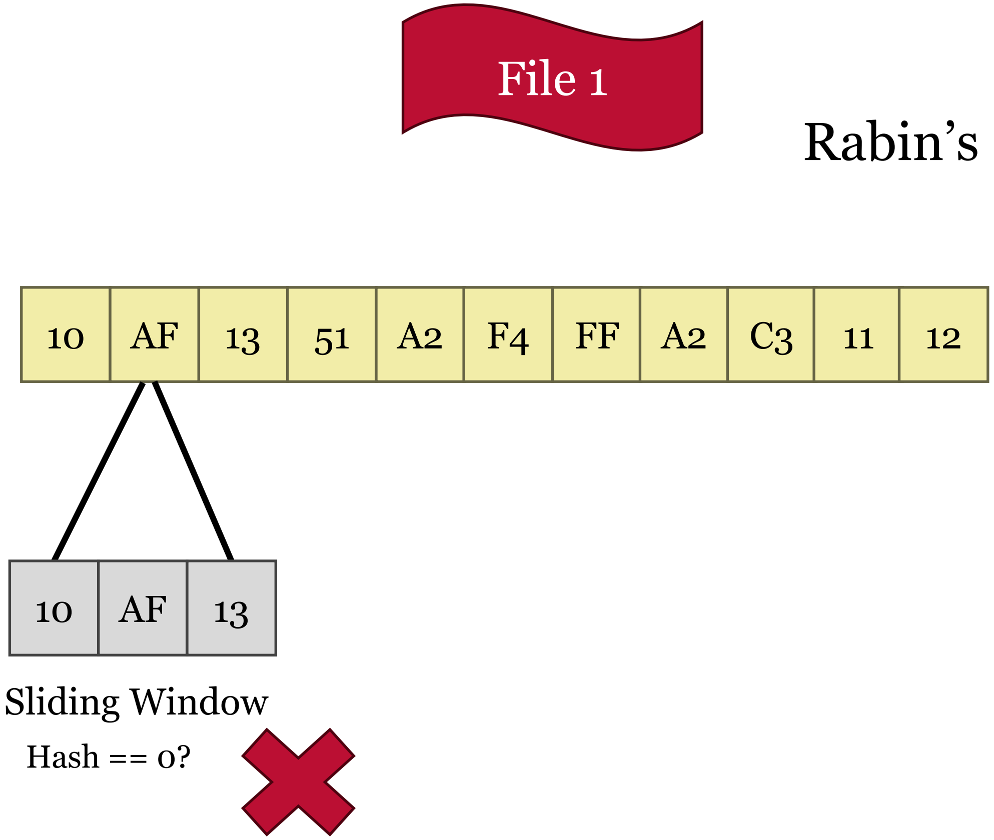
(a) SeqCDC

Outline

- Introduction
- Background
- Design
- Evaluation
- Conclusion

Background: Content-Defined Chunking

Rabin's chunking [7]



File 1

10	AF	13	51	A2	F4	FF	A2	C3	11	12

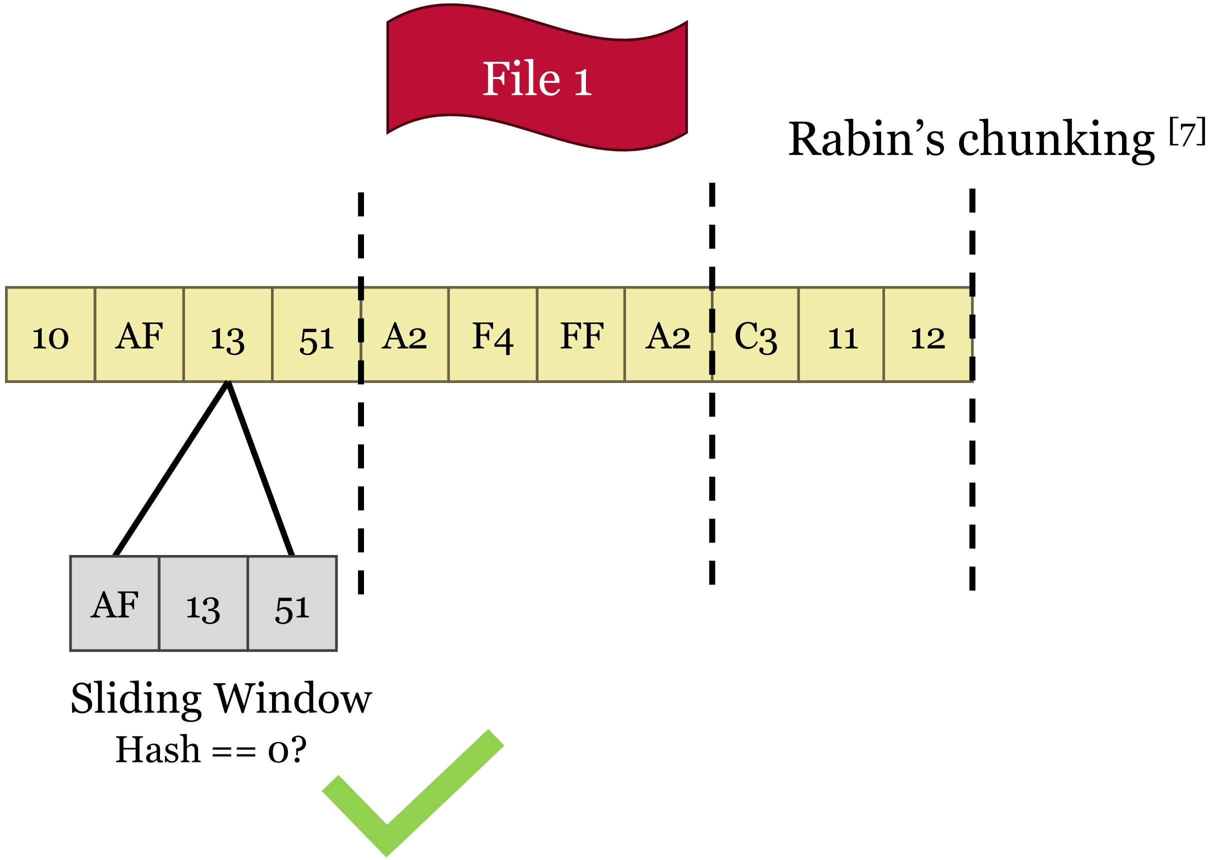
10	AF	13

Sliding Window

Hash == 0? ✗

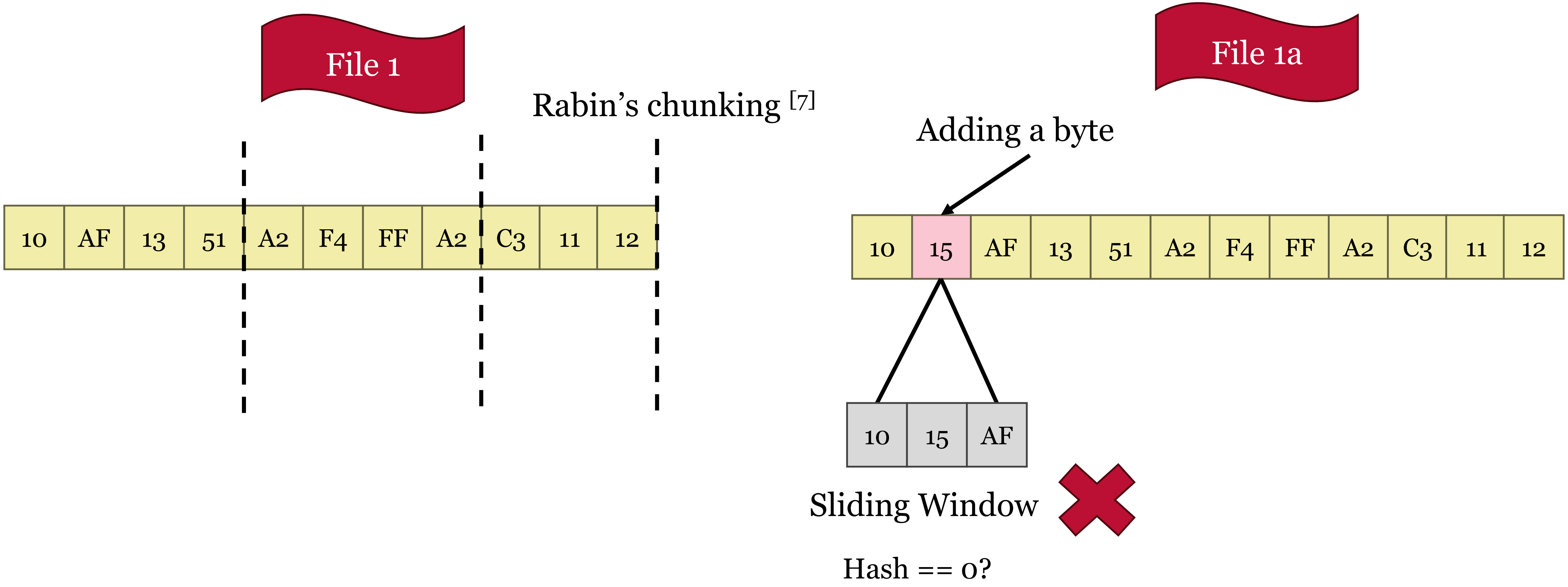
[7] Athicha Muthitacharoen et al. *A low-bandwidth network file system*. SOSP, 2001.

Background: Content-Defined Chunking



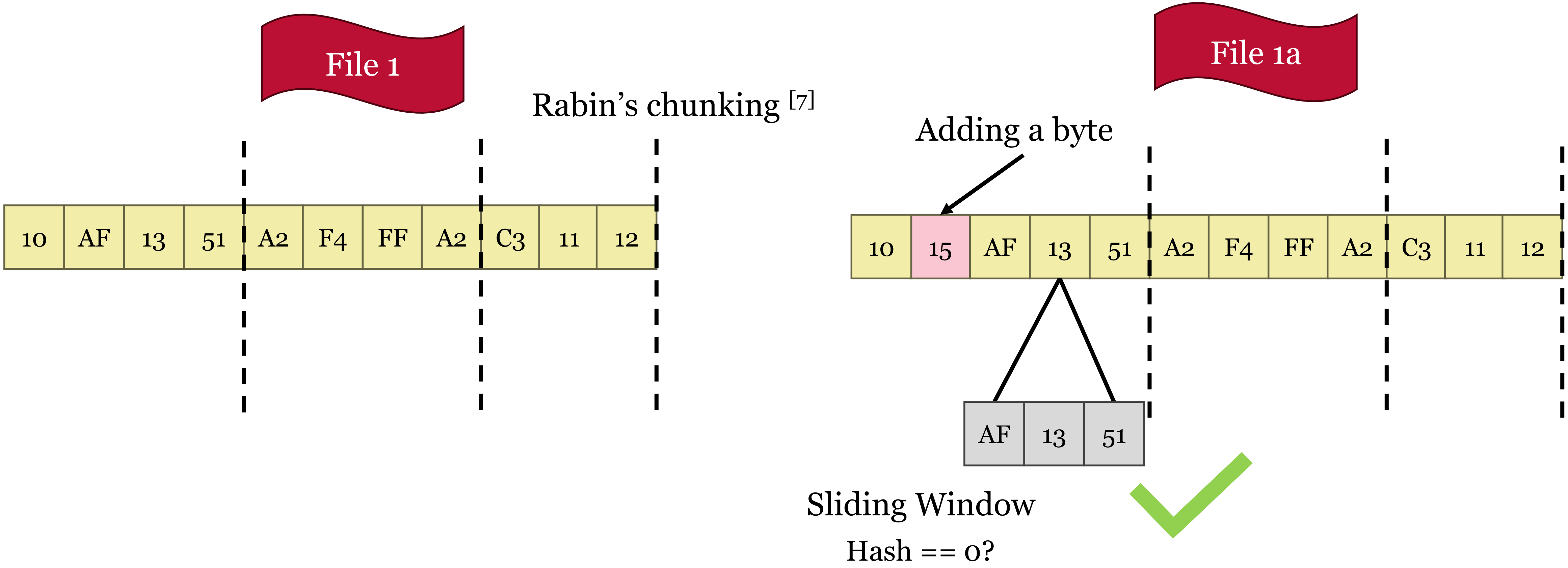
[7] Athicha Muthitacharoen et al. *A low-bandwidth network file system*. SOSP, 2001.

Background: Content-Defined Chunking


File 1
Rabin's chunking [7]
10 | AF | 13 | 51 | A2 | F4 | FF | A2 | C3 | 11 | 12
File 1a
Adding a byte
10 | 15 | AF | 13 | 51 | A2 | F4 | FF | A2 | C3 | 11 | 12
10 | 15 | AF
Sliding Window
Hash == 0?

[7] Athicha Muthitacharoen et al. *A low-bandwidth network file system*. SOSP, 2001.

Background: Content-Defined Chunking



Only one chunk is different, the rest are the same

[7] Athicha Muthitacharoen et al. *A low-bandwidth network file system*. SOSP, 2001.

Background: Issues with Traditional CDC

Traditional CDC

- Expensive boundary detection
- Large amount of data to scan
- Scanned amount does not change with chunk size

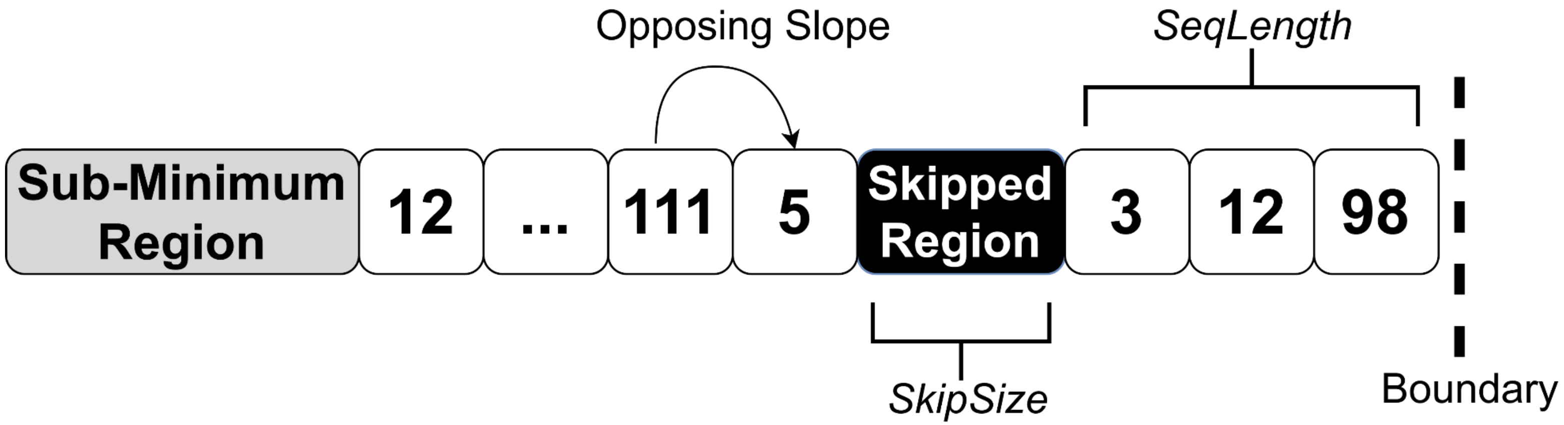
Can we chunk the data without scanning *all* of it?

Outline

- Introduction
- Background
- Design
- Evaluation
- Conclusion

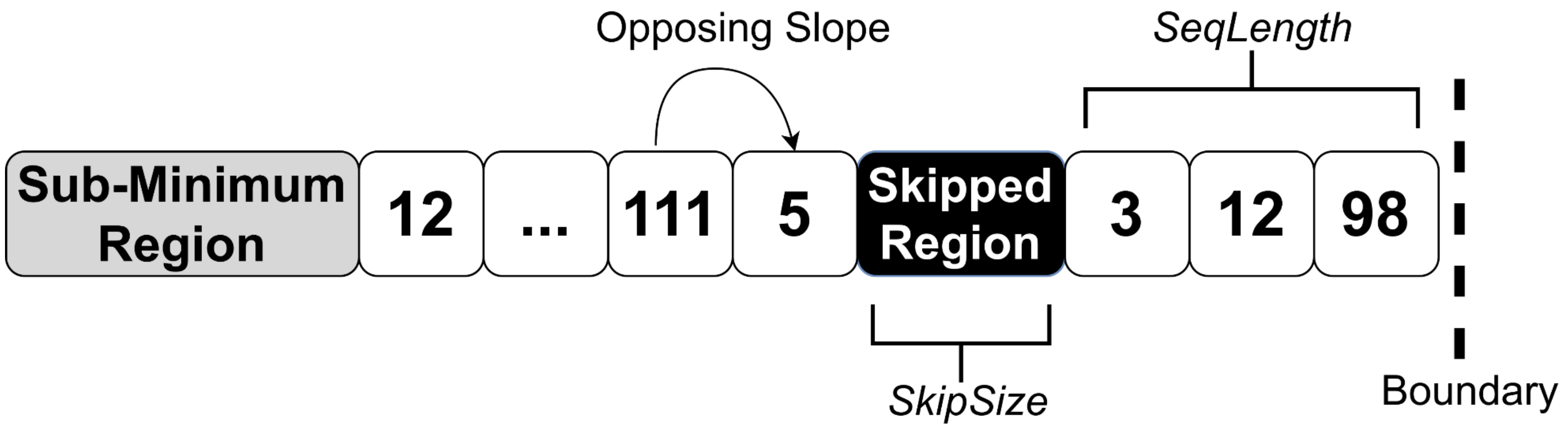
SeqCDC

- Insert chunk boundaries when fixed-length sequences are detected
    - Monotonically increasing / decreasing
    - *SeqLength*



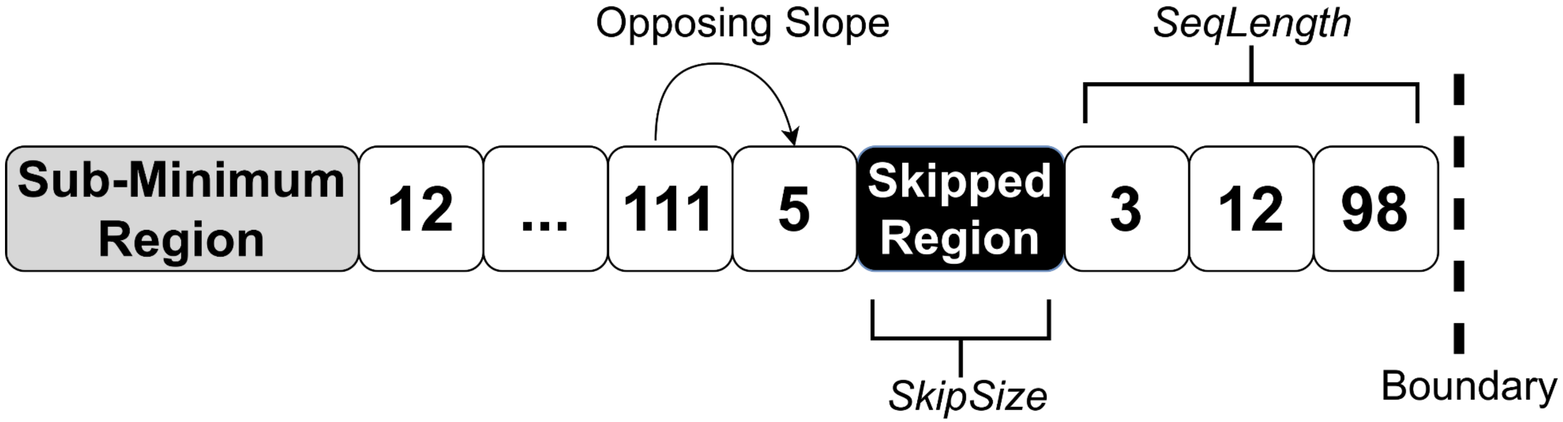
SeqCDC

- Skip scanning the sub-minimum region
  - Minimum chunk size
  - Similar to existing CDC algorithms



SeqCDC

- Skipped regions
  - *Unfavorable:* Opposing slope bytes
  - When triggered, skip scanning the next *SkipSize* bytes



SeqCDC

Traditional CDC

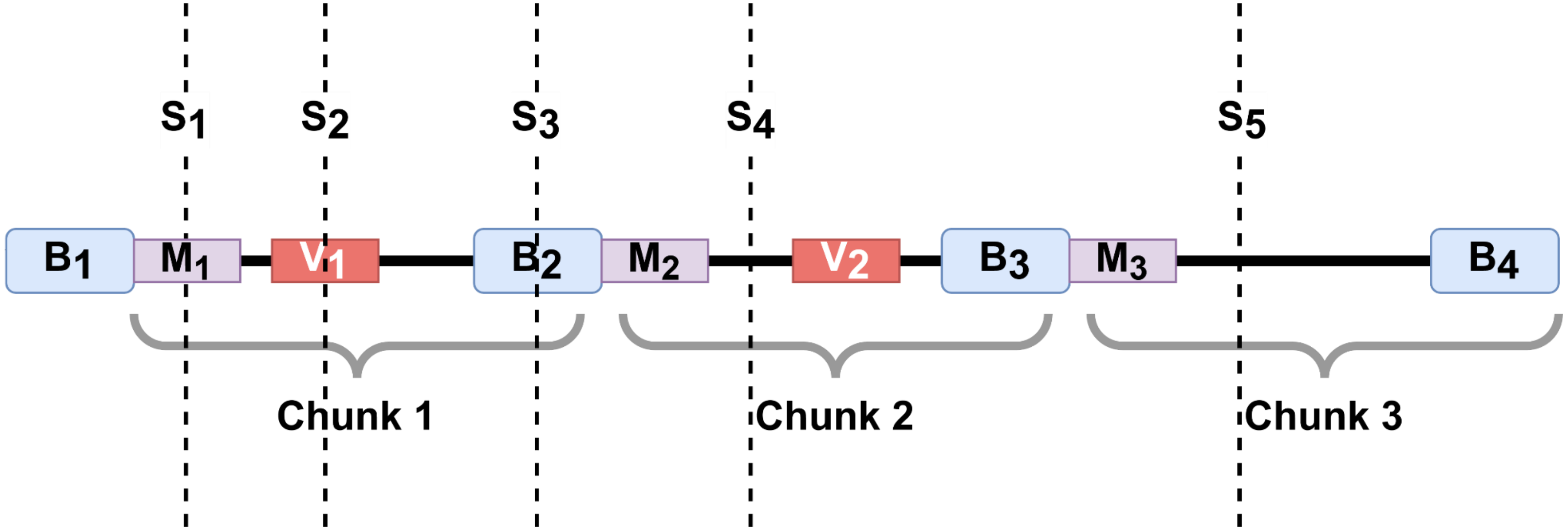
- Expensive boundary detection
- Large amount of data to scan
- Scanned amount does not change with chunk size

SeqCDC

- Lightweight and hashless
- Selectively skip *unfavorable regions*
- Larger chunk size => Larger *SkipSize*

SeqCDC

- How much is byte-shift detection affected?
  - Small amounts on real datasets
  - Detailed analysis in paper



a) Different kinds of byte shifts with SeqCDC

Evaluation

Datasets

Rust

bitnami by vmware

redis

Linux

MySQL

Metrics

Space Savings

Chunk Size Distribution

Speed / Throughput

Alternatives

AE [8]

FastCDC [12]

Rabin's Chunking [7]

RAM [13]

TTTD [9]

[7] Athicha Muthitacharoen et al. *A low-bandwidth network file system*. SOSP, 2001.

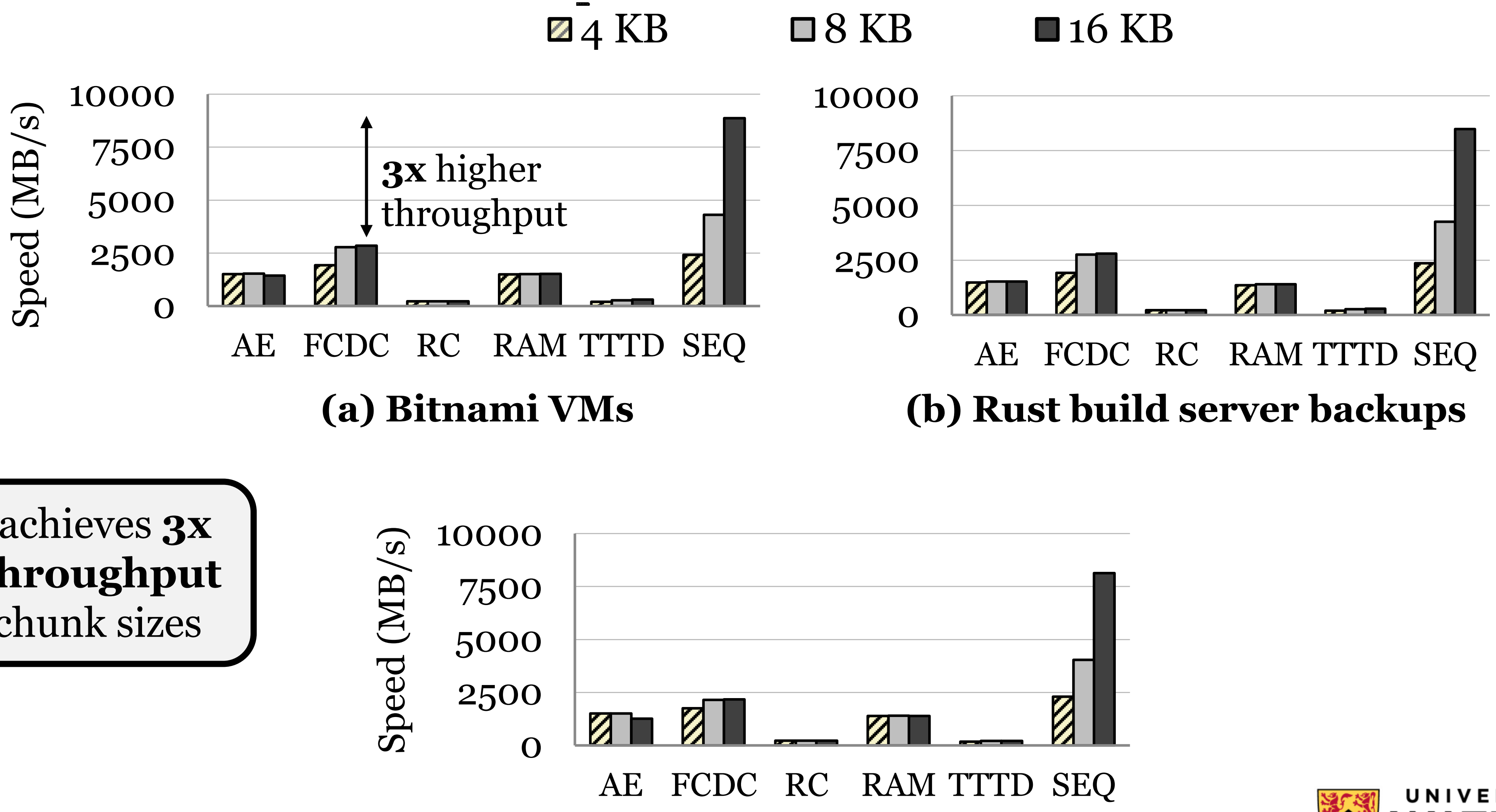
[8] Yucheng Zhang et al. *AE: An asymmetric extremum content defined chunking algorithm for fast and bandwidth-efficient data deduplication*. INFOCOM, 2015.

[9] Kave Eshghi et al. *A framework for analyzing and improving content-based chunking algorithms*. Hewlett-Packard Labs Technical Report, 2005

[12] Wen Xia et al. *FastCDC: A fast and efficient content-defined chunking approach for data deduplication*. USENIX ATC, 2016.

[13] Ryan NS Widodo et al. *A new content-defined chunking algorithm for data deduplication in cloud storage*. Future Generation Computer Systems, 2017.

Evaluation: Chunking Throughput



3x higher throughput

(a) Bitnami VMs

(b) Rust build server backups

(c) Redis backups

SeqCDC achieves **3x higher throughput** at large chunk sizes

Summary

- Data deduplication is used to improve storage efficiency
  - Content-defined chunking algorithms critical to system performance
- SeqCDC
  - Lightweight boundary detection and content-defined data skipping
  - 1.5x – 3x higher chunking throughput with similar space savings
- **Code:** https://github.com/UWASL/dedup-bench

Artifacts Available acm

Artifacts Evaluated acm Functional

UNIVERSITY OF WATERLOO