PC² PADERBORN CENTER FOR PARALLEL COMPUTING

Multi-Level Comparison of Data Deduplication in a Backup Scenario

Dirk Meister
André Brinkmann

Paderborn Center for Parallel Computing
University of Paderborn

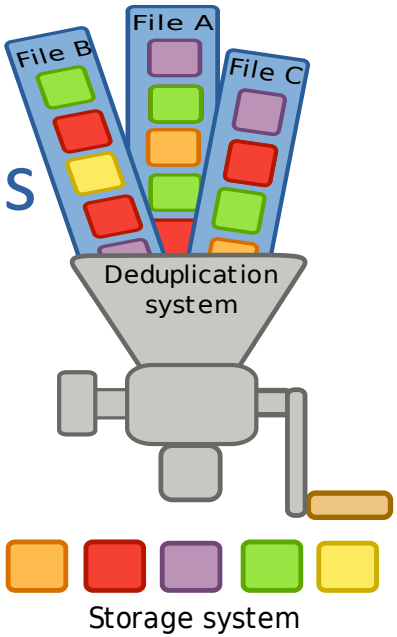
Structure

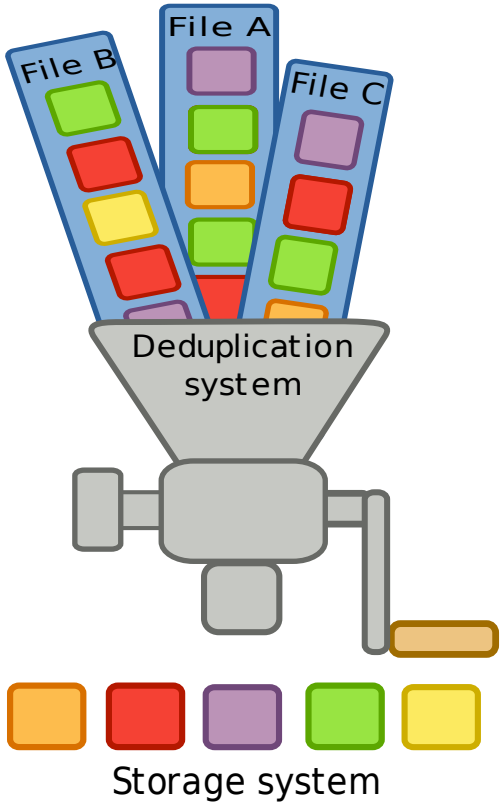
- Introduction
- Large-Scale: Real life system
  - Internal Redundancy
  - Temporal Redundancy
- Small-Scale:
  - Influence of Small Modifications
  - Different File Formats
- Conclusion

Motivation

- Backups critical for enterprises
- 26 full backups ➔ 26x storage demands
- Few changes ➔ high redundancies
- data deduplication :=

  Removal of redundant data to store only unique data





- Fingerprint-based Approaches
- Delta-Encoding-based Approaches

Fingerprinting-based Approaches

- Chunking
- Duplicate Detection
- Storage

Chunking

- Crucial for redundancy detection
- Approaches
  - Static Chunking
  - Content-Defined Chunking
  - Whole-File Chunking
  - Application-Specific Chunking

Static Chunking

- Chunks with a fixed size, e.g. 8 KB
- Very fast
- Problems with shiftings / insertions

```
14159 26535 89793 23846 26433 83279 50288 41971 37510
14159 A2653 58979 32384 62643 38327 95028 84197 13751 0
```

Content-defined Chunking

- Build chunks based on contents only
  - Fingerprint f_p for all substrings of size w
  - Chunk boundary at position p if f_p at the position holds that:

$$f_p \bmod n = c \text{ for a constant } 0 < c \leq n$$

- Chunk size
  - Expected value: n Byte (configurable)
- Also called: Rabin Fingerprint Chunking

Other Approaches

- Whole-File Chunking
  - Fingerprint over whole file contents
  - Finds only exact file duplicates
- Application-Specific Chunking
  - Use knowledge about specific file types
  - e.g. split mail files exactly add attachment boundaries

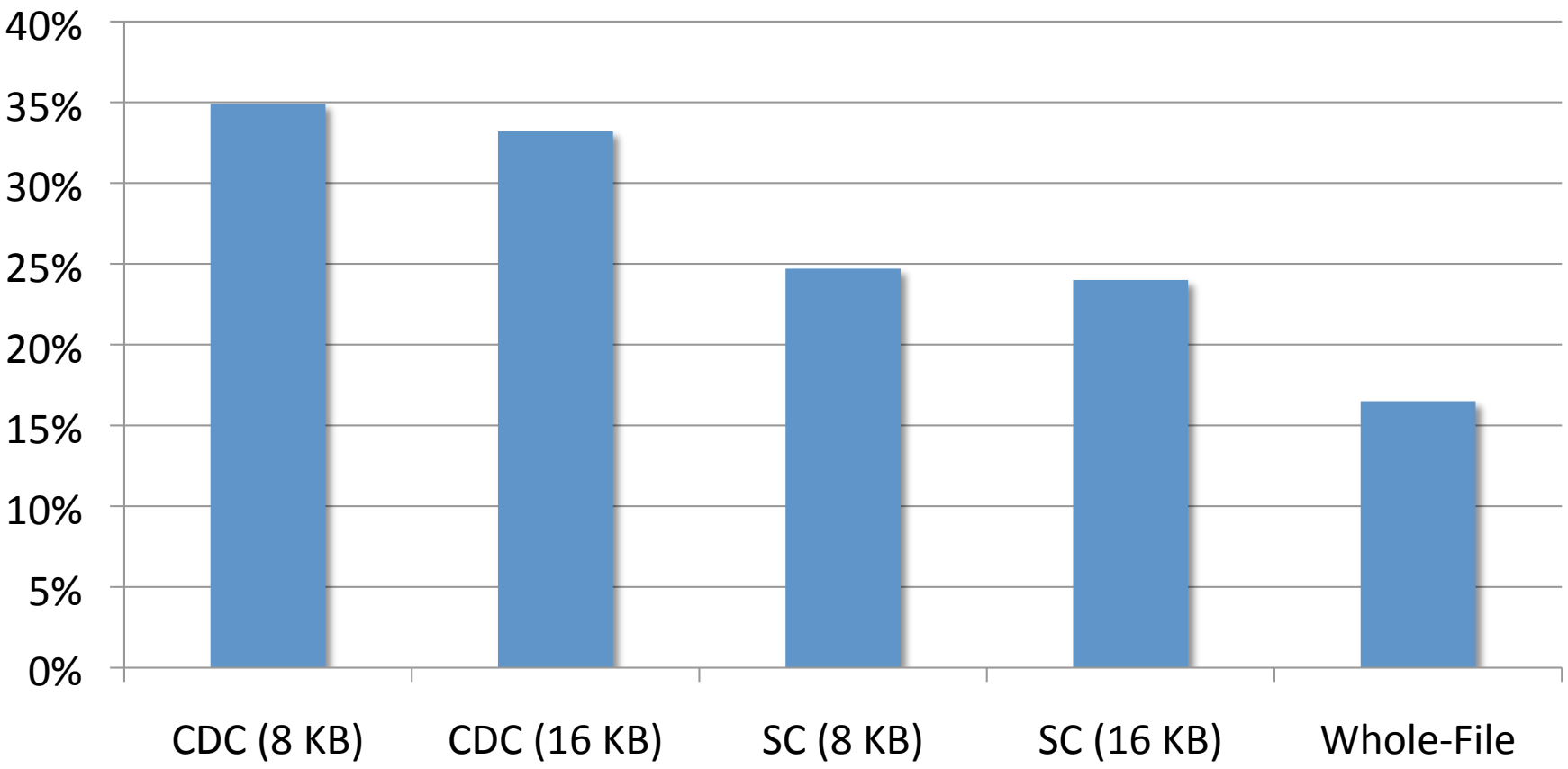
1. PART:
LARGE-SCALE COMPARISION
(OF FINGERPRINTING-BASED APPROACHES)

Real Life System

- User directories of the University of Paderborn
- Weekly measurements

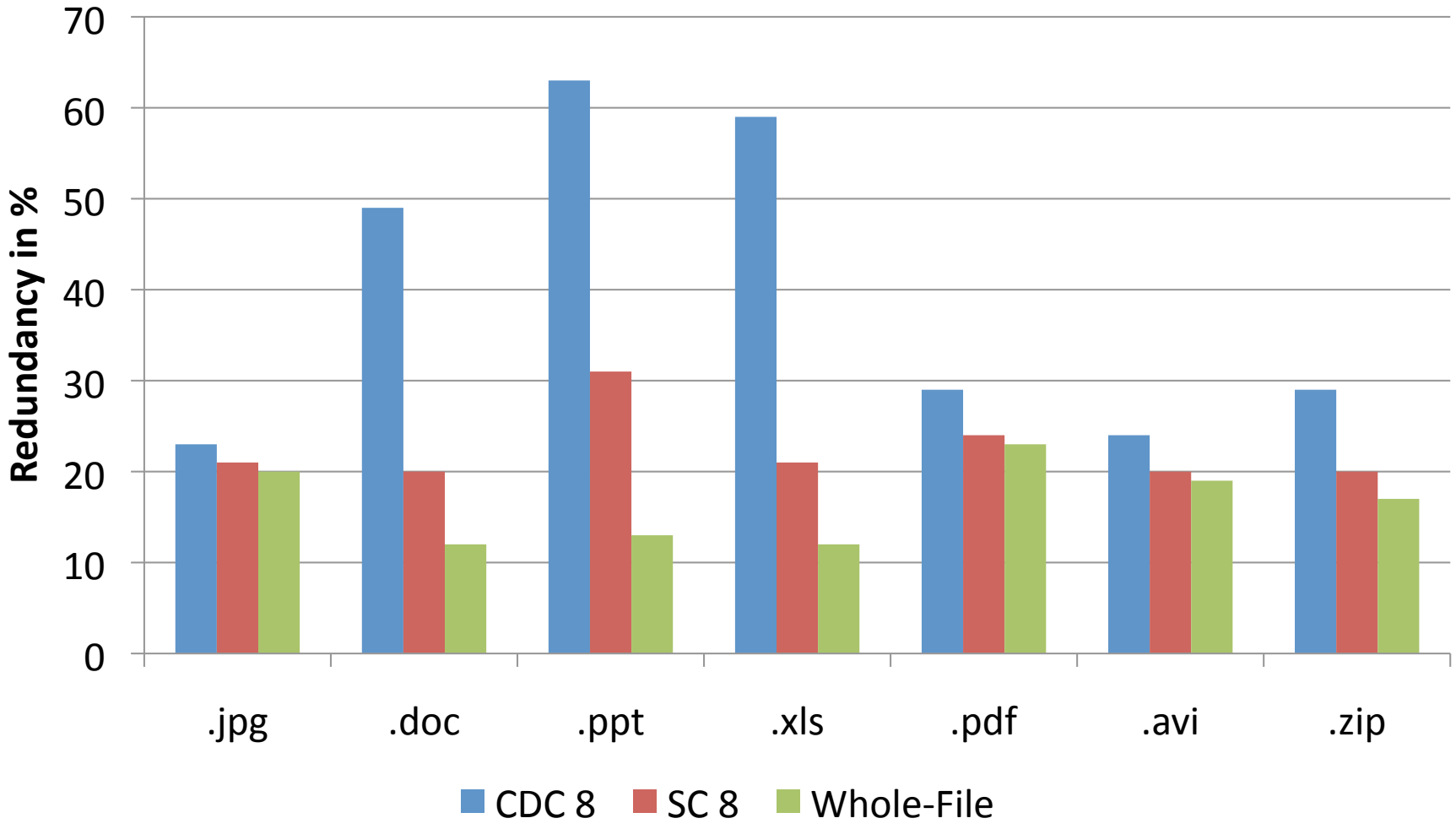
- Key Facts
  - 450 GB data
  - 3.6M files
  - 8.000 active users

Internal Redundancy

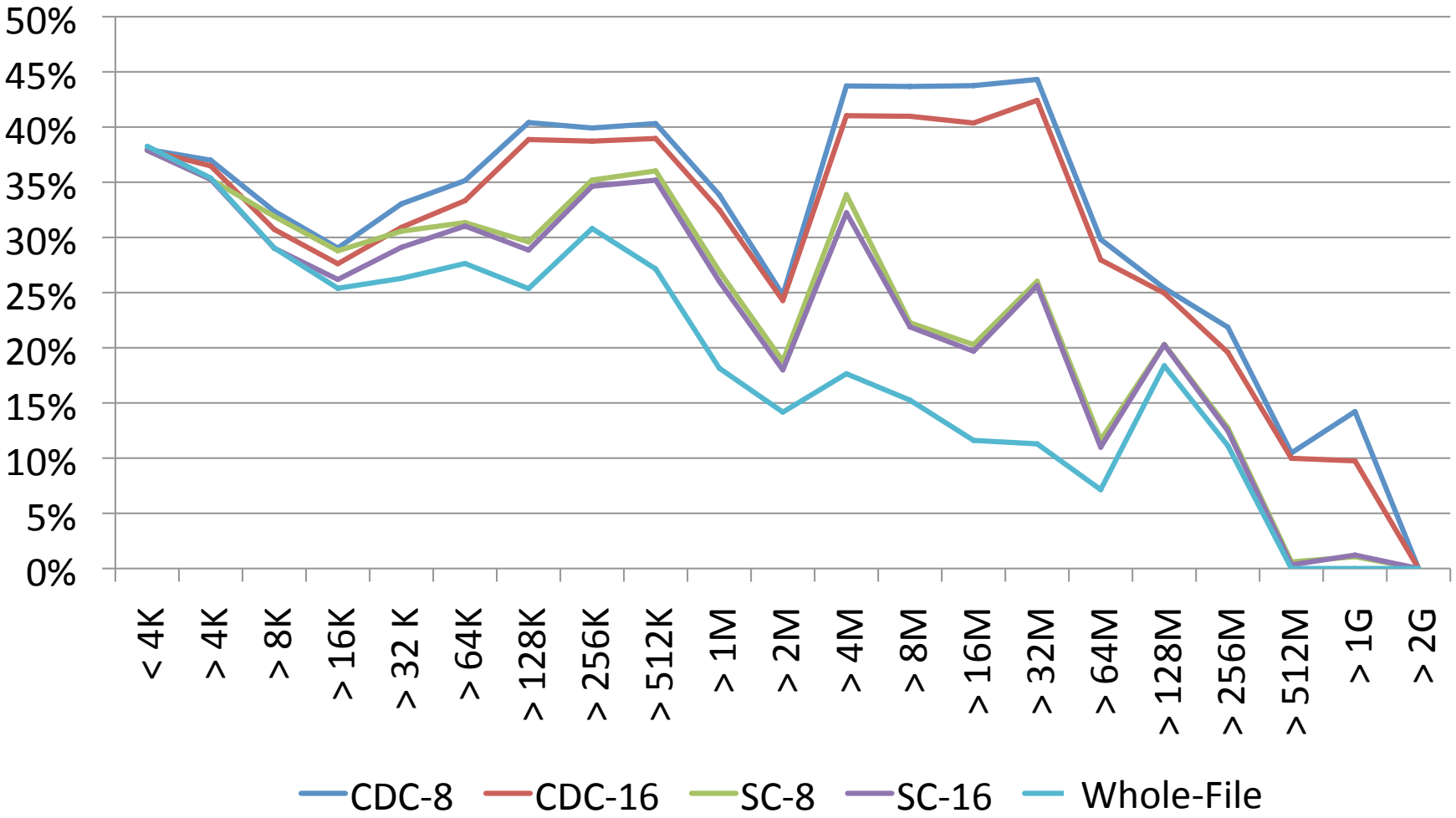


CDC = Content-defined Chunking
SC = Static Chunking
Whole-File = Whole-File Chunking

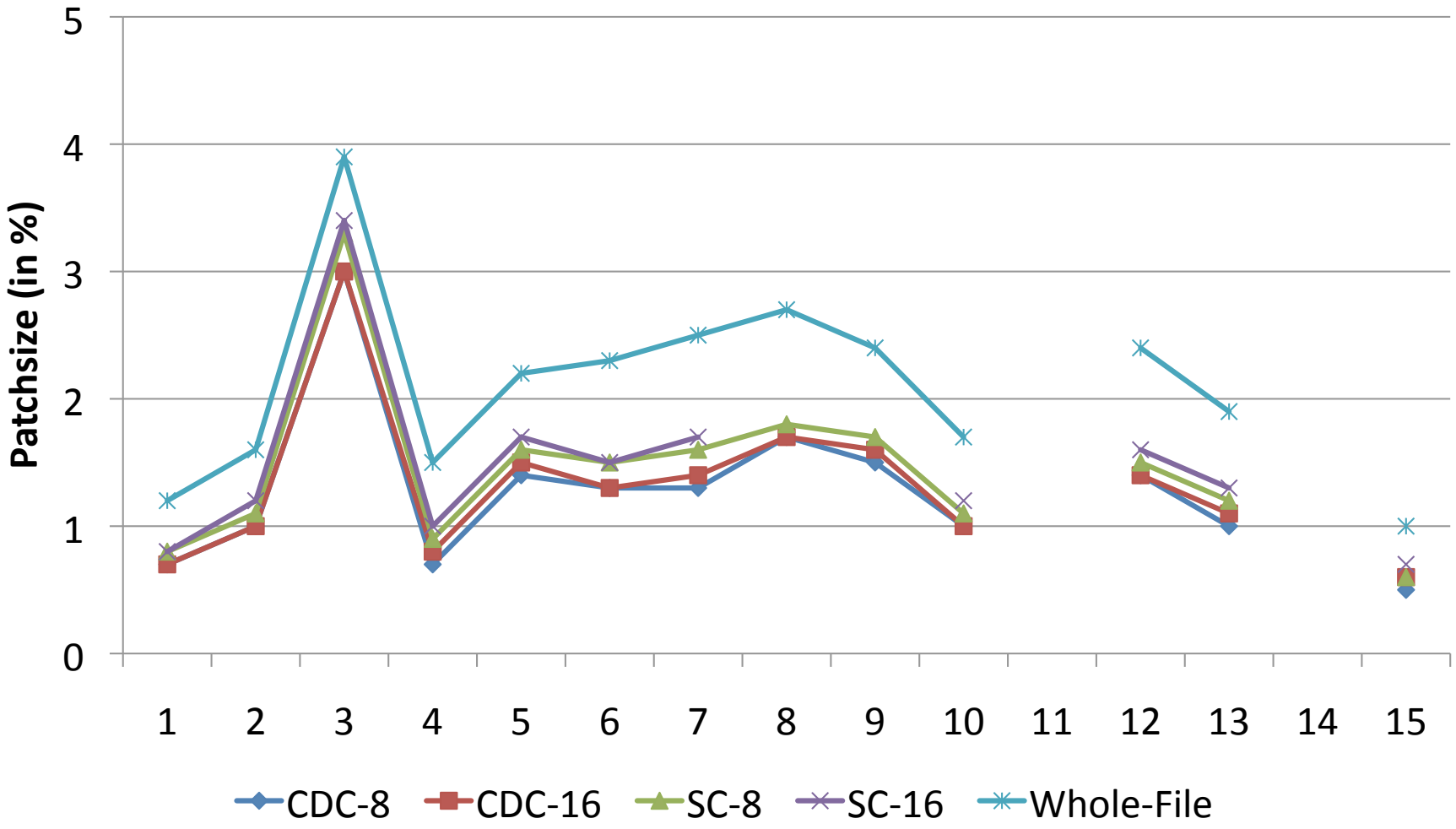
Int. Redundancy (file types)



Int. Redundancy (file size)



Temporal Redundancy

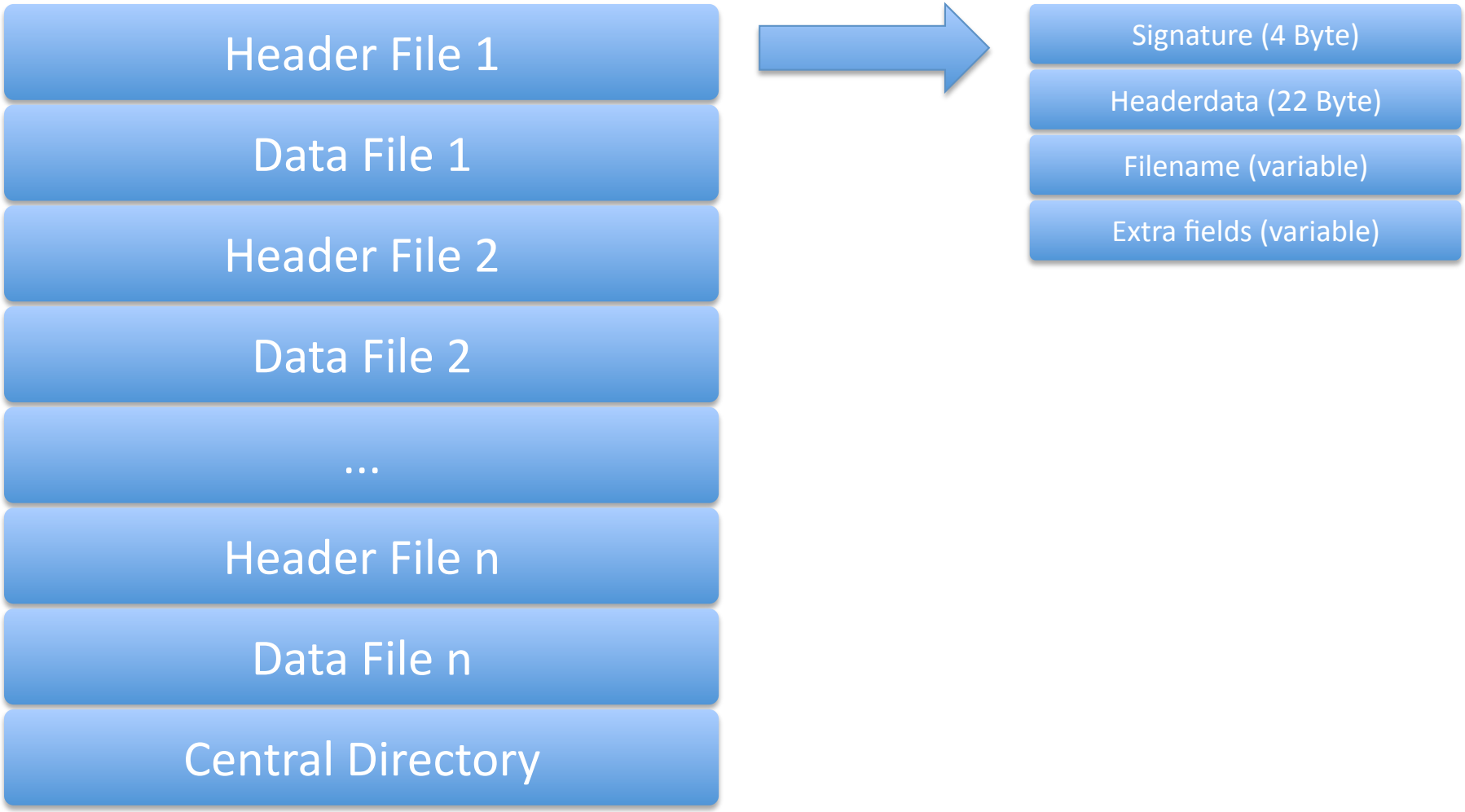


2. PART: SMALL-SCALE COMPARISION

Small-Scale Experiments

- Common Assumption:
  Small content modification
  ➔ *Small change in file*
- Experiments with various file types
  - .zip Archive
  - .ppt Presentation
  - More in full paper

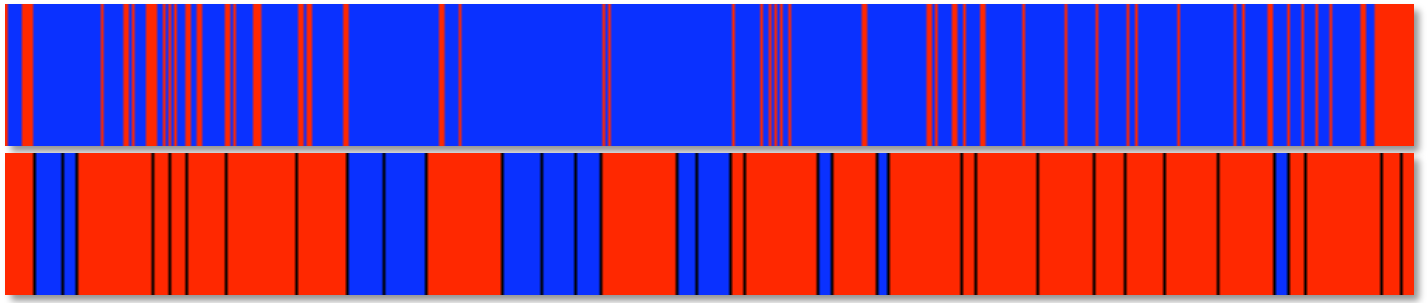
Structure .zip file



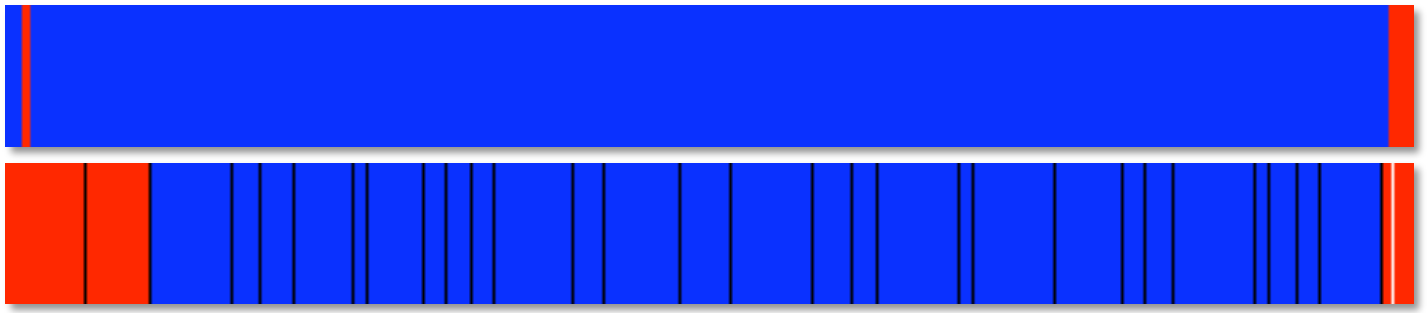
.ZIP File Experiment

- Experiment :
  - Archive of CIFS source code (Linux kernel)
  - Modification of AUTHORS file

Finder:



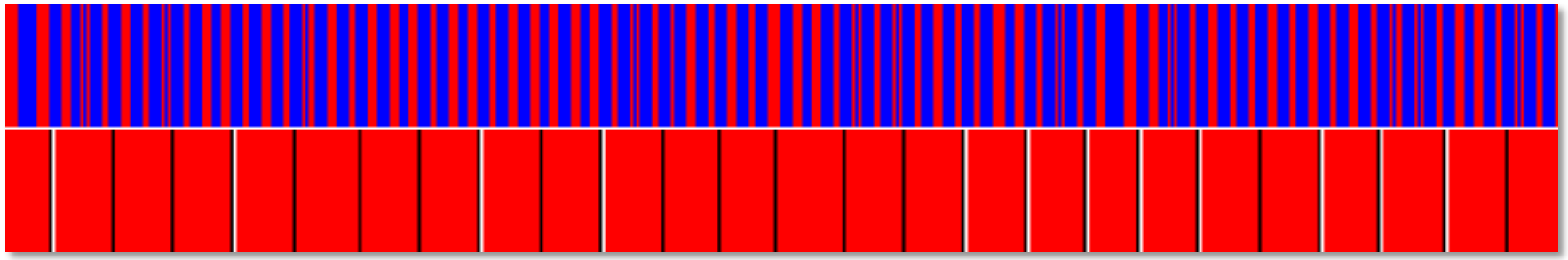
WinZIP:



.PPT File Experiment

- Experiment
  - Presentation without images
  - Name of author modified
- Powerpoint 2003/2007/2008: < 3% redundant

Powerpoint 2008 (Excerpt):



Conclusion

- Large-Scale:
  - ≈ 35% internal redundancy (CDC-8)
  - Advantage for Content-defined Chunking
  - > 98,5% temporal redundancy (all)
  - High variation between weeks
- Small-Scale:
  - Assumption often not correct
  - Application- and version-specific behavior

QUESTIONS?

SOFTWARE AVAILABLE UNDER CODE.GOOGLE.COM/P/FS-C/

Shiftings

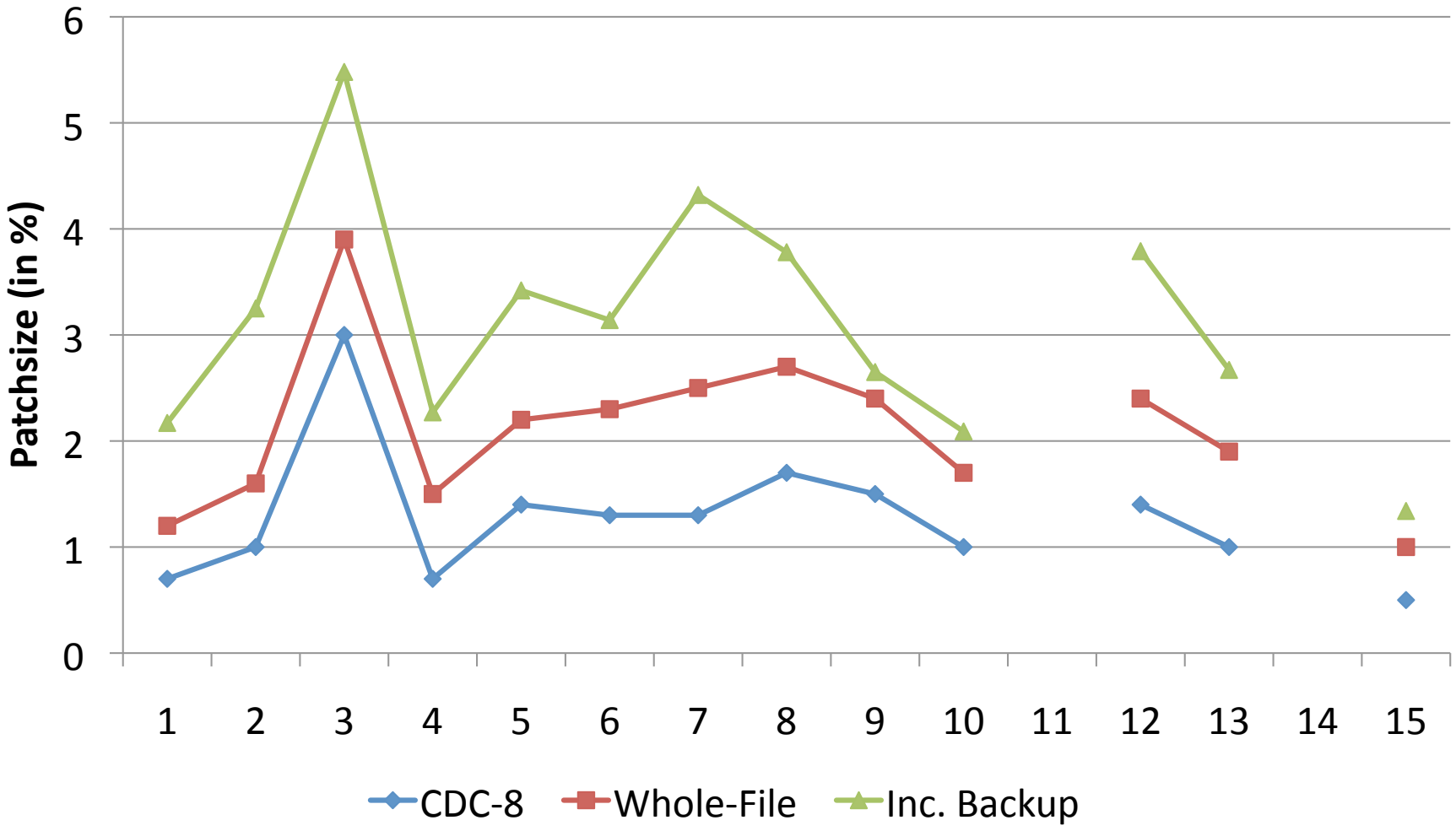
- Shiftings only have local influence

```
14159 26535  89793 23846 26433 83279 50288 41971 37510
14159 A265 3589793 23846 26433 83279 50288 41971 37510

14159 26535 89793 23846 26433 83279 50288 41971 37510
14159 A2653589793 23846 26433 83279 50288 41971 37510

14159 26535  89793 23846 26433 83279 50288 41971 37510
14159 A26535 89793 23846 26433 83279 50288 41971 37510
```

Duplicate Detection

- Check for every chunk: Duplicate
  - (Often) no exact comparision of contents
- Compare-by-Hash:
  - Fingerprint for each chunk (typically: SHA-1)
  - Check only if fingerprint is known
- Approach debated:
  - Data loss in case of hash collision

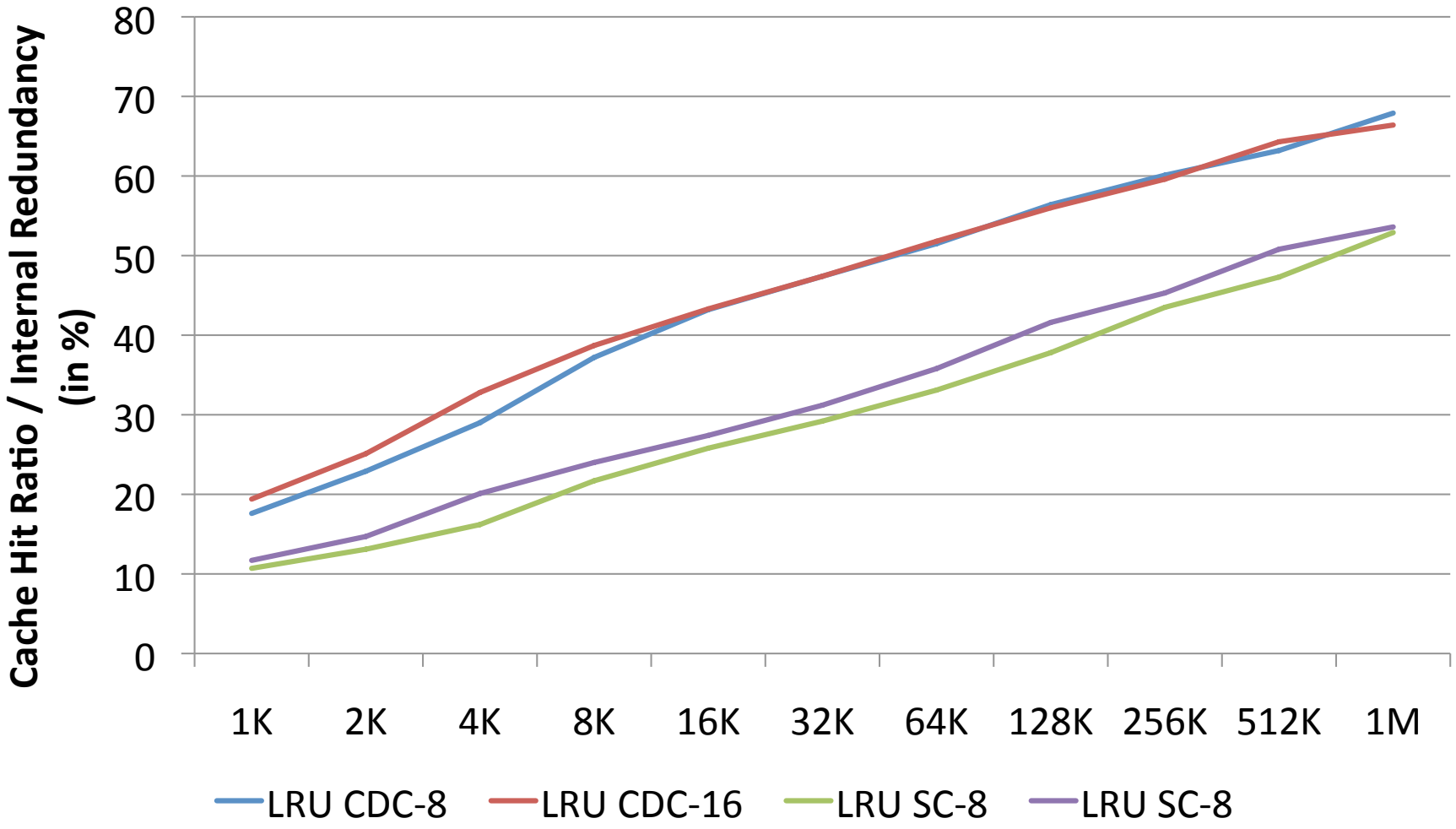
Comparision with Incr. Backup



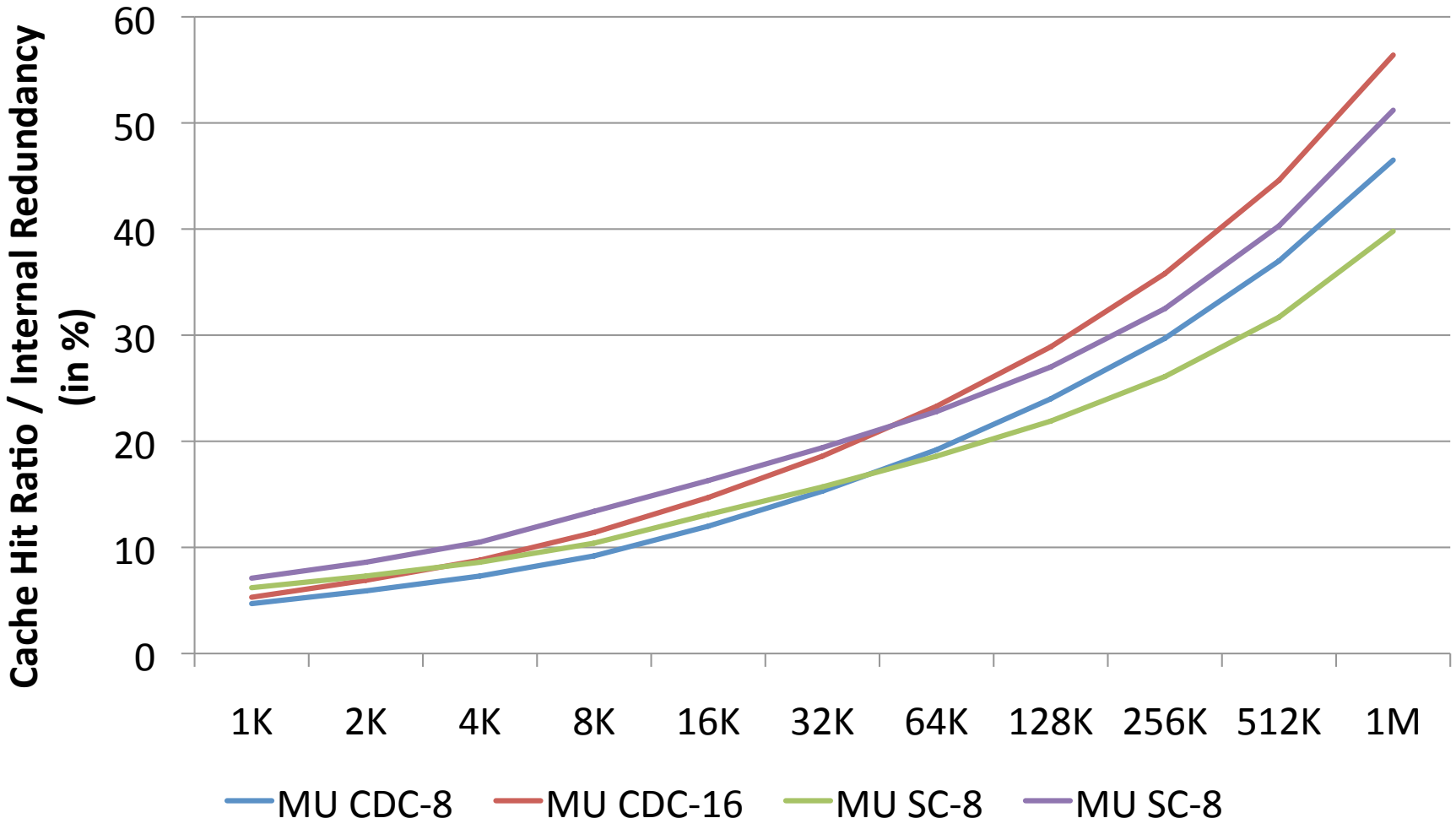
Caching Strategies

- Only 10% improvement in archival system Venti
- Backup scenario:
  - most chunks are redundant
  - reduced write accesses
- Caching also reduces read accesses
  - might have greater impact
- Comparison of (simple) caching strategies

LRU Cache


Cache Hit Ratio / Internal Redundancy (in %)
80
70
60
50
40
30
20
10
0
1K
2K
4K
8K
16K
32K
64K
128K
256K
512K
1M
LRU CDC-8
LRU CDC-16
LRU SC-8
LRU SC-8

MU Cache


Cache Hit Ratio / Internal Redundancy (in %)
60
50
40
30
20
10
0
1K
2K
4K
8K
16K
32K
64K
128K
256K
512K
1M
MU CDC-8
MU CDC-16
MU SC-8
MU SC-8

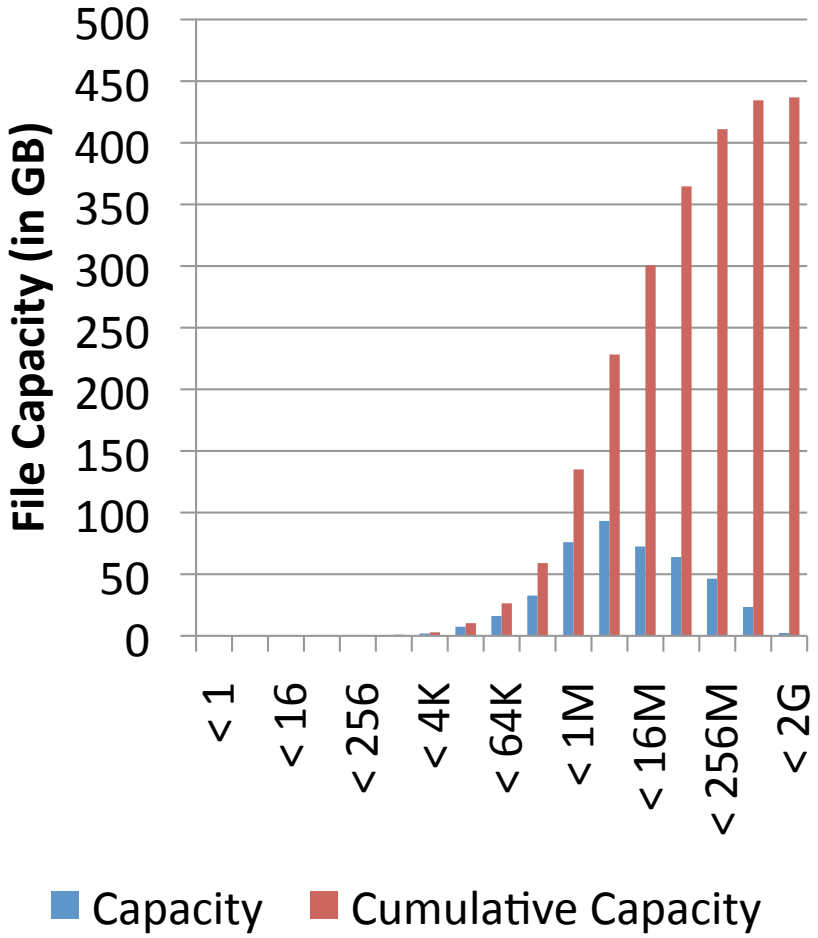
Related Work

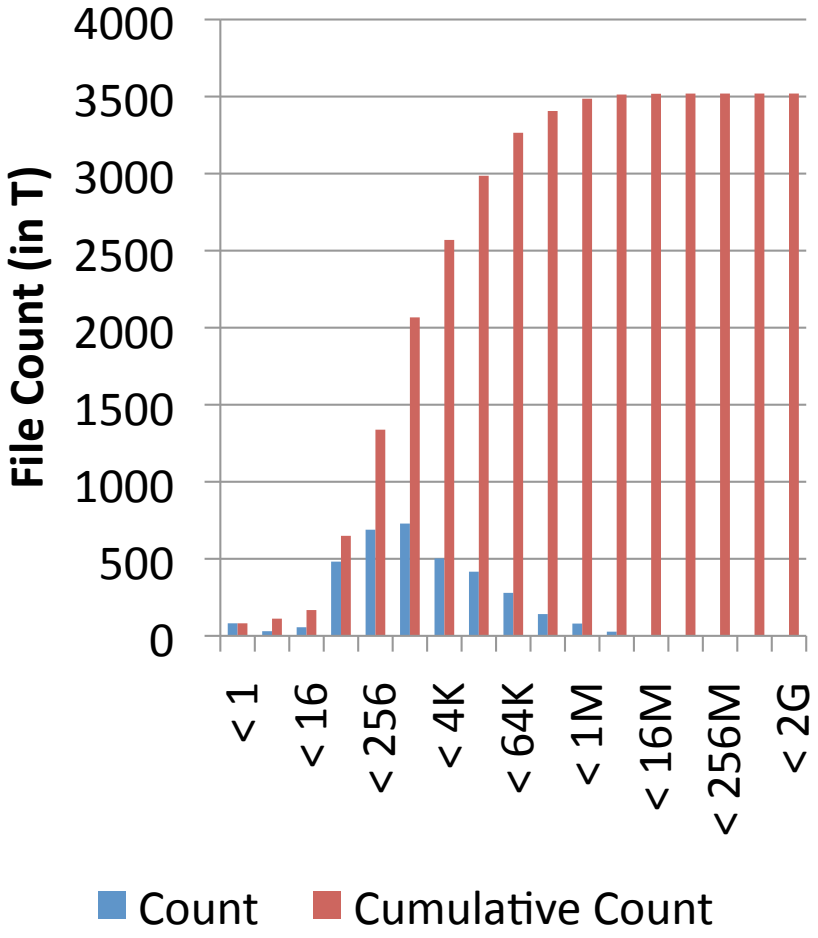
- Policoniades and Pratt (2004):
  - Mirror of sunsite.org.uk: 35 GB
  - Personal user files: 2.9 GB
  - Scratch directories: 100 GB
  - No temporal redundancies evaluated
- You and Karamanolis (2004):
  - 15.5 GB on various data sets not including personal user files

Related Work

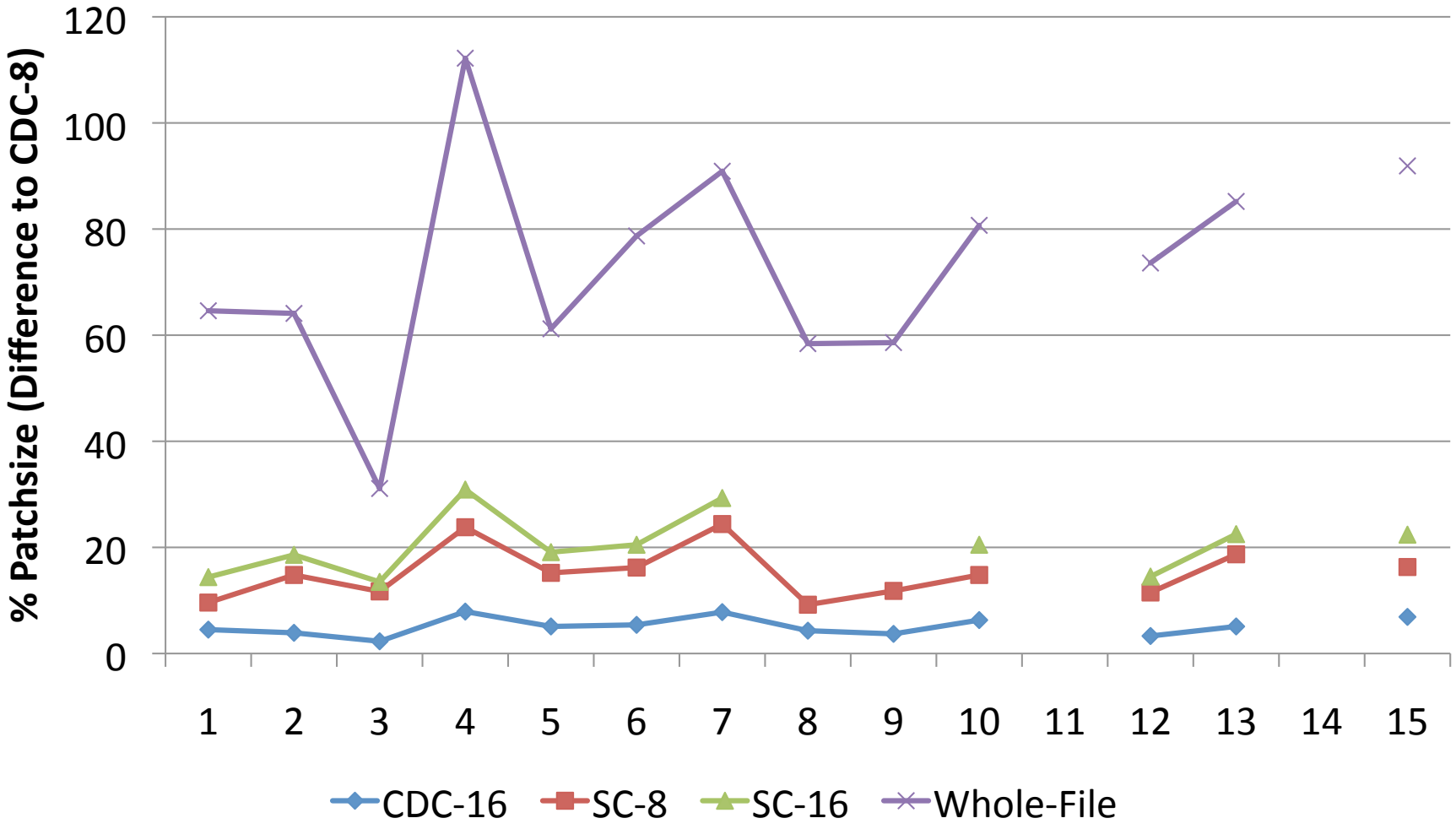
- Mandagere et al. (2009):
  - 450 GB personal user files
  - 16 users
  - No temporal redundancies
  - Includes throughput measurements
- Most publications about deduplication:
  - Include evaluations using various, often smaller data sets
  - Only some include personal user files and temporal redundancies

File size distribution





Difference to CDC-8



File Types (Excerpt)

Rank	File type	File Count	Storage
1	.jpg	3% (114 K)	13% (59.9 GB)
2	N/A	36% (1261 K)	11% (48.4 GB)
3	.doc	2% (85 K)	8% (36.8 GB)
4	.ppt	0% (9 K)	8% (33.2 GB)
5	.xls	1% (24 K)	8% (33.0 GB)
6	.pdf	1% (40 K)	7% (32.1 GB)
7	.avi	0% (0.6K)	3% (14.0 GB)
8	.mov	0% (0.2K)	3% (13.5 GB)
9	.dat	3% (97K)	2% (10.4 GB)
...			
13	.zip	0% (8,6 K)	2% (7.3 GB)
15	.gz	0% (2,8 K)	1% (5.4 GB)
	TOTAL	**3520 K**	**436,8 GB**

.PPTX

- New Powerpoint format
  - XML-Files (Content / Metadata)
  - Binary Files (Images / Videos)
  - ZIP-Container
- ➔Some files changed (minimally)
- ➔Compressed data changed

Powerpoint 2008: