Is Low Similarity Threshold A Bad Idea in Delta Compression?

Hongming Huang[1], **Chun Jason Xue[2]**, Nan Guan[1], Hong Xu[3]

[1]CITY UNIVERSITY OF HONG KONG

[2]MOHAMED BIN ZAYED UNIVERSITY OF ARTIFICIAL INTELLIGENCE

[3]THE CHINESE UNIVERSITY OF HONG KONG

香港城市大學
City University of Hong Kong

MOHAMED BIN ZAYED
UNIVERSITY OF
ARTIFICIAL INTELLIGENCE

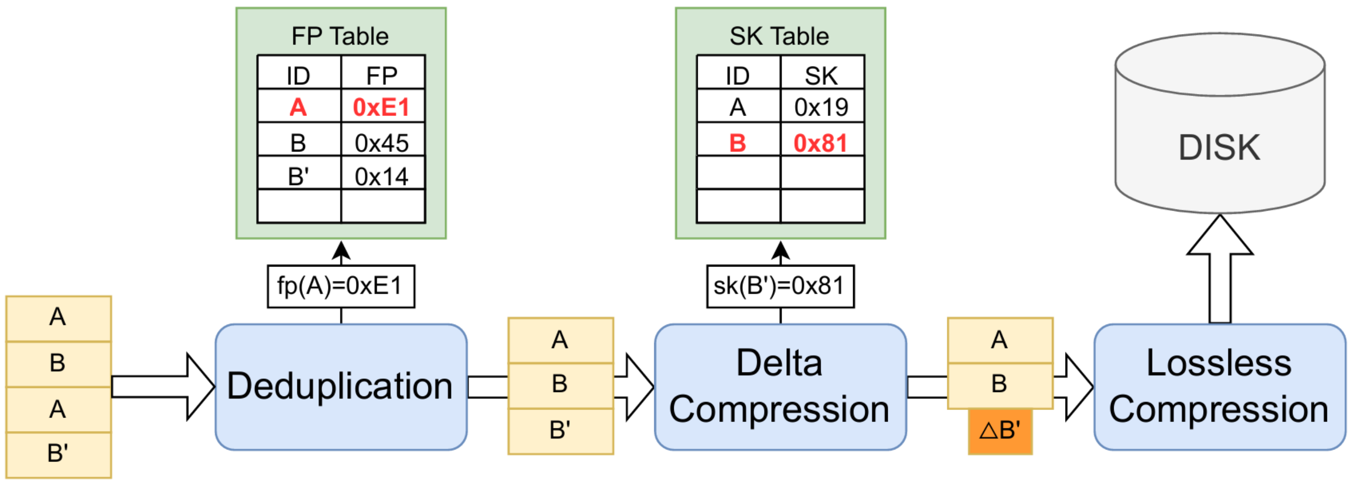
香港中文大學
The Chinese University of Hong Kong

Outline

- **Background and Motivation**
- Identify false similar blocks for delta compression
- How can low threshold help boundary shift problem
- Evaluation & Conclusion

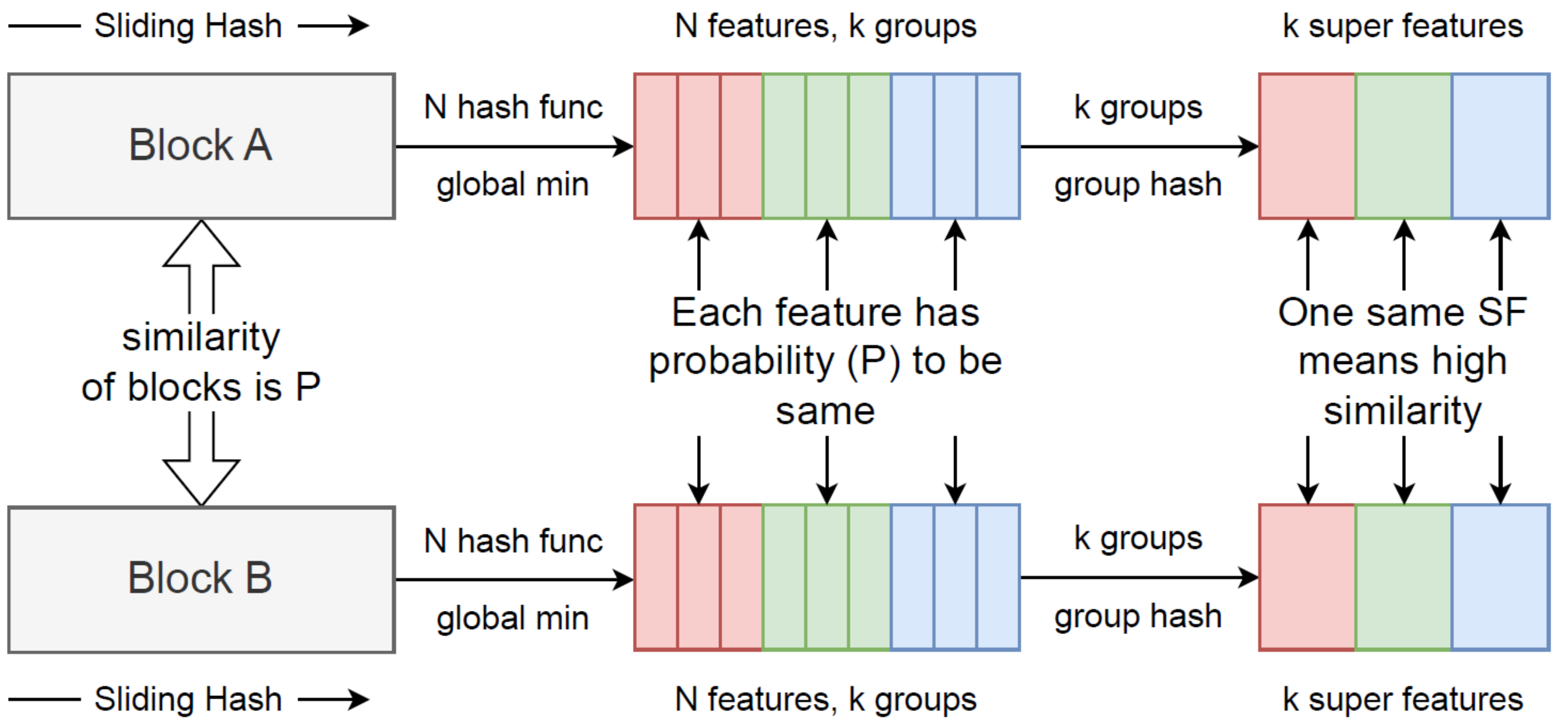
Intro: Modern Deduplication Techniques

- Deduplication is critical to modern storage systems.
- Modern deduplication system includes three techniques:
  - **Deduplication**: Calculates a fingerprint**(FP)** for each data block and uses the fingerprint to find and eliminate **identical** blocks
  - **Delta-compression**: uses sketch**(SK)** to find a **similar** base block for each new block, only storing the **difference** between these two blocks
  - **Lossless compression**: Using common lossless compression algorithm like ZSTD to finally compress all data before writing into the disk.



Super-Feature-Based Solution of Delta Compression

- Apply N(9) sliding hash function to the block, generate N min hash value as N **features**.
- Group features into k(3) groups, calculate the hash value for each group as **super features.**
- One matching SF means all features in a group is the same → most features should be the same → **high similarity**



Related Works

- **Finesse**: (FAST 2019)
  Use one hash function to generate N features in N sub-blocks. Greatly improves efficiency but losses over **15%** compression ratio.

- **Odess**: (ICDE 2021)
  First generate a small sample from the block, then run N-Transform. Greatly improves efficiency with a minor loss of compression ratio.

- **DeepSketch**: (FAST 2022)
  Use an NN model to generate features and uses Approximate Nearest Neighbor to find base blocks. 15% compression ratio gain over Finesse but efficiency is **low.**

- **Palantir:** (ASPLOS 2024)
  Generate a set of hierarchical super-features to detect more similar blocks, introduces heavy **computational overhead**

Rethinking on Existing Works

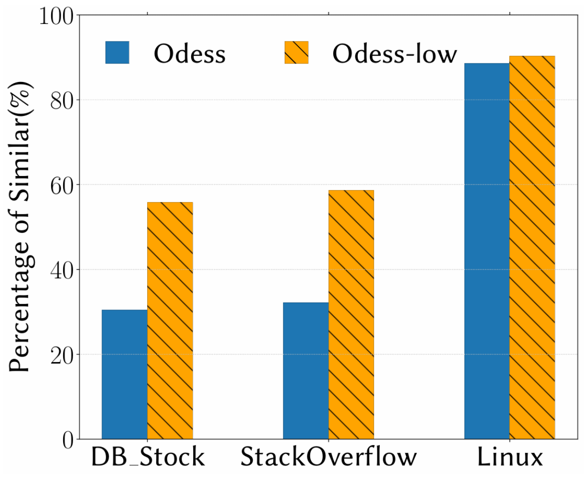
- Super-Feature-based approach relies on grouping parameters to determine a **single detection threshold.**

- Dilemma: **Accuracy vs. Coverage**
  - High threshold will miss potential similar blocks, low threshold will find false similar blocks;
  - Traditional wisdom choose the **high accuracy** to avoid false similar blocks (N-Transform, Odess);
  - Palantir overcome this dilemma at the cost of huge computational overhead

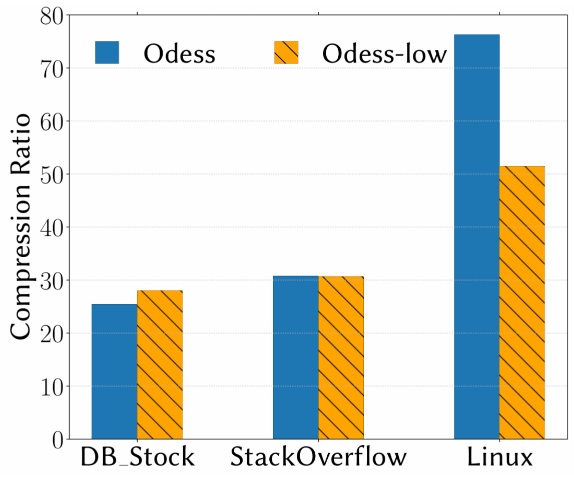
Is Low Threshold Truly Bad?

- We evaluated Odess algorithm with high and low thresholds, together with the deduplication and ZSTD lossless compression
- Low threshold can find much **more similar blocks**, but those false similar block will not bring compression ratio gain and even contribute negatively on some datasets.
- This paper's idea: If we filter out those **false similar blocks**, we can benefit from those extra found true similar blocks

These additional similar blocks may
Support other optimizations



(a) Percentage of similar



(b) Compression ratio

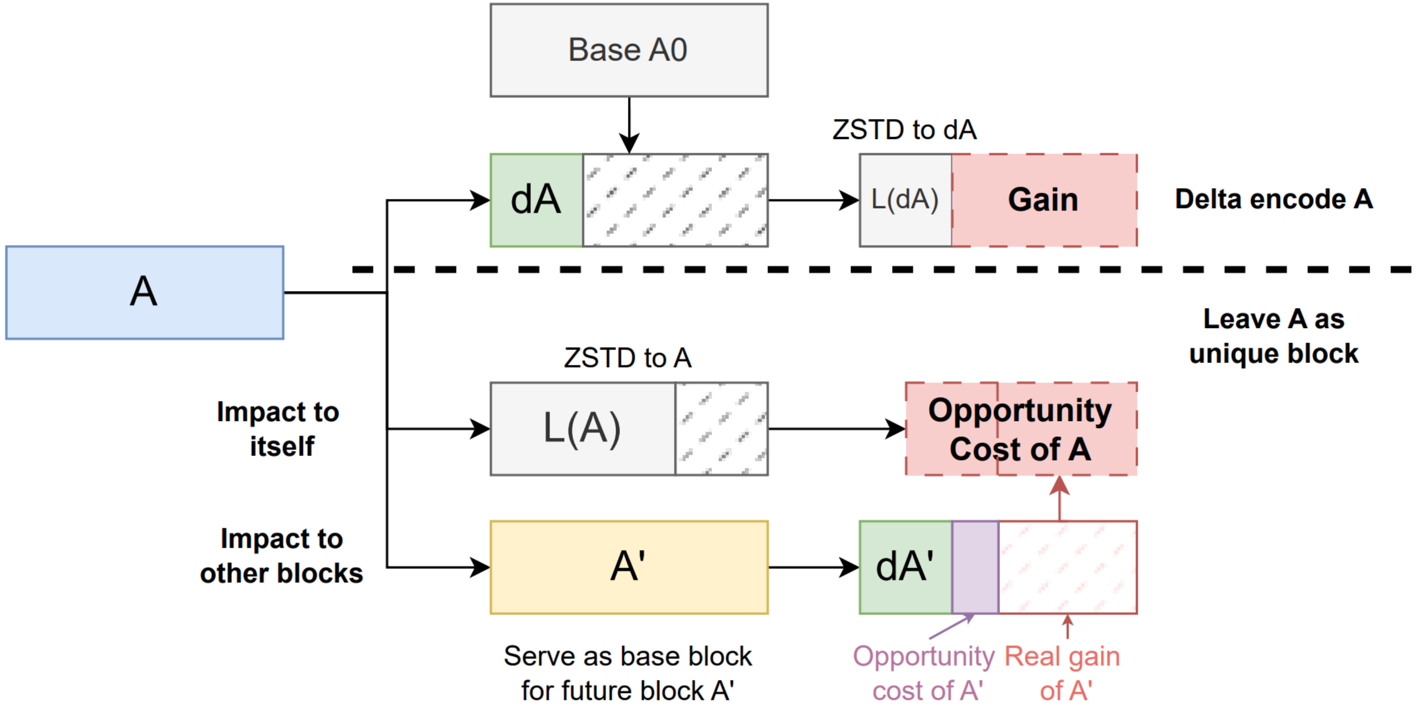
Outline

- Background and Motivation
- **Identify false similar blocks for delta compression**
- How can low threshold help boundary shift problem
- Evaluation & Conclusion

Identification of False Similar Blocks

- Challenge: “Similar enough” is hard to define.
- This paper’s idea:

  Quantify the **gain** and the **opportunity cost** of delta compression
- Gain of delta compression:
  - Conduct delta compression, calculate compressed data size
- Opportunity cost composed of two parts:
  - **Compression of current block**
  - **Impact to the other blocks**



Quantify Opportunity Cost

- Compression of itself:
  - The non-encoded block can be losslessly compressed.
  - Using the average lossless compression ratio C to estimate the lossless compression ratio of the current block:

$$Cost_{self} = block\_size - block\_size/C$$

- Impact on other blocks:
  - The non-encoded block can act as a **base block** and serve delta blocks in the future

$$Cost_{be\_base} = E\big(NUM(new\_delta)\big) * E(real\_gain)$$

  - The ***real gain*** of a delta block should discount its opportunity cost

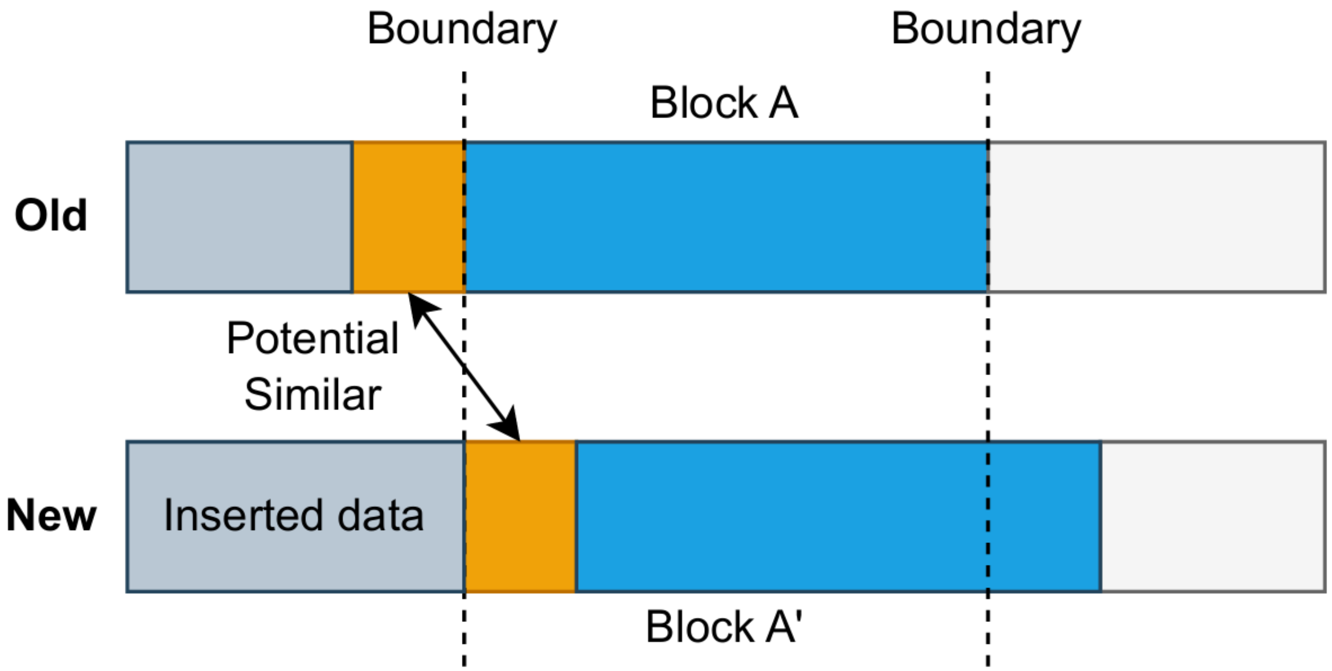
$$Cost_{be\_base} = \frac{\sum_{delta\ blocks}(delta\_gain - Opportunity\_Cost)}{NUM(unique\ blocks)}$$

Outline

- Background and Motivation
- Identify false similar blocks for delta compression
- **How can low threshold help boundary shift problem**
- Evaluation & Conclusion

Boundary Shift Problem

- When a little modification happens in a duplicated area, it will affect the chunking point of **all following blocks.**

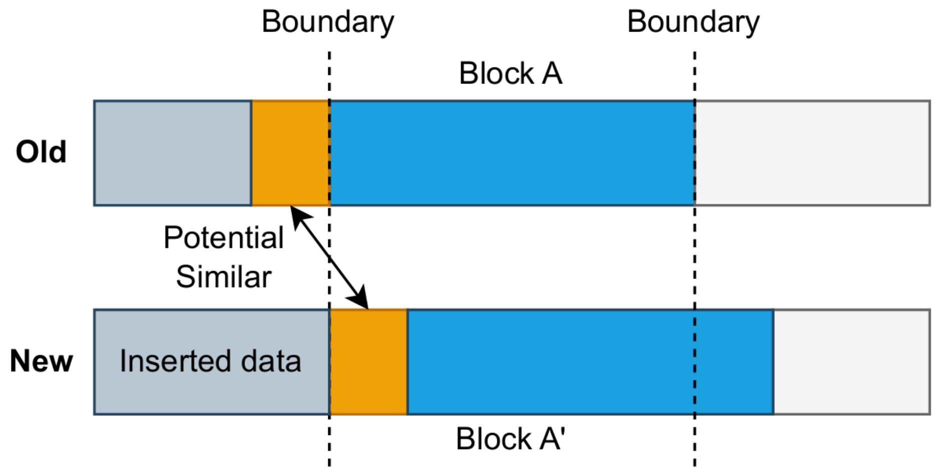


Insight to Boundary Shifting Problem

- Content-defined-chunking (CDC) CAN NOT perfectly solve the problem.
  - According to our experiment using fastCDC, **55%** of new unique blocks are actually shifted blocks, and may lead to up to **13.75%** compression ratio loss.
- Our insight:
  - The duplicate data is adjacent to the original block A
  - If we **extend** the base block A using the adjacent data, the shifted data(orange) can be eliminated by delta block
- High detection threshold solution does not help:
  - If the block is shifted by **20%**, original N-Transform or Odess algorithm only have **48%** chance to detect the original block
  - Without detecting the original block, we can do nothing to boundary shifting.

Base Block Extension with Low Threshold

- Using a lower detection threshold to find original blocks
- Extend the original block with **the previous** and **the next block** as a new base block
  - The shifted data (orange part) will be included in the extended base block
- Efficiency optimization:
  - Caching the extended base block and new identical blocks with its adjacent blocks in memory and SSD
  - No need to frequently decompress and load the adjacent block



Outline

- Background and Motivation
- Identify false similar blocks for delta compression
- How can low threshold help boundary shift problem
- **Evaluation & Conclusion**

Evaluation Setup

- Testbed: Xeon 8260 CPU, 512 GB RAM and full SSD disks.
- Datasets: five synthetic backup datasets and a Linux dataset (same as Palantir's evaluation)
  - We generate 20 backups by adding 1%, modifying 3.5% and deleting 0.5% of blocks in each backup version.
- Baseline: Odess solution with default high threshold
- Metrics:
  - E2E compression ratio: including deduplication and ZSTD
  - Delta Compression Coverage: Proportion of similar blocks

Result Overview

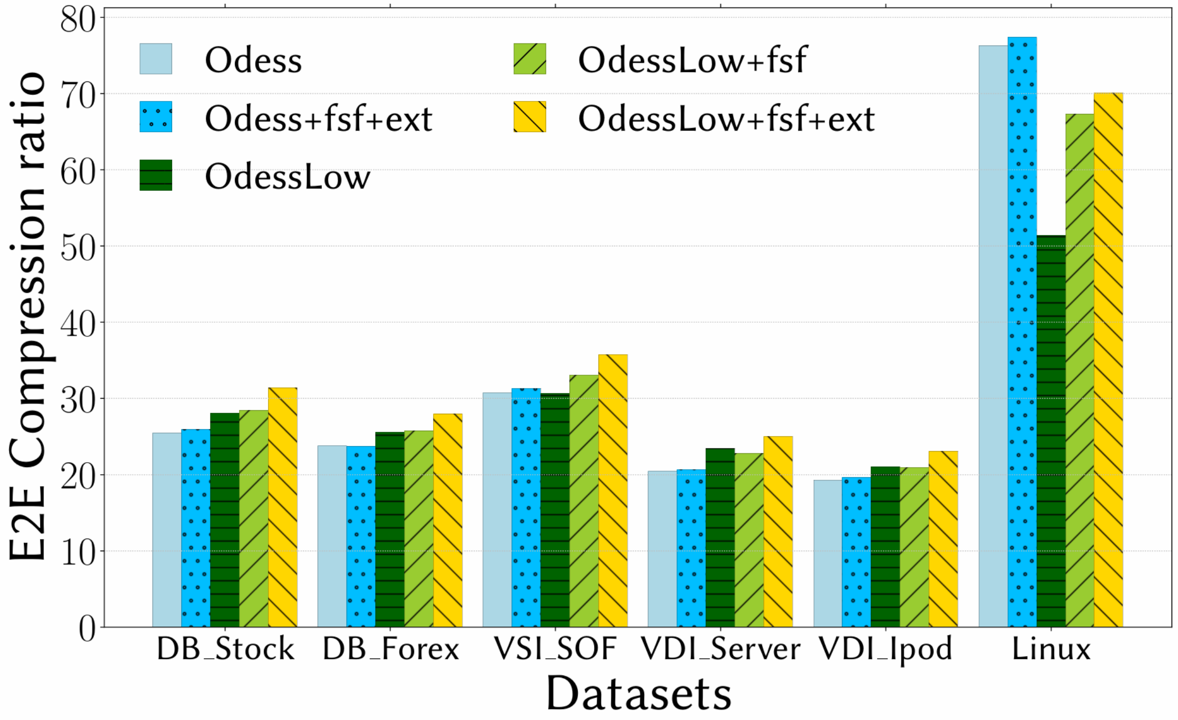
- Using lower threshold with two proposed optimizations achieves **15% higher** average compression ratio, with **3.7%** throughput loss.
- High threshold solution cannot benefit from the proposed optimizations.
- Both the false similar filter and the base block extension contributes to the E2E ratio

Algorithm	E2E ratio	DCC	Thrpt (MB/S)
Odess	1.000	0.446	181.558
Odess+fsf+ext	1.019	0.497	177.482
OdessLow	1.018	0.669	181.795
OdessLow+fsf	1.058	0.562	179.952
OdessLow+fsf+ext	**1.151**	**0.578**	**175.053**

Table 2. Average result over all datasets. The E2E ratio is normalized to Odess. **Fsf** means false similar filter, and **ext** means base block extension.

Result on All Datasets

- Directly using lower threshold may cause unstable result.
- Our two optimizations improve the system's robustness on different datasets.



Comparing Different False Similar Filter

- Palantir (ASPLOS'24) also proposed a naïve false similar filter algorithm.
- We compare two different algorithms to identify false similar blocks. **FSF** is this paper's filter, and **PFSF** is the algorithm proposed by Palantir.
- Our algorithm with comprehensive modeling outperforms Palantir's algorithm by **7.8%**, especially on datasets with more false similar blocks.

Dataset	OdessLow+PFSF	OdessLow+FSF	Gain(%)
DB_Stock	28.021	28.518	1.77%
DB_Forex	25.522	25.813	1.14%
VSI_SOF	32.056	33.631	**4.91%**
VDI_Server	23.747	23.530	-0.91%
VDI_Ipod	21.247	21.108	-0.23%
Linux	62.484	67.365	**7.81%**

Table 3. Compression ratio using different false similar filter algorithms with OdessLow. PFSF means the intuitive false similar filter of Palantir, FSF means this paper's false similar filter.

Conclusion

- Showed that low similarity threshold could be beneficial to Delta compression
- Proposed a false similar block filter and the base block extension algorithm to improve the delta compression with low detection threshold
- Presented a preliminary evaluation to prove the effectiveness and efficiency of low threshold solution and these two optimizations.

Thank You!

Boundary Shift Problem

- Content-defined-chunking (CDC) can not perfectly solve the problem.
  - We randomly adding 4.5% new data blocks to an original dataset with N blocks, which means 0.045*N of new unique blocks.
  - The fastCDC algorithm finds 0.1*N of new unique blocks.
  - (0.1-0.045)/0.1=55% of new unique blocks are actually “shifted blocks”
  - The average lossless compression ratio of unique block is 2.0
  - Assuming the average shifting size is 50% of the block size, the boundary-shift cause 55%*50%/2.0=13.75% of compression ratio loss