# Finesse: Fine-Grained Feature Locality based Fast Resemblance Detection for Post-Deduplication Delta Compression


**Yucheng Zhang** *Hubei University of Technology*

**Wen Xia** *Harbin Institute of Technology, Shenzhen & Peng Cheng Laboratory*

**Dan Feng** *Huazhong University of Science and Technology*

**Hong Jiang** *University of Texas at Arlington*

**Yu Hua** *Huazhong University of Science and Technology*

**Qiang Wang** *Huazhong University of Science and Technology*

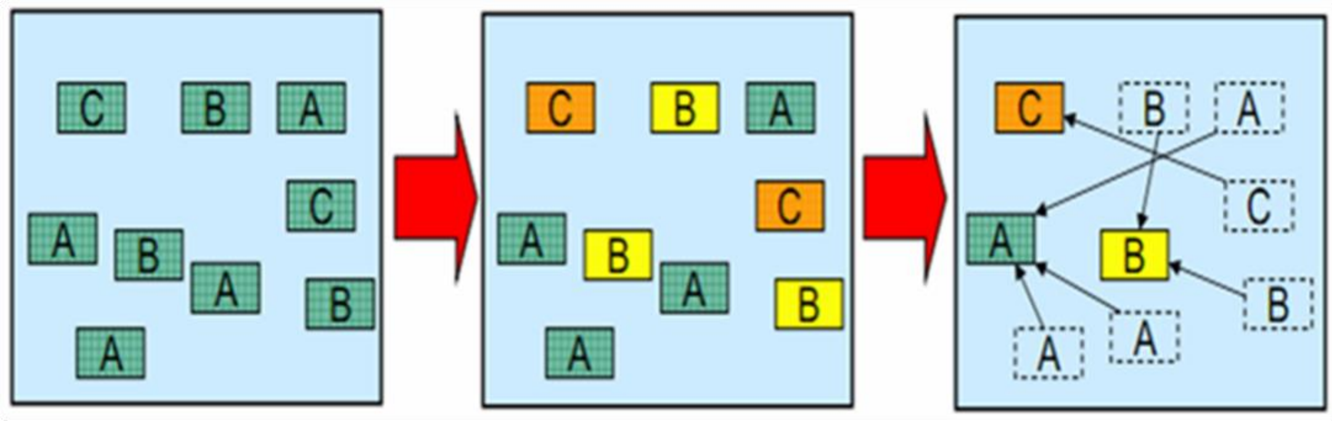
# Background


- Big data era
  - ➢ Amount of digital data in the world will reach 44 ZB by 2020

- Redundant data in backup systems
  - ➢ About 88-90% of the data in EMC and Symantec's backup systems are duplicate (FAST'12, USENIX ATC'15)

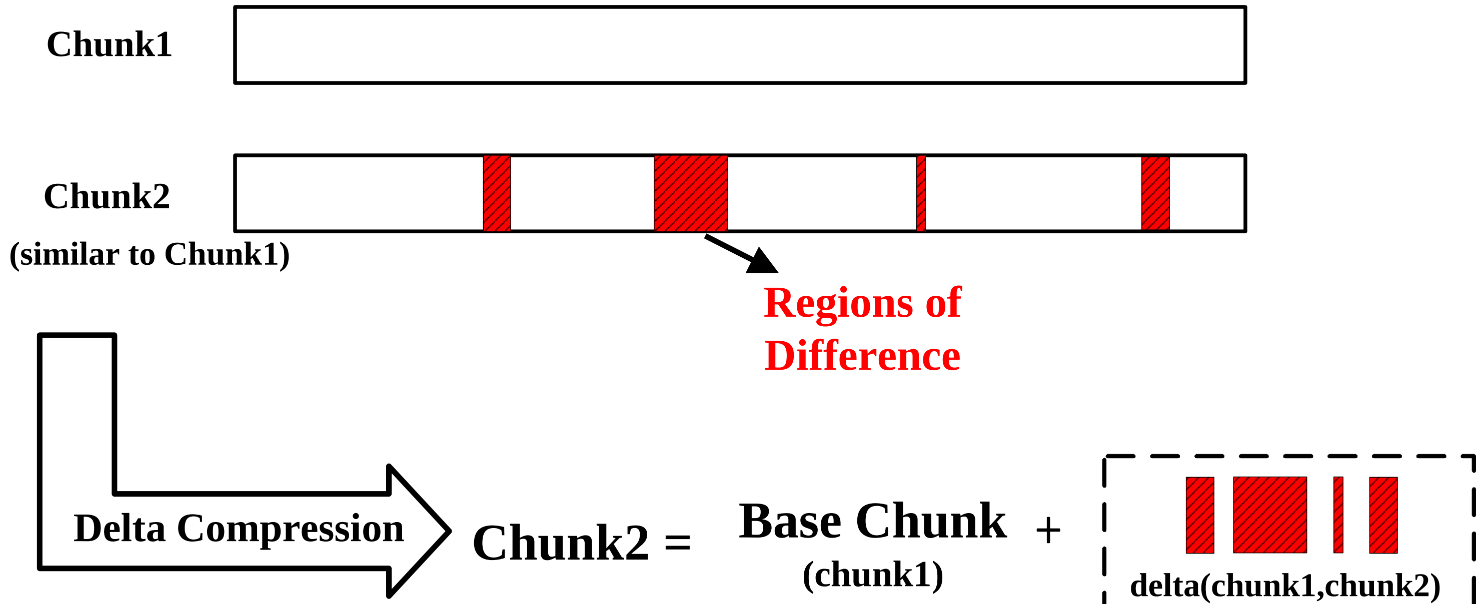
# Data Reduction Technologies

- Data deduplication
  - Remove duplicate chunks according to their fingerprints, only store one unique chunks
  - Drawback: cannot remove redundant data among non-duplicate but very similar chunks



# Data Reduction Technologies

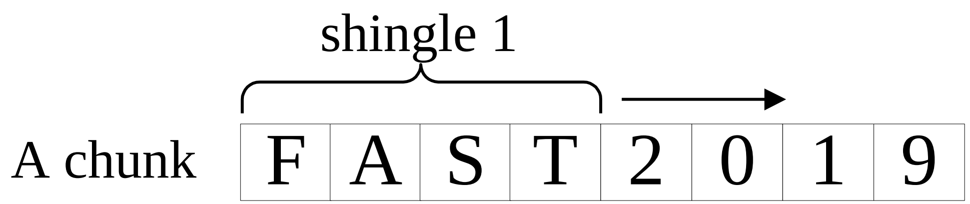
- Delta compression
  - Achieve 2X more compression ratio beyond deduplication (FAST'12, Performance'14, Sigmod'17)



# Resemblance Detection

- Detecting delta compression candidates
- Traditional **N-transform Super-Feature**
  - Generally, It extracts a fixed number of features from a chunk and grouping *N* features (*N*=12) into *M* SFs (e.g., *M*=3) for matching. One SF matching means the two chunks are very similar
  - Feature extraction is time-consuming: Requiring N linear transformations for each fingerprint to generate N-dimensional hash value sets (features)

# A simple example: extracting 4 features from a string




shingle 1 (S1): FAST
shingle 2 (S2): AST2
shingle 3 (S3): ST20
shingle 4 (S4): T201
shingle 5 (S5): 2019

**Rabin fingerprinting:**

S1 —> R1
S2 —> R2
S3 —> R3
S4 —> R4
S5 —> R5

# A simple example: extracting 4 features from a string

shingle 1

| A chunk | F | A | S | T | 2 | 0 | 1 | 9 |
|---|---|---|---|---|---|---|---|---|

shingle 1 (S1): FAST
shingle 2 (S2): AST2
shingle 3 (S3): ST20
shingle 4 (S4): T201
shingle 5 (S5): 2019

**Rabin fingerprinting:**

S1 —> R1
S2 —> R2
S3 —> R3
S4 —> R4
S5 —> R5

**4 Linear transformations:**

R1 —> R11, R12, R13, R14
R2 —> R21, R22, R23, R24
R3 —> R31, R32, R33, R34
R4 —> R41, R42, R43, R44
R5 —> R51, R52, R53, R54

## Feature extraction

Feature 1:max{R11,R21,R31,R41}
Feature 2:max{R12,R22,R32,R42}
Feature 3:max{R13,R23,R33,R43}
Feature 4:max{R14,R24,R34,R44}

# A simple example: extracting 4 features from a string

shingle 1

A chunk

| F | A | S | T | 2 | 0 | 1 | 9 |
|---|---|---|---|---|---|---|---|

shingle 1 (S1): FAST
shingle 2 (S2): AST2
shingle 3 (S3): ST20
shingle 4 (S4): T201
shingle 5 (S5): 2019

**Rabin fingerprinting:**

S1 —> R1
S2 —> R2
S3 —> R3
S4 —> R4
S5 —> R5

*N-transform Super-Feature* is compute-intensive because it requires *too many linear transformations*

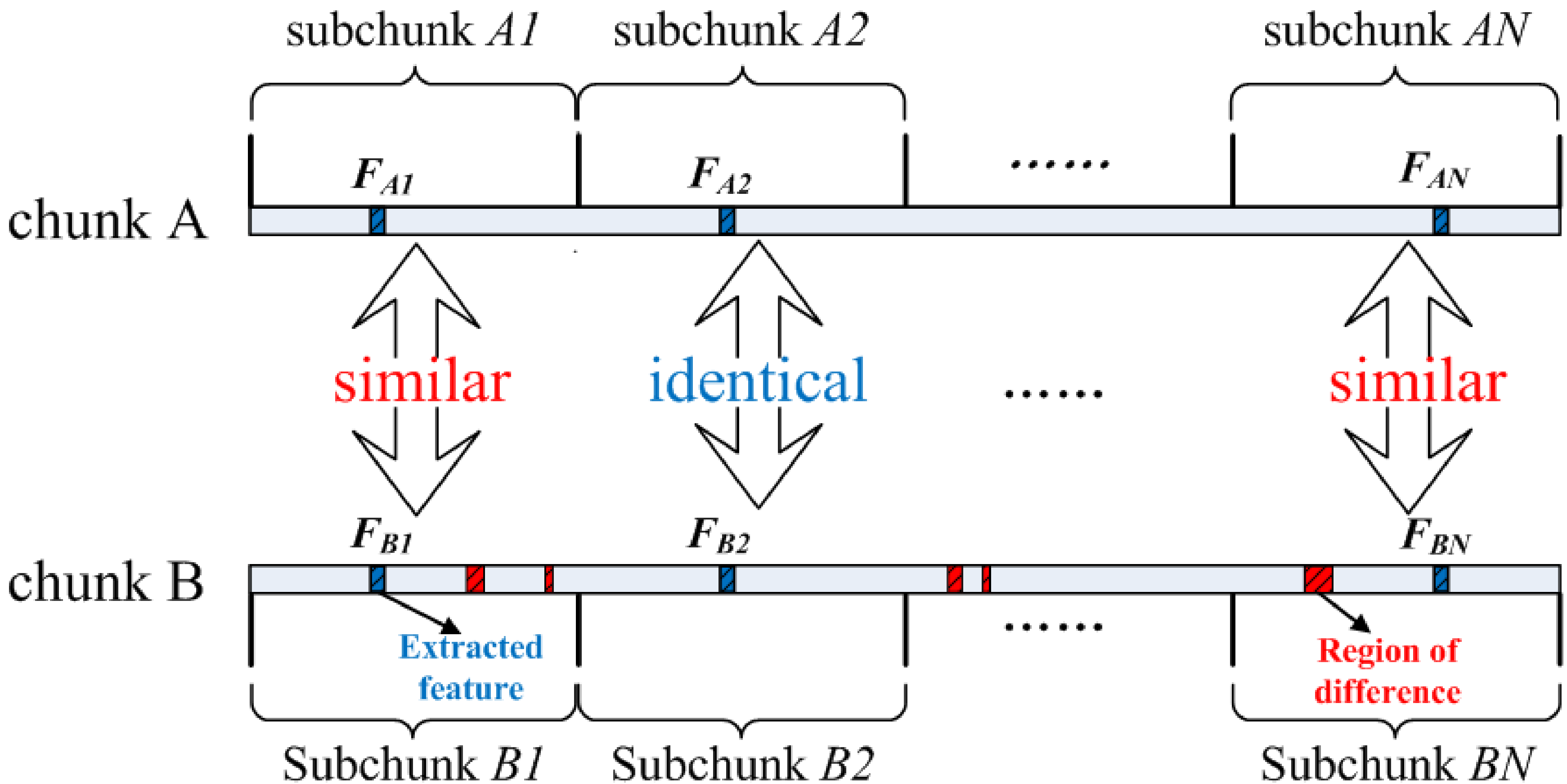
**4 Linear transformations:**

R1 —> R11, R12, R13, R14
R2 —> R21, R22, R23, R24
R3 —> R31, R32, R33, R34
R4 —> R41, R42, R43, R44
R5 —> R51, R52, R53, R54

## Feature extraction

Feature 1:max{R11,R21,R31,R41}
Feature 2:max{R12,R22,R32,R42}
Feature 3:max{R13,R23,R33,R43}
Feature 4:max{R14,R24,R34,R44}

# Observation: Fine-grained Feature Locality




$F_{A1} = F_{B1}$ $F_{A2} = F_{B2}$ … $F_{AN} = F_{BN}$

# Observation: Fine-grained Feature Locality

| Datasets | WEB | TAR | RDB | SYN | VMA | VMB |
|---|---|---|---|---|---|---|
| **Avg. # of identical subchunks** | 8.27 | 9.19 | 6.86 | 5.78 | 5.99 | 6.34 |
| **Avg. # of subchunks owning the same features** | 10.82 | 10.97 | 10.23 | 10.10 | 10.04 | 10.64 |

All identified chunks are all divided into 12 equal-sized subchunks

**Most of the corresponding subchunk pairs in the detected similar chunks have the same features**

# Design of Finesse

Finesse, a fast resemblance detection approach that exploits the fine-grained feature locality

Step1: Feature extraction

- Dividing a chunk into N equal-sized subchunks, computing Rabin fingerprints for all shingles in the chunk, and selecting the maximum fingerprints in each subchunk as features to obtain N features
- **Advantages:** Do not require the time-consuming linear transformations for extracting more features, only need to divide the chunk into more subchunks

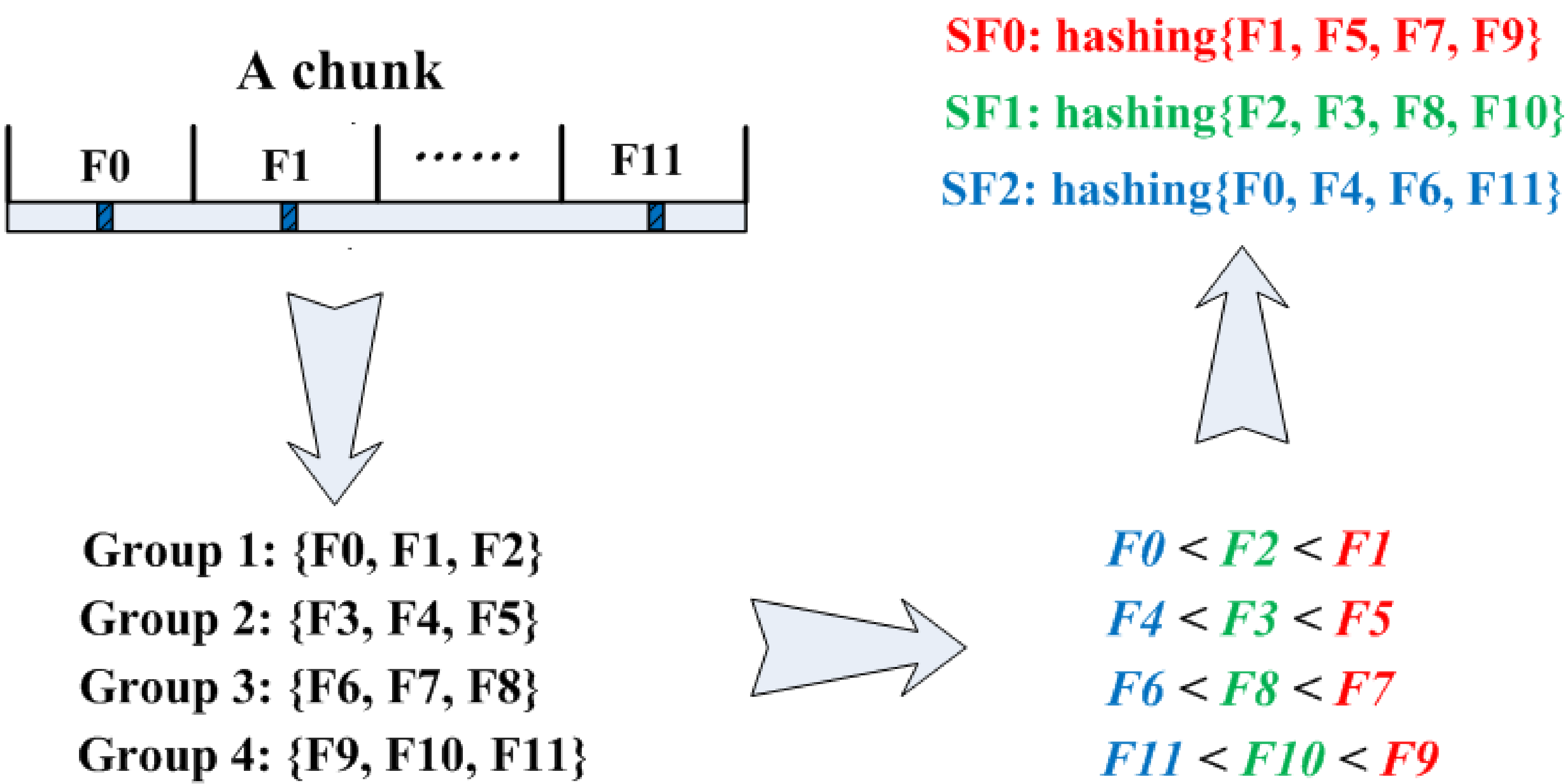
# Design of Finesse

Step2: Feature grouping

- **Principle**: features in an SF should be extracted from the subchunks distributed uniformly across the chunk.

# Design of Finesse

## Step2: Feature grouping



# Computational Overheads

| Approaches | Operations |
|---|---|
| **N-transform SF** | Rabin fingerprinting |
| | **N linear transformations** |
| | **N conditional branches for feature selection** |
| **Finesse** | Rabin fingerprinting |
| | **1 conditional branch for feature selection** |

Computational overhead required to process one shingle.

# Datasets

$$\text{DR} = \frac{\textit{total data size before deduplication}}{\textit{total data size after deduplication}}$$

| Name | Size | DR | Workload descriptions |
| --- | --- | --- | --- |
| WEB | 367 GB | 4.21 | 135 days' snapshots of the website: news.sina.com |
| TAR | 112 GB | 1.70 | 258 versions of Linux kernel source code. Each version is packaged as a tar file |
| RDB | 540 GB | 12.25 | 100 backups of the redis key-value store database |
| SYN | 330 GB | 13.07 | 176 synthetic backups by simulating file create/delete/modify operations |
| VMA | 117 GB | 1.61 | 78 virtual machine images of different OS release versions, including Fedora, CentOS, Debian, etc |
| VMB | 321 GB | 10.45 | 20 backups of an Ubuntu 12.04 VM image in use by a research group |

# Resemblance Detection Efficiency

| Dataset | Approaches | DCR | DCE |
|---|---|---|---|
| WEB | *N-transform SF* | 7.60 | 0.8749 |
| | *Finesse* | 7.52 (–1.05%) | 0.8795 (+0.53%) |
| TAR | *N-transform SF* | 15.00 | 0.9516 |
| | *Finesse* | 15.34 (+2.27%) | 0.9846 (+3.47%) |
| RDB | *N-transform SF* | 3.67 | 0.9129 |
| | *Finesse* | 3.94 (+7.36%) | 0.9448 (+3.49%) |
| SYN | *N-transform SF* | 1.75 | 0.9326 |
| | *Finesse* | 1.70 (–2.86%) | 0.9640 (+3.37%) |
| VMA | *N-transform SF* | 1.56 | 0.9088 |
| | *Finesse* | 1.51 (–3.21%) | 0.9161 (+0.80%) |
| VMB | *N-transform SF* | 1.30 | 0.9093 |
| | *Finesse* | 1.28 (–1.54%) | 0.9193 (+1.10%) |

$$DCR = \frac{\text{total size befoer delta compression}}{\text{total size after delta compression}}$$

$$DCE = \frac{\text{chunk data size after delta compression}}{\text{chunk data size before delta compression}}$$

# Resemblance Detection Efficiency

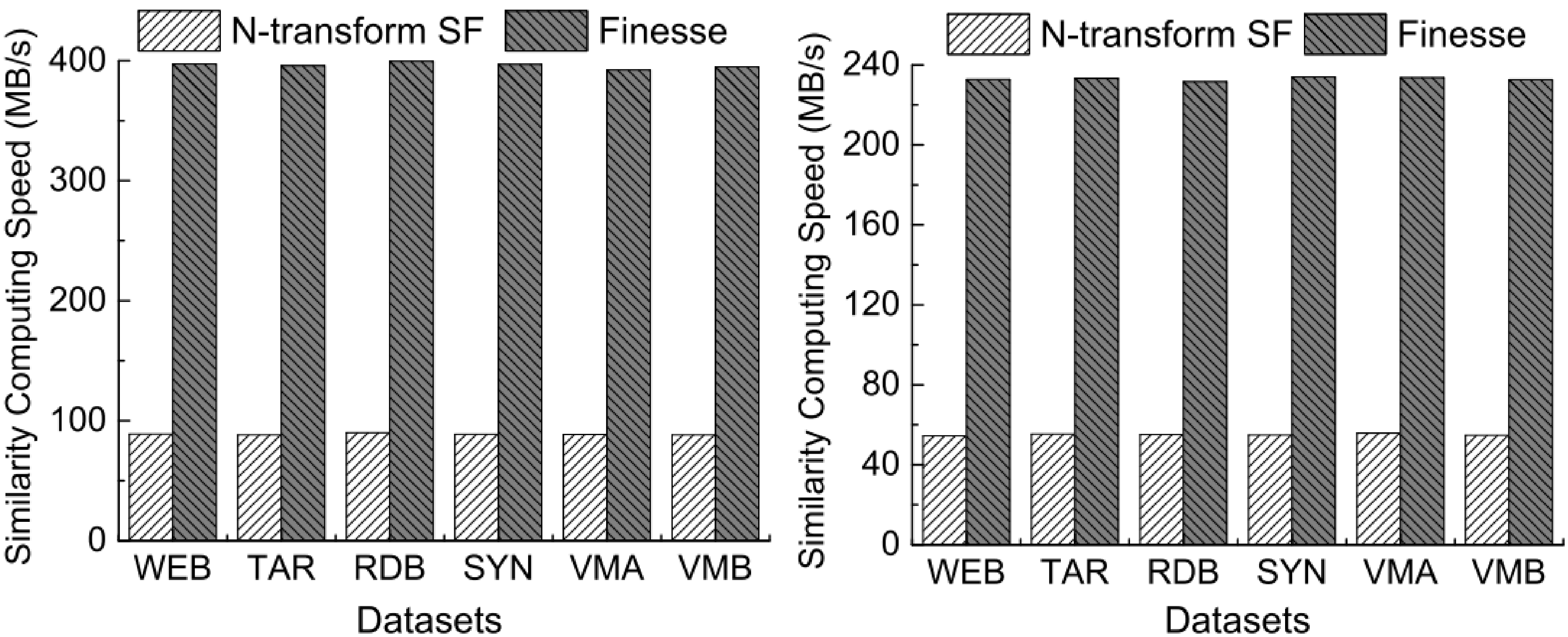
| Dataset | Approaches | DCR | DCE |
|---|---|---|---|
| WEB | *N-transform SF* | 7.60 | 0.8749 |
| | *Finesse* | 7.52 (–1.05%) | 0.8795 (+0.53%) |
| TAR | *N-transform SF* | 15.00 | 0.9516 |
| | *Finesse* | 15.34 (+2.27%) | 0.9846 (+3.47%) |
| RDB | *N-transform SF* | 3.67 | 0.9129 |
| | *Finesse* | 3.94 (+7.36%) | 0.9448 (+3.49%) |
| SYN | *N-transform SF* | 1.75 | 0.9326 |
| | *Finesse* | 1.70 (–2.86%) | 0.9640 (+3.37%) |
| VMA | *N-transform SF* | 1.56 | 0.9088 |
| | *Finesse* | 1.51 (–3.21%) | 0.9161 (+0.80%) |
| VMB | *N-transform SF* | 1.30 | 0.9093 |
| | *Finesse* | 1.28 (–1.54%) | 0.9193 (+1.10%) |

*Finesse achieves the similar resemblance detection efficiency as N-transform SF approach*

$$DCR = \frac{total\ size\ befoer\ delta\ compression}{total\ size\ after\ delta\ compression}$$

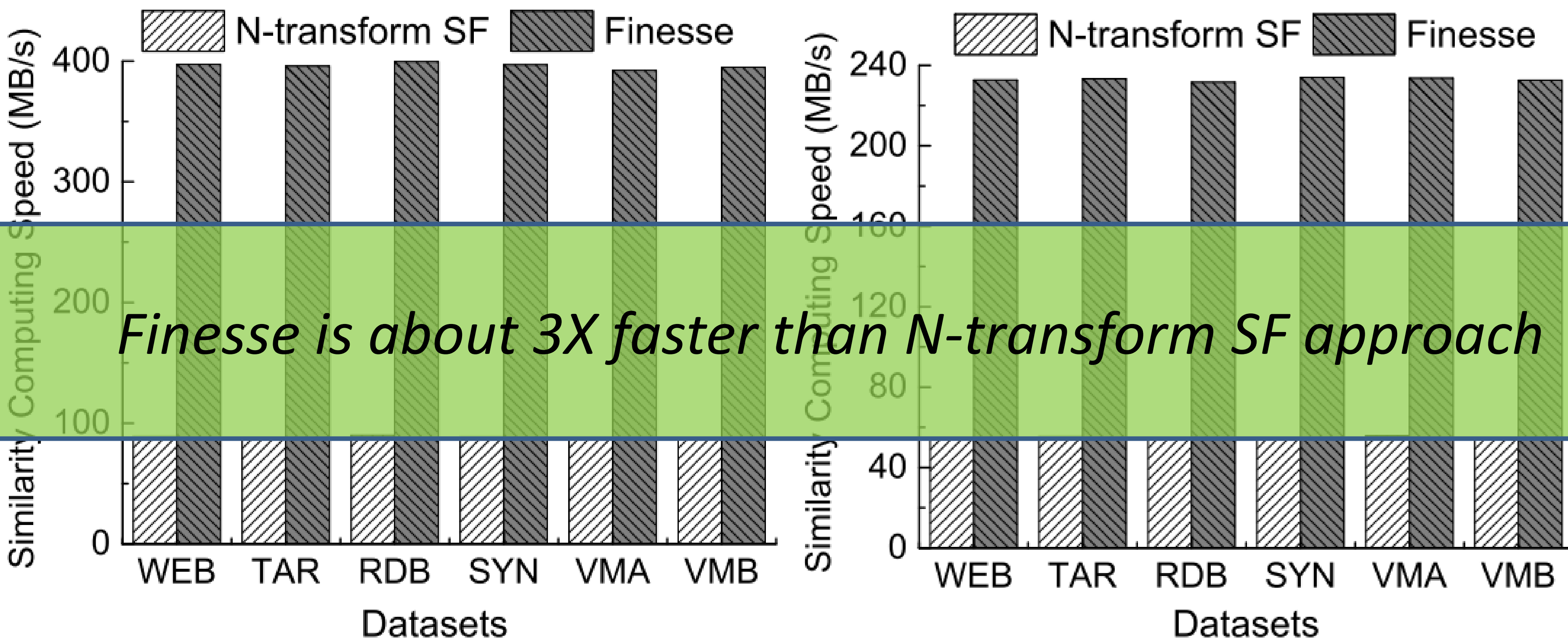
$$DCE = \frac{chunk\ data\ size\ after\ delta\ compression}{chunk\ data\ size\ before\ delta\ compression}$$

# Similarity Computing Speed




(a) Intel i7-4770 (b) Intel E5-2620

# Similarity Computing Speed

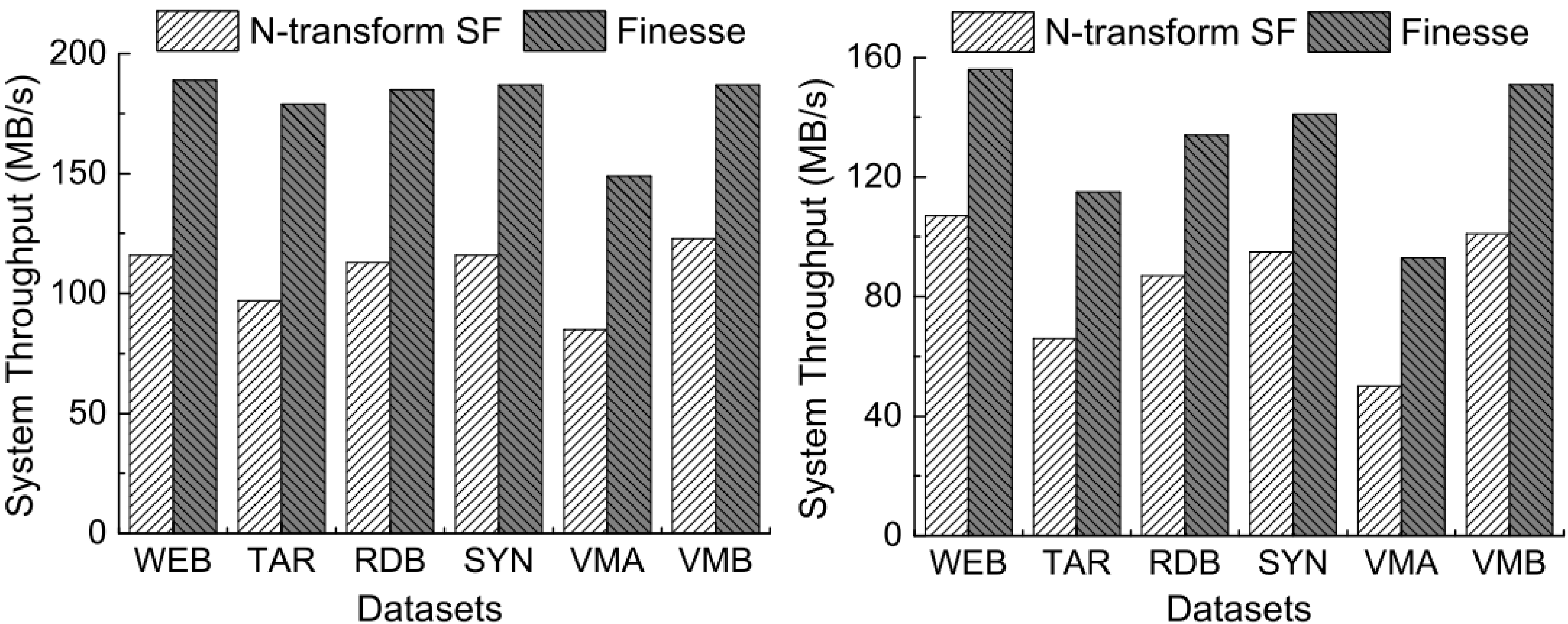



(a) Intel i7-4770

(b) Intel E5-2620

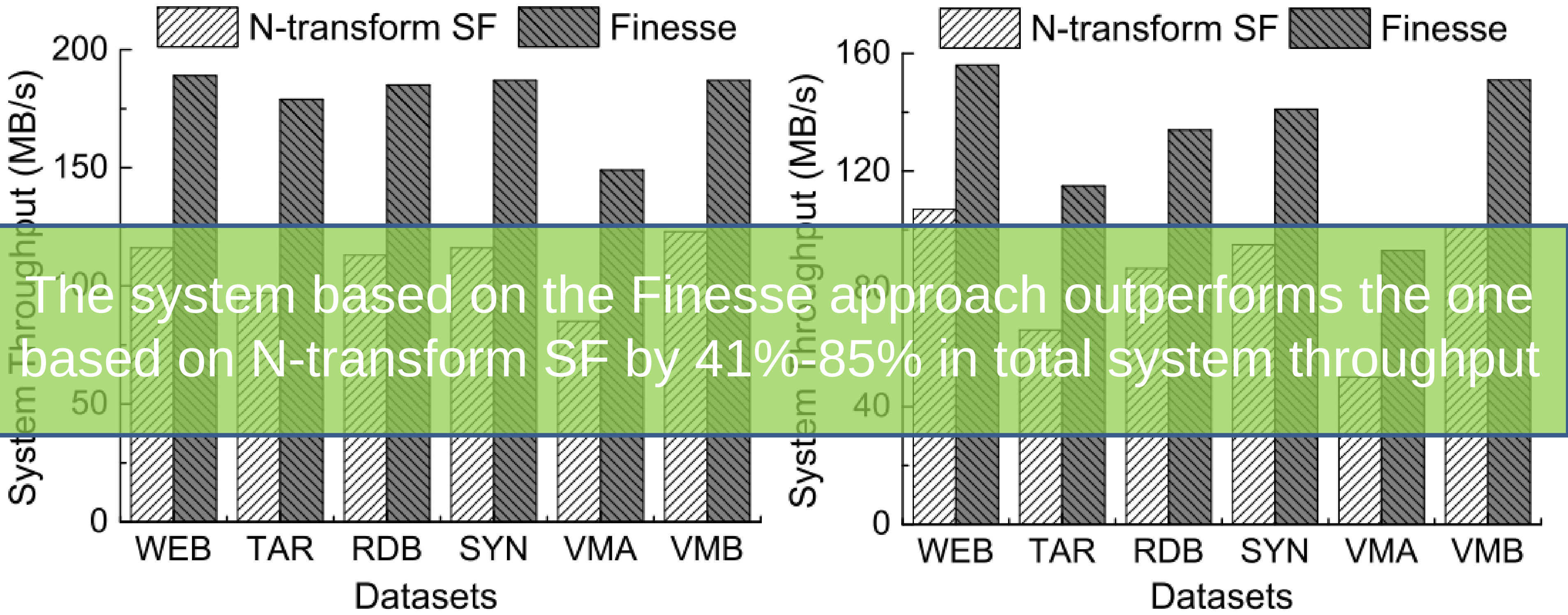
*Finesse is about 3X faster than N-transform SF approach*

# System Throughput




(a) Intel i7-4770 (b) Intel E5-2620

# System Throughput




The system based on the Finesse approach outperforms the one based on N-transform SF by 41%-85% in total system throughput

(a) Intel i7-4770 (b) Intel E5-2620

# Conclusion

- We observed fine-grained feature locality among similar chunks in backup workloads
- We proposed Finesse, a fast resemblance detection based on fine-grained feature locality
- Our experimental results suggest Finesse runs 3X faster than N-transform SF for resemblance detection

# ***Thank you!***

Questions?