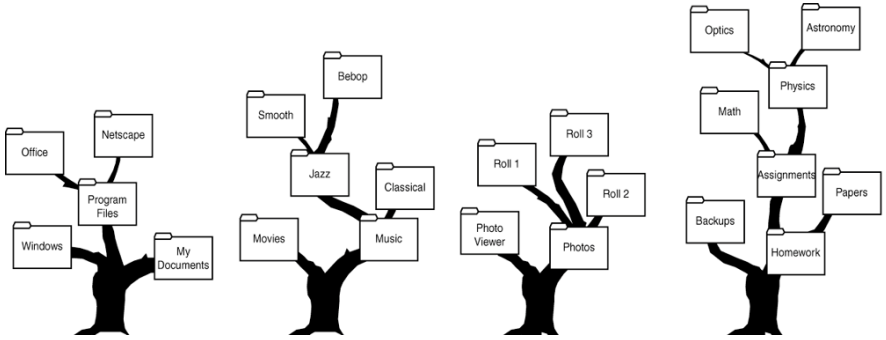
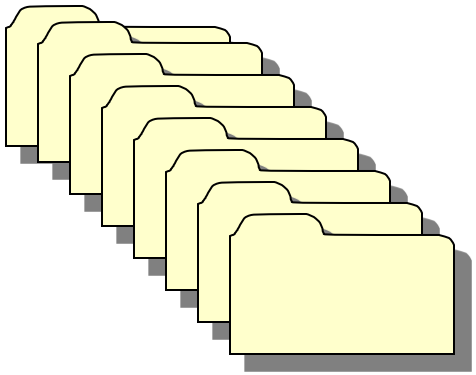
Generating Realistic Datasets for Deduplication Analysis

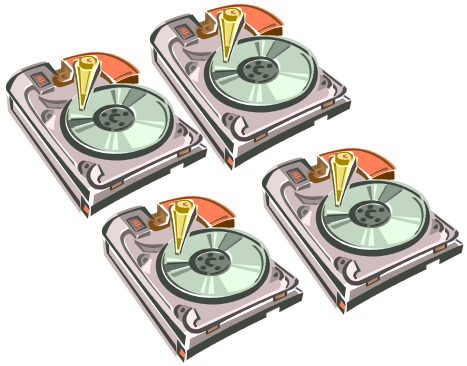


Vasily Tarasov[1], Amar Mudrankit[1], Will Buik[2], Philip Shilane[3], Geoff Kuenning[2], Erez Zadok[1]

[1]Stony Brook University [2]Harvey Mudd College [3]EMC Corporation

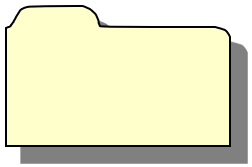
Overview





EMC² STONY BROOK UNIVERSITY

Overview

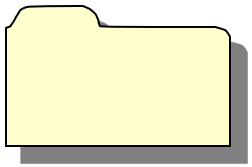


Deduplication



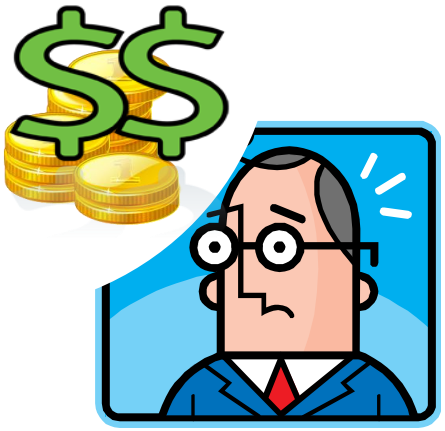
EMC² STONY BROOK UNIVERSITY

Overview



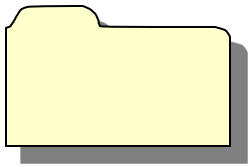
Deduplication





EMC² STONY BROOK UNIVERSITY

Overview



Deduplication





Wait, deduplication performance depends on the **data…**

Everyone uses different datasets

EMC² STONY BROOK UNIVERSITY

Outline

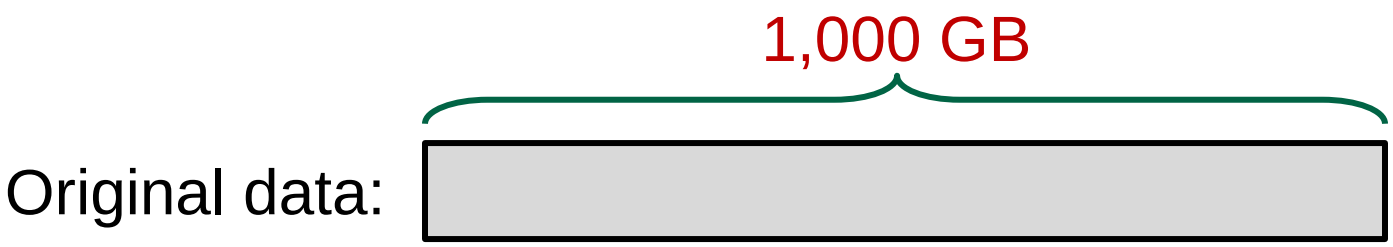
1. **Deduplication and datasets survey**
2. Framework design
3. Framework implementation
4. Evaluation
5. Conclusions & future work

Outline

1. **Deduplication and datasets survey**
2. Framework design
3. Framework implementation
4. Evaluation
5. Conclusions & future works

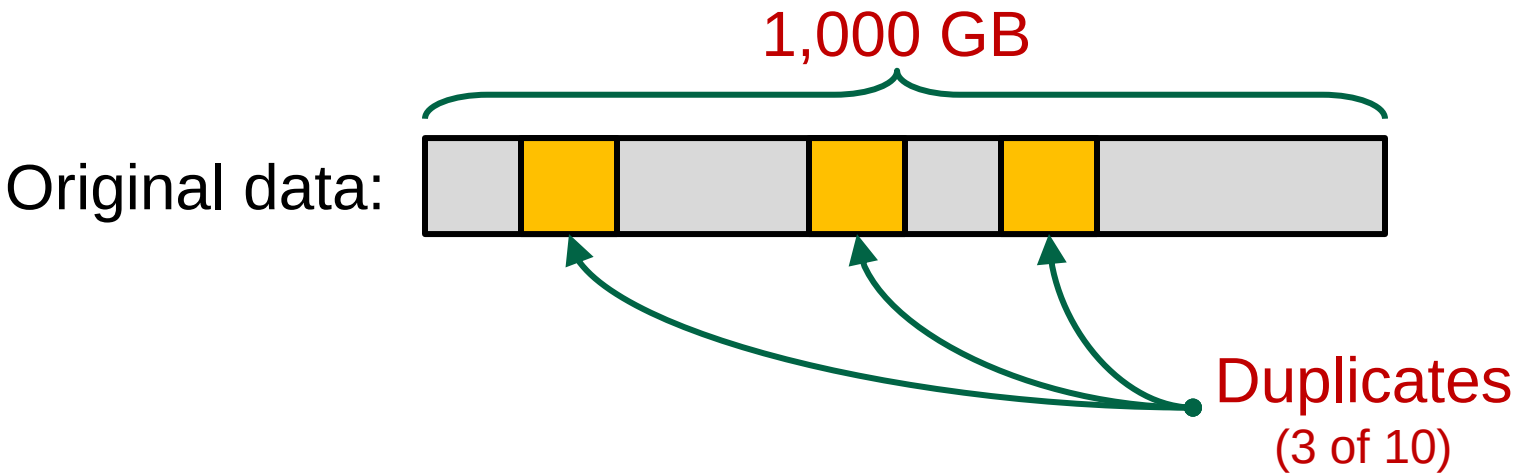
Deduplication Basics

Coarse-grained data compression technique:



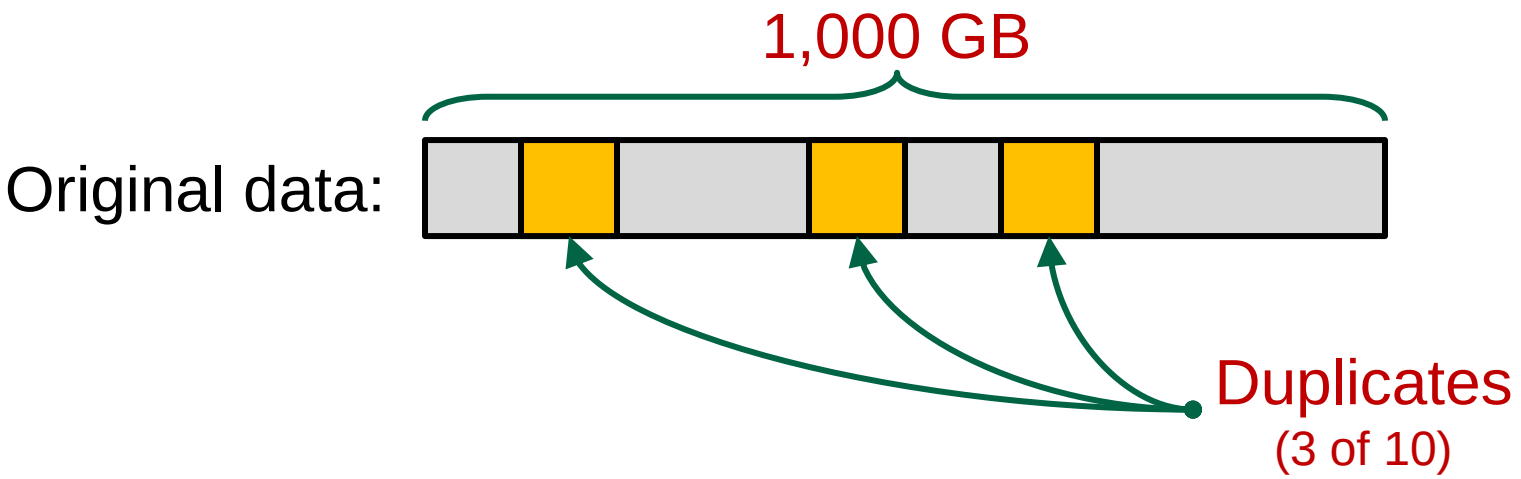
Deduplication Basics

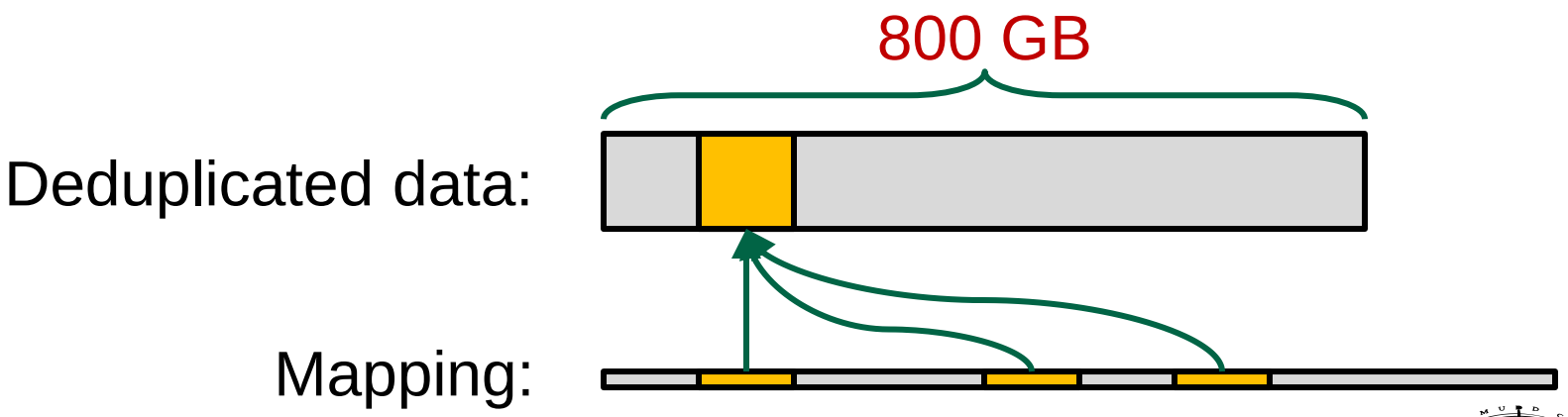
Coarse-grained data compression technique:



Deduplication Basics

Coarse-grained data compression technique:

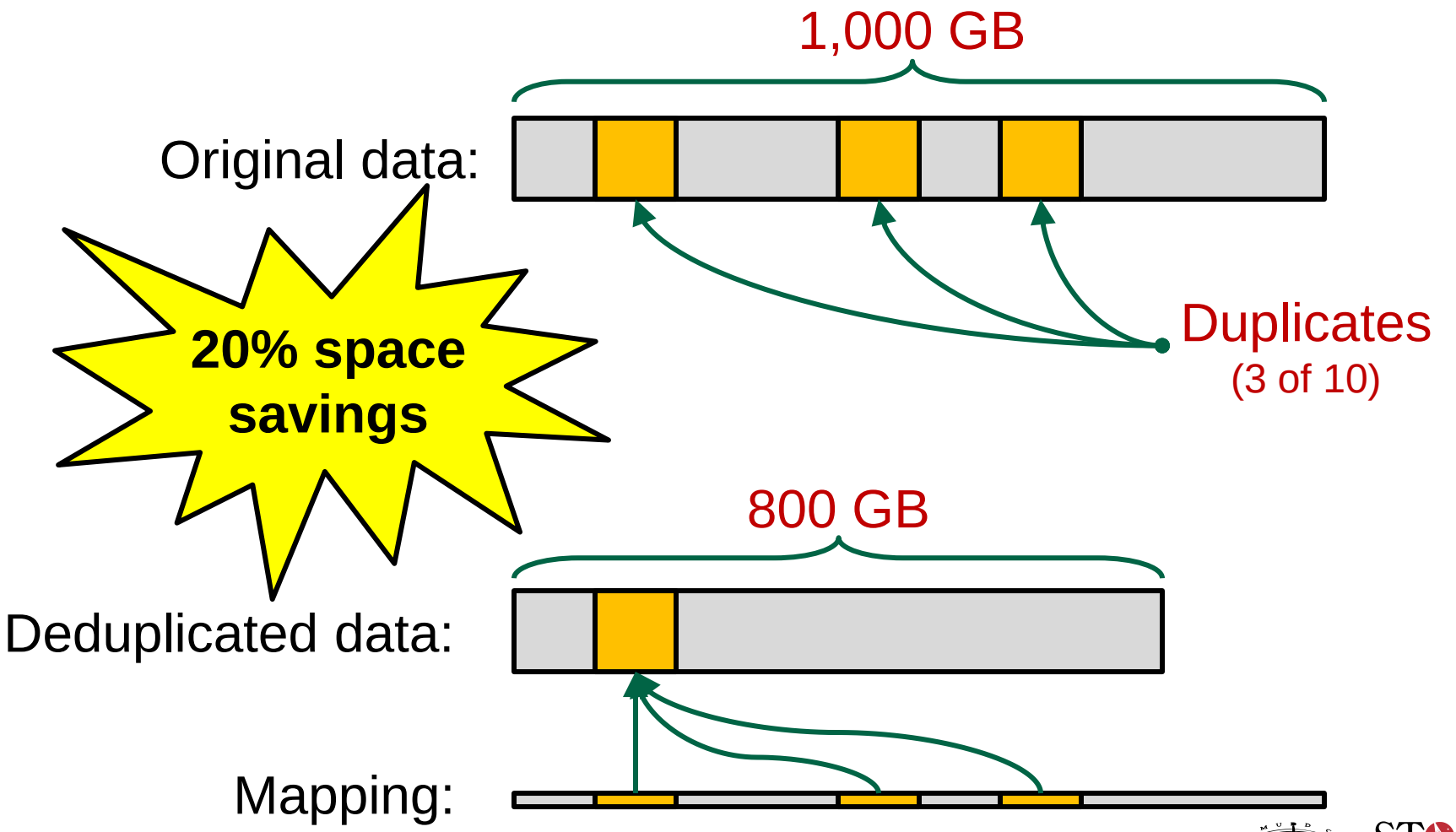




EMC² STONY BROOK UNIVERSITY

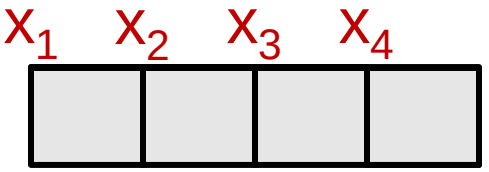
Deduplication Basics

Coarse-grained data compression technique:

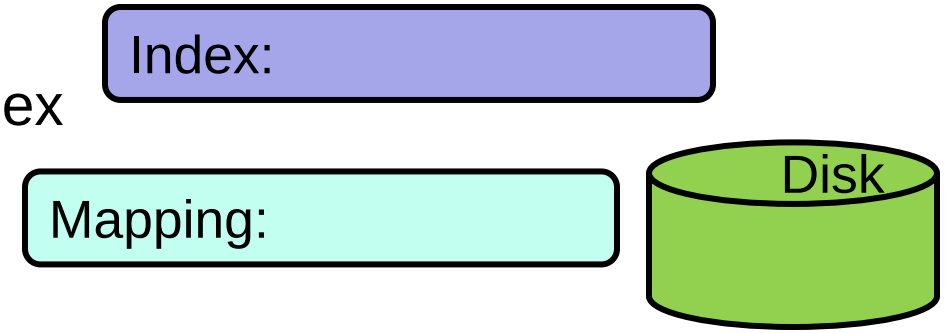


Typical Ingest Datapath

1. Chunking
   - Determine **boundaries** in the data
     - File boundaries
     - Fixed-size chunking
     - Variable chunking using Rabin fingerprints
     - More intelligent ways, e.g., content-type-aware
2. Hashing
   - Easy-to-compute chunk **identifiers**
   - Collision-resistant
   - Cryptographic
   - SHA256, MD5 (plus byte-by-byte comparison)
3. Indexing and mapping
   - Find duplicates in the index
   - Create mapping entries



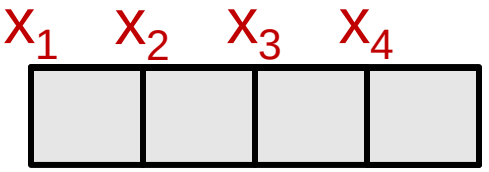
0xBE 0xEF 0xBE 0xEF



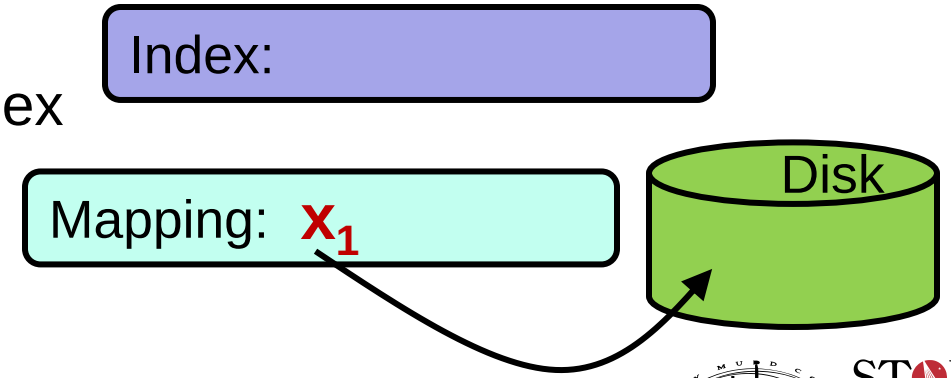
EMC² STONY BROOK UNIVERSITY

Typical Ingest Datapath

1. Chunking
   - Determine **boundaries** in the data
     - File boundaries
     - Fixed-size chunking
     - Variable chunking using Rabin fingerprints
     - More intelligent ways, e.g., content-type-aware
2. Hashing
   - Easy-to-compute chunk **identifiers**
   - Collision-resistant
   - Cryptographic
   - SHA256, MD5 (plus byte-by-byte comparison)
3. Indexing and mapping
   - Find duplicates in the index
   - Create mapping entries



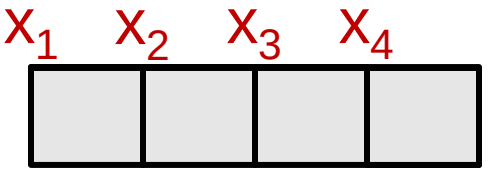
0xBE **0xEF** **0xBE** **0xEF**



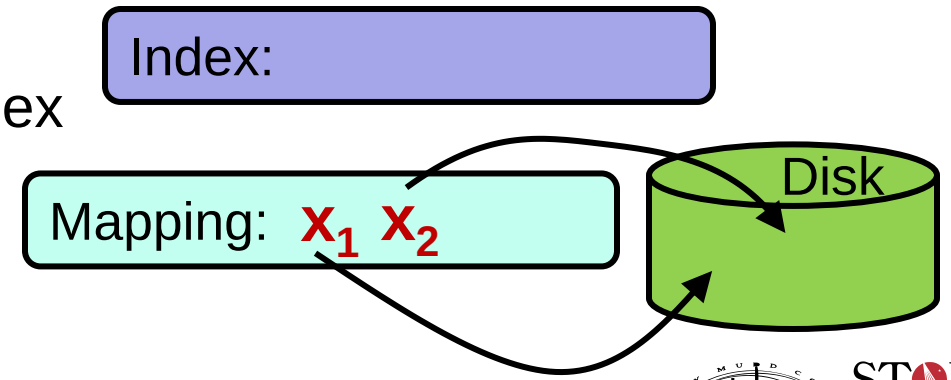
EMC² STONY BROOK UNIVERSITY

Typical Ingest Datapath

1. Chunking
   - Determine **boundaries** in the data
     - File boundaries
     - Fixed-size chunking
     - Variable chunking using Rabin fingerprints
     - More intelligent ways, e.g., content-type-aware
2. Hashing
   - Easy-to-compute chunk **identifiers**
   - Collision-resistant
   - Cryptographic
   - SHA256, MD5 (plus byte-by-byte comparison)
3. Indexing and mapping
   - Find duplicates in the index
   - Create mapping entries



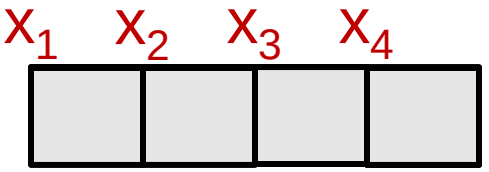
0xBE 0xEF 0xBE 0xEF

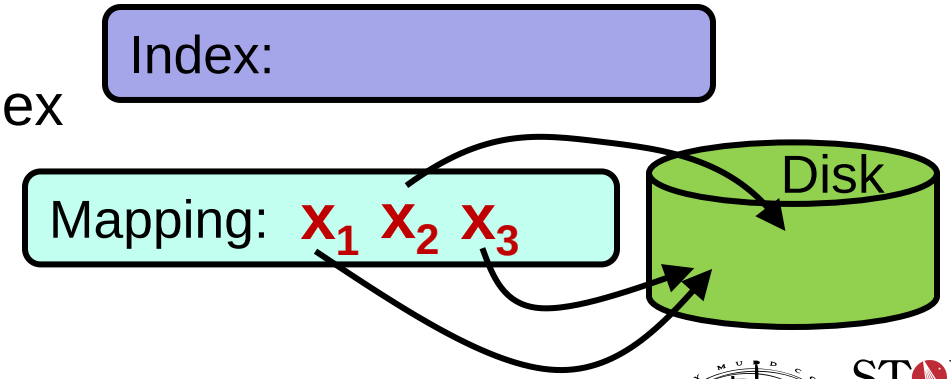


EMC² STONY BROOK UNIVERSITY

Typical Ingest Datapath

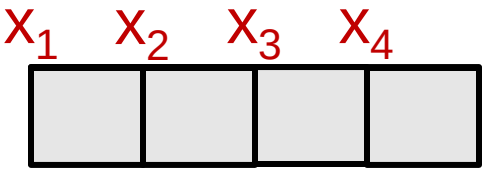
1. Chunking
   - Determine **boundaries** in the data
     - File boundaries
     - Fixed-size chunking
     - Variable chunking using Rabin fingerprints
     - More intelligent ways, e.g., content-type-aware
2. Hashing
   - Easy-to-compute chunk **identifiers**
   - Collision-resistant
   - Cryptographic
   - SHA256, MD5 (plus byte-by-byte comparison)
3. Indexing and mapping
   - Find duplicates in the index
   - Create mapping entries



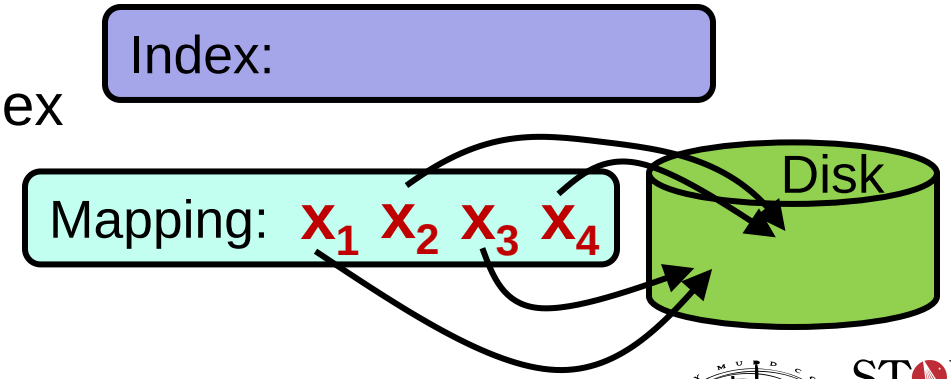


Typical Ingest Datapath

1. Chunking
   - Determine **boundaries** in the data
     - File boundaries
     - Fixed-size chunking
     - Variable chunking using Rabin fingerprints
     - More intelligent ways, e.g., content-type-aware
2. Hashing
   - Easy-to-compute chunk **identifiers**
   - Collision-resistant
   - Cryptographic
   - SHA256, MD5 (plus byte-by-byte comparison)
3. Indexing and mapping
   - Find duplicates in the index
   - Create mapping entries



0xBE **0xEF**



EMC² STONY BROOK UNIVERSITY

How Efficient is Deduplication?

- High deduplication ratios for certain datasets
  - Backups: **15×** [Wallace2012]
  - Virtual machine images: **7×** [Smith2008]
  - Multi-tenant shared storage: **3×** [Meyer2011]
- Fastest growing segment of storage industry [NetApp/IDC 2011]
  - **80%+** of corporations are exploring deduplication [IDC 2011]
- Challenges:
  - Performance
    - Computationally intensive
    - Out of RAM indexes
    - Fragmentation
  - Manageability, reliability

EMC² STONY BROOK UNIVERSITY

How Efficient is Deduplication?

- High deduplication ratios for certain datasets
  - Backups: **15×** [Wallace2012]
  - Virtual machine images: **7×** [Smith2008]
  - Multi-tenant shared storage: **3×** [Meyer2011]
- Fastest growing segment of storage industry [NetApp/IDC 2011]
  - **80%+** of corporations are exploring deduplication [IDC 2011]
- Challenges:
  - Performance
    - Computationally intensive
    - Out of RAM indexes
    - Fragmentation
  - Manageability, reliability

Variety of optimizations:

- Intelligent caching
- Intelligent prefetching
- Bloom filters
- Content-aware chunking
- …

EMC² STONY BROOK UNIVERSITY

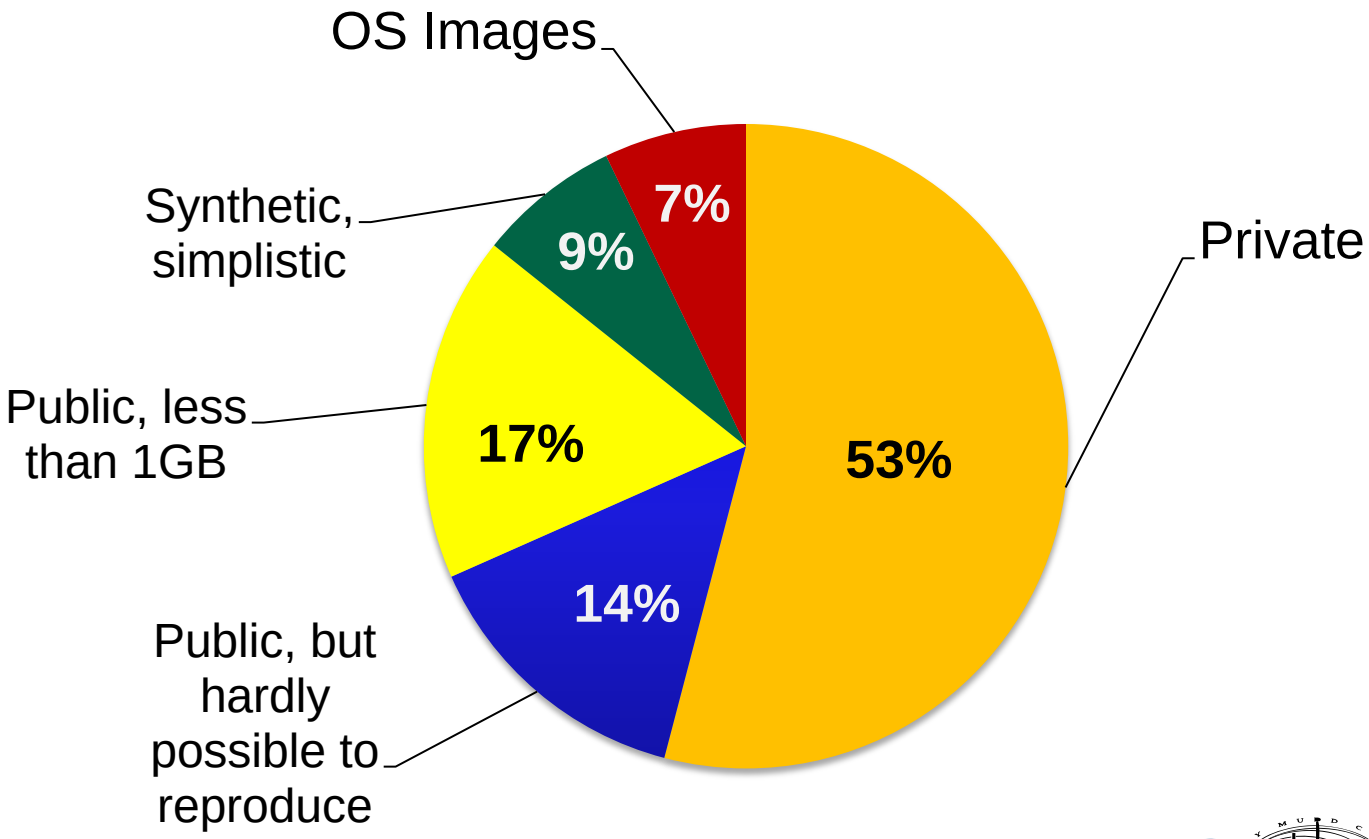
Survey of Datasets

- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets

EMC² STONY BROOK UNIVERSITY

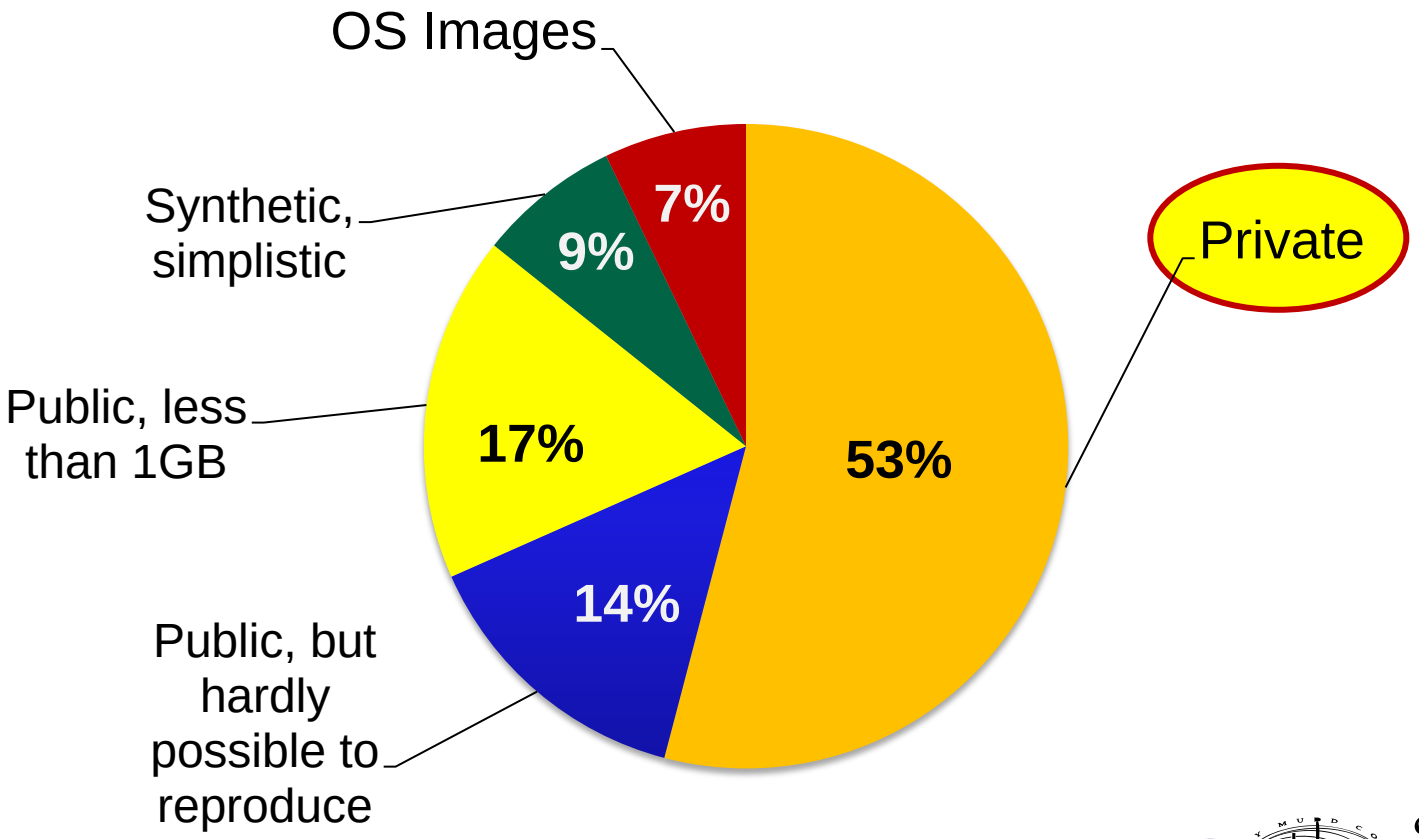
Survey of Datasets

- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets



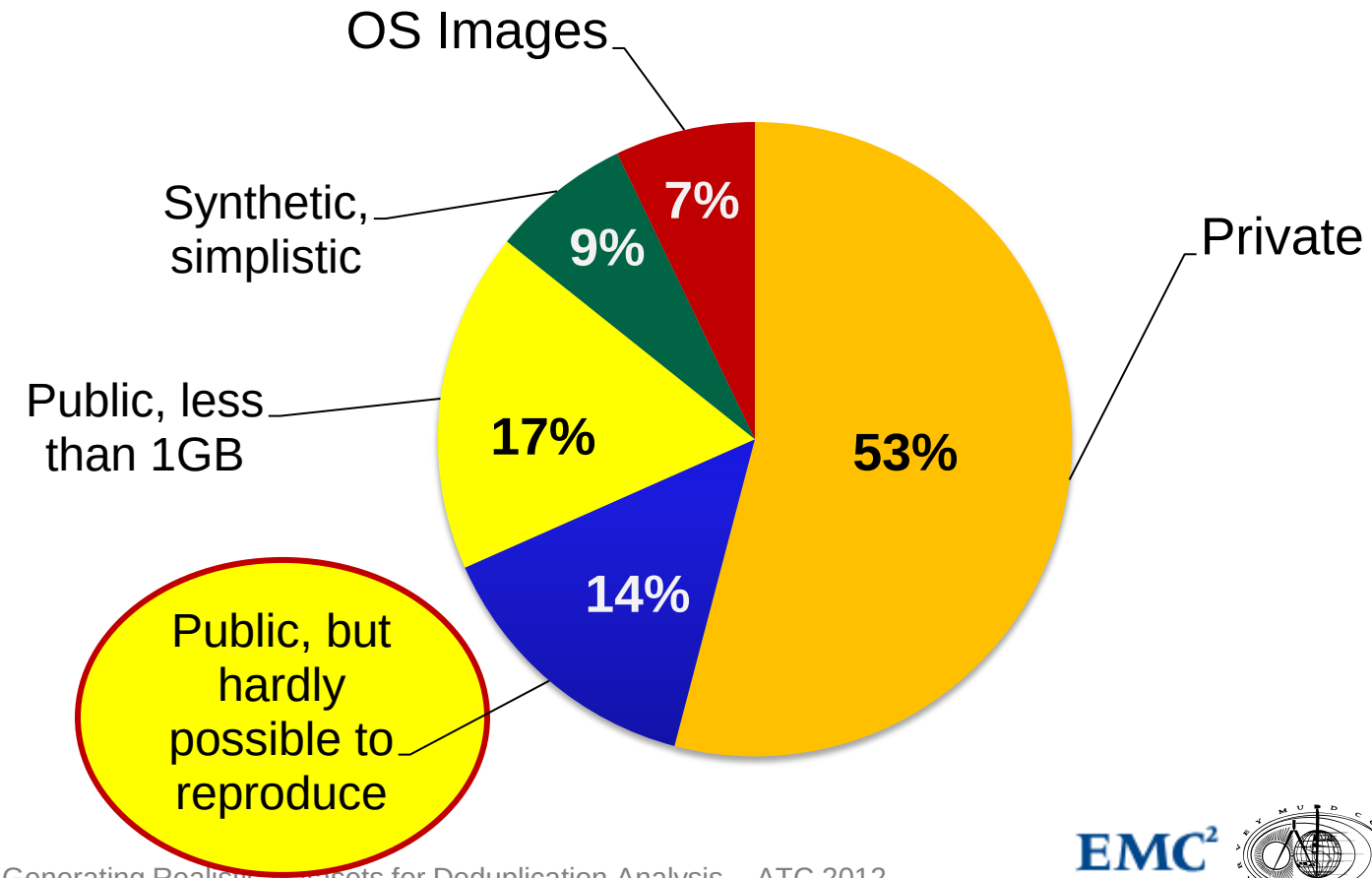
Survey of Datasets

- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets



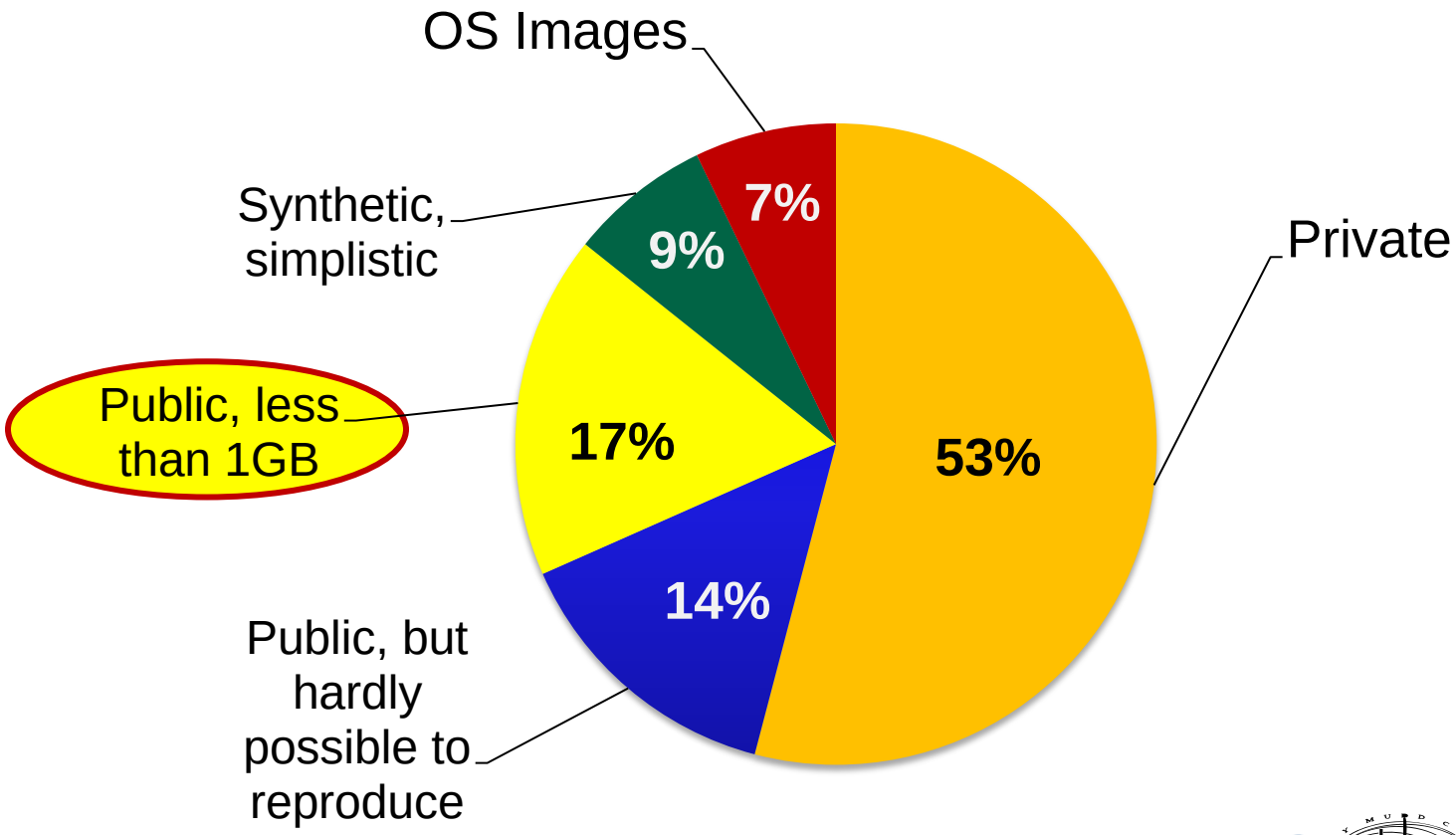
Survey of Datasets

- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets



Survey of Datasets

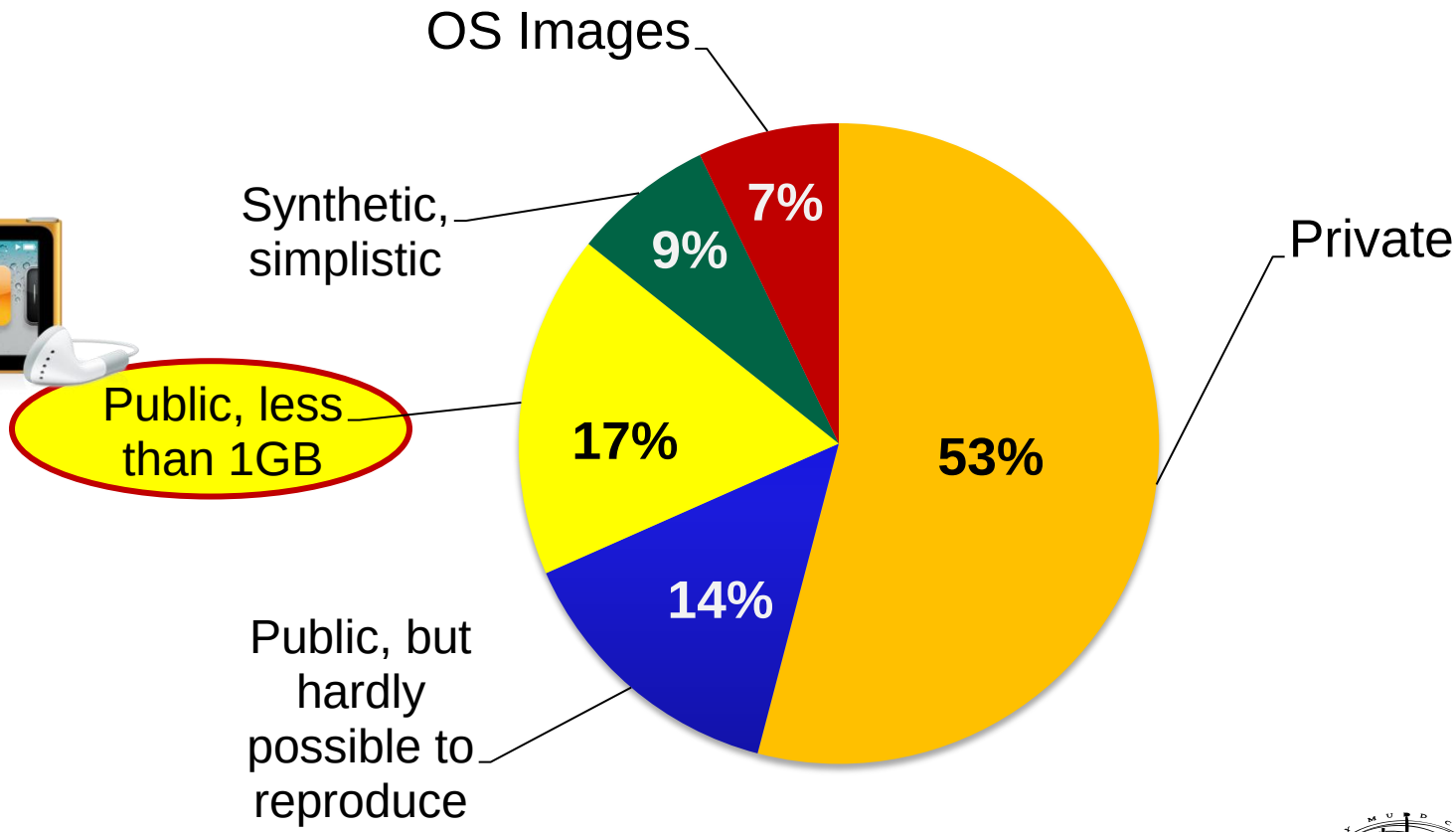
- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets



Survey of Datasets

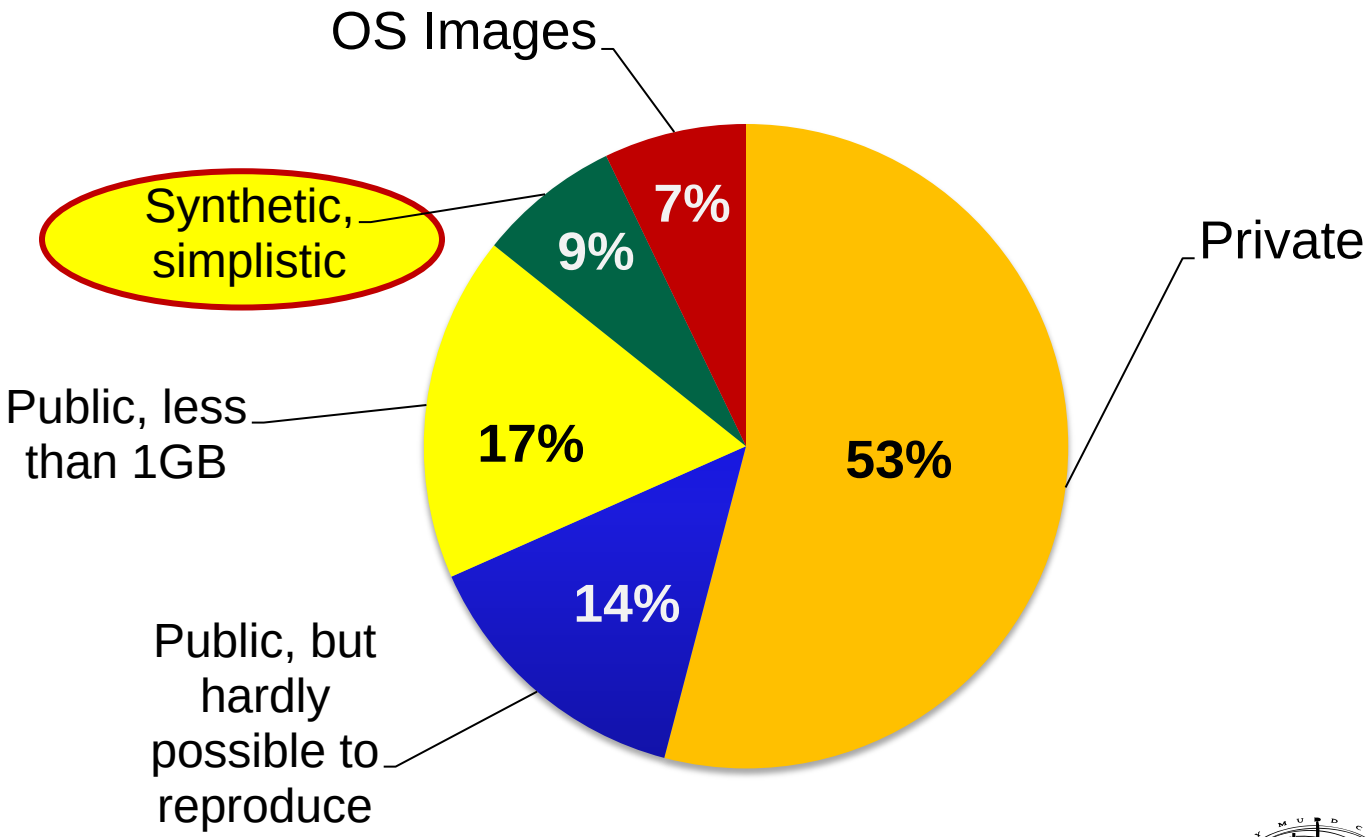
- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets





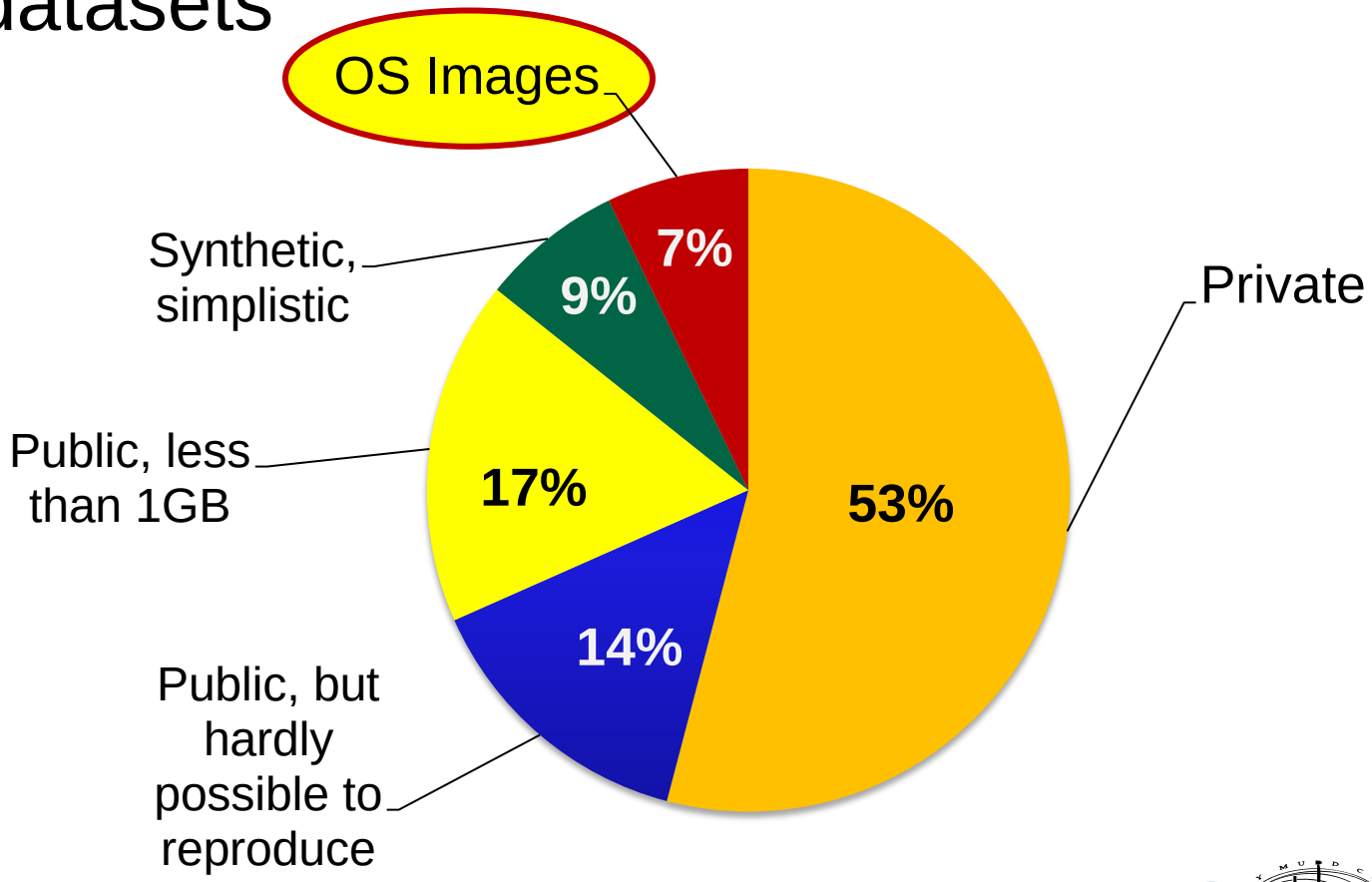
Survey of Datasets

- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets



Survey of Datasets

- **33** research papers
- **FAST**, **ATC**, **MSST**, **SYSTOR** conferences
- **120** datasets



The Need for Datasets

To perform **true**, **versatile**, and **easy** evaluation and comparison of deduplication systems

Requirements:

- Accessible
- Realistic
- Sufficiently large
- Easy to distribute
- With controllable characteristics

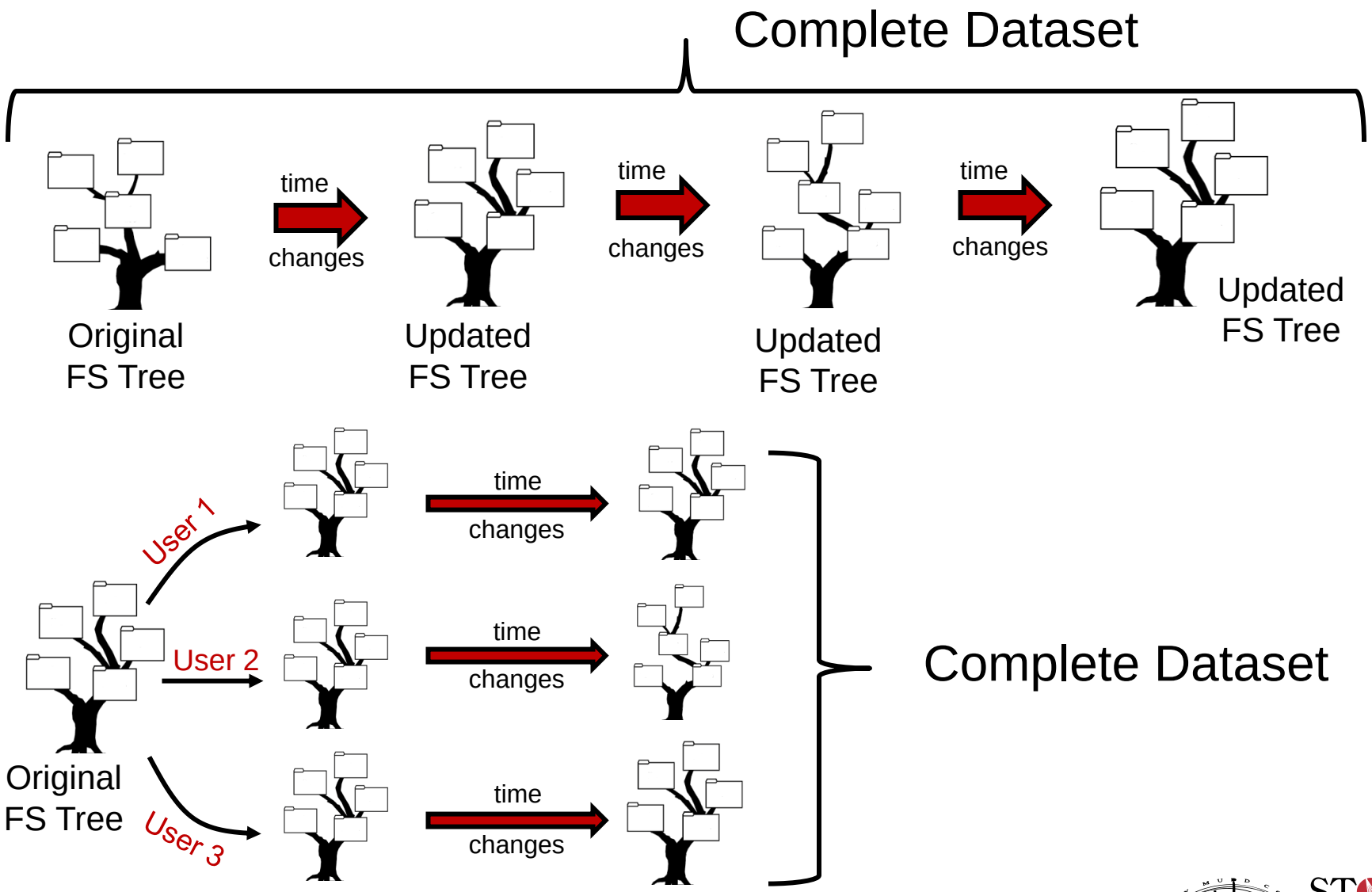


EMC² STONY BROOK UNIVERSITY

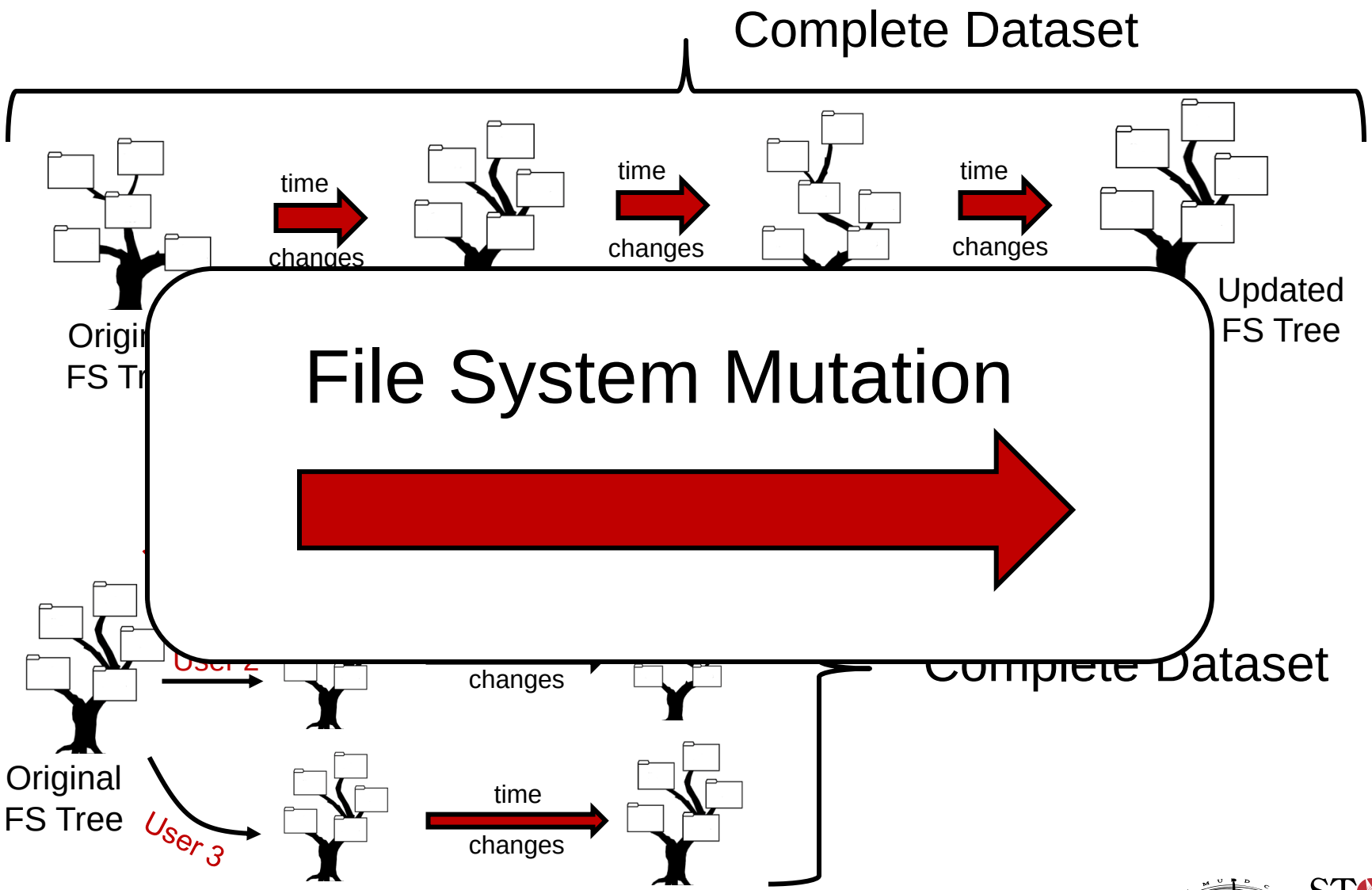
Outline

1. Deduplication and datasets survey
2. **Framework design**
3. Framework implementation
4. Evaluation
5. Conclusions & future works

Insight


Complete Dataset
time
changes
Original
FS Tree
Updated
FS Tree
Updated
FS Tree
Updated
FS Tree
User 1
User 2
User 3
Original
FS Tree
Complete Dataset

Insight


Complete Dataset
time
changes
time
changes
time
changes
Updated
FS Tree
File System Mutation
changes
Original
FS Tree
User 3
time
changes

Framework Objects

- FSTREE
  - Per-snapshot
  - In-memory
- DIRECTORY
- FILE
- CHUNK
  - Chunk identifiers
    - Roughly correspond to hashes
  - Per-file lists of chunks

Framework Objects

- FSTREE
  - Per-snapshot
  - In-memory
- DIRECTORY
- FILE
- CHUNK
  - Chunk identifiers
    - Roughly correspond to hashes
  - Per-file lists of chunks

2TB **snapshots,**
16KB file size, 10 files/directory

9GB RAM

EMC² STONY BROOK UNIVERSITY

Framework Objects

- FSTREE
  - Per-snapshot
  - In-memory
- DIRECTORY
- FILE
- CHUNK
  - Chunk identifiers
    - Roughly correspond to hashes
  - Per-file lists of chunks

2TB **snapshots,** **9GB** RAM
16KB file size, 10 files/directory

14TB **snapshots** **64GB** RAM

EMC² STONY BROOK UNIVERSITY

Framework Objects

- FSTREE
  - Per-snapshot
  - In-memory
- DIRECTORY
- FILE
- CHUNK
  - Chunk identifiers
    - Roughly correspond
  - Per-file lists of chunks

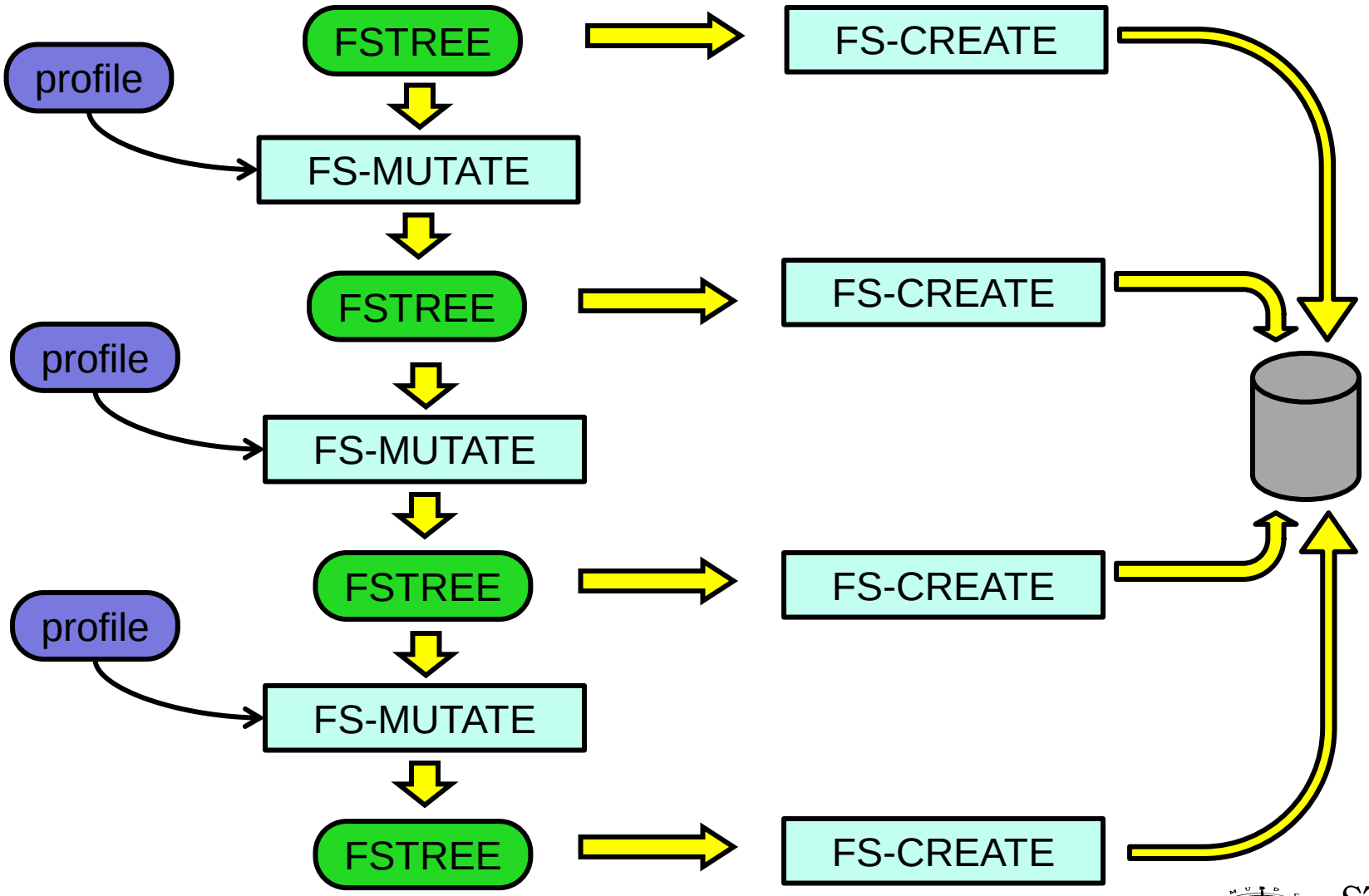
2TB **snapshots,** **9GB** RAM
16KB file size, 10 files/directory

14TB **snapshots** **64GB** RAM

71 snapshots
Total: **1PB** dataset
still 64GB of RAM

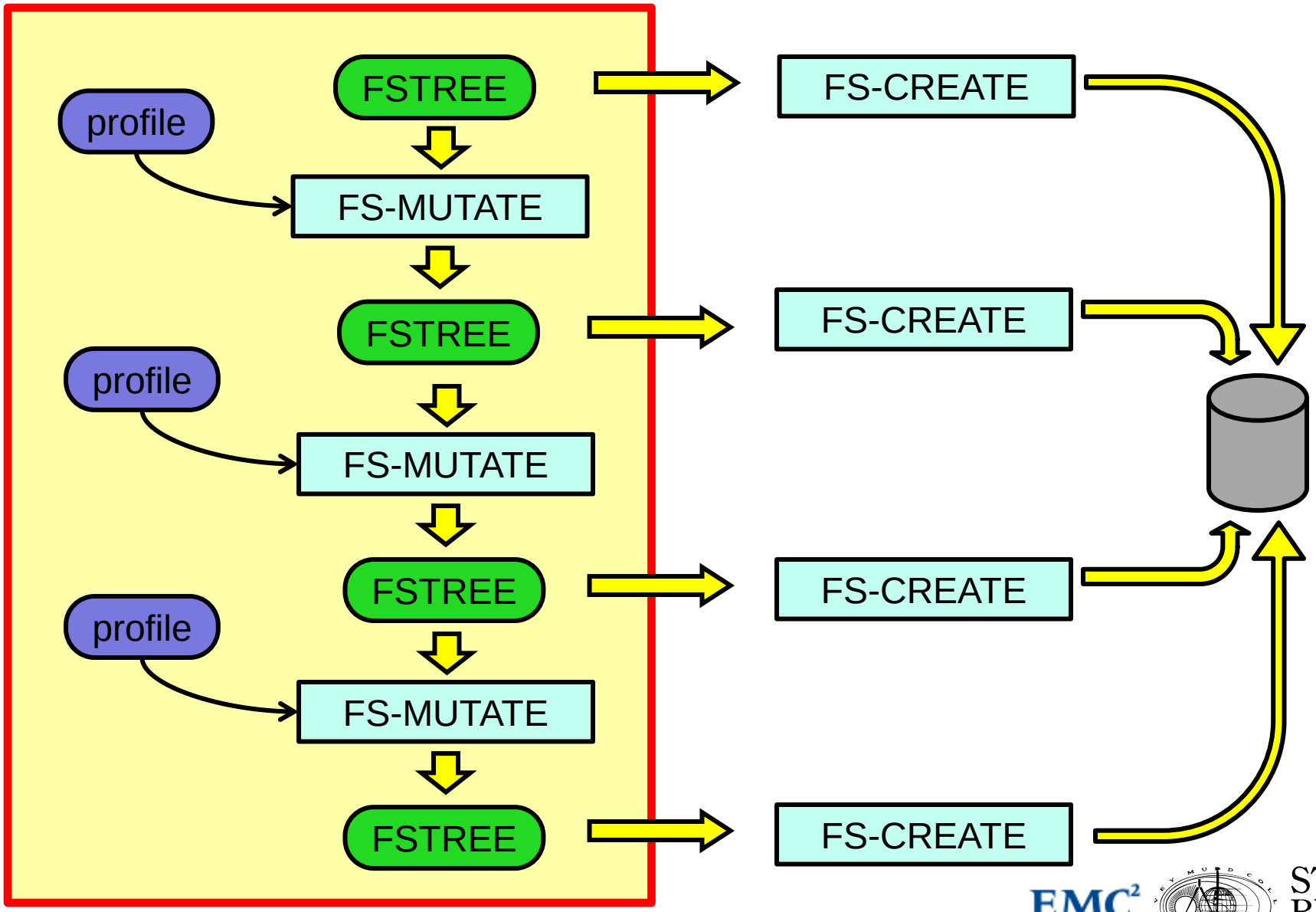
EMC² STONY BROOK UNIVERSITY

Mutation Conveyor

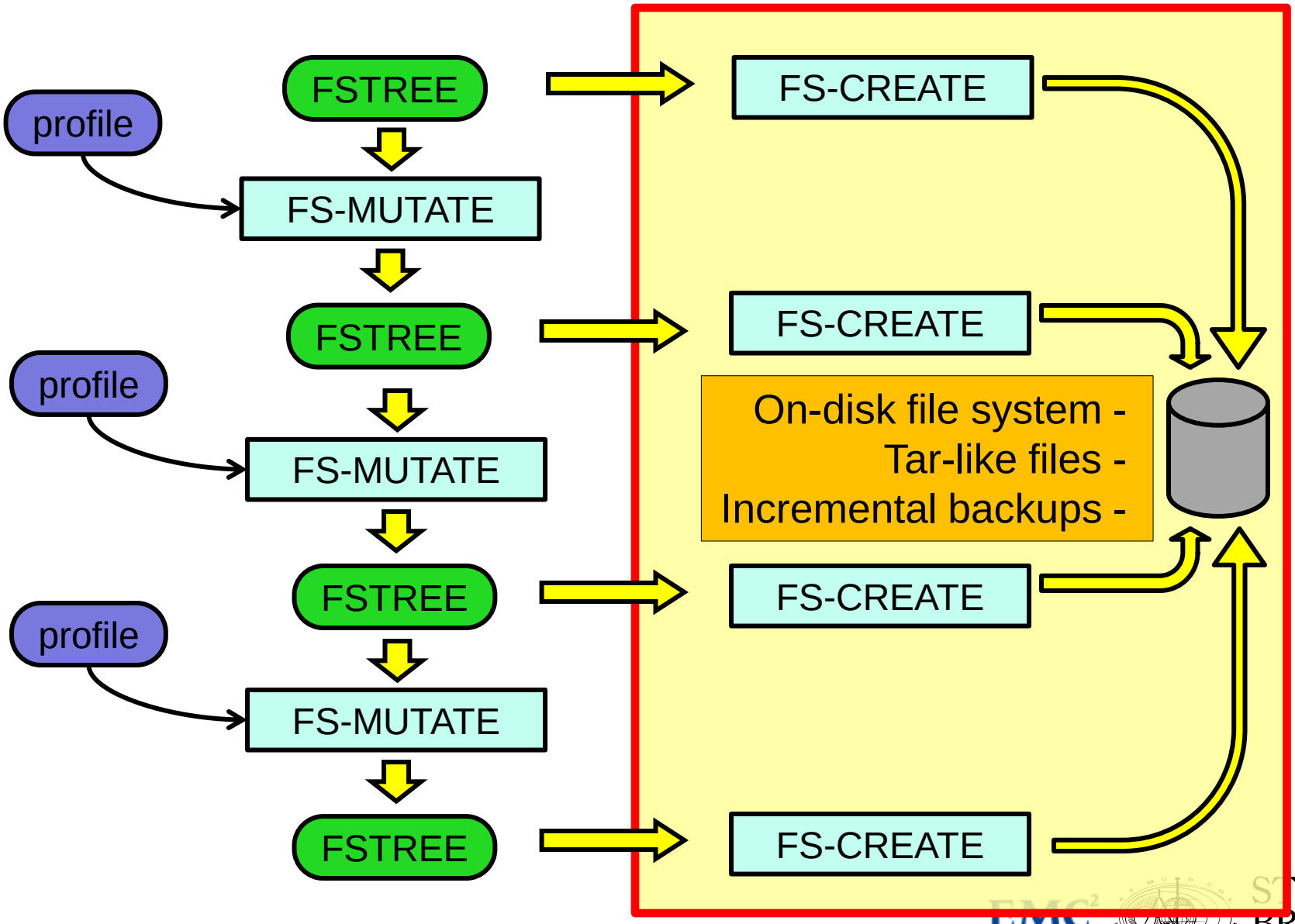


EMC² STONY BROOK UNIVERSITY

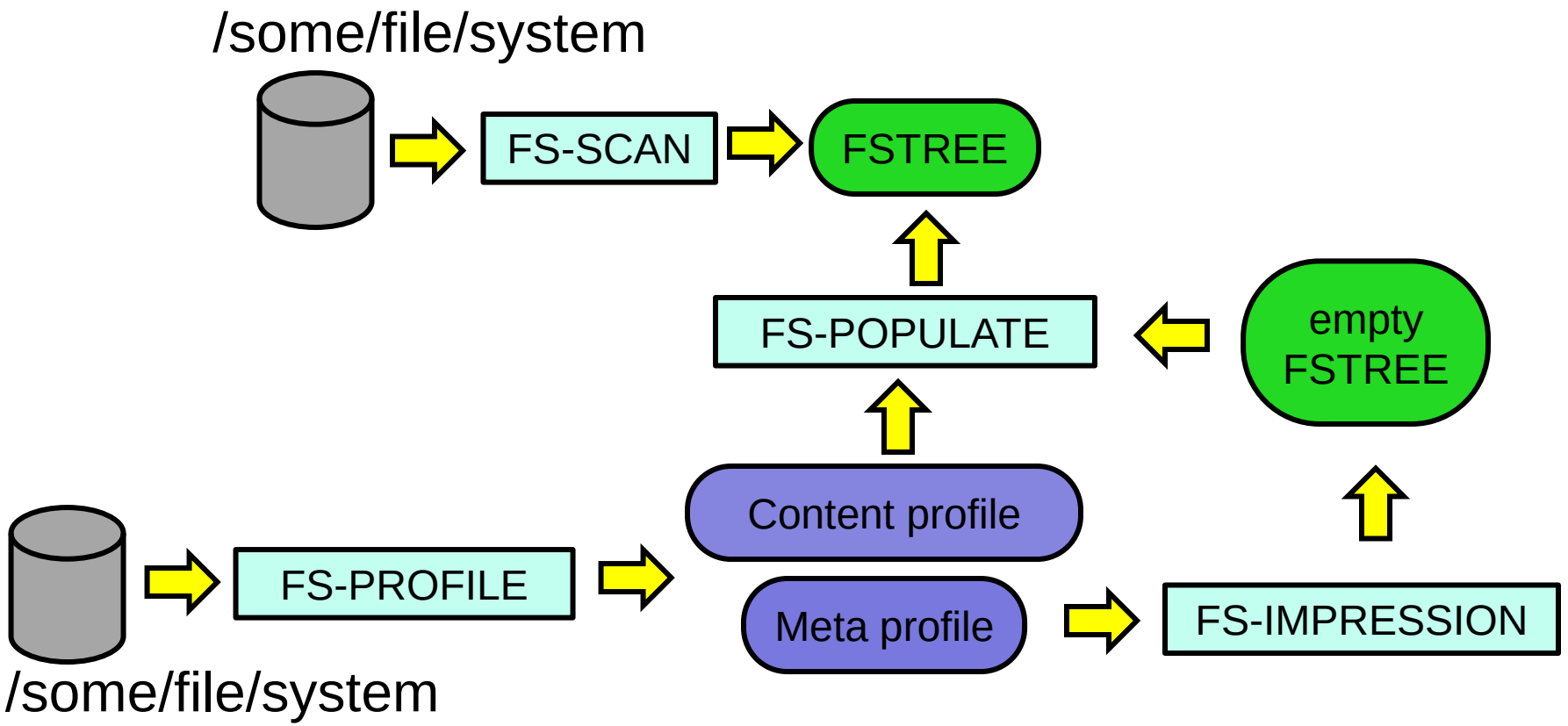
Mutation Conveyor



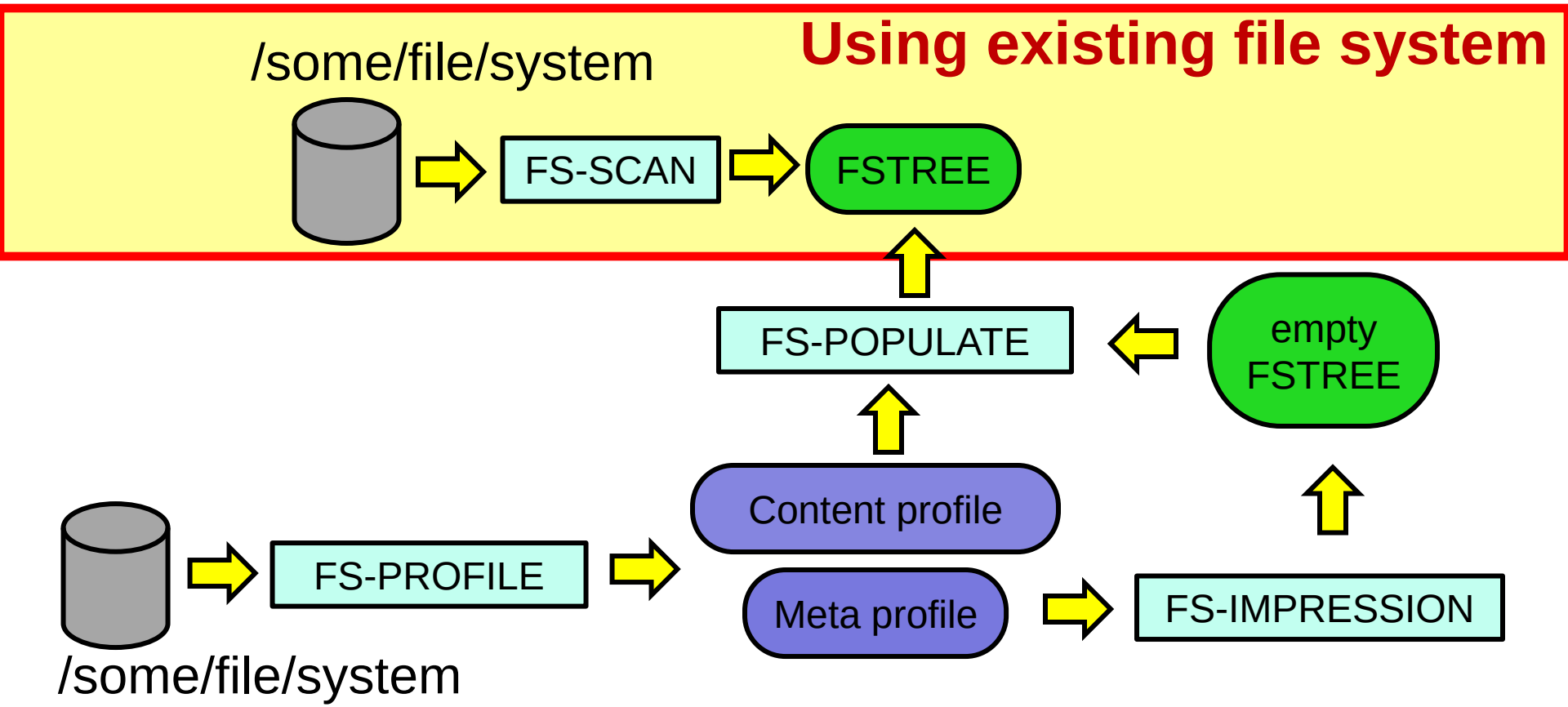
Mutation Conveyor



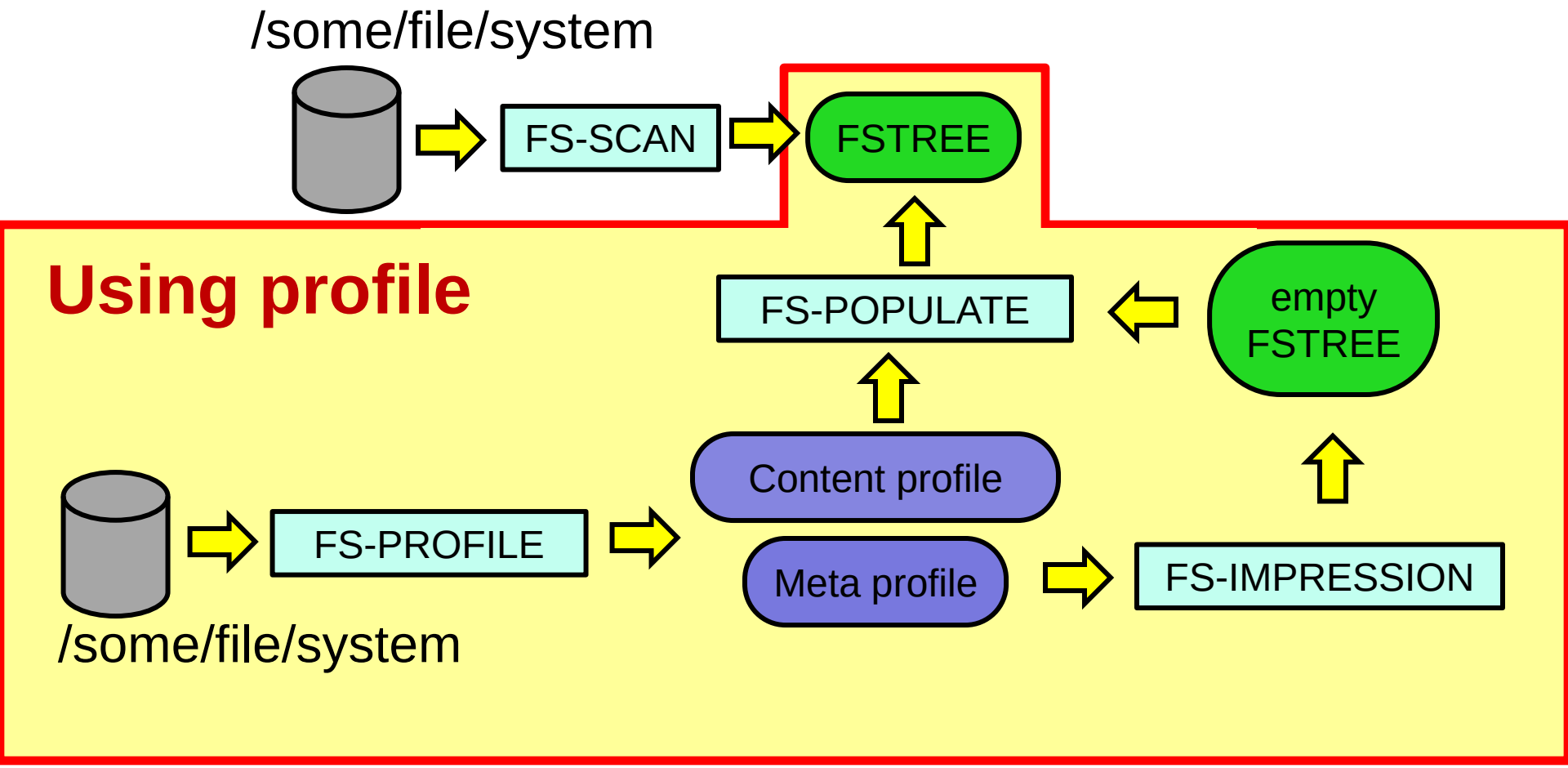
Initial File System Tree



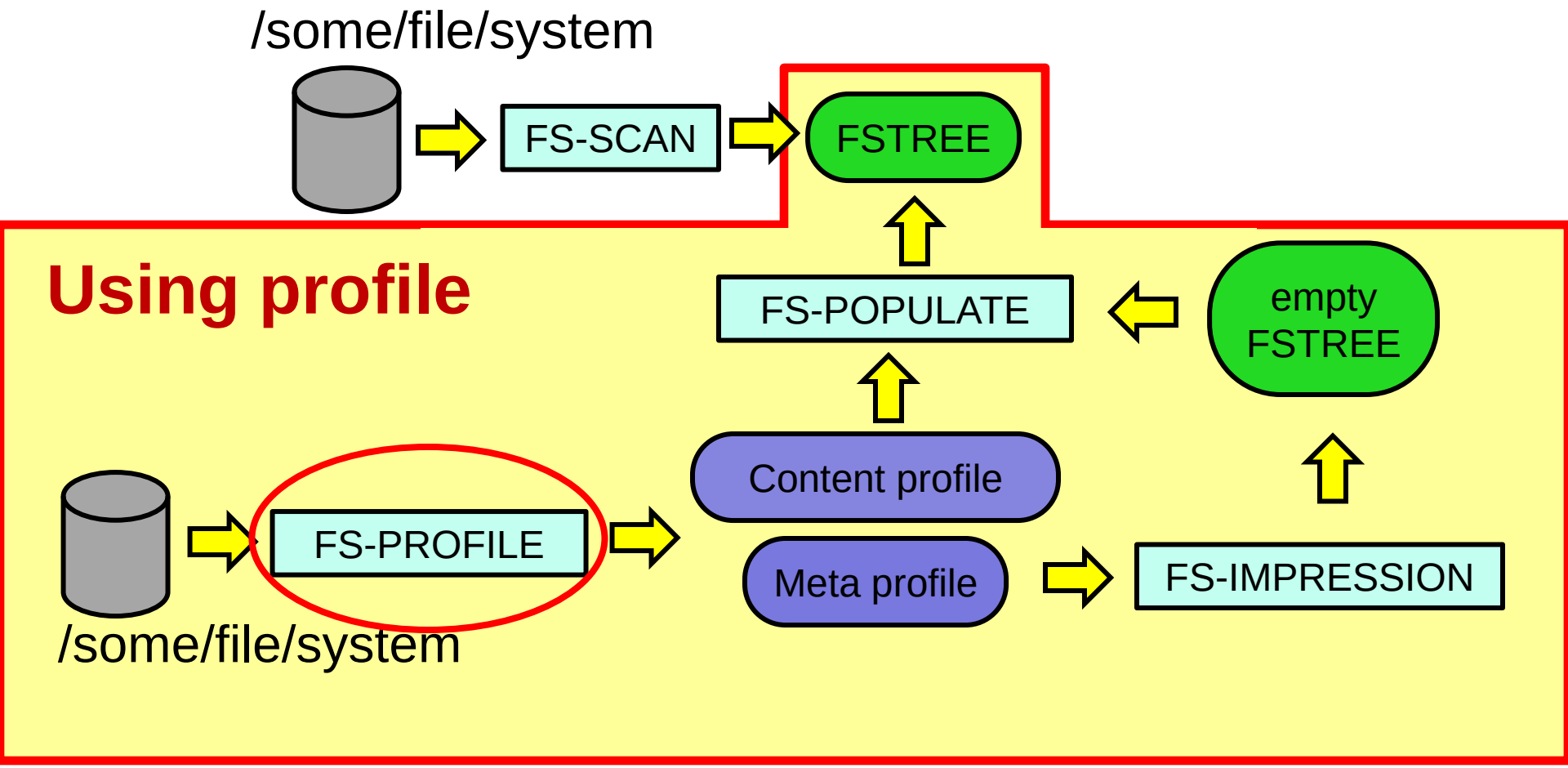
Initial File System Tree



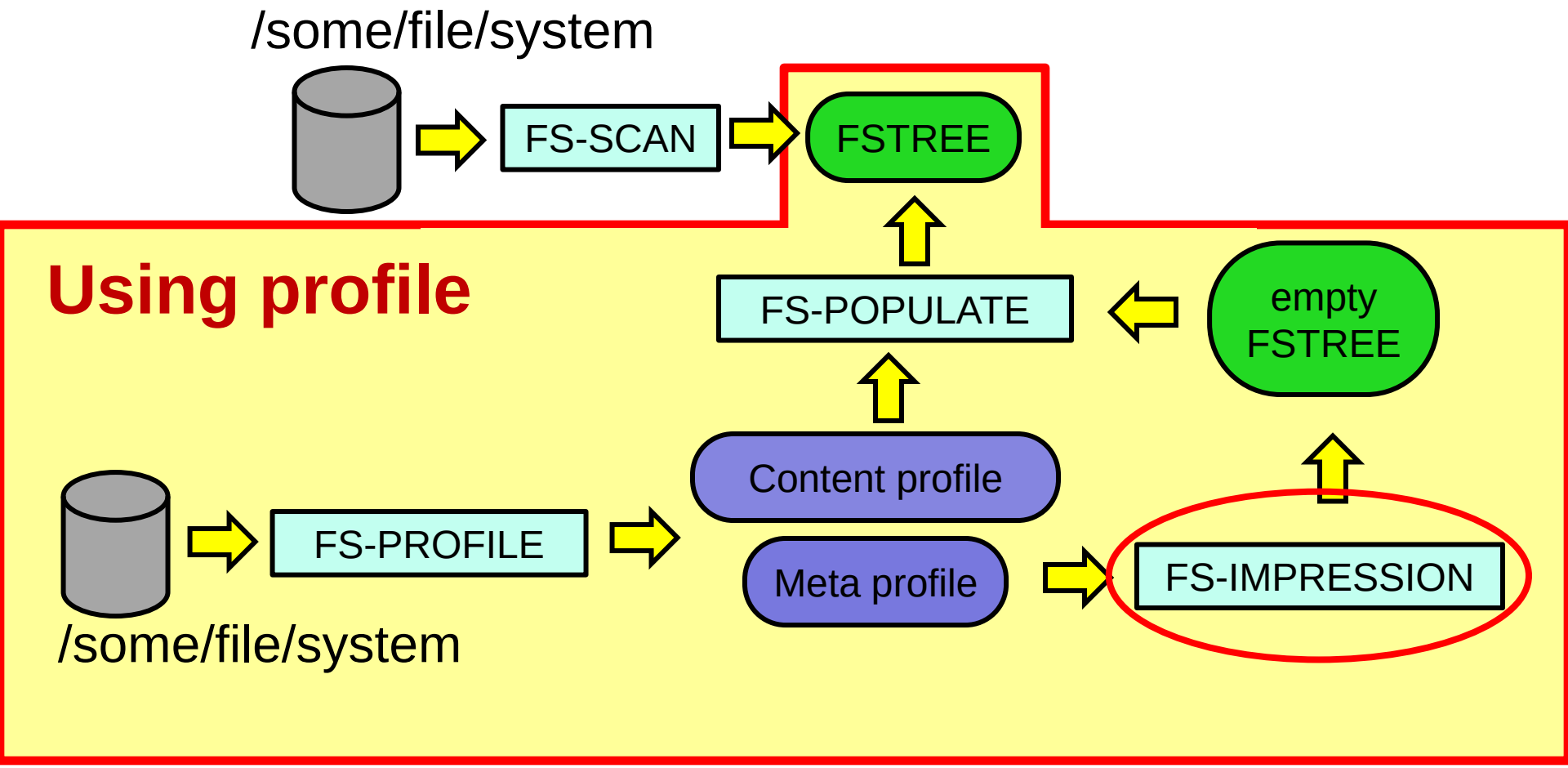
Initial File System Tree



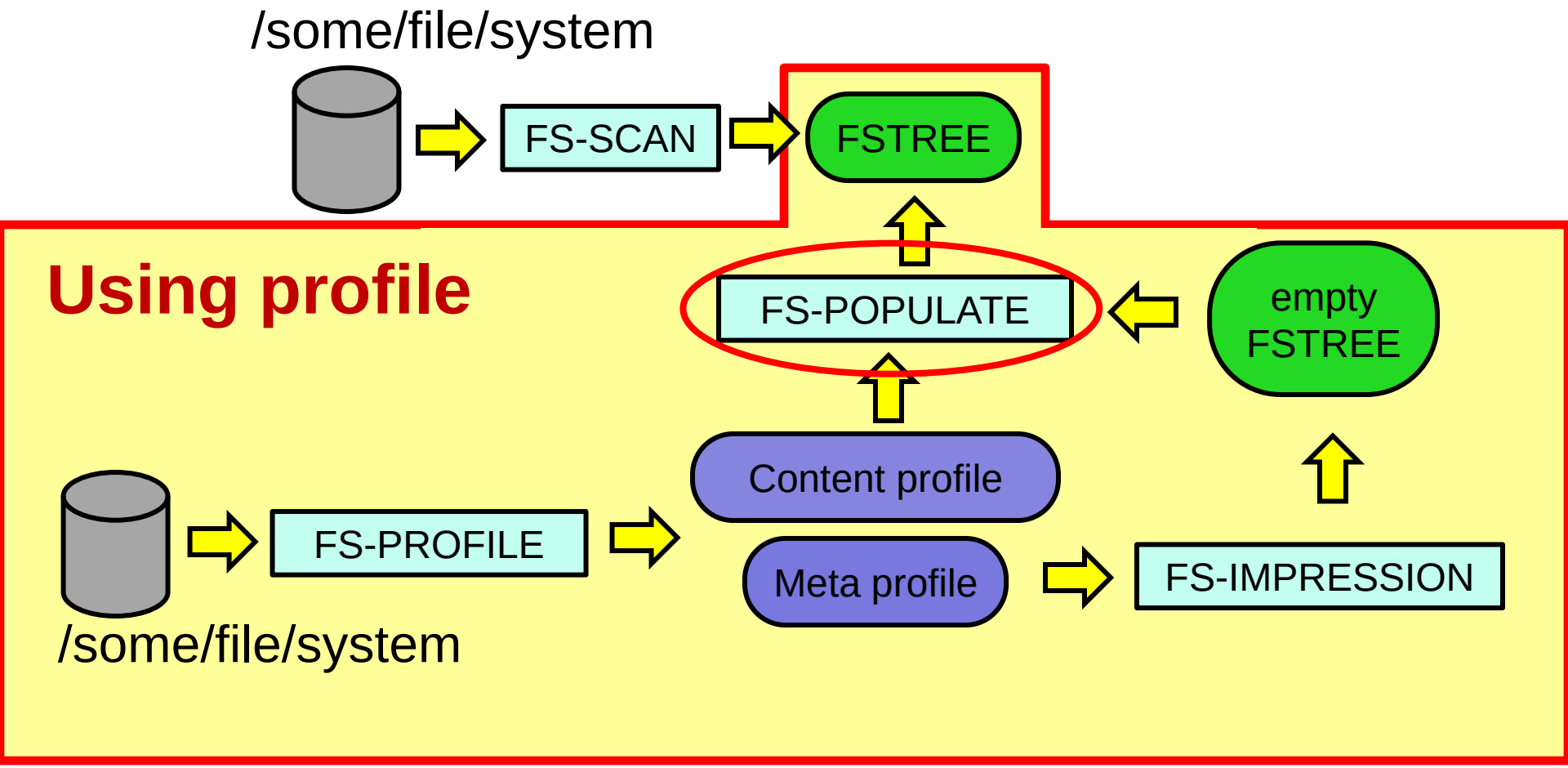
Initial File System Tree



Initial File System Tree



Initial File System Tree



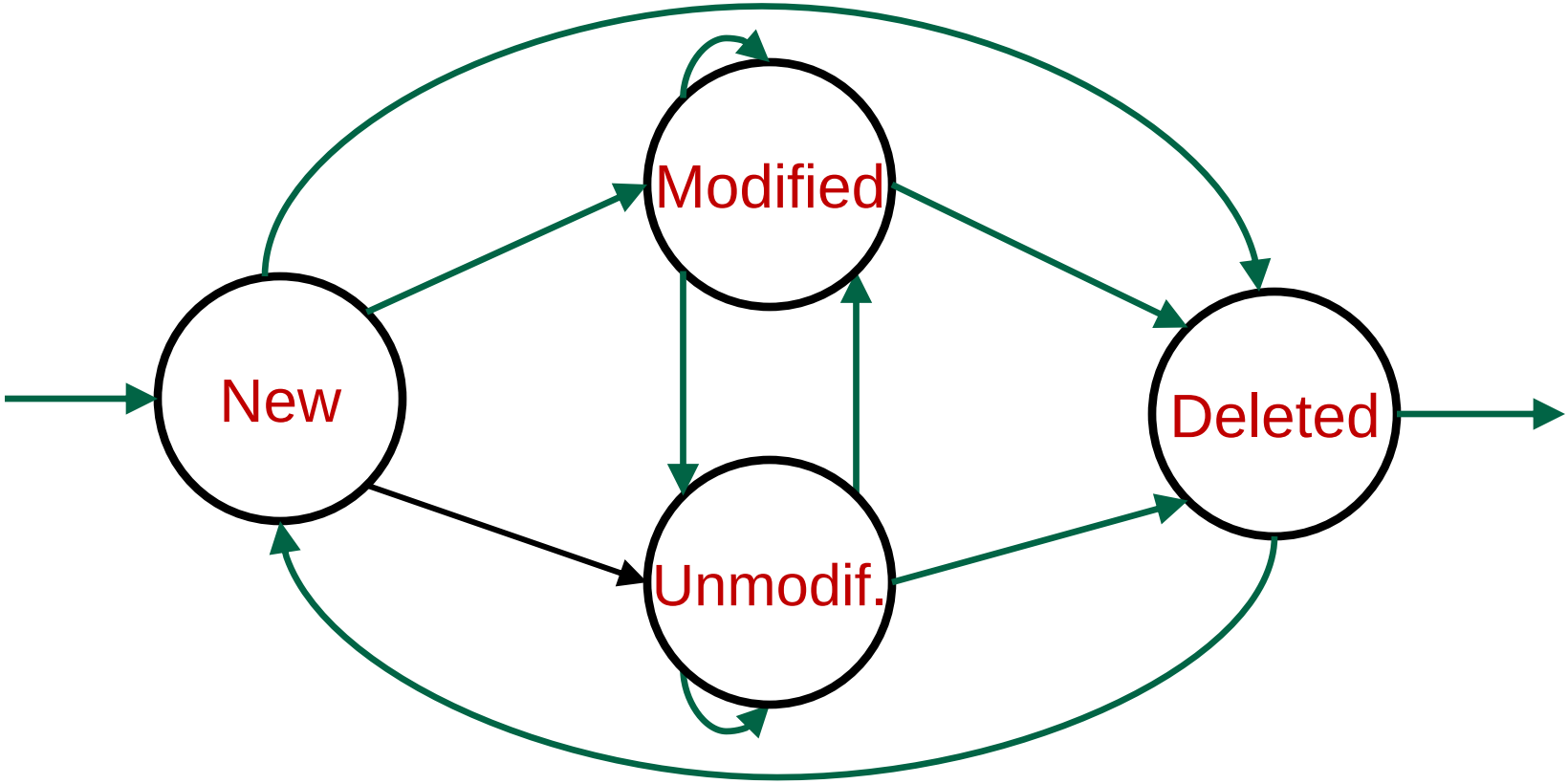
Outline

1. Deduplication and datasets survey
2. Framework design
3. **Framework implementation**
4. Evaluation
5. Conclusions & future work

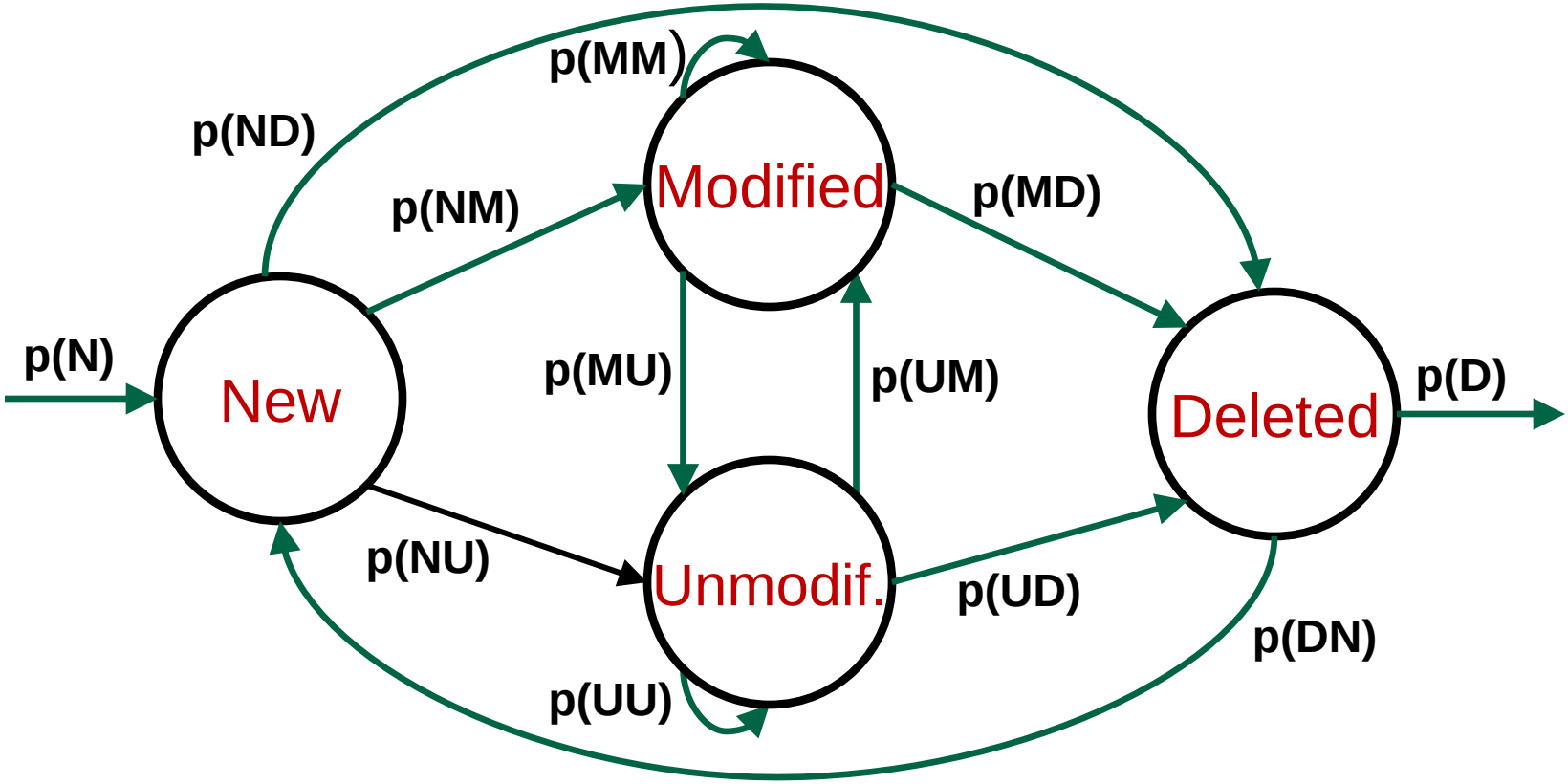
Main Idea

- **Consecutive snapshots** of real datasets
  - Scan live file systems periodically
  - Use existing backup data
  - Software or data releases (e.g., Linux kernels)
- Observe **statistics of changes**
  - Markov Model
  - Multi-dimensional distribution

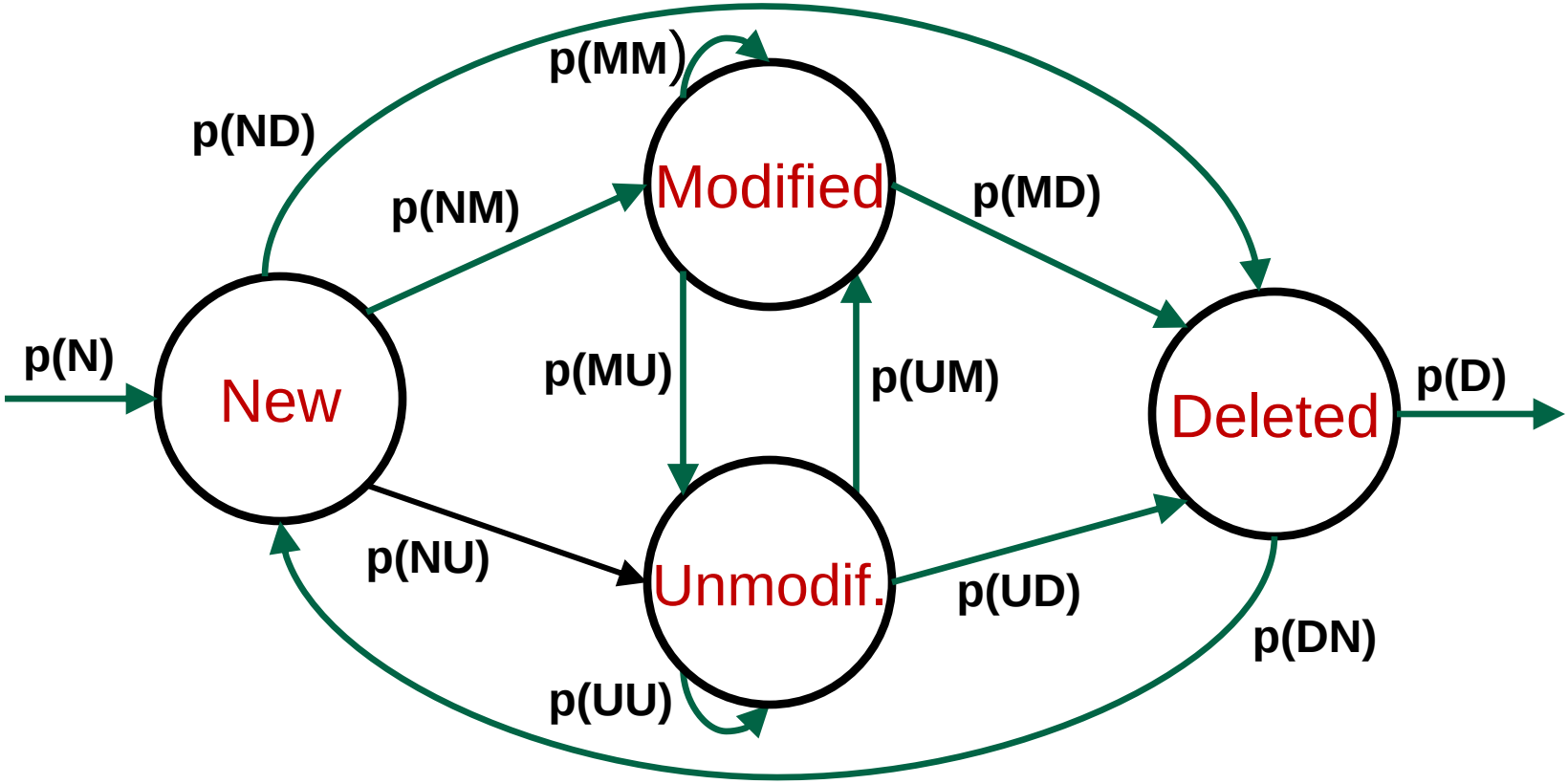
Markov Model: File States

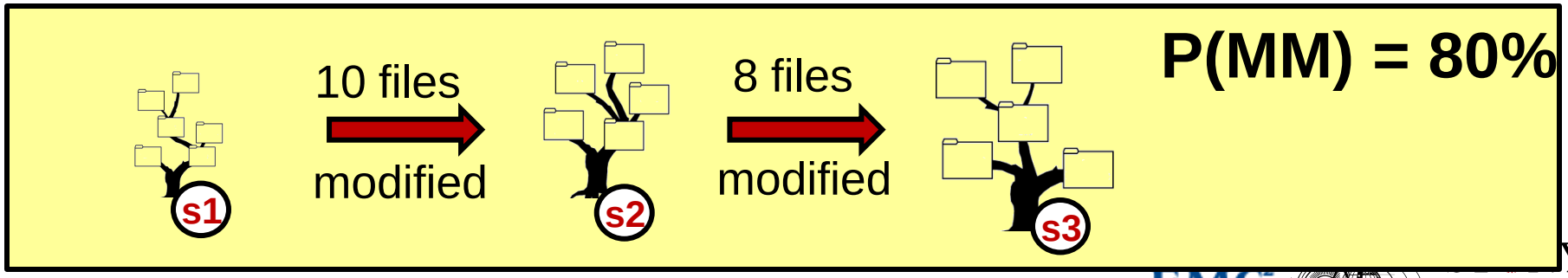


Markov Model: File States

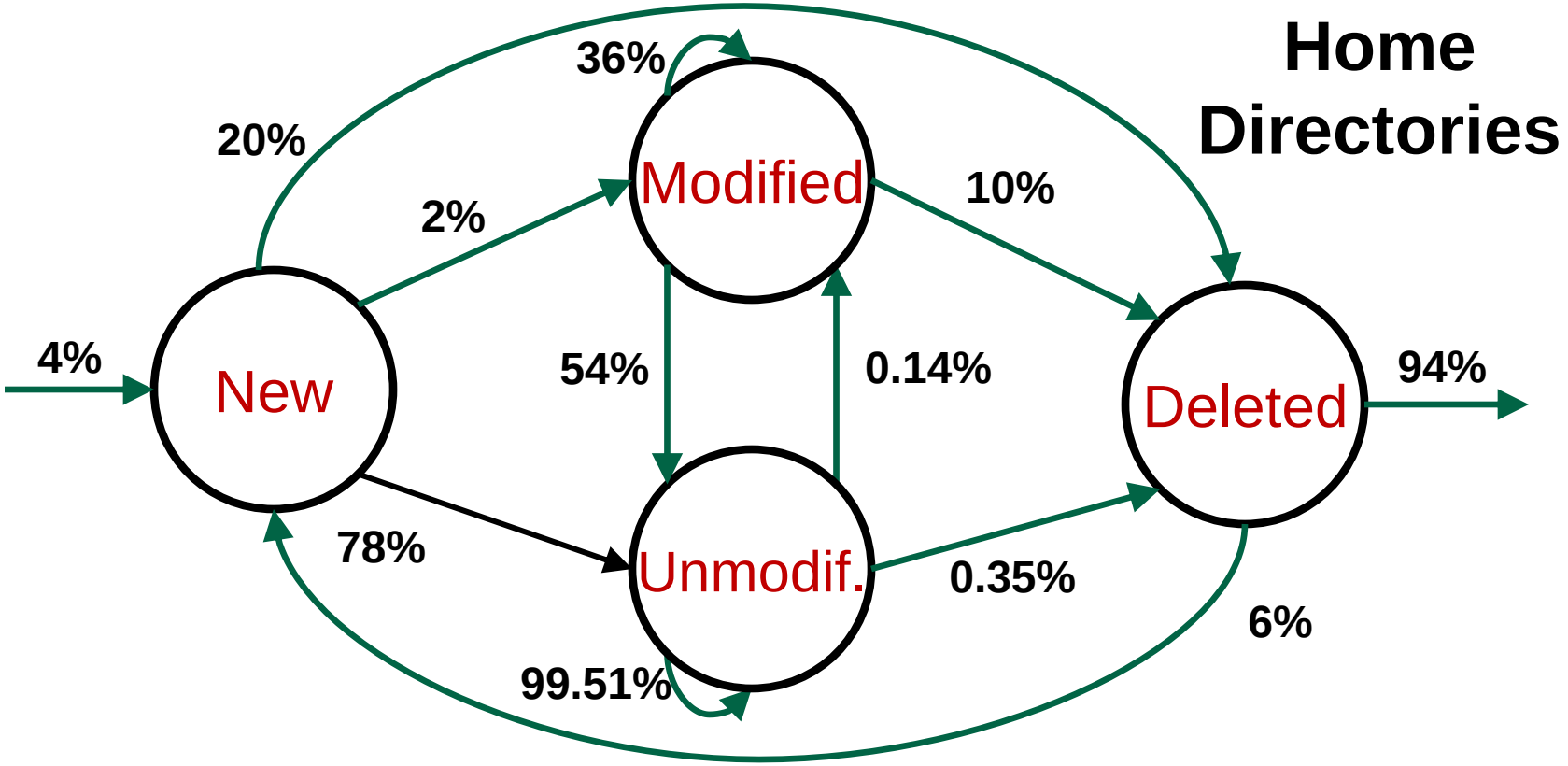


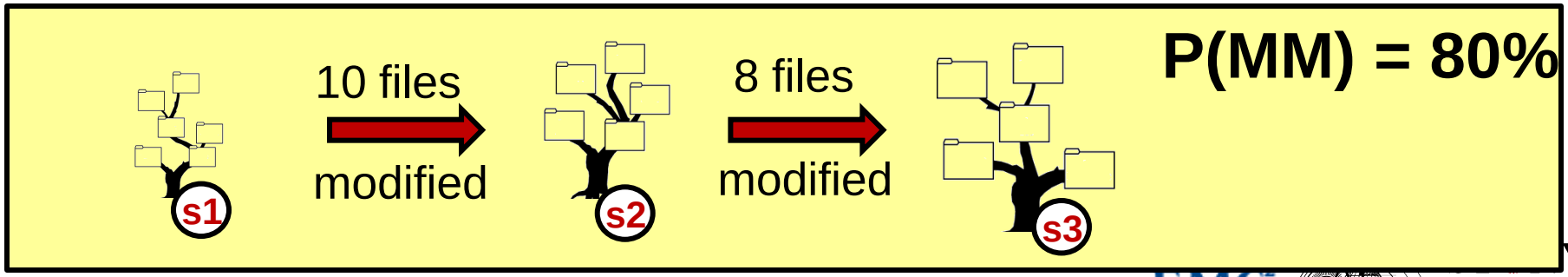
Markov Model: File States



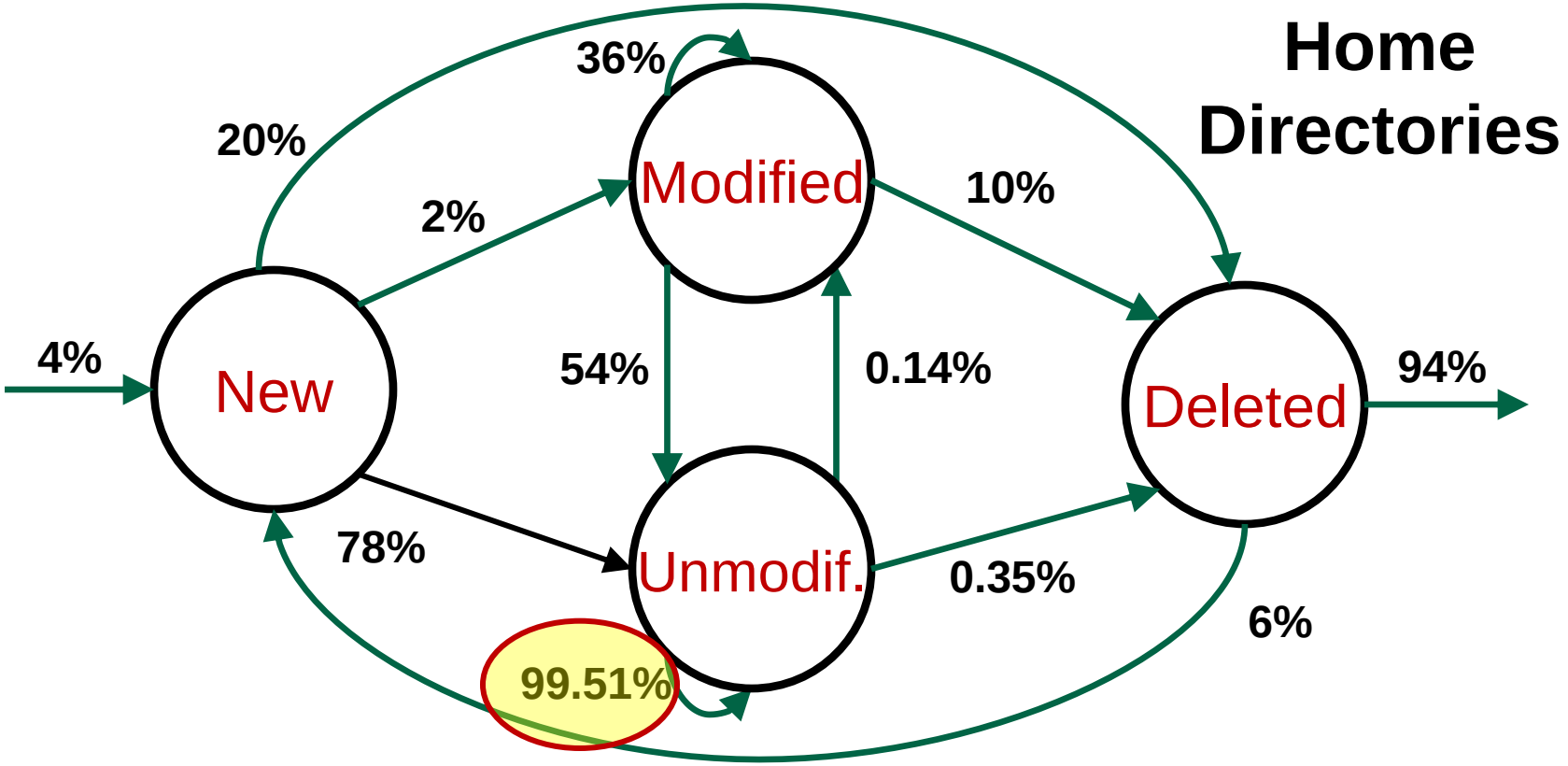


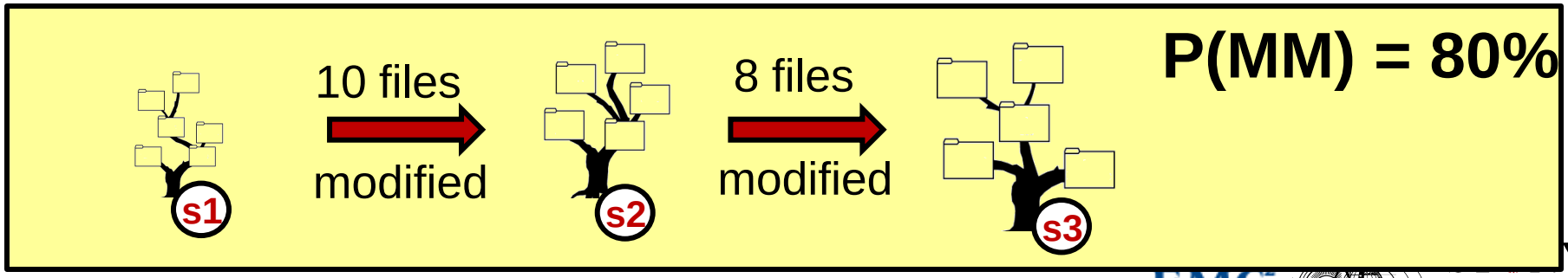
Markov Model: File States



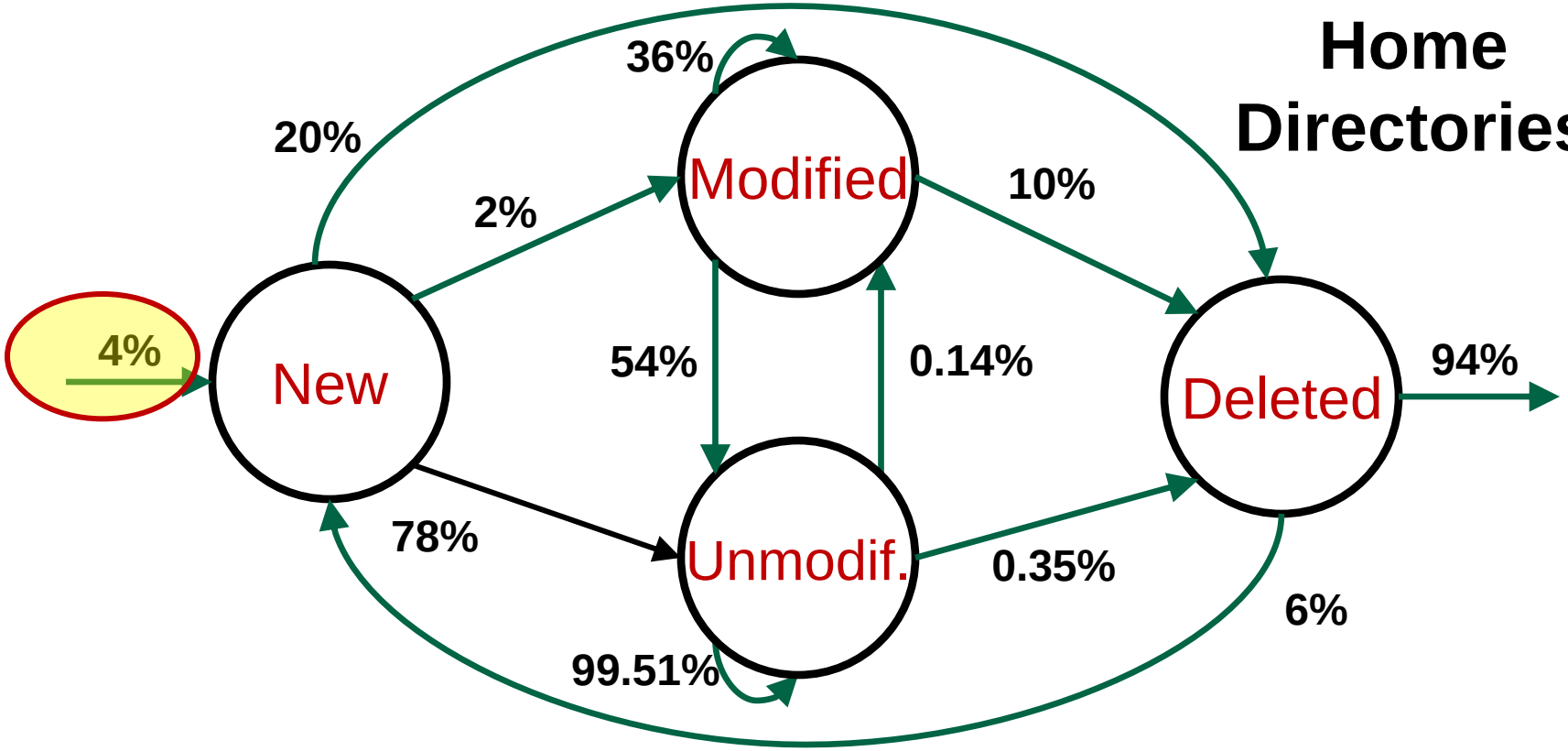


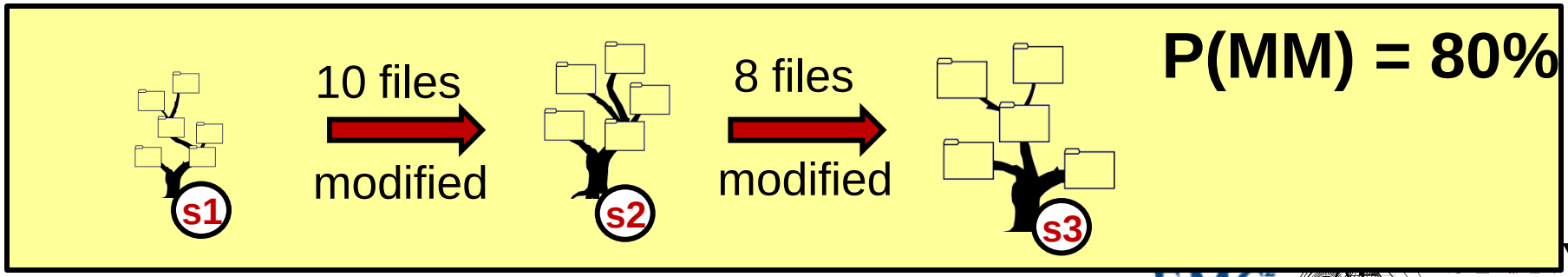
Markov Model: File States





Markov Model: File States

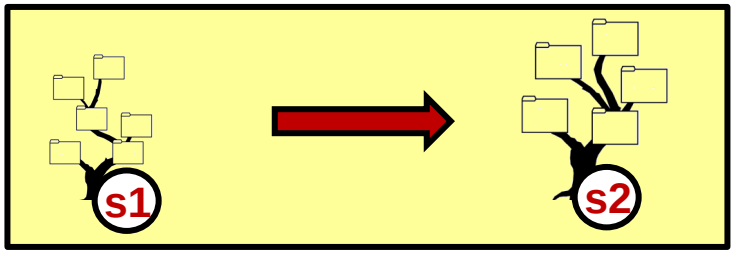




Multi-dimensional Distribution: **New** Files

Dimensions:

- Directory Depth
- File Extension
- Size (in chunks)
- Unique chunks
- Chunks with 1 duplicate
- Chunks with 2 duplicates

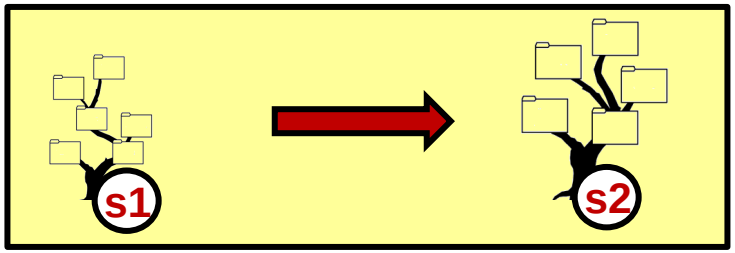


Compared to previous snapshot

Multi-dimensional Distribution: **New** Files

Dimensions:

- Directory Depth
- File Extension
- Size (in chunks)
- Unique chunks
- Chunks with 1 duplicate
- Chunks with 2 duplicates



Compared to previous snapshot

$\mathbf{M_{new}}$(depth, ext, size_chunks, uniq, dup1,dup2)
$\mathbf{M_{new}}$: number of new files with corresponding properties
Example: $\mathbf{M_{new}(2, ".c", 7, 3, 2,2) = 10}$

Outline

1. Deduplication and datasets survey
2. Framework design
3. Framework implementation
4. **Evaluation**
5. Conclusions & future work

Datasets

Name	Snapshot number	Files per snapshot (1000)	Snapshot size (GB)	Total Files (1000)	Total Size (GB)
Kernels (2.6.*)	40	23	0.3	903	13
CentOS (5.*)	8	195	4.5	1,559	36
Home	15 weekly	1,023	227	15,352	3,482
MacOS	71 daily	1,173	59	83,220	4,080
System Logs	8 weekly	334	78	2,672	626
Sources	8 weekly	139	162	1,112	1,331

Datasets

Name	Snapshot number	Files per snapshot (1000)	Snapshot size (GB)	Total Files (1000)	Total Size (GB)
Kernels (2.6.*)	40	23	0.3	903	13
CentOS (5.*)	8	195	4.5	1,559	36
Home	15 weekly	1,023	227	15,352	3,482
MacOS	71 daily	1,173	59	83,220	4,080
System Logs	8 weekly	334	78	2,672	626
Sources	8 weekly	139	162	1,112	1,331

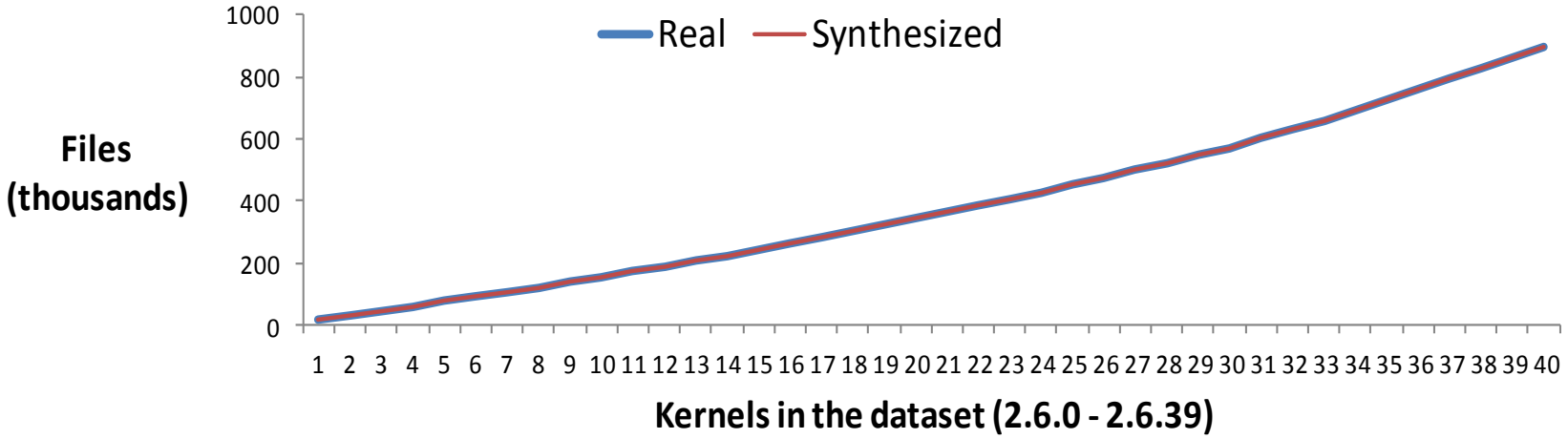
Datasets

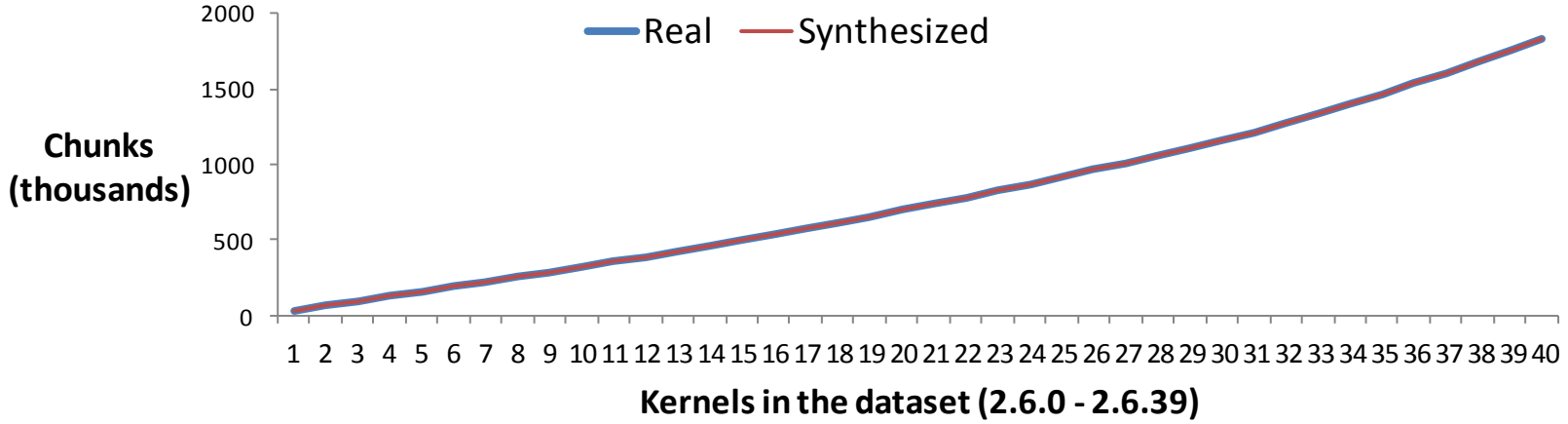
Name	Snapshot number	Files per snapshot (1000)	Snapshot size (GB)	Total Files (1000)	Total Size (GB)
Kernels (2.6.*)	40	23	0.3	903	13
CentOS (5.*)	8	195	4.5	1,559	36
Home	15 weekly	1,023	227	15,352	3,482
MacOS	71 daily	1,173	59	83,220	4,080
System Logs	8 weekly	334	78	2,672	626
Sources	8 weekly	139	162	1,112	1,331

Methodology

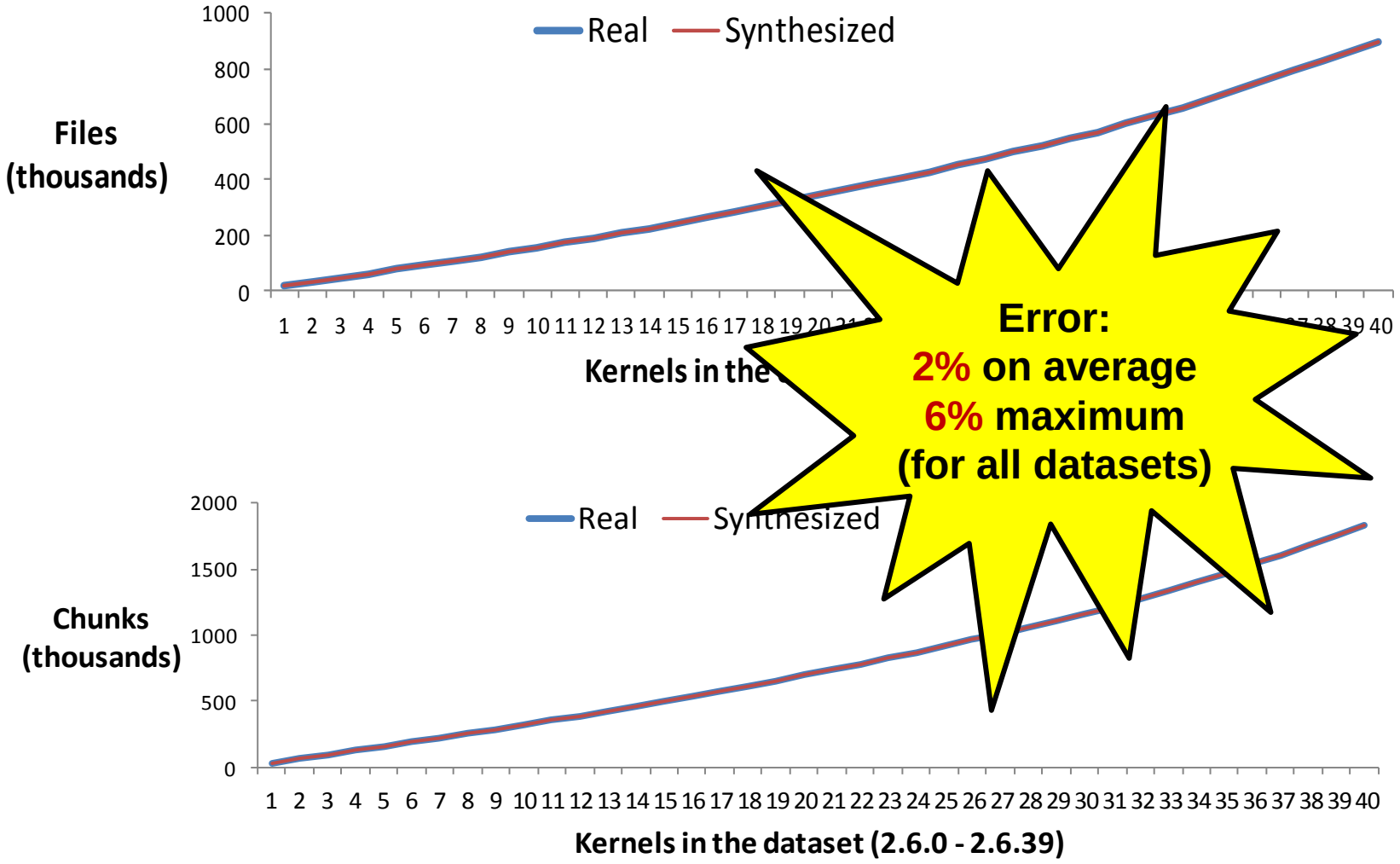
1. Scanned all snapshots
2. Created mutation profiles
3. Generated synthetic datasets with FS-MUTATE
4. Evaluated the error between emulated parameters and real parameters

File Number and Total Size



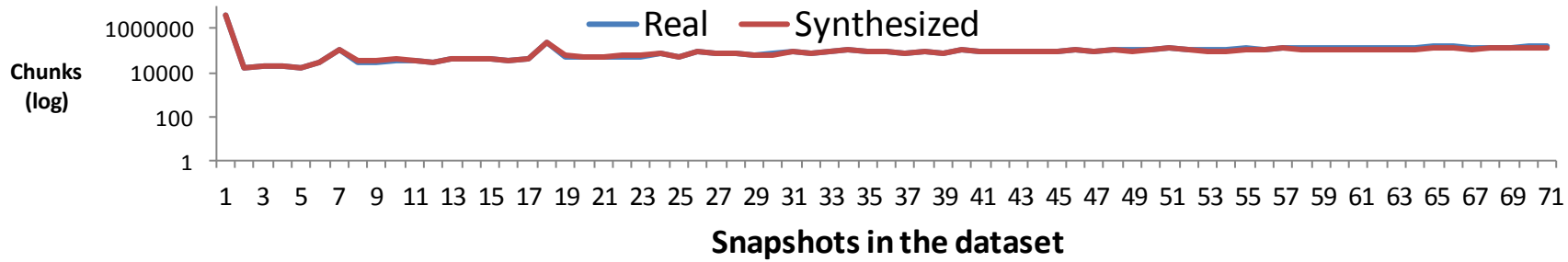


File Number and Total Size

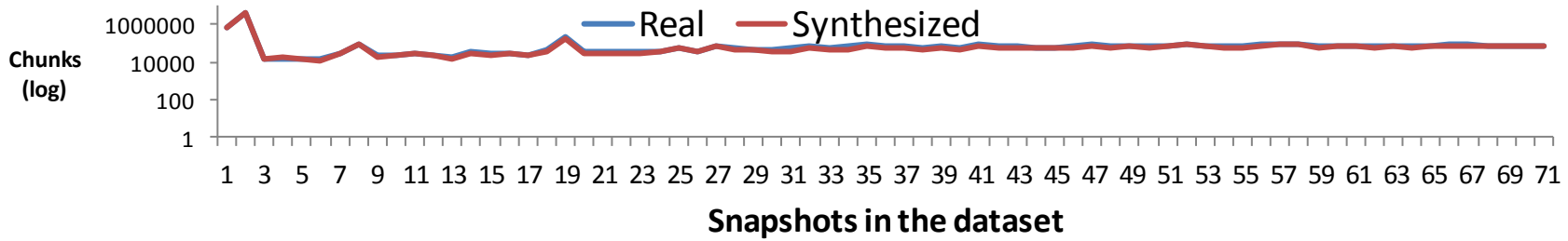


Chunk Duplicates: MacOS

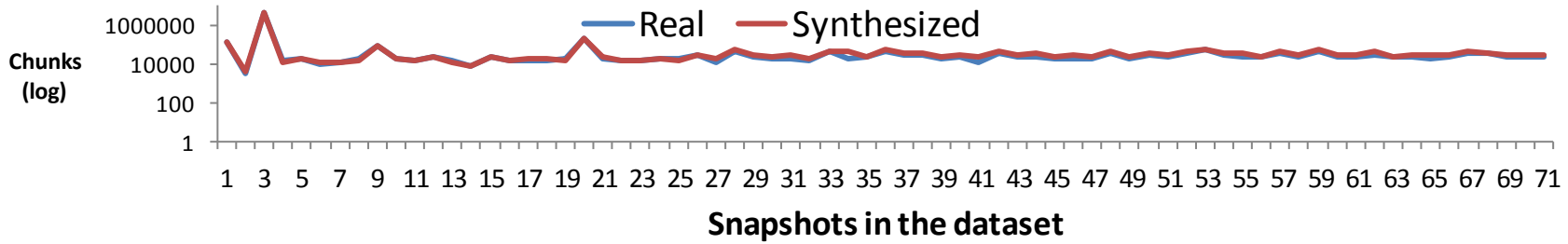
Unique chunks



Chunks with 1 duplicate

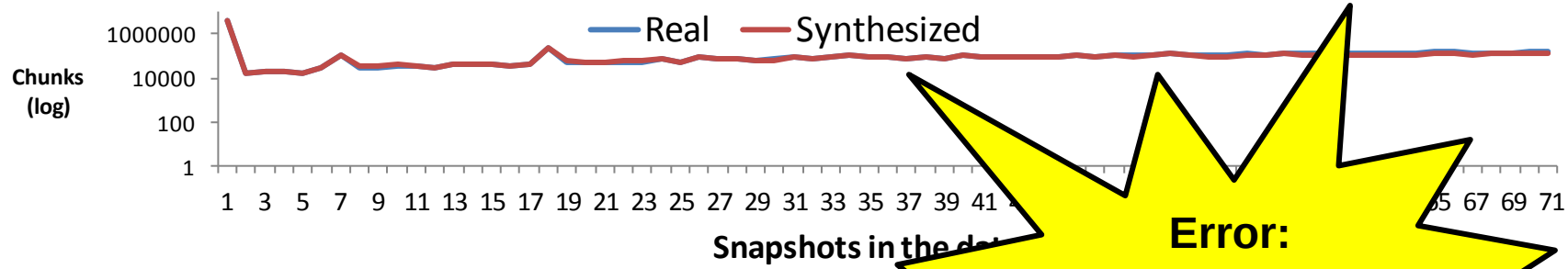


Chunks with 2 duplicates

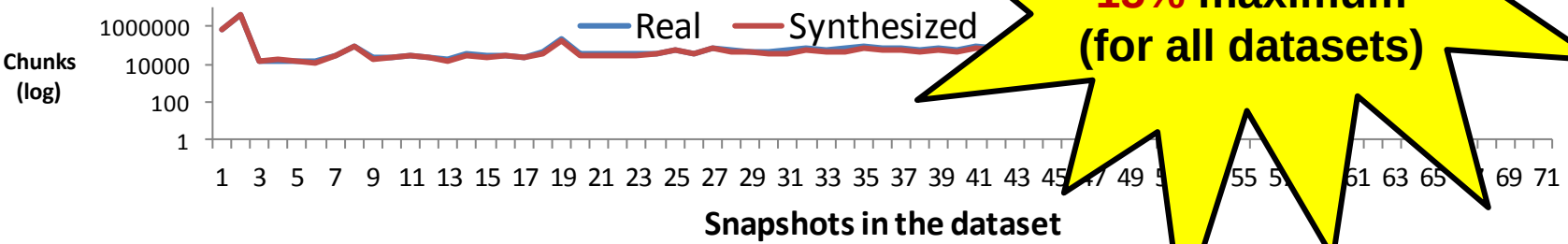


Chunk Duplicates: MacOS

Unique chunks

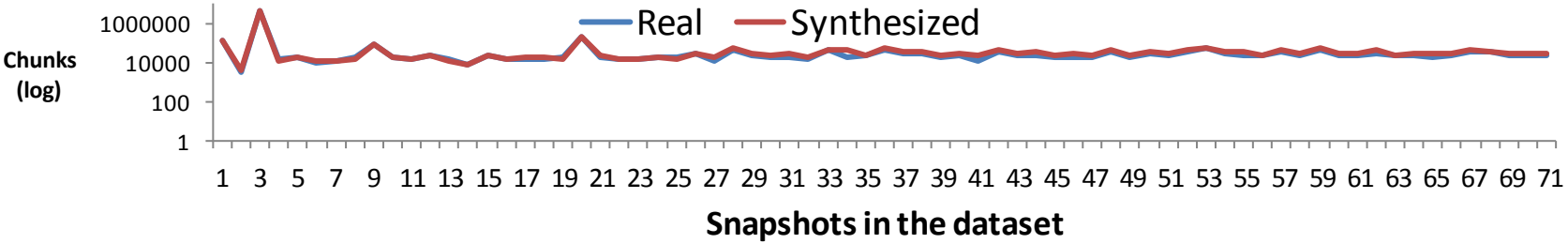


Chunks with 1 duplicate



Error:
9% on average
15% maximum
(for all datasets)

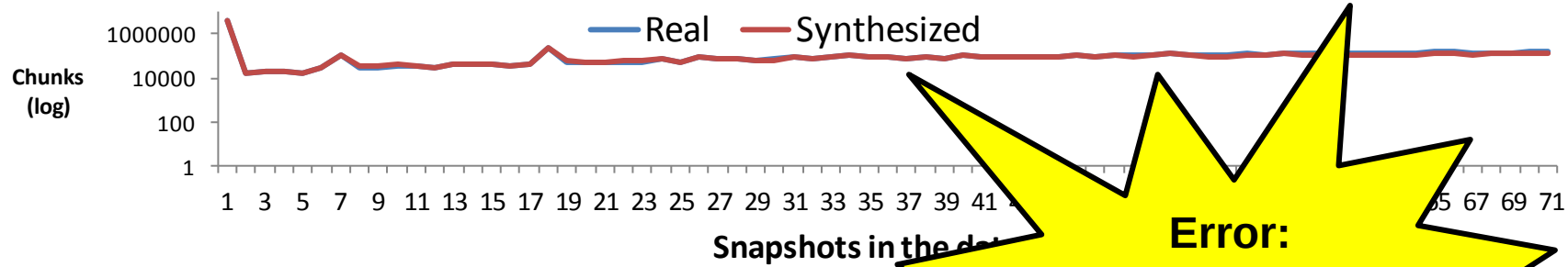
Chunks with 2 duplicates



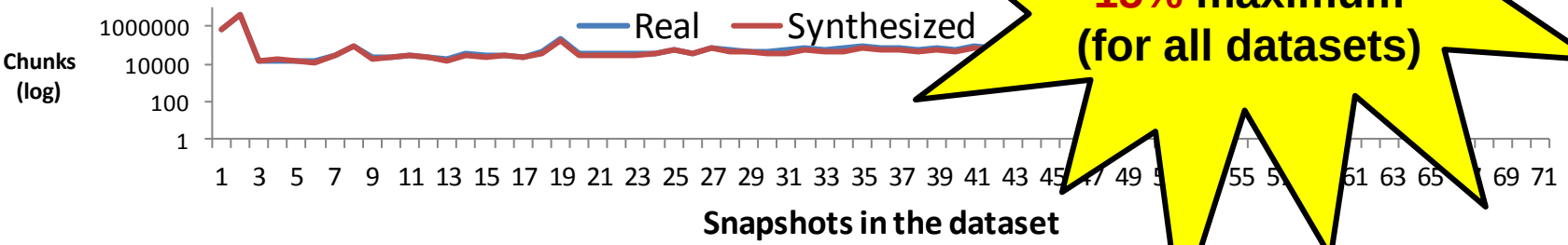
EMC² STONY BROOK UNIVERSITY

Chunk Duplicates: MacOS

Unique chunks

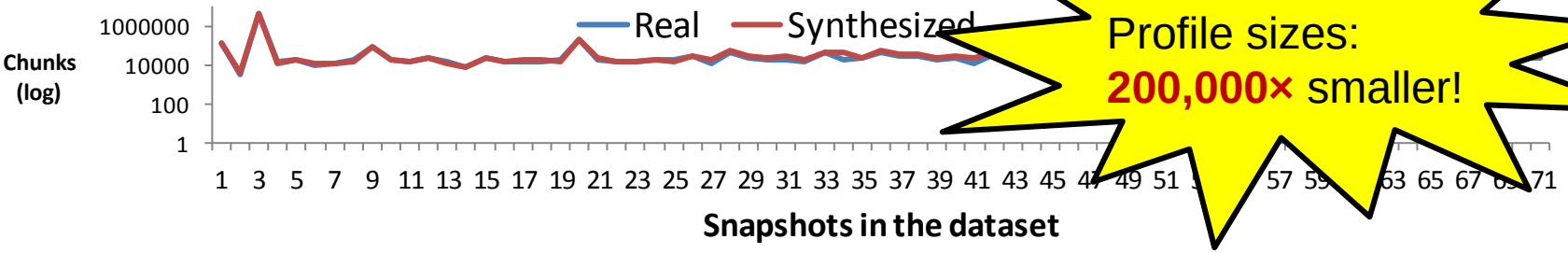


Chunks with 1 duplicate



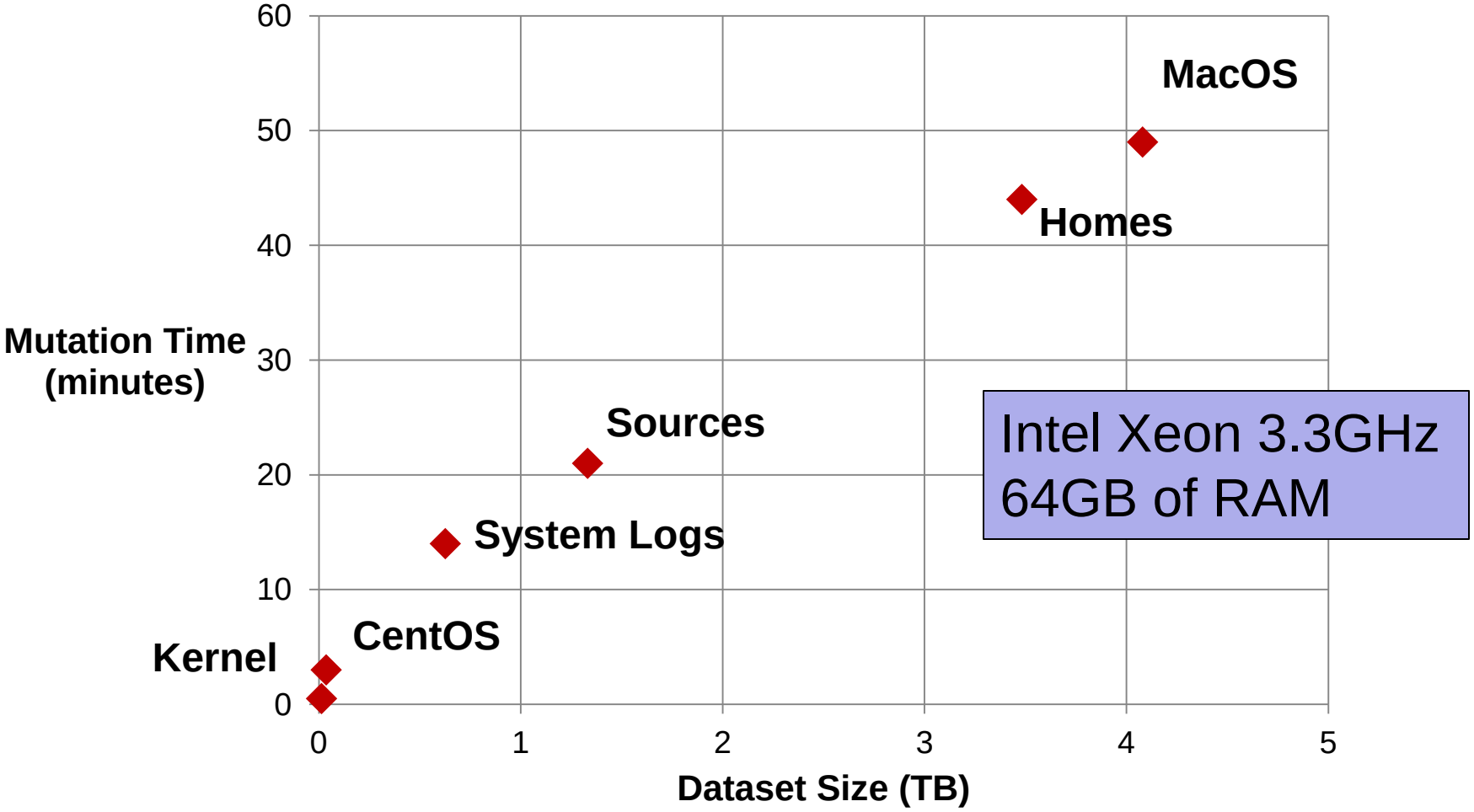
Error:
9% on average
15% maximum
(for all datasets)

Chunks with 2 duplicates



Profile sizes:
200,000× smaller!

Performance vs. Size



Outline

1. Deduplication and datasets survey
2. Framework design
3. Framework implementation
4. Evaluation
5. **Conclusions & future work**

Summary

- **First technique for generating deduplication datasets**
  - Realistic properties
  - Fair comparison of different deduplication techniques
  - Sharing among researchers
- Emulate file system changes
- Generic framework
  - Flexible, versatile, and extensible
- Statistical implementation
  - Markov Model
  - Multi-dimensional statistics
- High accuracy, small model size, high performance

Future work

- Study existing deduplication systems
- Create exhaustive list of parameters
  - E.g., local chunk compression control
- Initial file system generation
  - Convenient usage
- Detect global trend lines
  - Statistical clustering
- Analyze other datasets

Generating Realistic Datasets for Deduplication Analysis

Q&A

Download sources and profiles:

https://avatar.fsl.cs.sunysb.edu/groups/deduplicationpublic/

Vasily Tarasov[1], Amar Mudrankit[1], Will Buik[2],
Philip Shilane[3], Geoff Kuenning[2], Erez Zadok[1]
[1]Stony Brook University [2]Harvey Mudd College [3]EMC Corporation

NSF NetApp EMC² STONY BROOK UNIVERSITY