Metadata Considered Harmful ... to Deduplication

Xing Lin, Fred Dougis, Jim Li, Xudong Li, Robert Ricci, Stephen Smaldone, and Grant Wallace

University of Utah, EMC Corporation, NanKai University
Deduplication

Idea: identify duplicate data blocks and store a single copy
Deduplication

**Idea**: identify duplicate data blocks and store a single copy

<table>
<thead>
<tr>
<th>Chunks</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chunks</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F'</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deduplication

Idea: identify duplicate data blocks and store a single copy

Chunks:

v1.0

A B C D E F G H

v2.0

A B C D E F’ G H

Stored on disk:

A B C D E F G H F’
Deduplication

**Idea:** identify duplicate data blocks and store a single copy

```
Chunks  
A | B | C | D | E | F | G | H 

v1.0

A | B | C | D | E | F | G | H |

v2.0

F'

Stored on disk

A | B | C | D | E | F | G | H | F'

Eliminate nearly all of v2.0 on disk
Expected Deduplication

What if we have many versions?

v1.0

v2.0
What if we have many versions?

Dedup ratio = \frac{\text{original size}}{\text{post_dedup size}}

- v1.0
- v2.0

\sim 2 \times
Expected Deduplication

What if we have many versions?
Dedup ratio = original_size/post_dedup_size

v1.0
Input

v2.0
Input

v3.0
Input

~2 x
~3 x
Expected Deduplication

What if we have many versions?
Dedup ratio = original_size/post_dedup_size

v1.0
Input

v2.0
Input

v3.0
Input

...
Great (Deduplication) Expectations

• In Reality
  – 308 versions of Linux source code: 2 x
  – Other examples of awful deduplication
Great (Deduplication) Expectations

• **In Reality**
  – **308** versions of Linux source code: \(2 \times\)
  – Other examples of awful deduplication

• **Contributions:**
  – Identify and categorize barriers to deduplication
  – Solutions
    • EMC Data Domain (industrial experience)
    • GNU tar (academic research)
Metadata Impacts Deduplication
Metadata Impacts Deduplication

• Case 1: metadata changes
  – the input is an aggregate of many small user files, interleaved with file metadata
  – Metadata: file path, timestamps, etc.
Metadata Impacts Deduplication

• Case 1: metadata changes
  – the input is an aggregate of many small user files, interleaved with file metadata
  – Metadata: file path, timestamps, etc.
Metadata Impacts Deduplication

• Case 1: metadata changes
  – the input is an aggregate of many small user files, interleaved with file metadata
  – Metadata: file path, timestamps, etc.
Metadata Impacts Deduplication

• Case 1: metadata changes
  – the input is an aggregate of many small user files, interleaved with file metadata
  – Metadata: file path, timestamps, etc.
Metadata Impacts Deduplication

• Case 1: metadata changes
  – the input is an aggregate of many small user files, interleaved with file metadata
  – Metadata: file path, timestamps, etc.
  – GNU tar, EMC NetWorker, Oracle RMAN backups
Metadata Impacts Deduplication

• Case 1: metadata changes
  – the input is an aggregate of many small user files, interleaved with file metadata
  – Metadata: file path, timestamps, etc.
  – GNU tar, EMC NetWorker, Oracle RMAN backups
  – Videos suffer from a similar problem (next talk)
Metadata Impacts Deduplication

• Case 2: metadata location changes
  – The input is encoded in (fixed-size) blocks and metadata is inserted for each block
Metadata Impacts Deduplication

• Case 2: metadata location changes
  – The input is encoded in (fixed-size) blocks and metadata is inserted for each block
  – Data insertion/deletion lead to metadata shifts
  – Tape format
Metadata Impacts Deduplication

• Case 2: metadata location changes
  – The input is encoded in (fixed-size) blocks and metadata is inserted for each block
  – Data insertion/deletion lead to metadata shifts
  – Tape format
Case 2: metadata location changes

- The input is encoded in (fixed-size) blocks and metadata is inserted for each block
- Data insertion/deletion lead to metadata shifts
- Tape format
Metadata Impacts Deduplication

• Case 2: metadata location changes
  – The input is encoded in (fixed-size) blocks and metadata is inserted for each block
  – Data insertion/deletion lead to metadata shifts
  – Tape format

v1.0

v2.0
Case 2: metadata location changes
- The input is encoded in (fixed-size) blocks and metadata is inserted for each block
- Data insertion/deletion lead to metadata shifts
- Tape format

![Diagram showing metadata impact on deduplication]

v1.0 → v2.0
Metadata Impacts Deduplication

• Case 2: metadata location changes
  – The input is encoded in (fixed-size) blocks and metadata is inserted for each block
  – Data insertion/deletion lead to metadata shifts
  – Tape format

![Diagram of chunk changes from v1.0 to v2.0]
• Case 2: metadata location changes
  – The input is encoded in (fixed-size) blocks and metadata is inserted for each block
  – Data insertion/deletion lead to metadata shifts
  – Tape format

Locations of block markers are shifted, leading to different chunks
Solution: Separate Metadata from Data
Solution: Separate Metadata from Data

• Three approaches:
  – *Recommended*: design deduplication-friendly formats
    • Case study: EMC NetWorker
Solution: Separate Metadata from Data

• Three approaches:
  – *Recommended*: design deduplication-friendly formats
    • Case study: EMC NetWorker
  – *Transparent*: application-level post-processing
    • Case study: GNU tar
Solution: Separate Metadata from Data

• Three approaches:
  – **Recommended**: design deduplication-friendly formats
    • Case study: EMC NetWorker
  – **Transparent**: application-level post-processing
    • Case study: GNU tar
  – **Last resort**: format-aware deduplication
    • Case studies: 1) virtual tape libraries (VTL)
    2) Oracle RMAN backups
Solution: Separate Metadata from Data

• Three approaches:
  – *Recommended*: design deduplication-friendly formats
    • Case study: EMC NetWorker
  – *Transparent*: application-level post-processing
    • Case study: GNU tar
  – *Last resort*: format-aware deduplication
    • Case studies: 1) virtual tape libraries (VTL)
      2) Oracle RMAN backups
Application-level Post-processing

• tar (tape archive)
  – Collects files into one archive file
  – File system archiving, source code distribution, ...

• GNU tar format
  – A sequence of entries, one per file, each containing a file header and data blocks
  – Header block: file path, size, modification time
Application-level Post-processing

• tar (tape archive)
  – Collects files into one archive file
  – File system archiving, source code distribution, ...

• GNU tar format
  – A sequence of entries, one per file, each containing a file header and data blocks
  – Header block: file path, size, modification time
Application-level Post-processing

- **tar (tape archive)**
  - Collects files into one archive file
  - File system archiving, source code distribution, ...

- **GNU tar format**
  - A sequence of entries, one per file, each containing a file header and data blocks
  - Header block: file path, size, modification time
Metadata Changes with GNU tar

SHA1s

<table>
<thead>
<tr>
<th>Directory</th>
<th>SHA1</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux-2.6.39.3/README</td>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
</tr>
<tr>
<td>linux-2.6.39.4/README</td>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
</tr>
</tbody>
</table>
## Metadata Changes with GNU tar

<table>
<thead>
<tr>
<th>SHA1s</th>
<th>Linux-2.6.39.3/README</th>
<th>Linux-2.6.39.4/README</th>
</tr>
</thead>
<tbody>
<tr>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
<td></td>
</tr>
</tbody>
</table>
Metadata Changes with GNU tar

<table>
<thead>
<tr>
<th>SHA1s</th>
<th>Version 1</th>
<th>Version 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux-2.6.39.3/README</td>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
<td></td>
</tr>
<tr>
<td>linux-2.6.39.4/README</td>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
<td></td>
</tr>
</tbody>
</table>

- rw-rw-r-- 1 root root 17525 Jul 9 2011
- rw-rw-r-- 1 root root 17525 Aug 3 2011

File 1

File 2

Header block

Data blocks

Modified Header block
Metadata Changes with GNU tar

**SHA1s**

- `linux-2.6.39.3/README`  
  - SHA1: `a735c31cef6d19d56de6824131527fdce04ead47`

- `linux-2.6.39.4/README`  
  - SHA1: `a735c31cef6d19d56de6824131527fdce04ead47`

**Files**

- `linux-2.6.39.3/README`  
  - Permissions: `-rw-rw-r--`
  - Size: `17525`
  - Time: `Jul 9 2011`

- `linux-2.6.39.4/README`  
  - Permissions: `-rw-rw-r--`
  - Size: `17525`
  - Time: `Aug 3 2011`

**Diagram**

- **Version 1**
  - File 1
    - Chunk1
    - Chunk2
    - Chunk3
  - File 2
    - Chunk1
    - Chunk2
    - Chunk3

- **Version 2**
  - File 1
    - Chunk1
    - Chunk2
    - Chunk3
  - File 2
    - Chunk1
    - Chunk2
    - Chunk3

**Notes**

- Chunk1 and Chunk2 in version 2 are different.

**Modified Header block**
Metadata Changes with GNU tar

SHA1s

<table>
<thead>
<tr>
<th>File Path</th>
<th>SHA1</th>
</tr>
</thead>
<tbody>
<tr>
<td>linux-2.6.39.3/README</td>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
</tr>
<tr>
<td>linux-2.6.39.4/README</td>
<td>a735c31cef6d19d56de6824131527fdce04ead47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File Path</th>
<th>Permissions</th>
<th>Owner</th>
<th>Size</th>
<th>Date</th>
</tr>
</thead>
</table>

Version 1

- File 1
  - Chunk1
  - Chunk2
  - Chunk3

Version 2

- File 1
  - Modified Header block
  - Data blocks
  - Chunk1 and Chunk2 in version 2 are different.

- File 2
  - Header block
  - Data blocks

2x for 308 versions of Linux
Migratory tar (mtar)

tar

File 1

File 2

Header block

Data blocks
Migratory tar (mtar)

**tar**

- File 1
- File 2
- Header block
- Data blocks

**Number of header blocks**: 2

**mtar**

Migrate
Migratory tar (mtar)

```
Migrate

Restore

File 1

File 2

Header block

Data blocks

Number of header blocks

2

File 1

File 2

Migrate

Restore

Number of header blocks

2

mtar
```
mtar - Evaluation

• 9 GNU software and Linux kernel source code
• Many versions: 13 ~ 308
mtar - Evaluation

• 9 GNU software and Linux kernel source code
• Many versions: 13 ~ 308
mtar - Evaluation

- 9 GNU software and Linux kernel source code
- Many versions: 13 ~ 308

Improvements:
1. across all datasets
2. huge: 1.5-5.3×
Virtual Tape Library (VTL)

- VTL: a disk-based storage system to emulate a tape library
- Tape recorder: write in fixed-size blocks and attach a block marker for each block
  - Block marker: which device and the timestamp a file is written
Virtual Tape Library (VTL)

- VTL: a disk-based storage system to emulate a tape library
- Tape recorder: write in fixed-size blocks and attach a block marker for each block
  - Block marker: which device and the timestamp a file is written
- Data insertion/deletion leads to marker shifts.
Format-aware Deduplication

• Modify deduplication systems to be aware of file formats; extract metadata and store it separately
Format-aware Deduplication

- Modify deduplication systems to be aware of file formats; extract metadata and store it separately

Input received
Format-aware Deduplication

• Modify deduplication systems to be aware of file formats; extract metadata and store it separately.

Input received

When storing, remove markers

Input stored
VTL Evaluation

75% improvement
More in the Paper

• **Design deduplication-friendly formats**
  – **Case study: EMC NetWorker**

• Application-level post-processing
  – Case study: GNU tar

• Format-aware Deduplication
  – Case studies: 1) VTL
  
  2) Oracle RMAN backups

• mtar available at [https://github.com/xinglin/mtar](https://github.com/xinglin/mtar)
Conclusion

• **Metadata impacts deduplication**
  – Metadata changes more frequently, introducing many *unnecessary* unique chunks

• Solution: separate metadata from data
  – Up to 5× improvements in deduplication
Conclusion

• **Metadata impacts deduplication**
  – Metadata changes more frequently, introducing many *unnecessary* unique chunks

• Solution: separate metadata from data
  – Up to 5× improvements in deduplication

*Metadata eats away your dedupe.*
Backup slides
Data vs. Metadata vs. File Content-only

Deduplication Ratio (X)

Datasets

Metadata block
Datablock
Concatenated

amake bash coreu fdisk gcc gdb glibc stalk tar linux
Metadata is STILL There!!!
Metadata is STILL There!!!

Before-dedup
4.39%
95.61%

After-dedup
4.39%
5.75%

We need better solutions to store metadata!
### mtar - Evaluation

<table>
<thead>
<tr>
<th>Software</th>
<th>Versions</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>automake</td>
<td>64</td>
<td>304.72</td>
</tr>
<tr>
<td>bash</td>
<td>23</td>
<td>276.69</td>
</tr>
<tr>
<td>coreutils</td>
<td>37</td>
<td>1284.49</td>
</tr>
<tr>
<td>fdisk</td>
<td>13</td>
<td>21.61</td>
</tr>
<tr>
<td>gcc</td>
<td>68</td>
<td>20315.45</td>
</tr>
<tr>
<td>gdb</td>
<td>32</td>
<td>4004.77</td>
</tr>
<tr>
<td>glibc</td>
<td>43</td>
<td>3811.48</td>
</tr>
<tr>
<td>smalltalk</td>
<td>33</td>
<td>685.39</td>
</tr>
<tr>
<td>tar</td>
<td>21</td>
<td>219.86</td>
</tr>
<tr>
<td>linux</td>
<td>308</td>
<td>98444.58</td>
</tr>
</tbody>
</table>

**Dataset**
Networker Evaluation

![Graph showing deduplication ratio improvements for different datasets. The x-axis represents datasets, and the y-axis represents the deduplication ratio (X). There are bars for original and CDF improvements, with some datasets marked as N.N.]