Chunk Fragmentation Level
An Effective Indicator for Read Performance Degradation in Deduplication Storage

Youngjin Nam, Guanlin Lu, Nohhyun Park
Weijun Xiao, David H. C. Du

University of Minnesota
http://cris.cs.umn.edu
Talk Outline

• **Background & Motivation**
• Our goal & Solution
• Performance analysis
• Summary & Future work
Digital Data Growth @ Worldwide

• “Digital Universe” from 2009 to 2020
  – data: 44 times 800K PB(’09) → 35,200K PB(’20)
  – # files: 67 times, storage capacity: 30 times

(source: iView: The Digital Universe Decade – Are You Ready?, IDC 2010.5)
Good News is ...

- 75% of digital data are copies!
  - 25% are unique!
  - high chance to reduce the storage demand/supply gap!
  - but, regulations for multiple copies (reliability/availability)

- Data de-duplication with some challenges
  - mainly applied in 2nd-tier storage (archive, backup..)
Background

Deduplication Storage Overview

- Data stream, dedupe appliance, underlying storage
- Dedupe appliance: chunking + deduplication
Background

How Deduplication Works?

• Dividing data(object) into (variable/fixed-sized) small chunks & computing hash (SHA-1) for each chunk
Background

How Deduplication Works?

• If a chunk has a copy (already-existing chunk) by looking up hash index, eliminate it (store its pointer)
• Otherwise, store the new (unique) chunks into storage
Background

Beauty of Deduplication

- Reuse of the already-existing chunks as many as possible, in order to
  - reduce (backup) storage capacity usage
  - increase write I/O performance by reducing actual write I/O requests to underlying physical storage
Motivation

Degraded Read Performance

- More chunks are deduplicated, read performance decreases (200MB/s → 140MB/s, 30%↓) [Zhu08]
  - reason: original write sequence (sequential write) is fragmented by eliminating the duplicate chunks
Motivation

Why Read Performance Matters?

- Rebuild performance in secondary storage
  - critical with the secondary storage [Zhu08]
  - recovery window time & system availability with ever-growing data

- Long-term digital preservation requirements
  - SNIA DPCO(snia.org/forums/dpco)
  - LTDP Reference Model (http://www.ltdprm.org)

- Dedupe gets used for primary storage
  - read IOs will be as many as write IOs
  - eg: storing VM(virtual machine) images
Talk Outline

• Background & Motivation
• **Our goal & Solution**
• Performance analysis
• Summary & Future work
Research Objectives

• Invent an indicator (CFL) for read performance degradation for deduplication storage
• Investigate effectiveness of our CFL indicator
• Examine CFL values with realistic workloads
Understanding Chunk Fragmentation

- Very initially, all “unique chunks” are grouped into a container to preserve a spatial locality (read sequence is the same as write; sequential)

```
D_A  
A0  A1  A2  A3  
A0  A1  A2  A3  

data stream
a series of “unique” chunks from D_A
logged into a container
container (fixed size, as large as RAID stripe size)
do a “LARGE” write
```
Understanding Chunk Fragmentation

- Chunks of a data stream – **distributed into a more # of containers**

```plaintext
A0  A1  A2  A3
D_A
```

- Container 3

```plaintext
A0' A1' A2' A3
D_A'
```

- Container 7

“read from ONE container”

“read from TWO containers”

A1 is deduped

pointing
**Chunk Fragmentation Level**

- Given a data stream,
  - **Optimal chunk fragmentation (OCF)**
    - sum of all chunk sizes / container size
  - **Current chunk fragmentation (CCF)**
    - # of containers where chunks are stored

- **Chunk Fragmentation Level = OCF / CCF**
  - overuse ratio of containers w.r.t. OCF
Chunk Fragmentation Level

- Example

OCF of $D_A = 1$
CCF of $D_A = 1$
CFL of $D_A = 1$

$OCF$ of $D_A' = 1$
$CCF$ of $D_A' = 2$
$CFL$ of $D_A' = 0.5$

A1 is deduped

pointing

Container 3

Container 7
CFL vs. Read Performance

- **CFL** – good indicator for read performance degradation with deduped data stream

**Under optimal conditions** (CFL=1, CCF=OCF), to read the entire data stream, approximately, there will be (OCF-1) short seeks (betw. different containers)

**Under non-optimal conditions** (CFL < 1, CCF > OCF), there will be (OCF-1) short seeks + (CCF-OCF) long seeks

⇒ Long seeks contribute to the read perf. degradation
CFL’s Performance Model

- Relative read performance w.r.t. optimal case

\[ k = \frac{CS}{R \cdot t_s} \]

\[ P_m = \begin{cases} \frac{k+1}{k-2(\alpha_m-1)+\frac{(2\alpha_m-1)}{CFL_i}} & \text{if } CFL_i > 0.5 \\ \frac{k+1}{k+\frac{\alpha_m}{CFL_i}} & \text{otherwise,} \end{cases} \]

**Relative read performance of a data stream decreases**

1. CFL decreases;
2. \( m \) (# of the concurrent data streams) increases
Talk Outline

• Background & Motivation
• Our goal & Solution
• **Performance analysis**
• Summary & Future work
Performance Analysis

• Experimental setup
  – underlying storage: 3-disk RAID0 (stripe unit size=128KB)
  – deduplication appliance: Linux box (2.6.18, 3.0GHz dual cores and 4GB RAM)
  – each container is stripped over three disks (RAID0)

• Evaluation sketch
  – correlation: CFL vs. read performance
  – impacts of other parameters (CS, m) to performance
  – CFL variation with realistic workloads
Performance Analysis

CFL vs. Read Performance

- Strong relationship was observed:
  - 63.7% performance degradation when CFL=0.3

![Graph showing performance degradation with CFL]
Performance Analysis

Impacts of Other Parameters

- **Container size**: max. 27.6% degradation (CS\textsubscript{10})
- **# of concurrent streams**: max. 18.4% (m=6)
Performance Analysis

CFL Variation with Realistic Workloads

- **Four realistic workloads**: exchange server (.7), workstation (.6), RCS system (.5), /var dir (.8)

- **CFL (fredp4-RCS)**, dropped to 0.65 (35% degradation in read performance)
Talk Outline

- Background & Motivation
- Our goal & Solution
- Performance analysis
- **Summary & Future work**
Summary

• Proposed **CFL (Chunk Fragmentation Level)** to measure read performance degradation for each data stream in deduplication storage

• **Experiments** showed *CFL could be used as an effective indicator* (more important than other parameters: 63.7% >> 27.6%, 18.4%)

• **Realistic workloads** exhibited *remarkable drops in CFL* (max. 0.65)
Future(On-going) Work

• **Chunk re-organization** to preserve the optimal(initial) read performance based on the observed CFL values
  - validating & improving accuracy of the theoretical model
  - evaluating selective migration scheme, selective migration & duplication scheme

• Read cache management
  - how to effectively cache & prefetch **multiple read data streams** from underlying storage?
Thank you!
Question & Answering

Youngjin Nam, Guanlin Lu, Nohhyun Park
Weijun Xiao, David H. C. Du

University of Minnesota
http://cris.cs.umn.edu