I/O Deduplication: Utilizing Content Similarity to Improve I/O Performance

Ricardo Koller       Raju Rangaswami

School of Computing and Information Sciences
College of Engineering and Computing

September 9, 2010
I/O Deduplication

It is a storage solution that uses content similarity for improving I/O: eliminate duplicated I/O’s and reduce seek times.

- It is not Data Deduplication, used in archival storage [Venti], COW disks [QEMU].

- It consists of 3 techniques:
  - Content based cache
  - Dynamic replica retrieval
  - Selective duplication
Outline

1. Content Based Cache
2. Dynamic Replica Retrieval
3. Selective Duplication
4. Related Work
5. Limitations & Future work
6. Conclusions
Workloads

Block traces collected downstream of an active page cache for three weeks.

- **web-vm**: Two VM's hosting web-servers: web-mail & online course management.
- **mail**: Our department mail server.
- **homes**: NFS server that serves the home directories of our research group.
Motivation 1: Frequency
Motivation 2: Recency

![Graph showing reuse distance for Web-vm, Mail, and Homes categories with Read and Write operations. The y-axis represents reuse distance with values ranging from 1000 to 1e+07, and the x-axis represents Read, Write, and Read+Write operations. The graph shows different trends for each category, indicating varying reuse distances for each operation.]
Design: Content based Cache

- Reads sectors are searched for hits and in case of miss, content is searched and possibly inserted into the cache.
- Placed at the block layer
  - Write-through cache to maintain semantics.
  - ARC for second level cache.
Evaluation

- The I/O deduplication system was implemented as a module for kernel 2.6.20

- Traces replayed at 100X using a modified version of *btreplay*

- Measurements of I/O time were taken using *blktrace*

- Performed on a single Intel(R) Pentium 4 CPU 2.00GHZ with 1GB of memory and a WD disk running at 7200RPM
Evaluation: Content Addressed Cache

- Sector 4MB
  - Content 4MB
- Sector 200MB
  - Content 200MB

Read hit ratio
- web-vm
- mail
- homes
Evaluation: Hits versus Cache Size

![Graphs showing read hit ratio versus cache size for ARC and LRU algorithms for read and read/write operations.](image-url)
Outline

1. Content Based Cache
2. Dynamic Replica Retrieval
3. Selective Duplication
4. Related Work
5. Limitations & Future work
6. Conclusions
Motivation: Duplication

<table>
<thead>
<tr>
<th>Workloads</th>
<th>web-vm</th>
<th>mail</th>
<th>homes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique 4K pages (millions)</td>
<td>1.9</td>
<td>27</td>
<td>62</td>
</tr>
<tr>
<td>Total 4K pages (millions)</td>
<td>5.2</td>
<td>73</td>
<td>183</td>
</tr>
<tr>
<td>Disk Static similarity</td>
<td>2.67</td>
<td>2.64</td>
<td>2.94</td>
</tr>
</tbody>
</table>
**Motivation: Duplication**

### Workloads

<table>
<thead>
<tr>
<th></th>
<th>web-vm</th>
<th>mail</th>
<th>homes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique 4K pages (millions)</td>
<td>1.9</td>
<td>27</td>
<td>62</td>
</tr>
<tr>
<td>Total 4K pages (millions)</td>
<td>5.2</td>
<td>73</td>
<td>183</td>
</tr>
<tr>
<td>Disk Static similarity</td>
<td>2.67</td>
<td>2.64</td>
<td>2.94</td>
</tr>
</tbody>
</table>

#### Diagram

- **Workload static similarity**
  - web-vm
  - mail
  - homes

- **Maximum number of copies**
  - 10
  - 100
  - 1000
  - no limit
Design: Dynamic Replica Retrieval

- Reduce seek times by indirecting requests based on head position: choose the duplicate that’s closer to the head.
- The yellow entries share an uncached page.
- Current head position based on completed reads.
- Placed above the I/O scheduler:
  - Indirect only if there are no adjacent requests.
Evaluation: Dynamic Replica Retrieval

![Diagram showing performance comparison]

- **vanilla dedup**
- **Per-request disk I/O read time (sec)**
  - web-vm
  - mail
  - homes
Motivation: Working Set Overlap

![Content access overlap chart]

- **web-vm**: Light blue bars
- **mail**: Red bars
- **homes**: Gray bars

**X-axis**: Intervals (1 to 7)

**Y-axis**: Content access overlap (%)

The chart shows the overlap of content access for different intervals across web-vm, mail, and homes.
Data is duplicated at scratch spaces interspersed across the disk.
Dynamic Replica Retrieval

### Without Selective Duplication

**v d**

- **web-vm**: 0.005, 0.01, 0.015, 0.02
- **mail**: 0.005, 0.01, 0.015, 0.02
- **homes**: 0.005, 0.01, 0.015, 0.02

### With Selective Duplication

**v d**

- **web-vm**: 0.005, 0.01, 0.015, 0.02
- **mail**: 0.005, 0.01, 0.015, 0.02
- **homes**: 0.005, 0.01, 0.015, 0.02

Per-request disk I/O read time (sec)
## All Together

<table>
<thead>
<tr>
<th>Workload</th>
<th>Vanilla (rd sec)</th>
<th>I/O dedup (rd sec)</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>web-vm</td>
<td>3098.61</td>
<td>1641.90</td>
<td>47%</td>
</tr>
<tr>
<td>mail</td>
<td>4877.49</td>
<td>3467.30</td>
<td>28%</td>
</tr>
<tr>
<td>home</td>
<td>1904.63</td>
<td>1160.40</td>
<td>39%</td>
</tr>
</tbody>
</table>
Overhead

- Memory

\[ \text{mem}(P, WSS, HTB) = 13 \times P + 36 \times P \times WSS + 8 \times HTB \]

For a content cache of 1GB, static similarity of 4 and a hash table of a million buckets, the metadata is 48MB (4.6%).

- CPU

  ✓ if \( HTB = 1e3 \), \( \text{cpu\_read\_miss}(P) = O(P) + 100000 \) cycles.
  ✓ if \( HTB = 1e6 \), \( \text{cpu\_read\_miss}(P) = O(1) + 100000 \) cycles.

For our machine running at 2GHz, the 100000 + 1000 cycles are 90\( \mu \)s.
Outline

1. Content Based Cache
2. Dynamic Replica Retrieval
3. Selective Duplication
4. Related Work
5. Limitations & Future work
6. Conclusions
Related Work

- I/O Performance Optimization
  - Duplication of popular data: FS2, Borg

- Content Addressed Storage
  - Archival storage: Venti

- I/O Deduplication
  - Satori (COW-disk sharing mode)
Outline

1. Content Based Cache
2. Dynamic Replica Retrieval
3. Selective Duplication
4. Related Work
5. Limitations & Future work
6. Conclusions
Limitations & Future Work

- Integration with the page cache
- Multiple disks
- Variable sized chunks
- Write requests “special handling”, leave them for later? pdflush?
Limitations & Future Work

- Page replacement strategies for content
  - LRU and ARC are further from the optimal when used with content

- I/O scheduling based on duplicated blocks
  - Greedy option is sub-optimal
  - I/O scheduling with duplicated blocks is NP-Hard (as shown for regular I/O scheduling by [Andrews, 1996].)
  - Design an approximation algorithm with some nice approx. ratio
Outline

1. Content Based Cache
2. Dynamic Replica Retrieval
3. Selective Duplication
4. Related Work
5. Limitations & Future work
6. Conclusions
Summary and Conclusions

- For systems where content is more frequent than sector and reuse distances are shorter for content compared to sector, content based caches can be more effective than sector ones.

- On disk duplications can be used for reducing I/O times.
Questions?

http://dsrl.cs.fiu.edu/projects/iodedup/