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

FIU | FLORIDA INTERNATIONAL UNIVERSITY

February 26, 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 arhival 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.

Outline

1 Content Based Cache

2 Dynamic Replica Retrieval

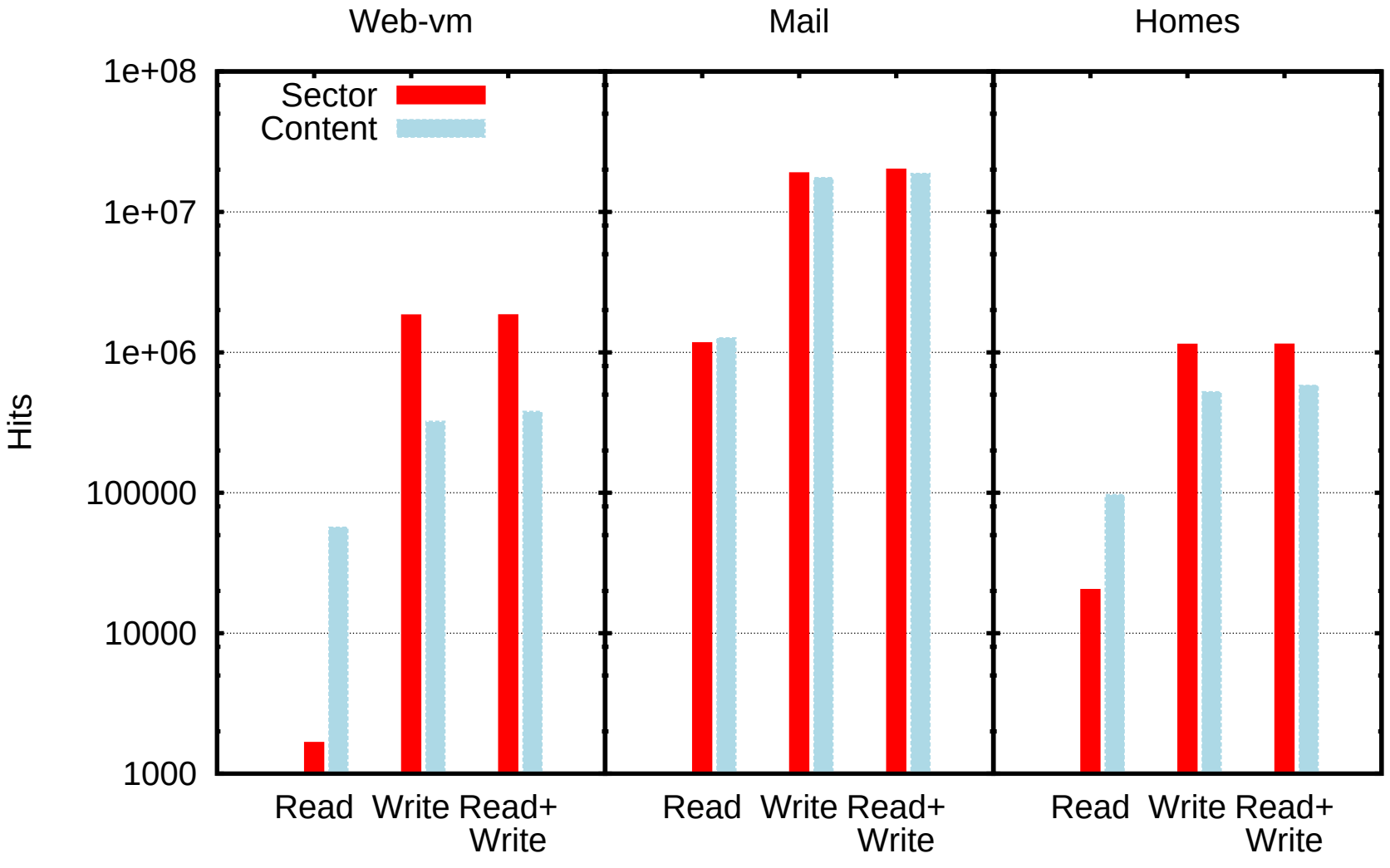
3 Selective Duplication

4 Related Work

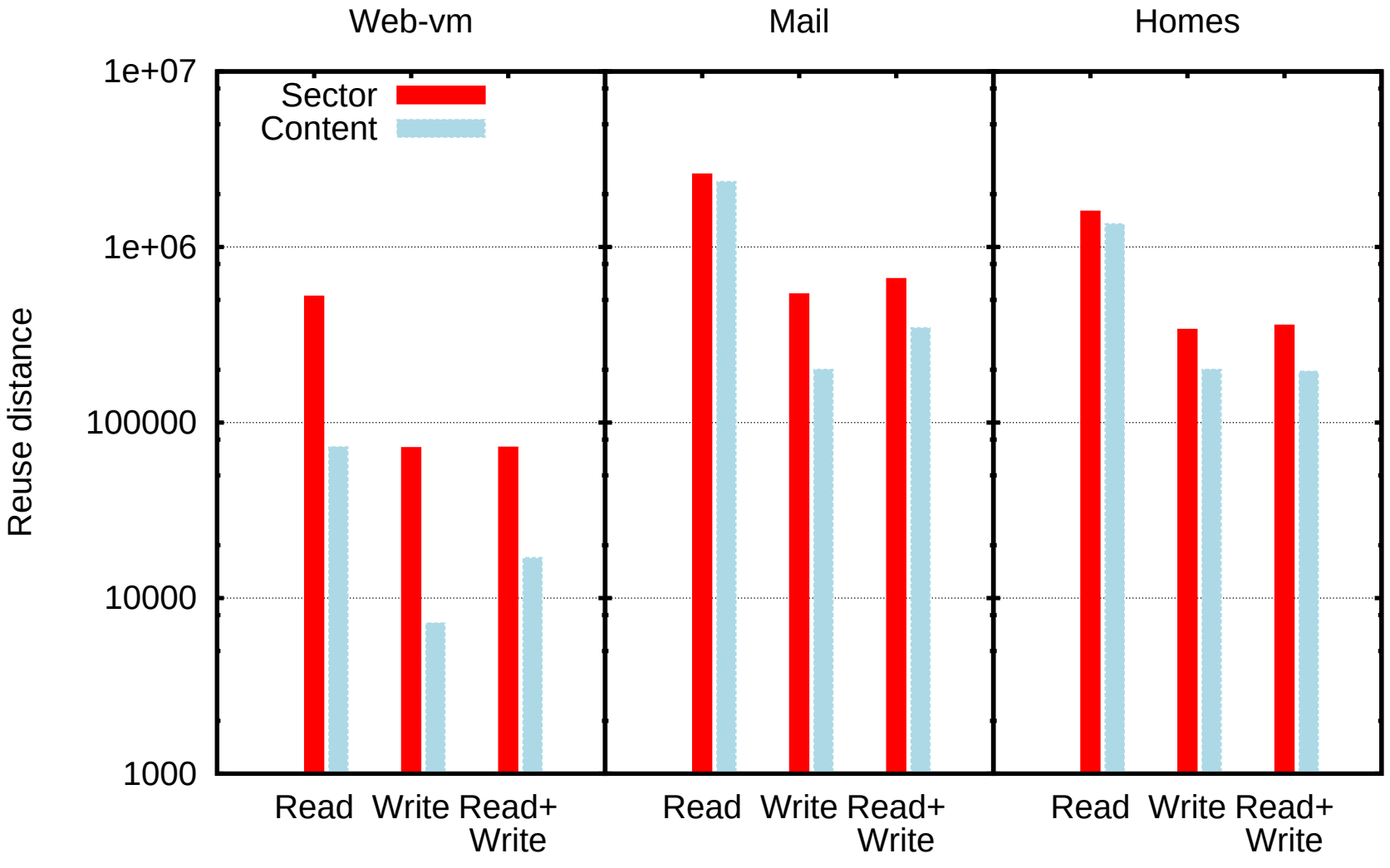
5 Limitations & Future work

6 Conclusions

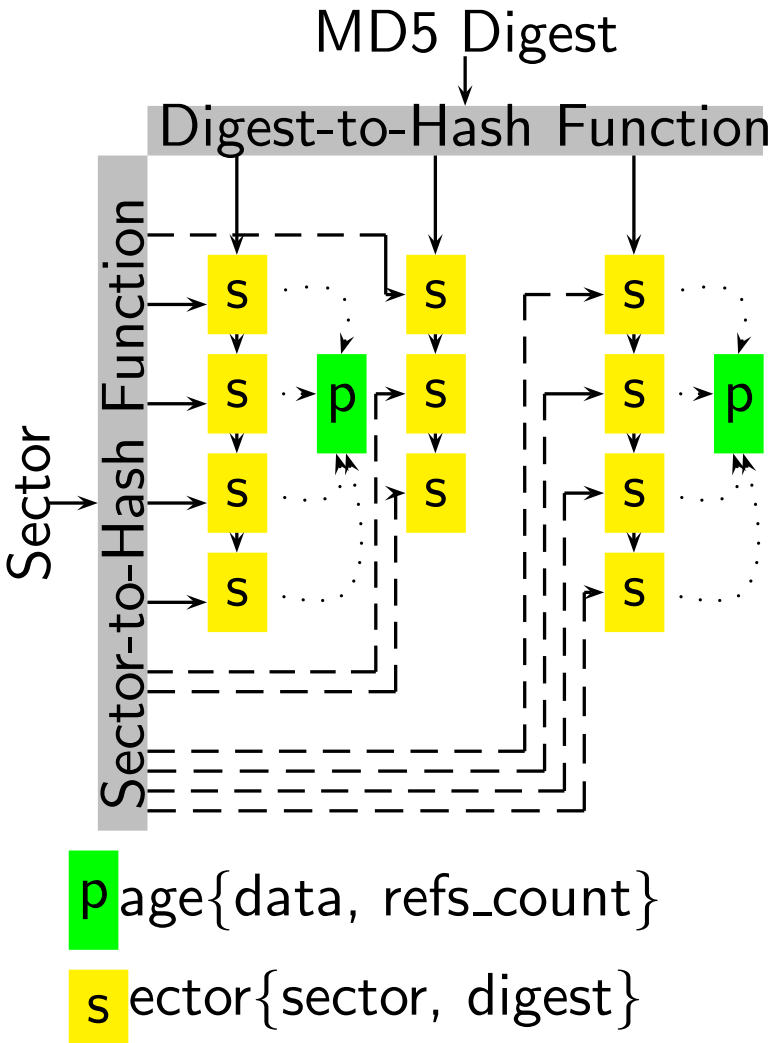
Motivation 1: Frequency



Motivation 2: Recency



Design: Content based Cache

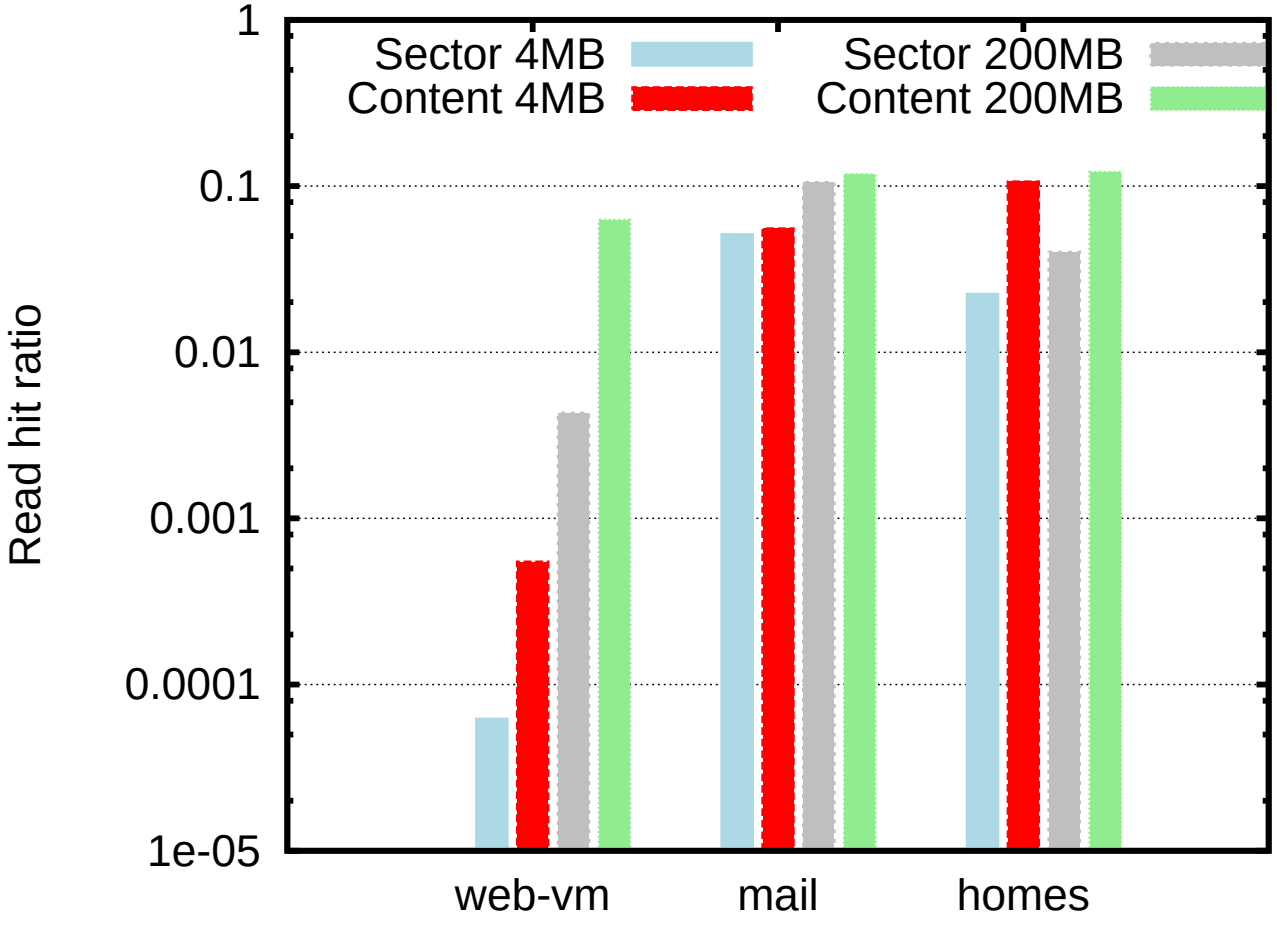


- Reads sectors are searched for hits and in case of miss, content is searched and possibly inserted into the cache.
- Placed at the block layer
  - ✓ Write-though cache to maintain semantics.
  - ✓ ARC for second level cache.

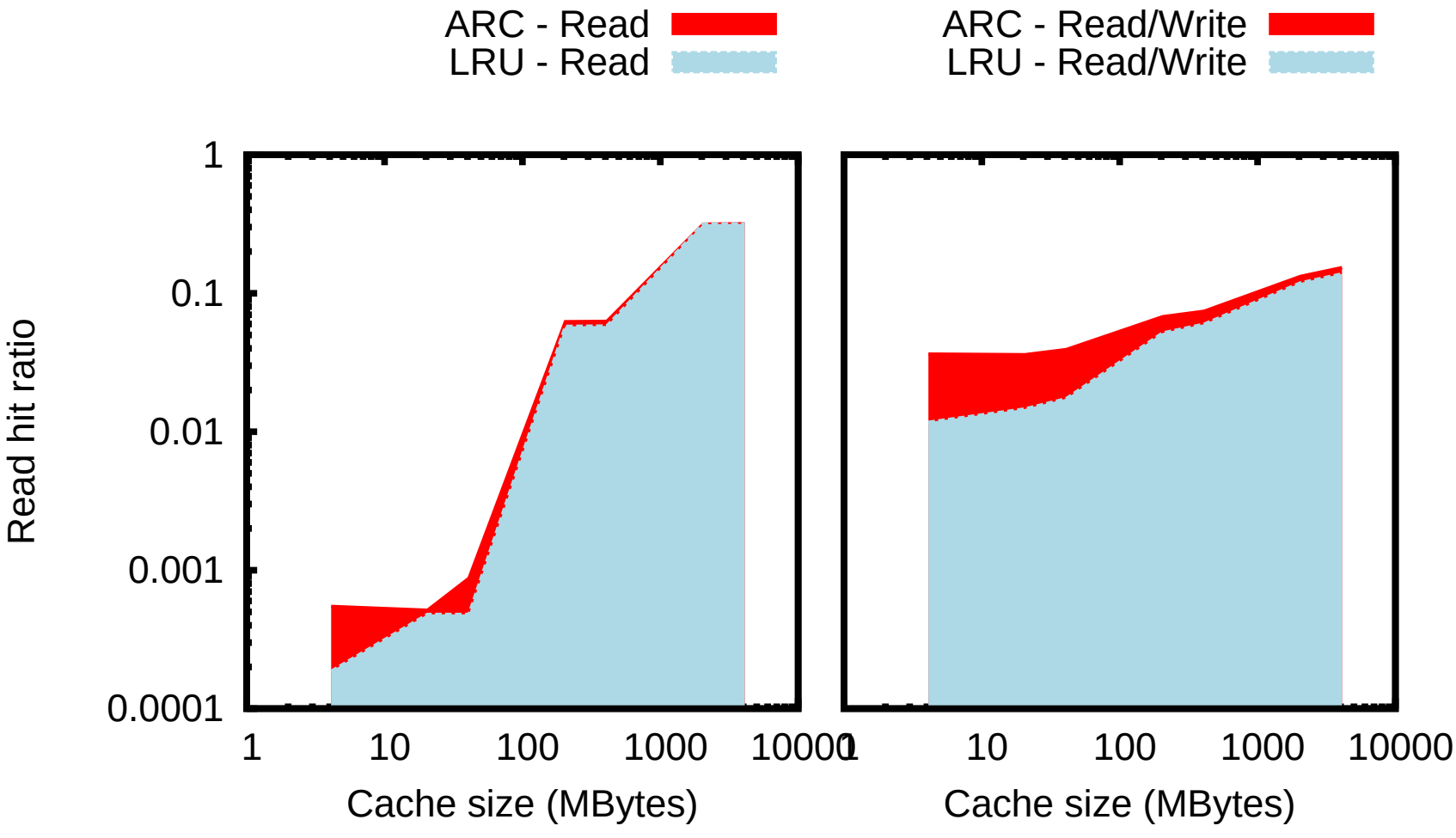
Evaluation

- 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



Evaluation: Hits versus Cache Size



Outline

1 Content Based Cache

2 Dynamic Replica Retrieval

3 Selective Duplication

4 Related Work

5 Limitations & Future work

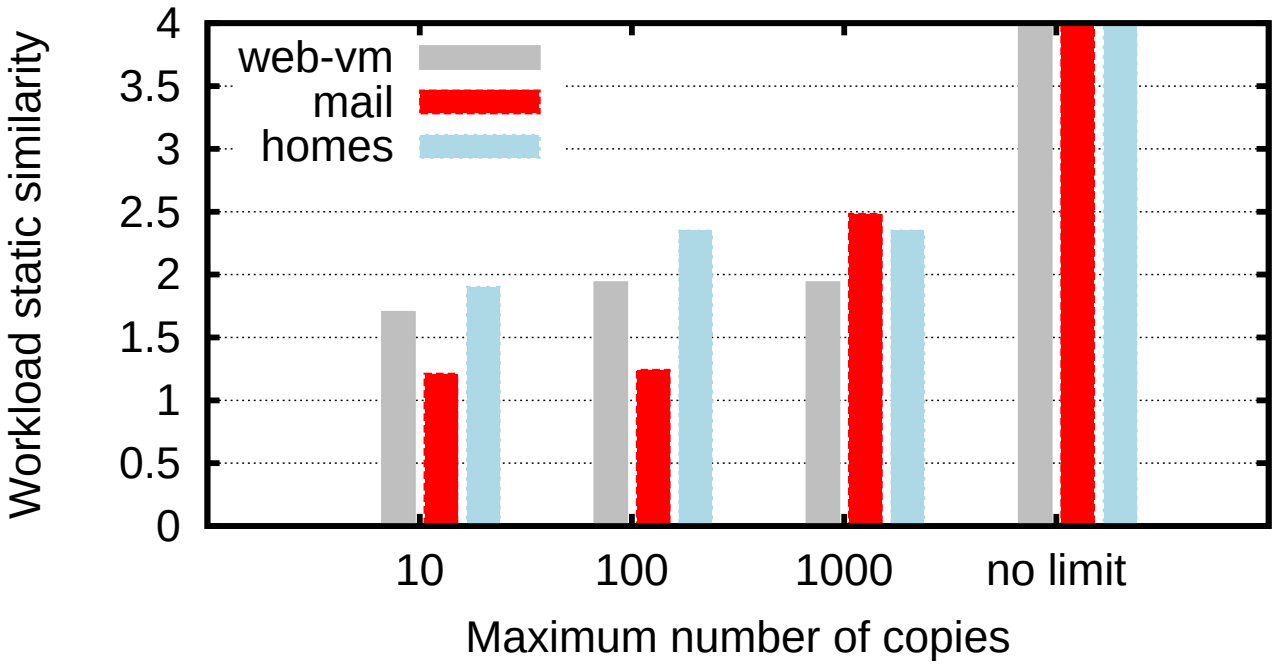
6 Conclusions

Motivation: Duplication

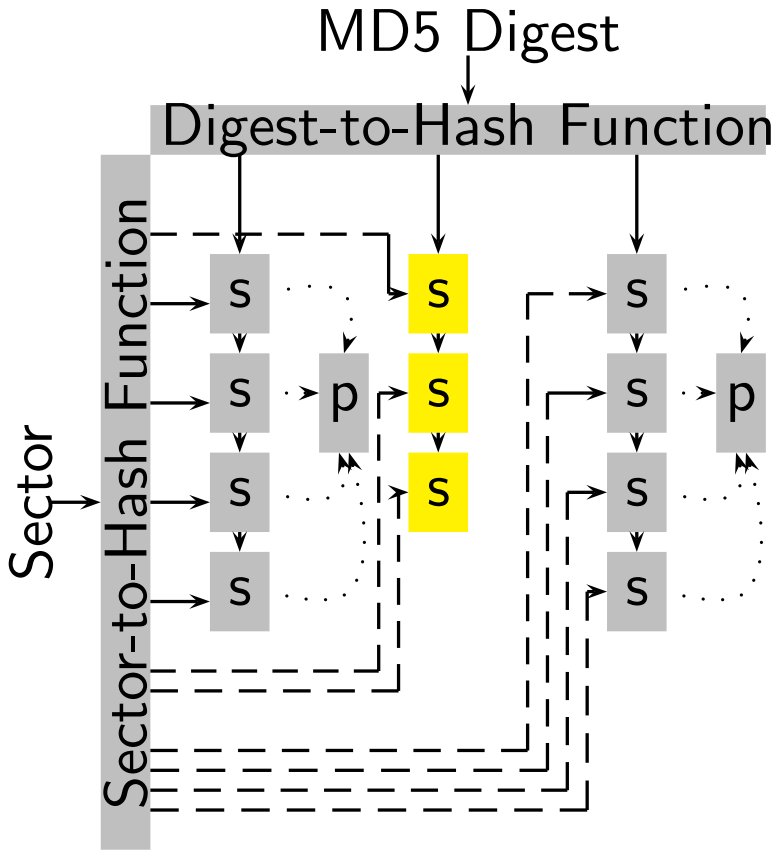
Workloads	**web-vm**	**mail**	**homes**
Unique 4K pages (millions)	1.9	27	62
Total 4K pages (millions)	5.2	73	183
Disk Static similarity	2.67	2.64	2.94

Motivation: Duplication

Workloads	**web-vm**	**mail**	**homes**
Unique 4K pages (millions)	1.9	27	62
Total 4K pages (millions)	5.2	73	183
Disk Static similarity	2.67	2.64	2.94

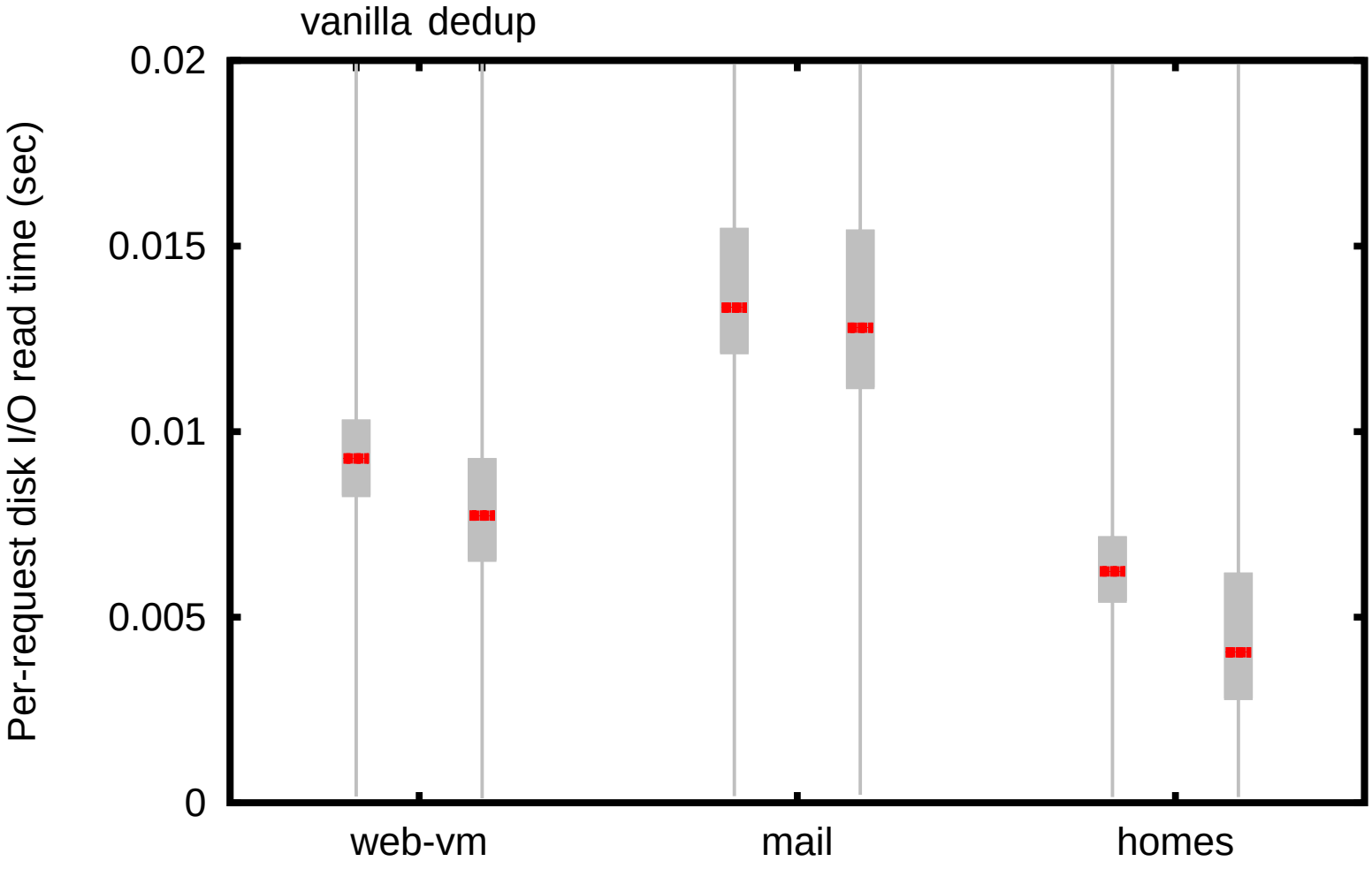


Design: Dynamic Replica Retrieval



- 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



Outline

1 Content Based Cache

2 Dynamic Replica Retrieval

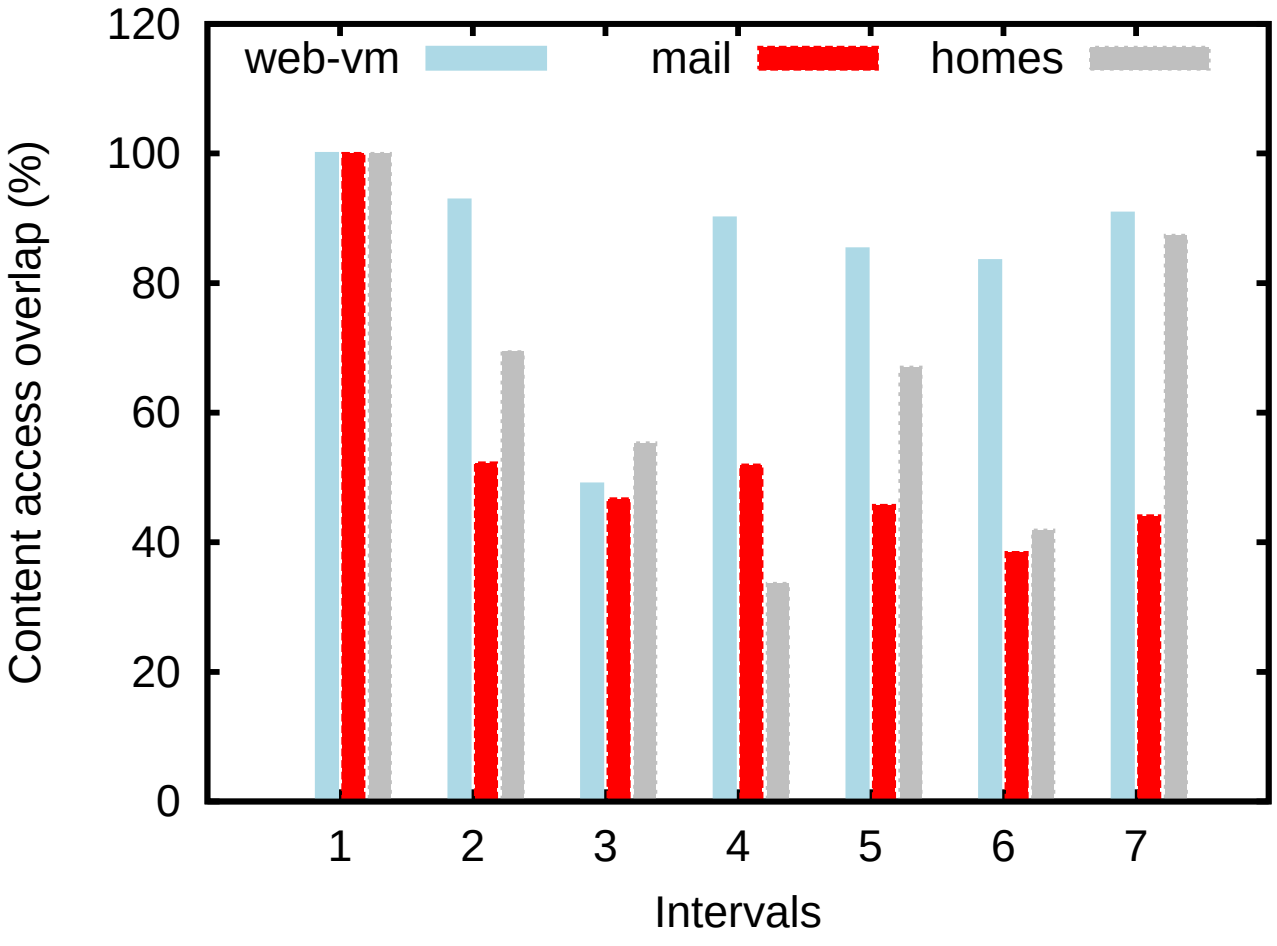
3 Selective Duplication

4 Related Work

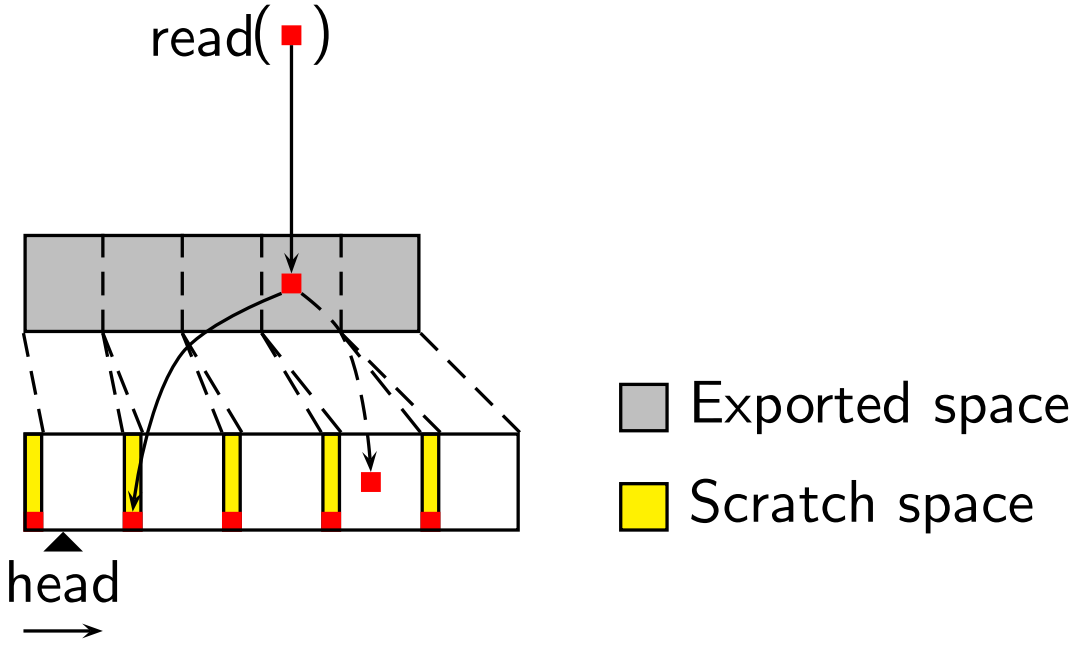
5 Limitations & Future work

6 Conclusions

Motivation: Working Set Overlap

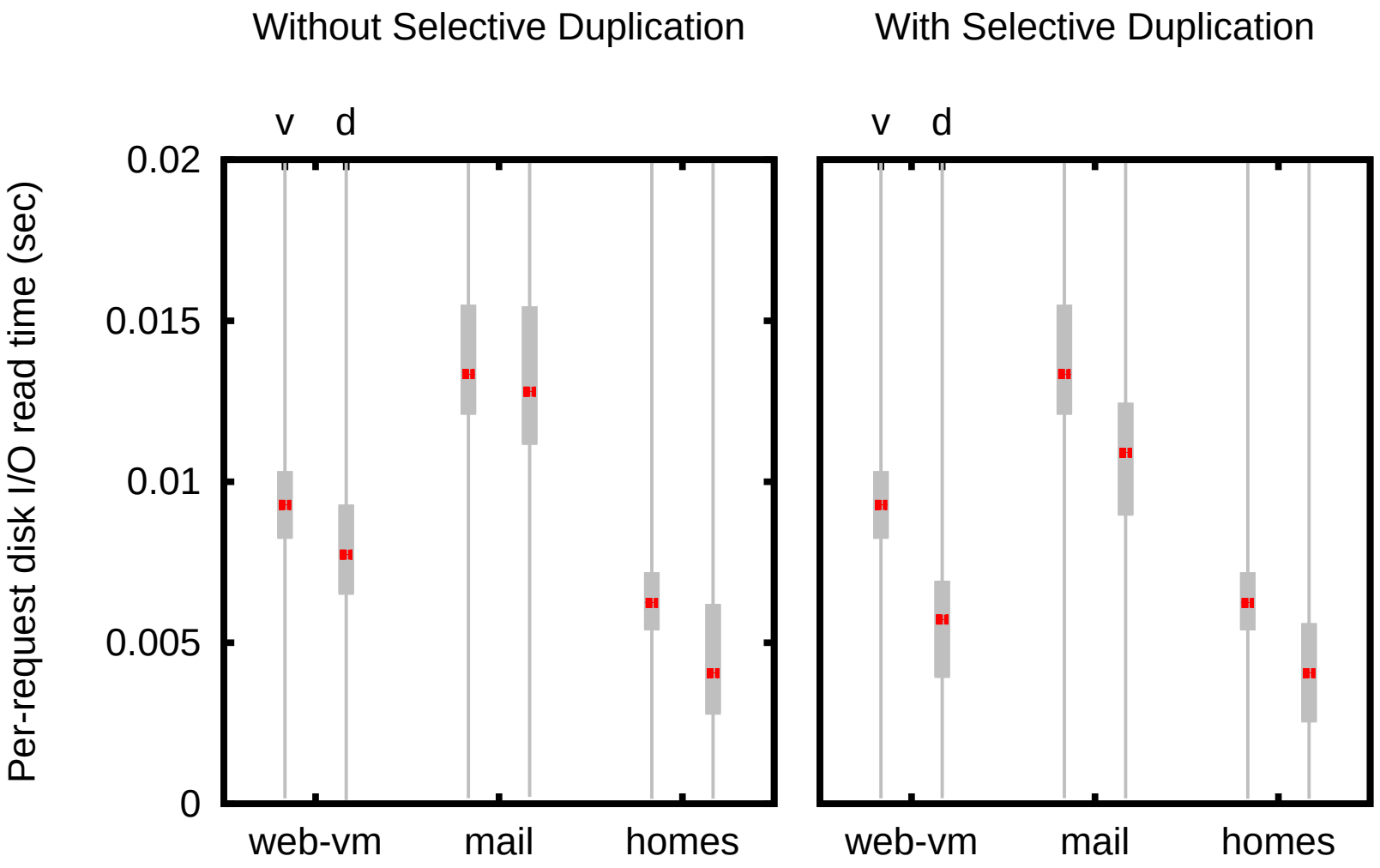


Selective Duplication



Data is duplicated at scratch spaces interspersed across the disk.

Dynamic Replica Retrieval



All Together

Workload	Vanilla (rd sec)	I/O dedup (rd sec)	Improvement
web-vm	3098.61	1641.90	47%
mail	4877.49	3467.30	28%
home	1904.63	1160.40	39%

Overhead

- Memory
  $mem(P, WSS, HTB) = 13 * P + 36 * P * WSS + 8 * HTB$

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 $HTB = 1e3$, $cpu\_read\_miss(P) = O(P) + 100000$ cycles.
  - ✓ if $HTB = 1e6$, $cpu\_read\_miss(P) = O(1) + 100000$ 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
- Page replacement strategies for content
- I/O scheduling based on duplicated blocks
- Write requests "special handling", leave them for later? pdflush?

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 sertor, 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/