To ARC or not to ARC

Ricardo A. Santana, Steven Lyons, Ricardo Koller*, Raju Rangaswami, Jason Liu

Florida International University * IBM TJ Watson
Host-side Caching

![Diagram showing the relationship between host, applications, SSD, DRAM, network, and centralized storage.]

Data Path

Non-Data Path

Centralized Storage
Benefits of Selective Caching

Extend the lifetime of the SSD

Improve hit-rate

Reduce system latency
Traditional Caching Algorithms

Designed to optimize hit-rate

Always perform cache replacement on a miss

Performance can be penalized

Adversary workloads for SSD caching
**LARC - Previous Solutions**

- **Cache (Data)**
  - LRU Eviction Policy
  - Insert data only if it is in the filter

- **Filter (Metadata)**
  - LRU: Recently Missed Pages
  - Insert metadata only if it is not in the filter

- HIT
- MISS
- Check the Filter
LARC - Observations

Avoids cache pollution

Maintains working set in cache

Slow to renew working set
mARC

Cache (Data) -> ARC Eviction Policy

Filter is not always used

Filter (Metadata) -> LRU: Recently Missed Pages
mARC - State Machine

Filter On
Stable

Performance Degradation

Better Performance
New Working Set

Filter Off
Unstable

Stable Performance
Great Performance

Poor Performance

Filter On
Unique Access
EVALUATION
FIU Traces

Mean Hit-Rate

- ARC
- LARC
- mARC

Cache size as a % of workload footprint

Mean Write-Rate

- ARC
- LARC
- mARC

Cache size as a % of workload footprint

Normalized vs. ARC

13% increase

23% increase
MSR Cambridge Traces

- **Left Graph:**
  - Title: Mean Hit-Rate
  - X-axis: Cache size as a % of workload footprint (5, 10, 15, 20, 25)
  - Y-axis: Mean Hit-Rate (0.0 to 0.6)
  - Legend: ARC, LARC, mARC

- **Right Graph:**
  - Title: Mean Write-Rate
  - X-axis: Cache size as a % of workload footprint (5, 10, 15, 20, 25)
  - Y-axis: Mean Write-Rate (0.0 to 0.9)
  - Legend: ARC, LARC, mARC

- Observations:
  - **Red Arrow (1%)** indicates a 1% difference in hit-rate between ARC and LARC.
  - **Green Arrow (25%)** indicates a 25% difference in write-rate between ARC and LARC.

- Summary:
  - The graphs compare the performance of ARC, LARC, and mARC caches under varying cache sizes.
  - ARC consistently outperforms LARC and mARC in terms of hit-rate and write-rate.
  - There is a notable difference in write-rate, with ARC showing a significant improvement over LARC and mARC at certain cache sizes.
Conclusions

Non-datapath caches are mainstream

Selective caching algorithms help

Improved performance
&
Improved device lifetime
Future Work

Generalizing algorithm

Explore other filtering mechanisms

Adaptive mechanisms

Kernel implementation
Acknowledgments

Anonymous reviewers

Ashvin Goel (for shepherding!)

NSF

NetApp
Thank you

Questions?
What is Churning?

Continuous Replacement

ITEMS SORTED BY ACCESS FREQUENCY

ACCESS FREQUENCY

S(WS_i)

S(C)

S(WS_{i+1})
# mARC - State Transition Table

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstable</td>
<td>Turn Filter On (Go to Stable)</td>
<td>Latest $HR_{Sample}$ is almost the same as the $HR_{State}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Latest $HR_{Sample}$ is much better than the $HR_{State}$</td>
</tr>
<tr>
<td></td>
<td>Turn Filter On (Go to Stable)</td>
<td>Latest $HR_{Sample}$ is significantly worse than the $HR_{State}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(ARC is incurring in high misses lately, possible streaming)</td>
</tr>
<tr>
<td>Stable</td>
<td>Turn Filter OFF (Go to Unstable)</td>
<td>Latest Sample hit-rate is much worse than the cache hit-rate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Filtering stopped working)</td>
</tr>
<tr>
<td>Unique Access</td>
<td>Turn Filter OFF (Go to Unstable)</td>
<td>Detect a new working set being cached</td>
</tr>
</tbody>
</table>