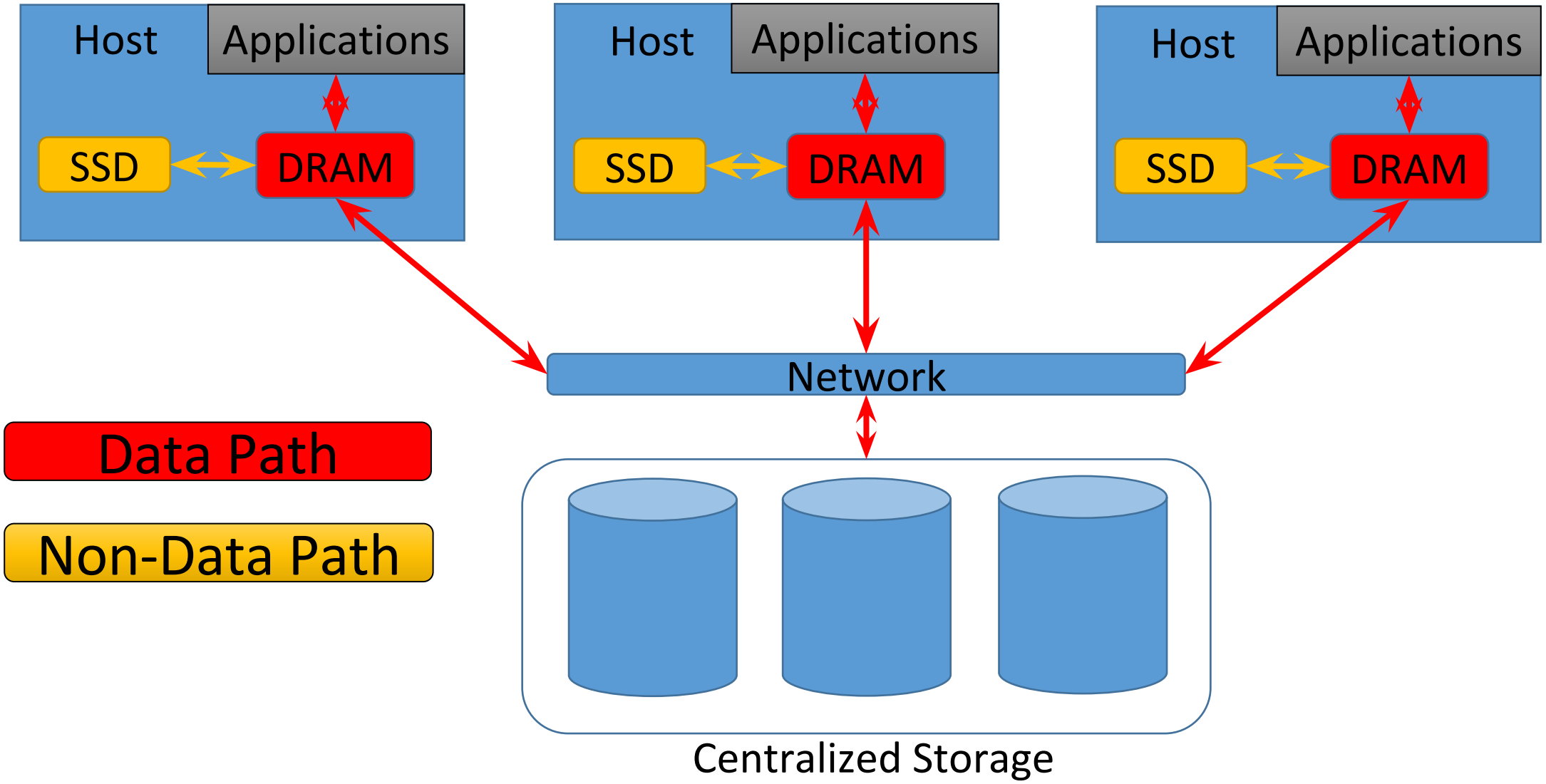
To ARC or not to ARC

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

Florida International University *** IBM TJ Watson**

Host-side Caching



Benefits of Selective Caching

Extend the lifetime of the SSD

Improve hit-rate

Reduce system latency

Traditional Caching Algorithms

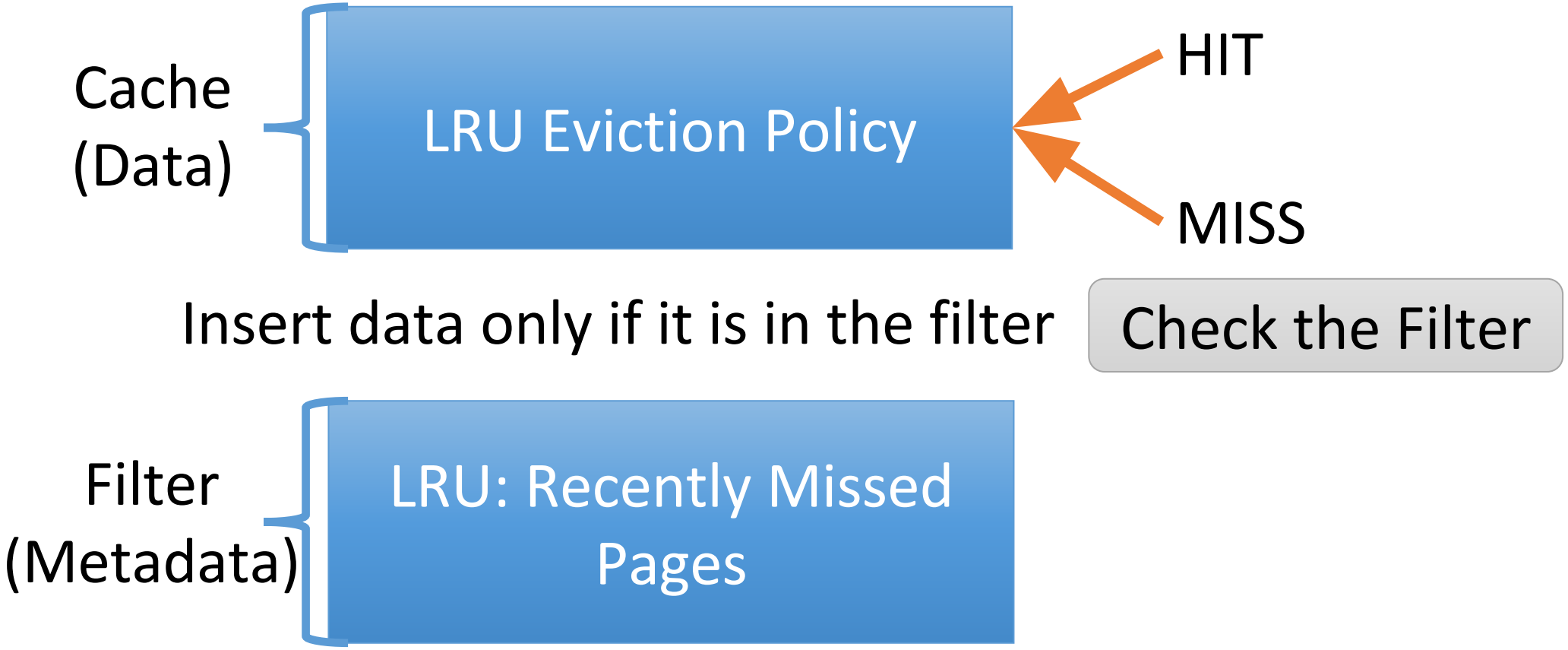
Designed to optimize hit-rate

Always perform cache replacement on a miss

Performance can be penalized

Adversary workloads for SSD caching

LARC - Previous Solutions



LARC - Observations

Avoids cache pollution

Maintains working set in cache

Slow to renew working set

mARC

Cache
(Data)

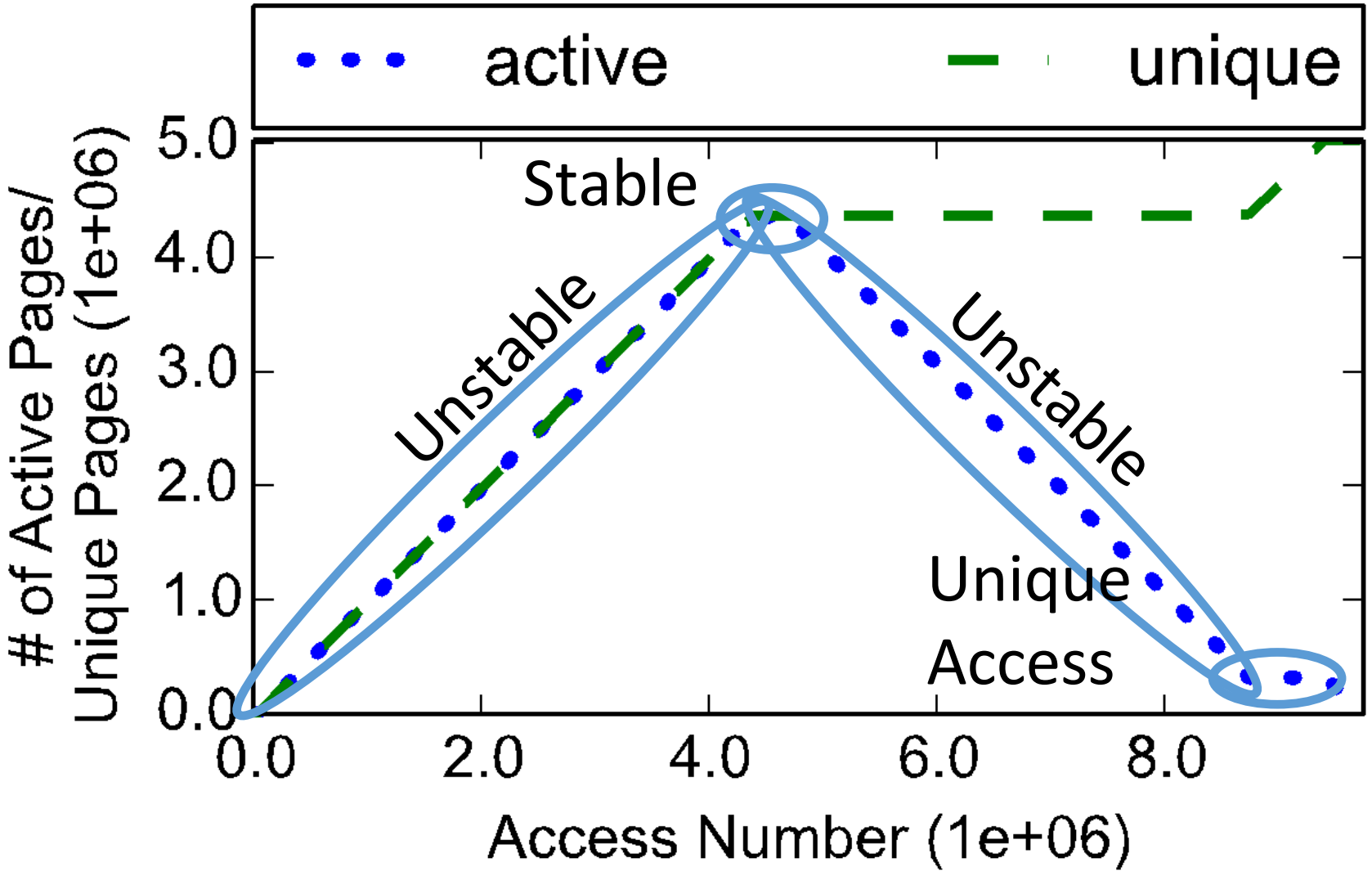
ARC Eviction Policy

Filter is not always used

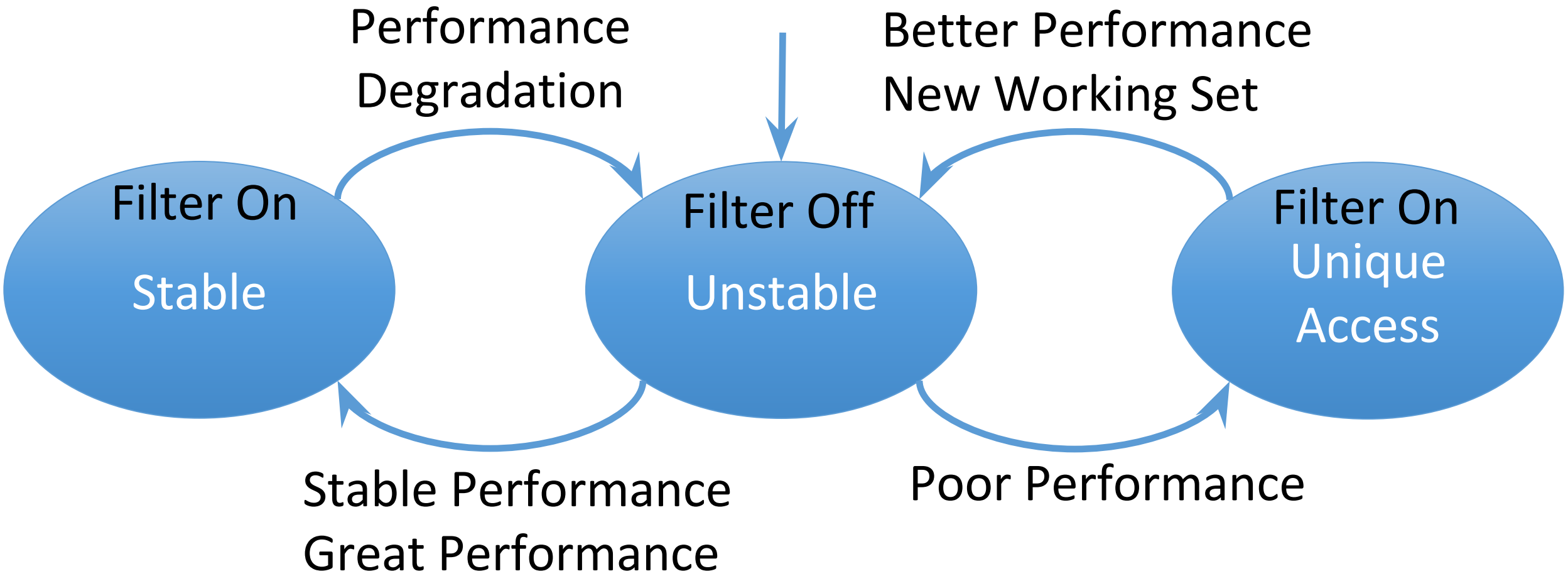
Filter
(Metadata)

LRU: Recently Missed Pages

Workload States A-U Modeling

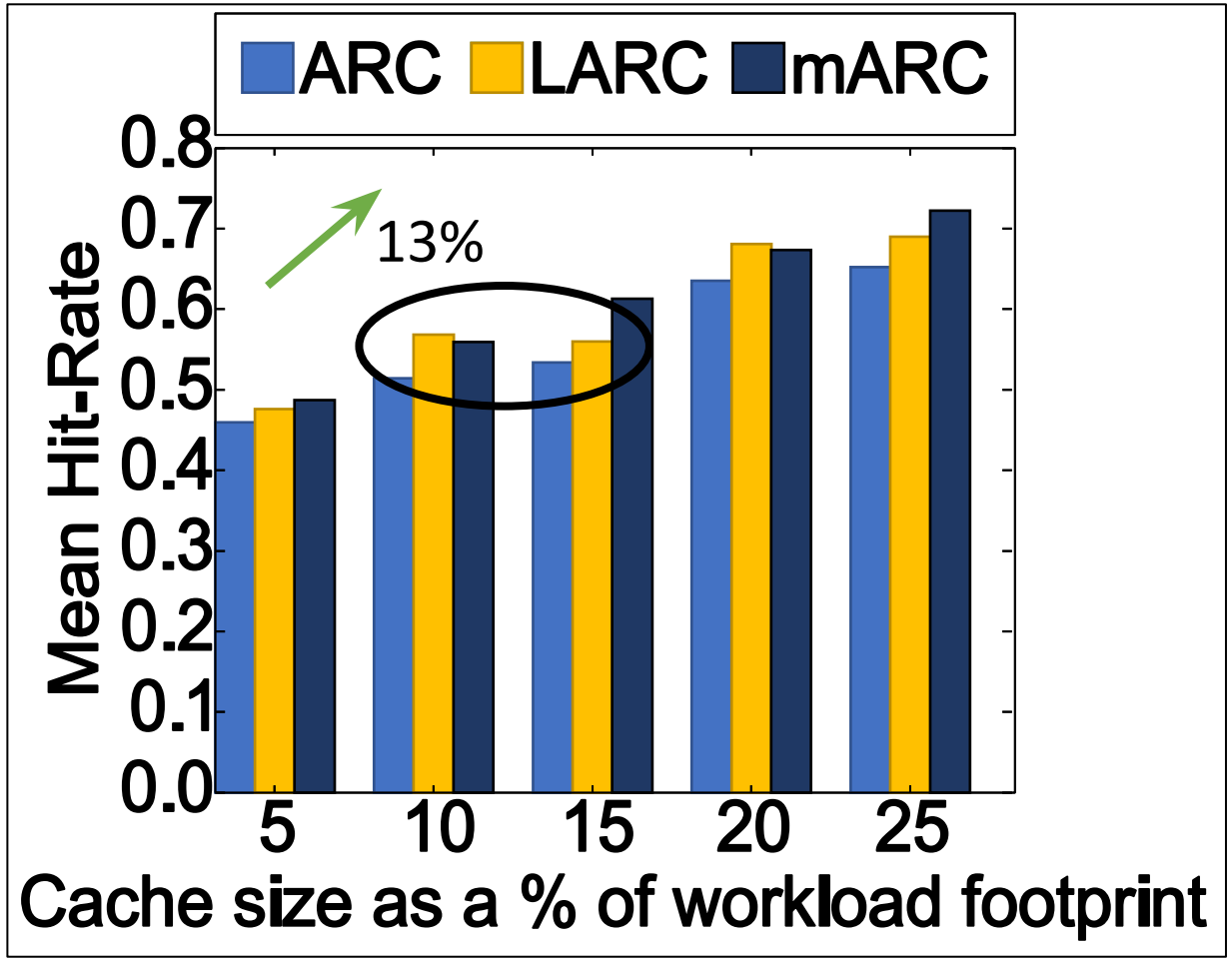


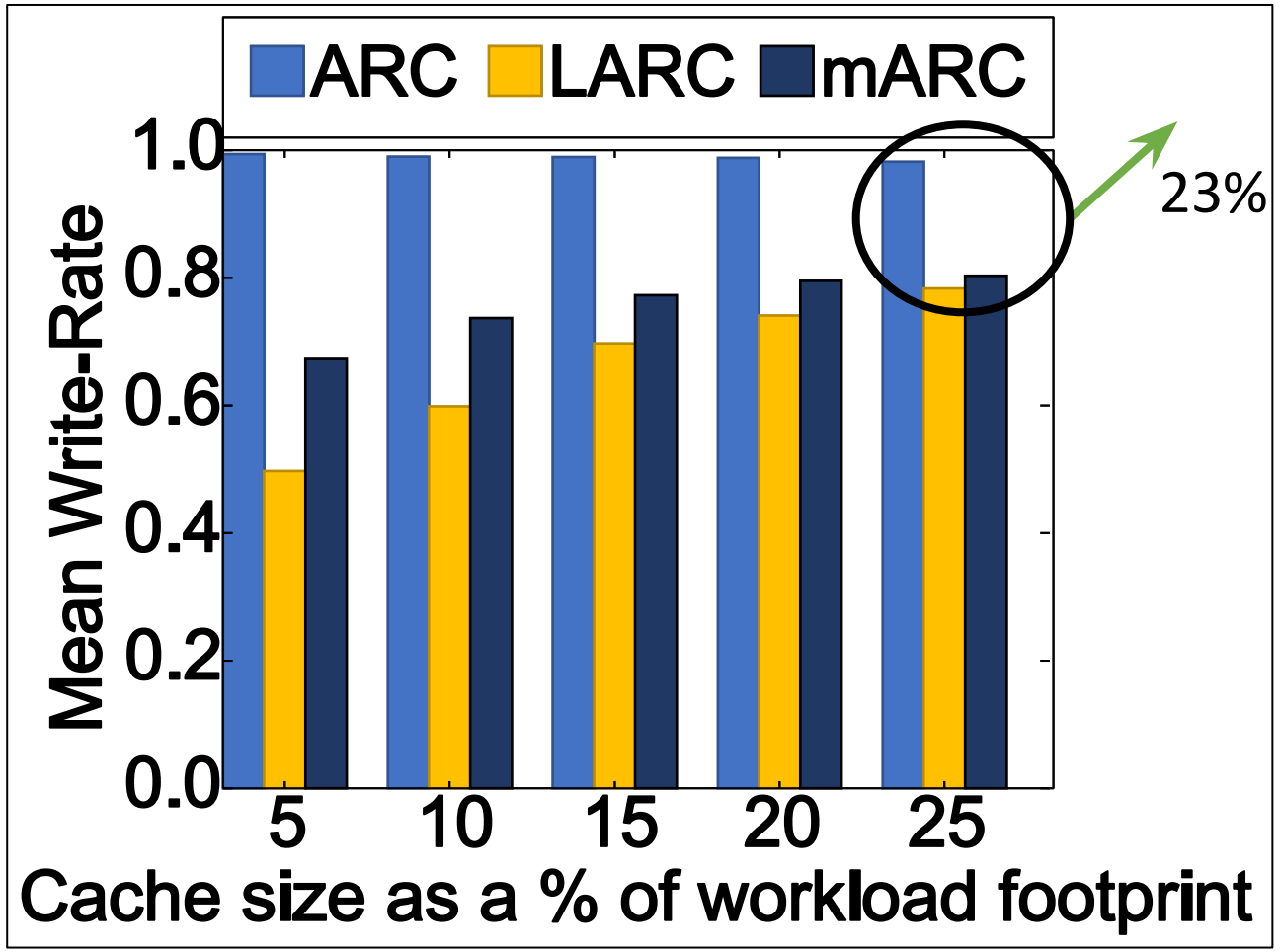
mARC - State Machine



EVALUATION

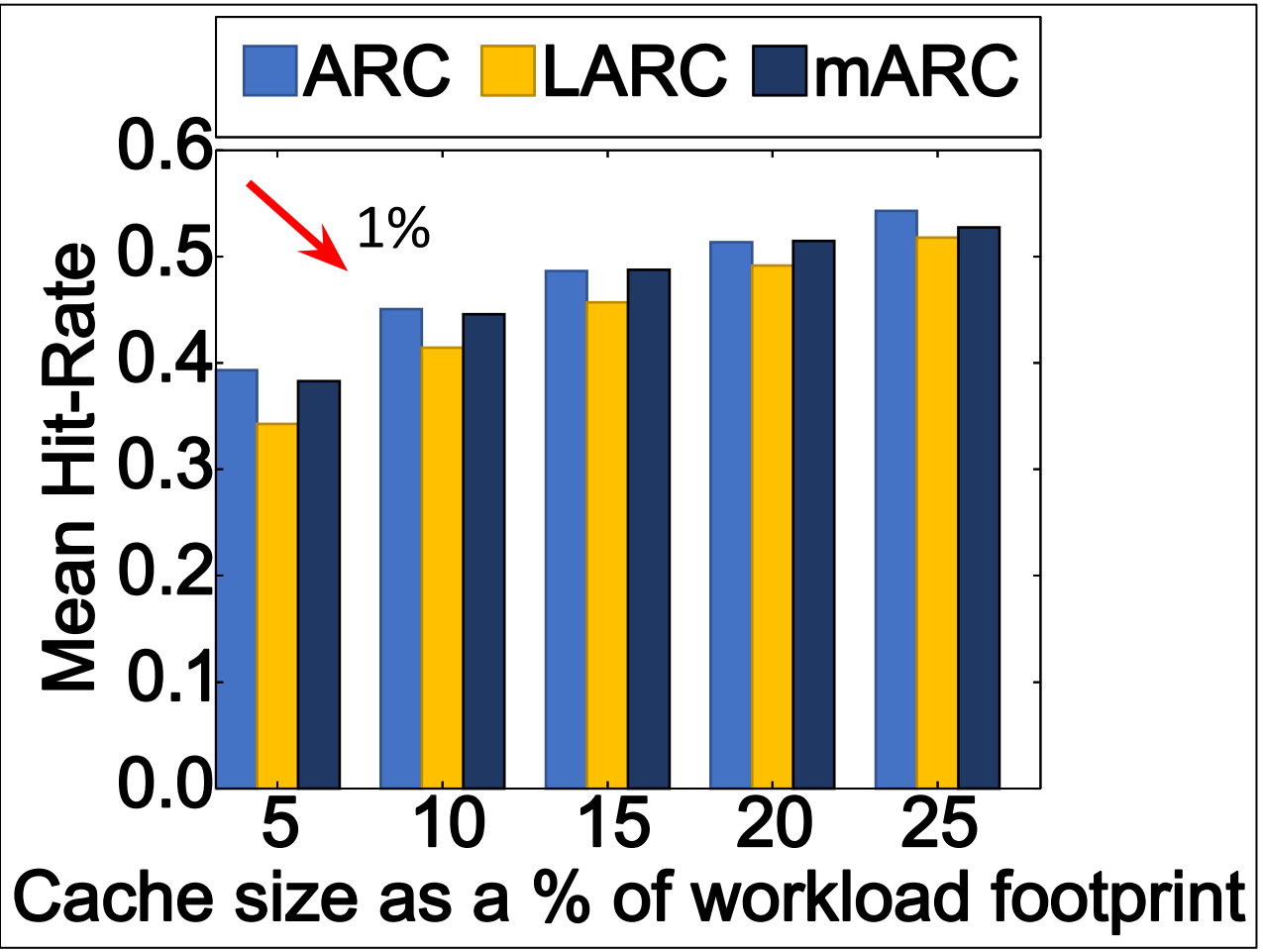
FIU Traces

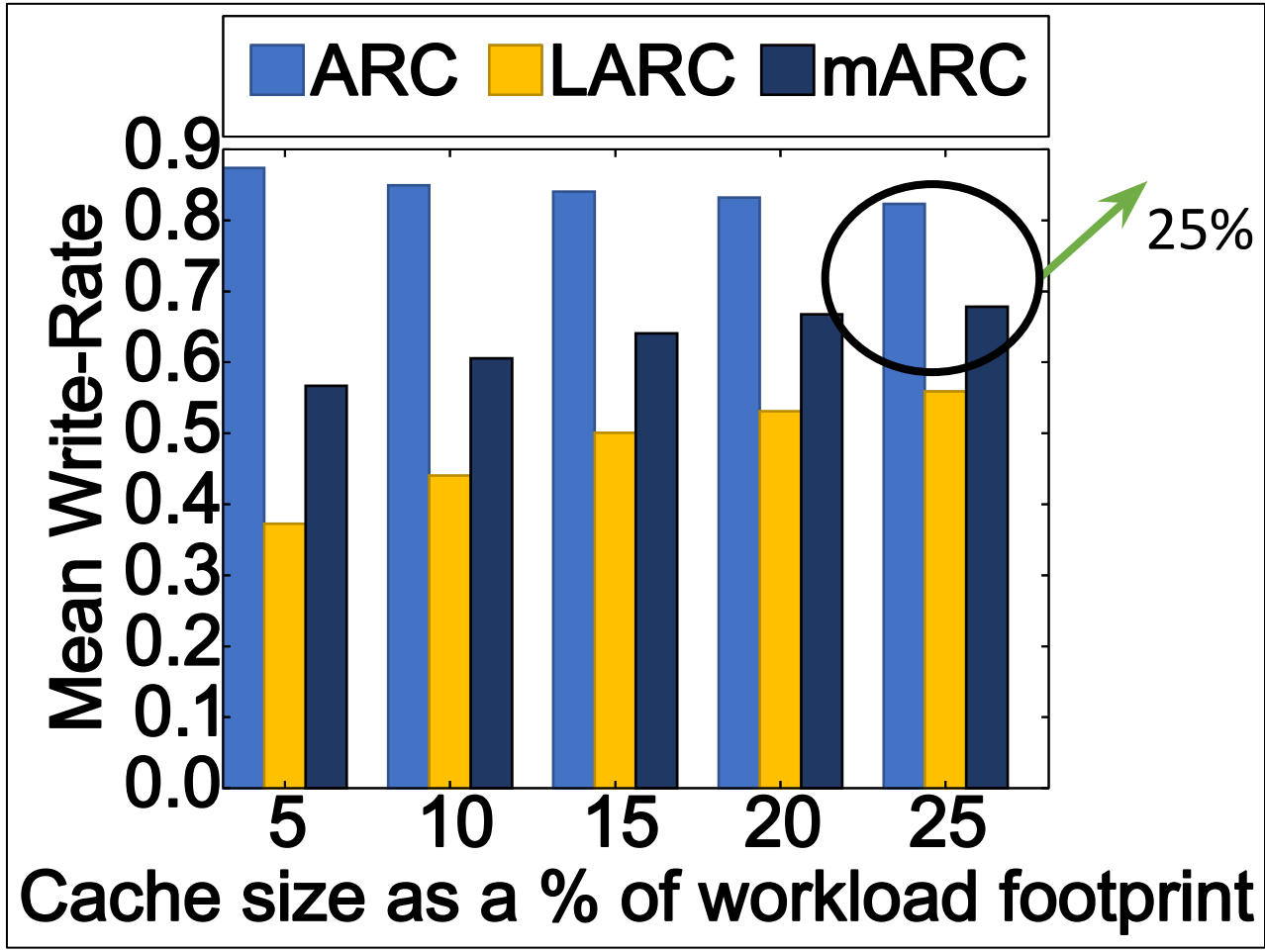




Normalized vs. ARC

MSR Cambridge Traces





Normalized vs. ARC

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!)

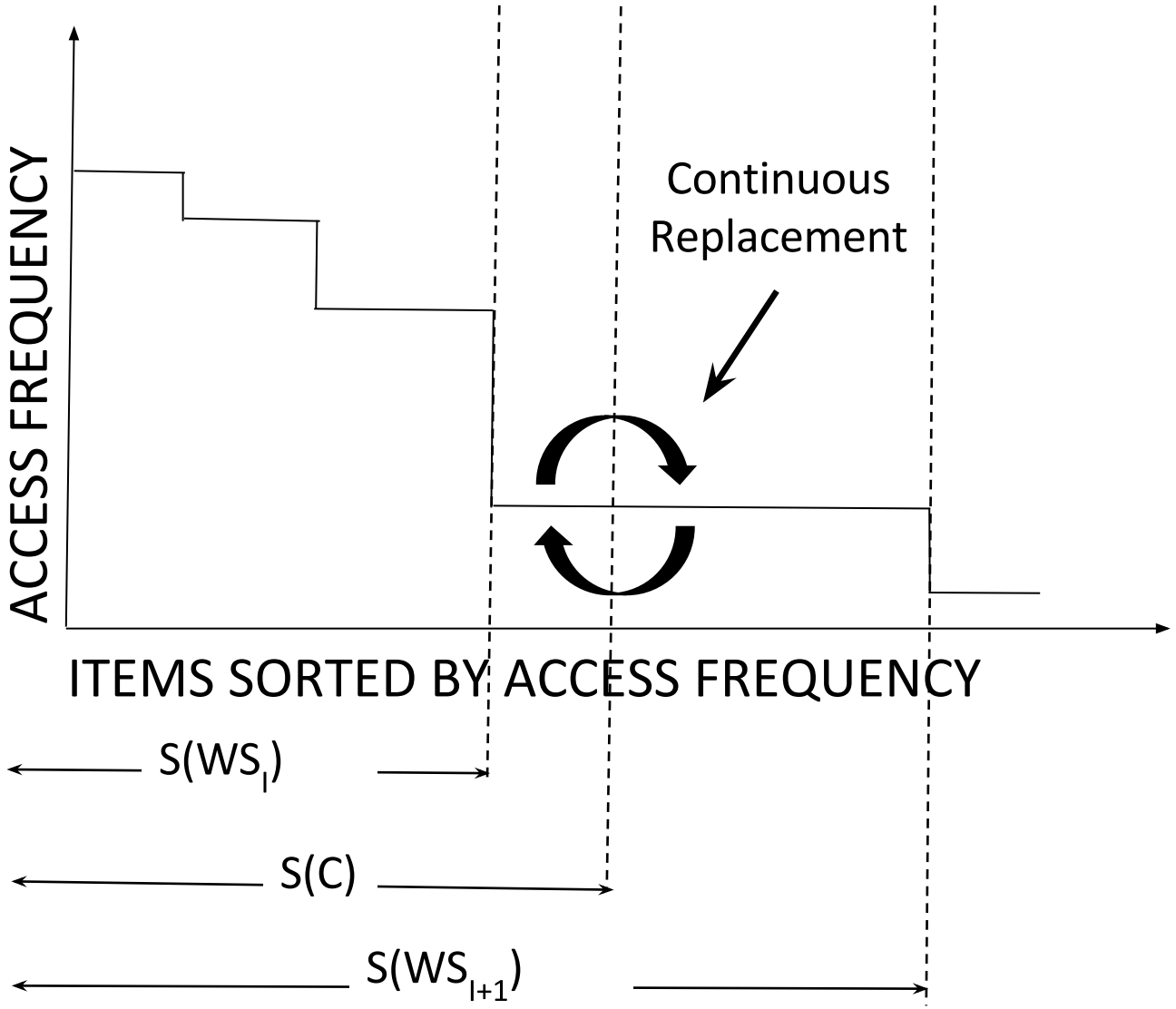
NSF

NetApp

Thank you

Questions?

What is Churning?



mARC - State Transition Table

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