Warming up Storage-level Caches with Bonfire

Yiying Zhang

Gokul Soundararajan

Mark W. Storer

Lakshmi N. Bairavasundaram

Sethuraman Subbiah

Andrea C. Arpaci-Dusseau

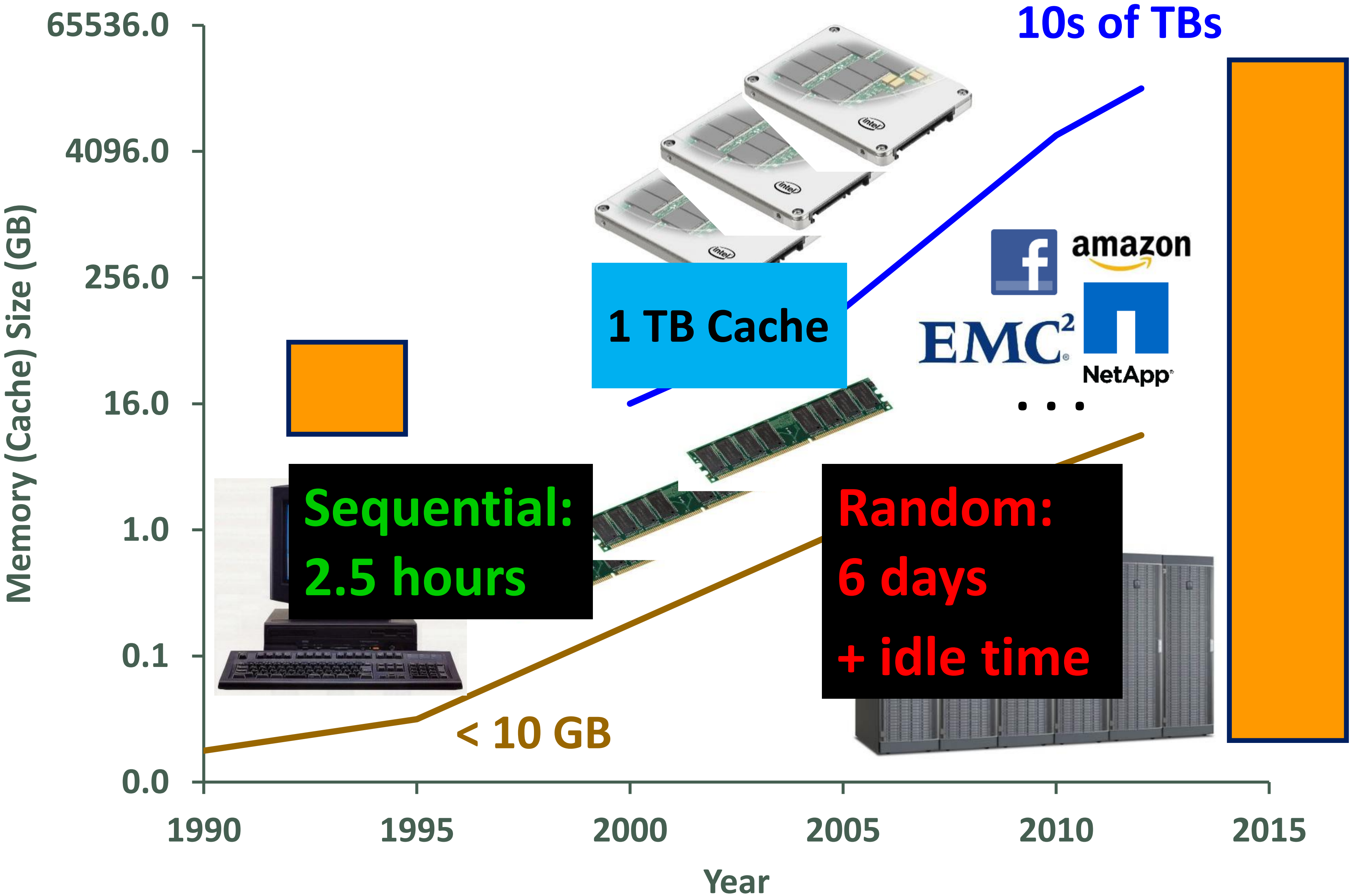
Remzi H. Arpaci-Dusseau

NetApp®

WISCONSIN

UNIVERSITY OF WISCONSIN–MADISON

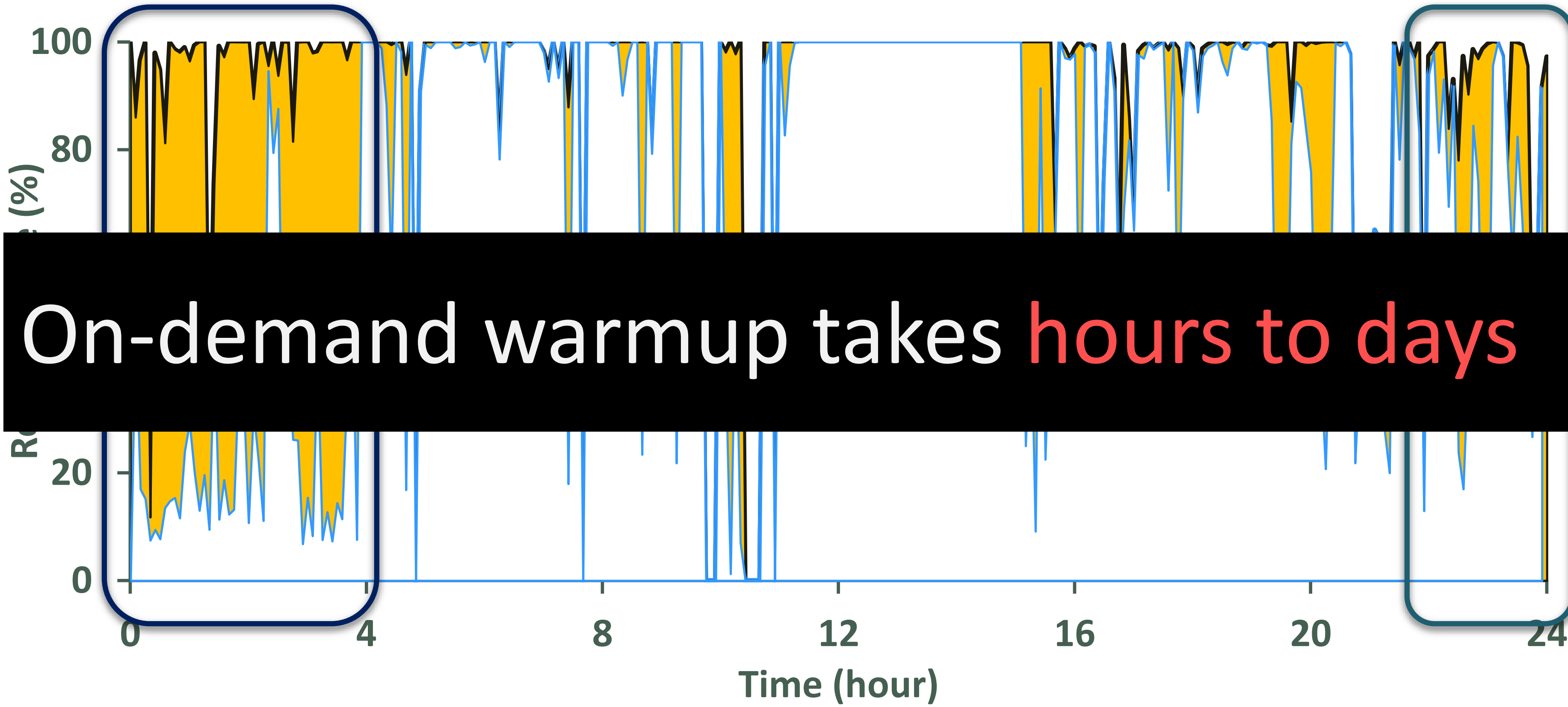
Does on-demand cache warmup still work ?



10s of TBs
1 TB Cache
amazon
EMC²
NetApp
. . .
Sequential:
2.5 hours
Random:
6 days
+ idle time
< 10 GB
65536.0
4096.0
256.0
16.0
1.0
0.1
0.0
Memory (Cache) Size (GB)
1990
1995
2000
2005
2010
2015
Year

How Long Does On-demand Warmup Take?

- Read hit rate difference between warm cache and on-demand



On-demand warmup takes hours to days

* Simulation results from a project server trace

To Make Things Worse

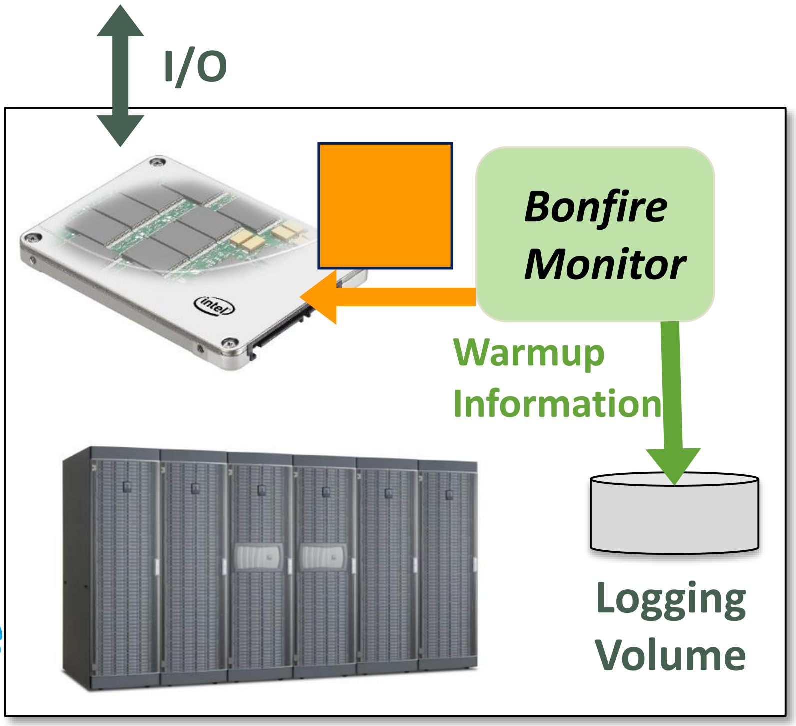
- Caches are critical
  - Key component to meet application SLAs
  - Reduce storage server I/O load

- Cache warmup happens often
  - Storage server restart
  - Storage server take-over
  - Dynamic caching [Narayanan'08, Bairavasundaram'12]

On-demand Warmup Doesn't Work Anymore What Can We Do?

- *Bonfire*
  - Monitors and logs I/Os
  - Load warmup data in bulk

- Challenges
  - What to monitor & log? **Effective**
  - How to monitor & log? **Efficient**
  - How to load warmup data? **Fast**
  - General solution



Storage System

Summary of Contributions

- **Trace analysis** for storage-level cache warmup
  - Temporal and spatial patterns of reaccesses
- Cache warmup **algorithm design and simulation**
- **Implementation and evaluation** of Bonfire
  - Up to **100%** warmup time improvement over on-demand
  - Up to **200%** more server I/O load reduction
  - Up to **5 times** lower read latency
  - Low overhead

Outline

- Introduction
- Trace analysis for cache warmup
- Cache warmup algorithm study with simulation
- Bonfire architecture
- Evaluation results
- Conclusion

Workload Study – Trace Selection

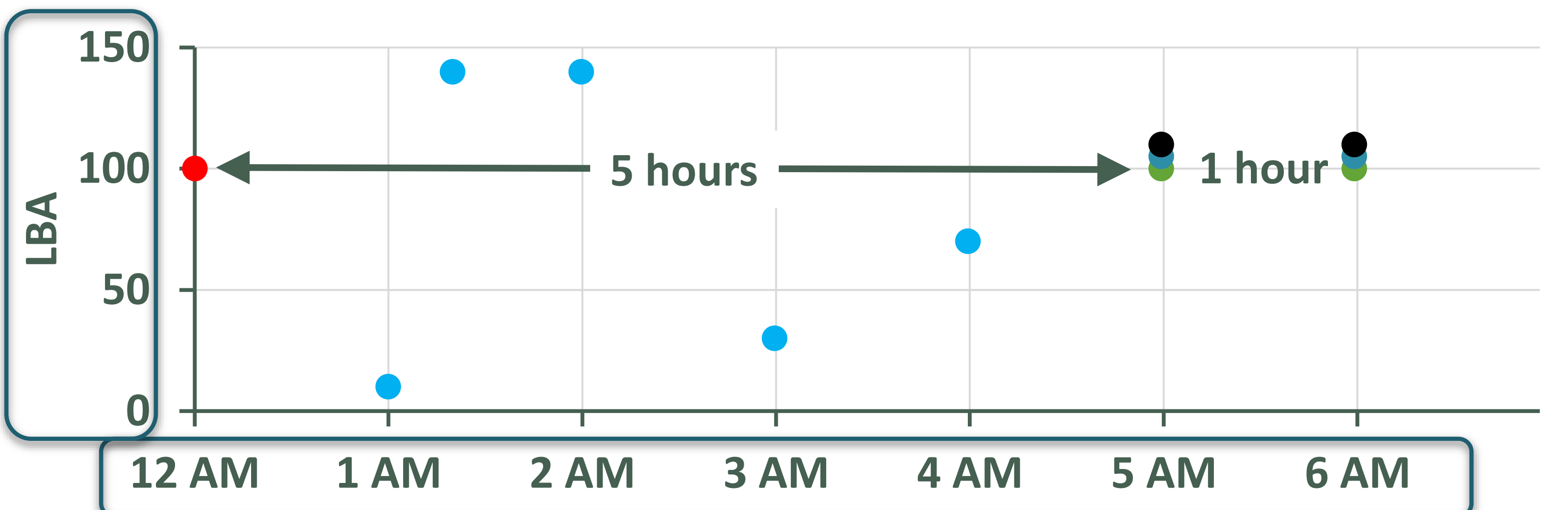
- MSR-Cambridge [Narayanan'08]
  - 36 one-week block-level traces from MSR-Cambridge data center servers
  - Filter out write-intensive, small working set, and low reaccess-rate

Server	Function	#volumes
mds	Media server	1
prn	Print server	1
proj	Project directories	3
src1	Source control	1
usr	User home directories	2
web	Web/SQL server	1

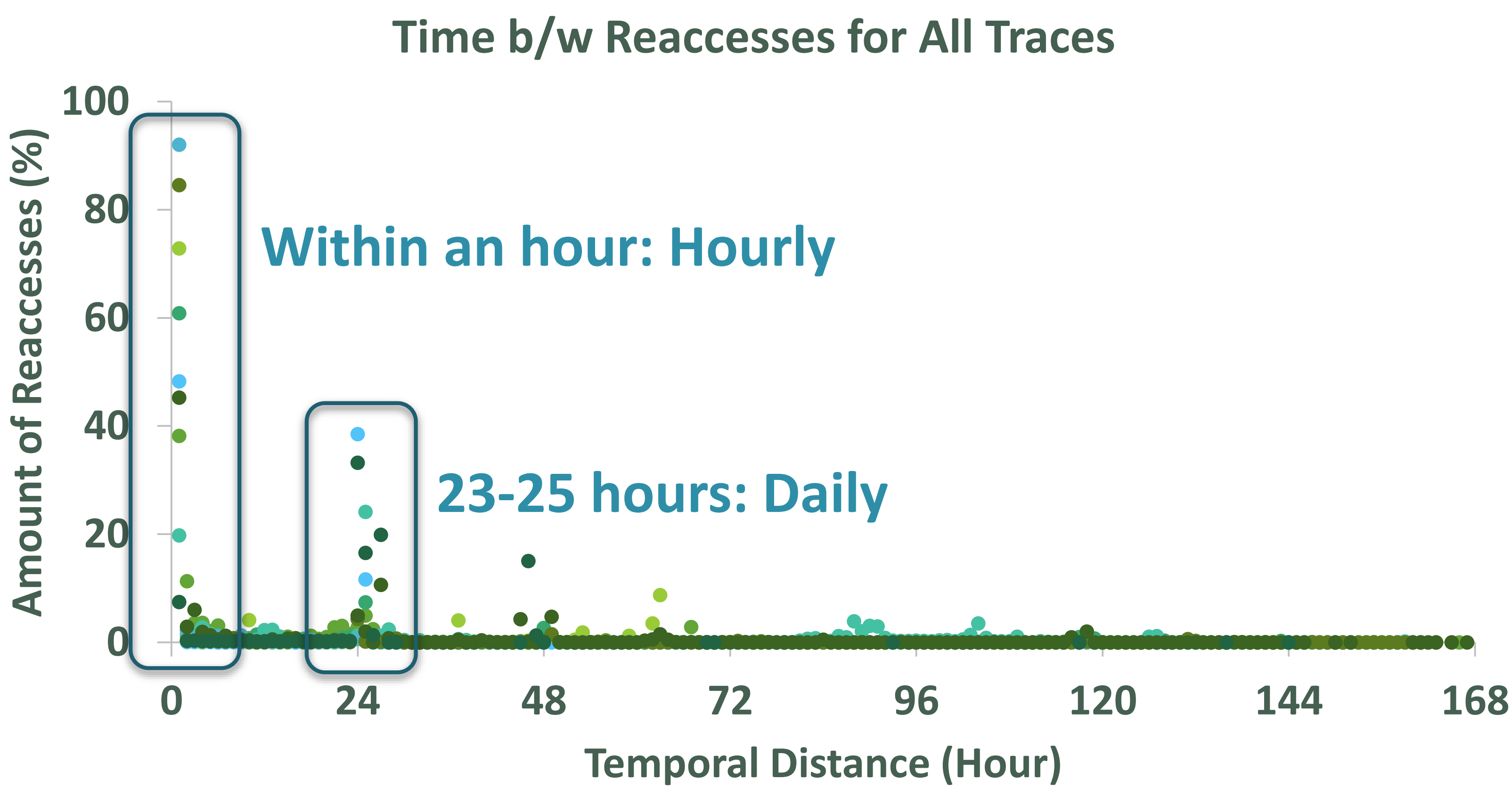
Reaccesses: Read After Reads and Read After Writes

Questions for Trace Study

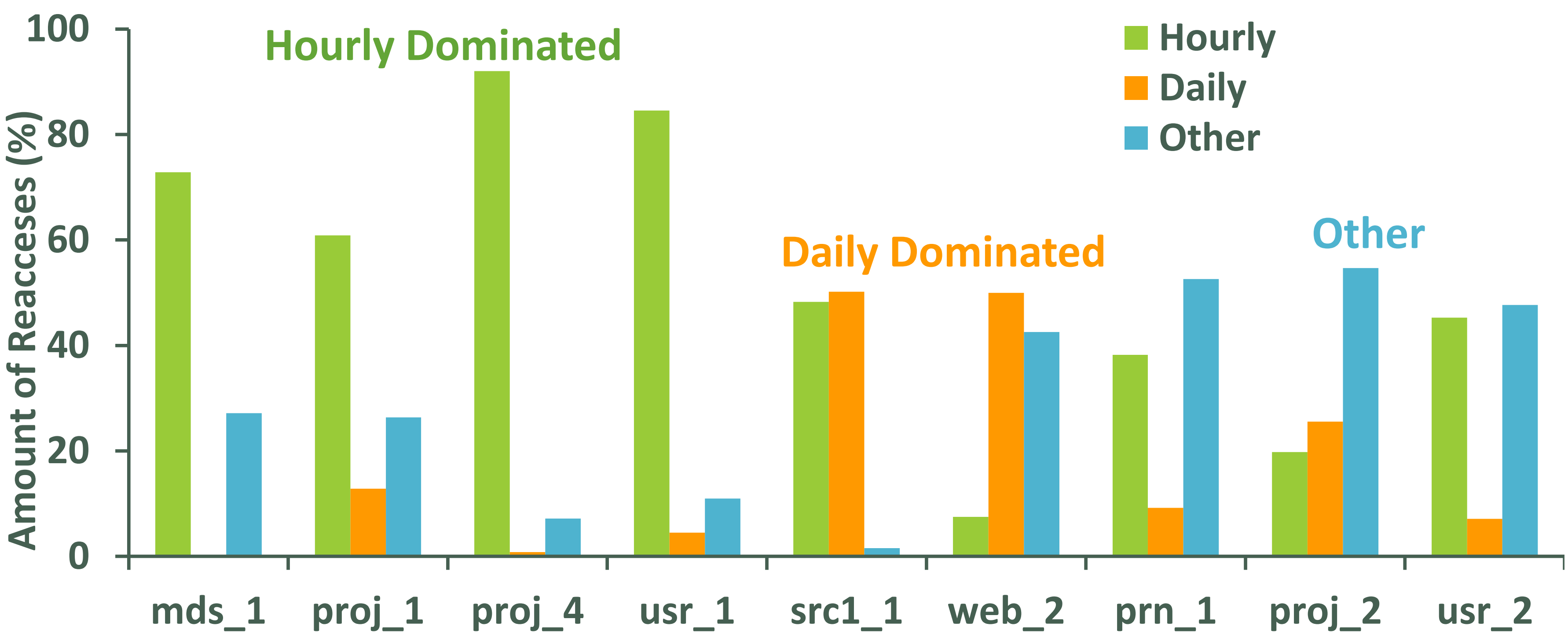
	Time	Space
Relative	Q1: What's the temporal distance?	Q3: What's the spatial distance? Any clustering of reaccesses?
Absolute	Q2: When do reaccesses happen (in terms of wall clock time)?	Q4: Where do reaccesses happen (in terms of LBA)?



Q1: What is the Temporal Distance?



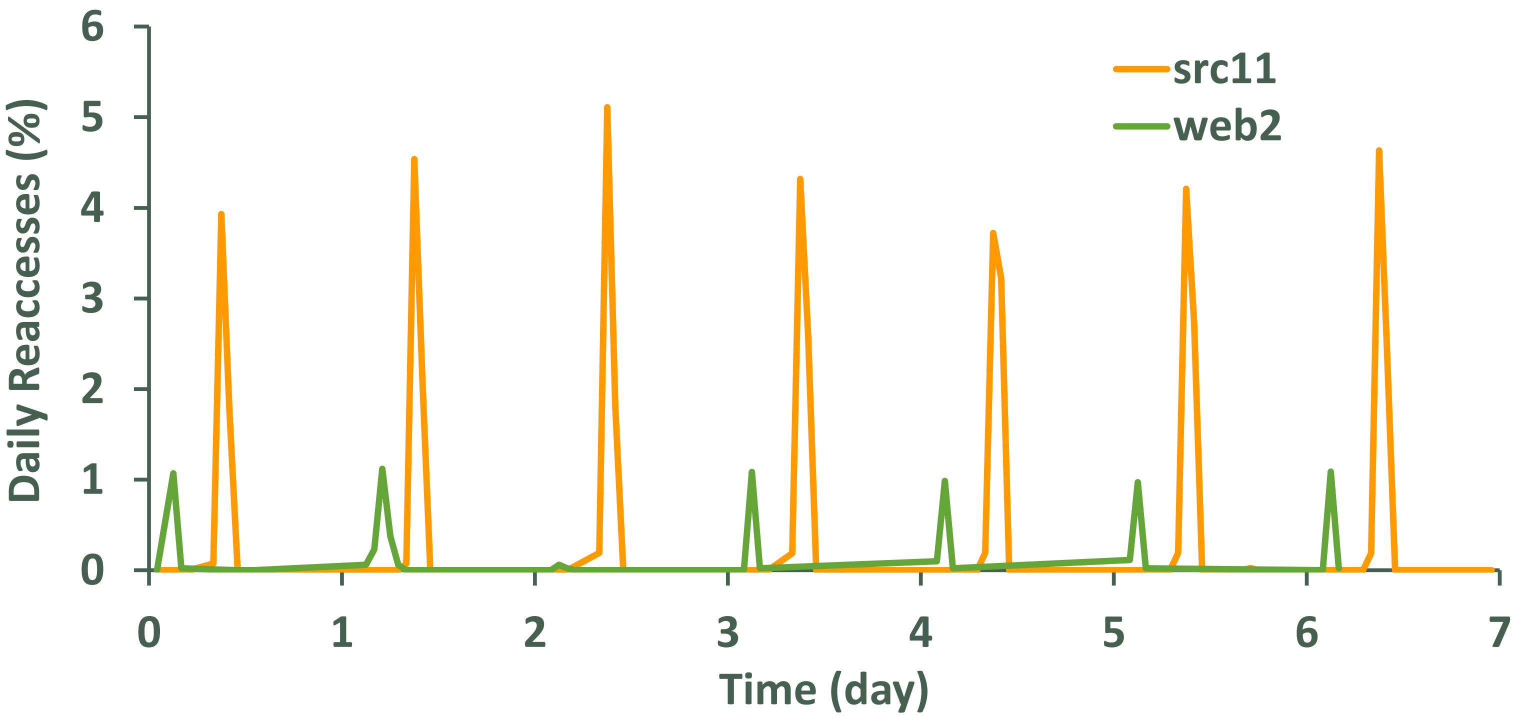
Q1: What is the Temporal Distance?



A1: Two main reaccess patterns: Hourly & Daily

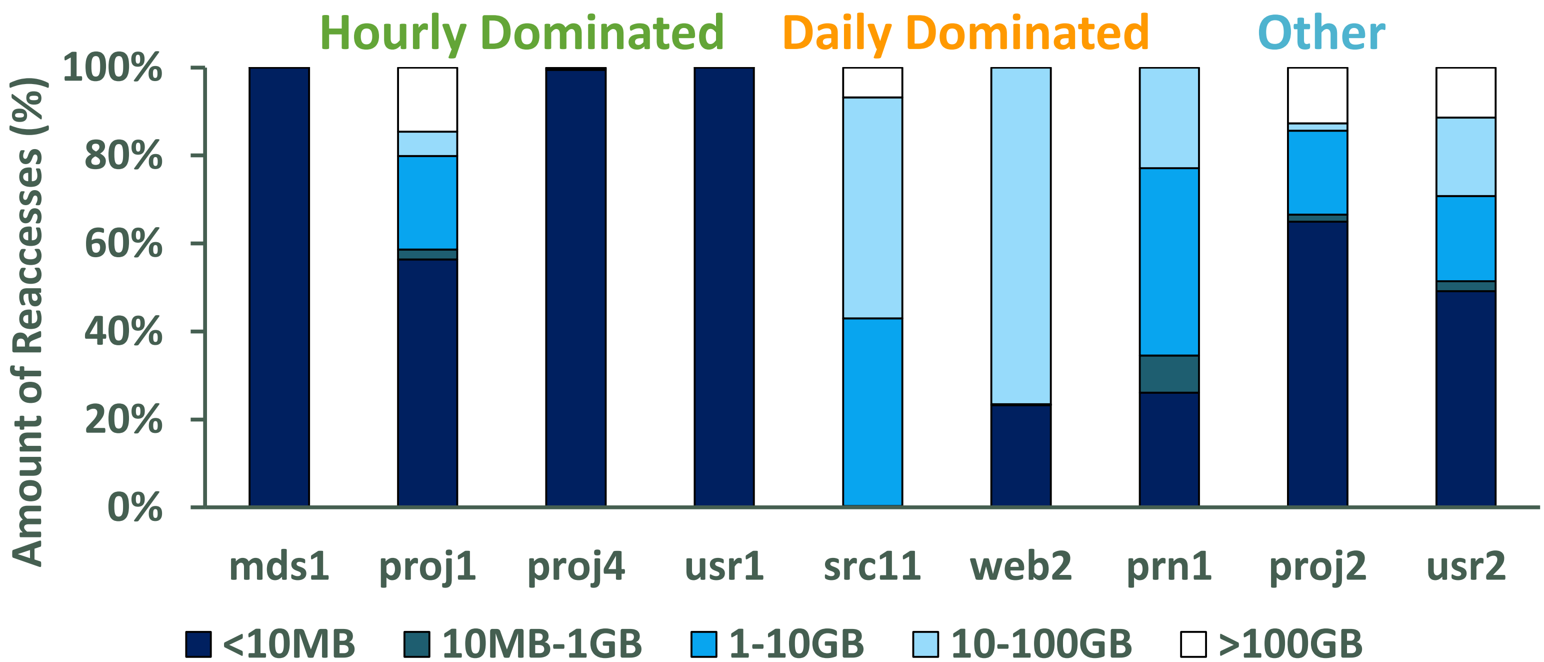
In an hour, recent blocks more likely reaccessed

Q2: When Do Reaccesses Happen (Wall Clock Time)?



A2: Daily reaccesses at same time every day

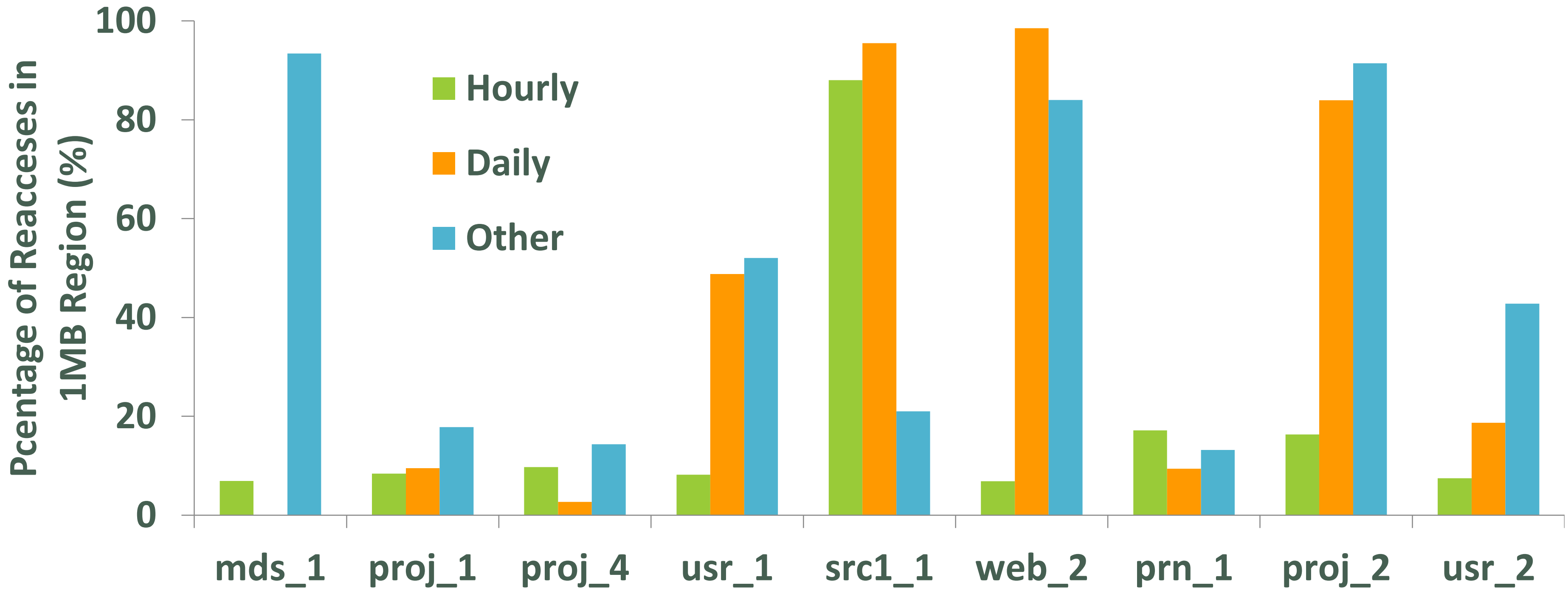
Q3: What is the Spatial Distance?



A3: Spatial distance usually small for hourly sometimes small for other reaccesses

Q3: Any spatial clustering among reaccesses?

- Percentage of 1MB regions that have reaccesses



A3: Daily reaccesses more spatially clustered

Trace Analysis Summary and Implications

	Time	Space
Relative	A1: Reaccesses have two main temporal patterns: within 1 hour, around 1 day	A3: Hourly reaccesses are close in spatial distance. Daily reaccesses exhibit spatial clustering.
Absolute	A2: Daily reaccesses correlate with wall clock time	A4: No hot spot of reaccesses in LBA space

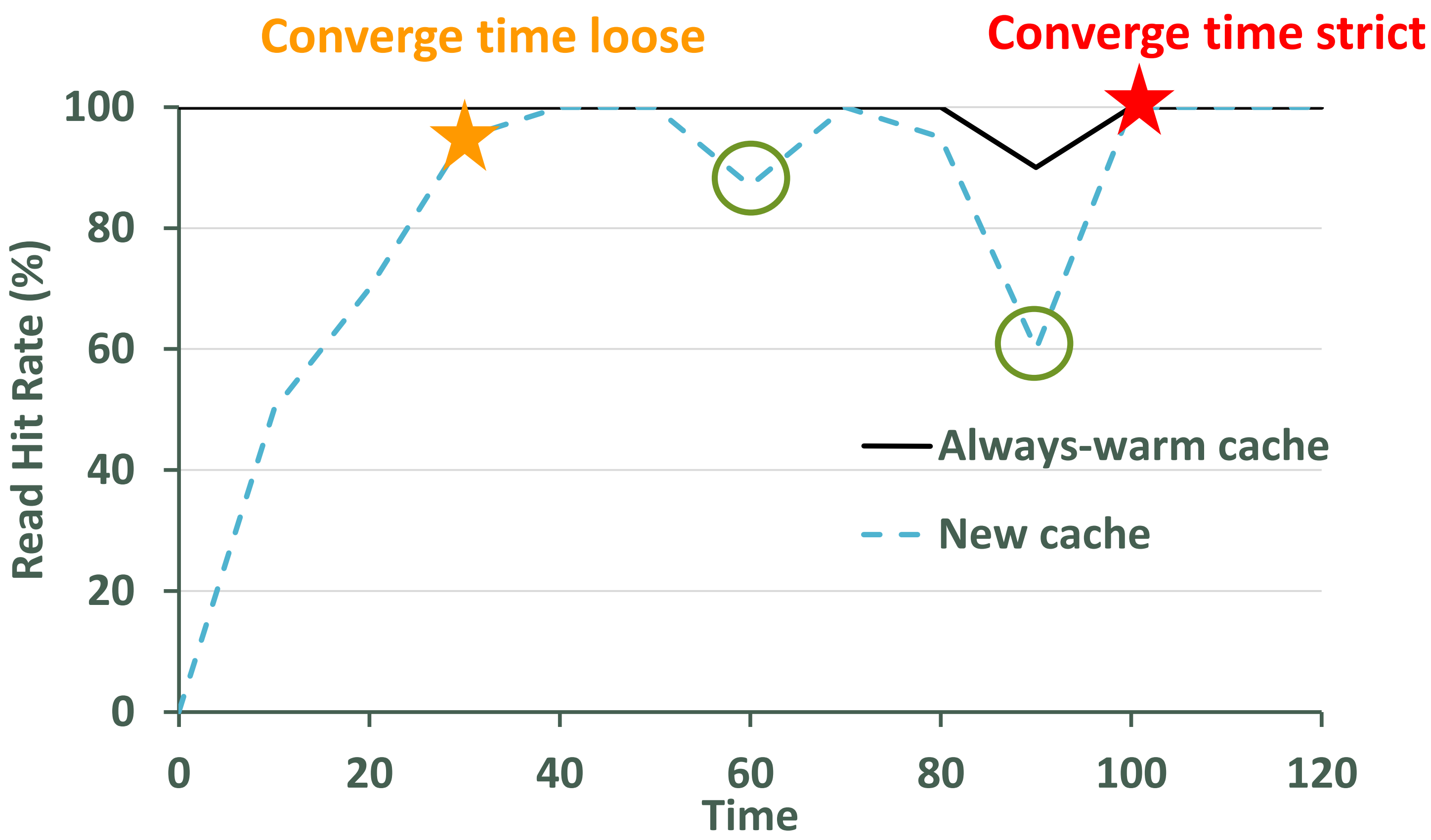
- A1 *Hourly*: Use recently accessed blocks
- A1 and A2 *Daily*: Use same period from previous day
- A3 Small spatial distance: Size of monitoring buffer is small

Outline

- Introduction
- Trace analysis for cache warmup
- Cache warmup algorithm study with simulation
- Bonfire architecture
- Evaluation results
- Conclusion

Metrics: Warmup Time

- Warmup period: Hit-rate convergence time



Metrics: Server I/O Reduction

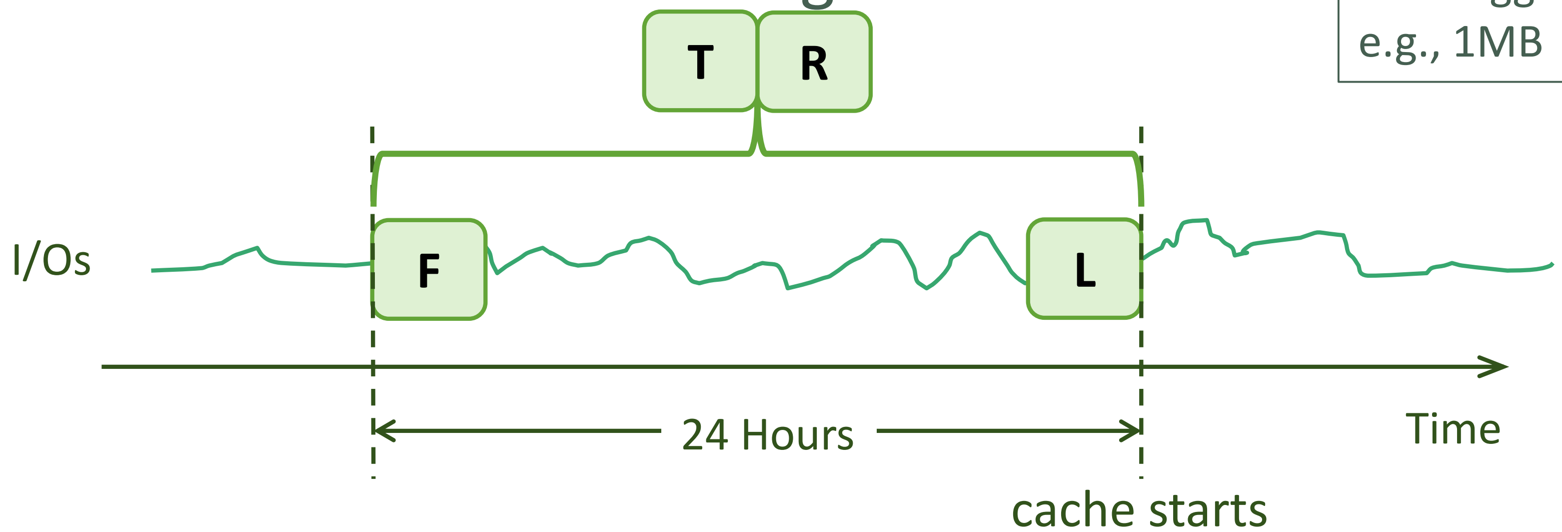
- Storage server I/O load reduction
  - $\frac{Amount\ of\ I/Os\ going\ to\ cache}{Total\ I/Os}$ during convergence time
- Improvement in server I/O load reduction
  - $\frac{Server\ I/O\ load\ reduction\ of\ Bonfire}{Server\frac{I}{O}load\ reduction\ of\ On{-}demand}$

Cache Warmup Algorithms

- ***Last-K***: Last K regions accessed in the trace
- ***First-K***: First K regions in the past 24 hours

- ***Top-K***: K most frequent regions
- ***Random-K***: Random K regions

Region: granularity of monitoring and logging e.g., 1MB



Simulation Results - Overall

- LRU cache simulator with four warmup algorithms
- Convergence time
  - Improves **14% to 100%**
- Server I/O load reduction
  - Improves **44% to 228%**
- In general, ***Last-K*** is the best
- ***First-K*** works for special case (known patterns)

Outline

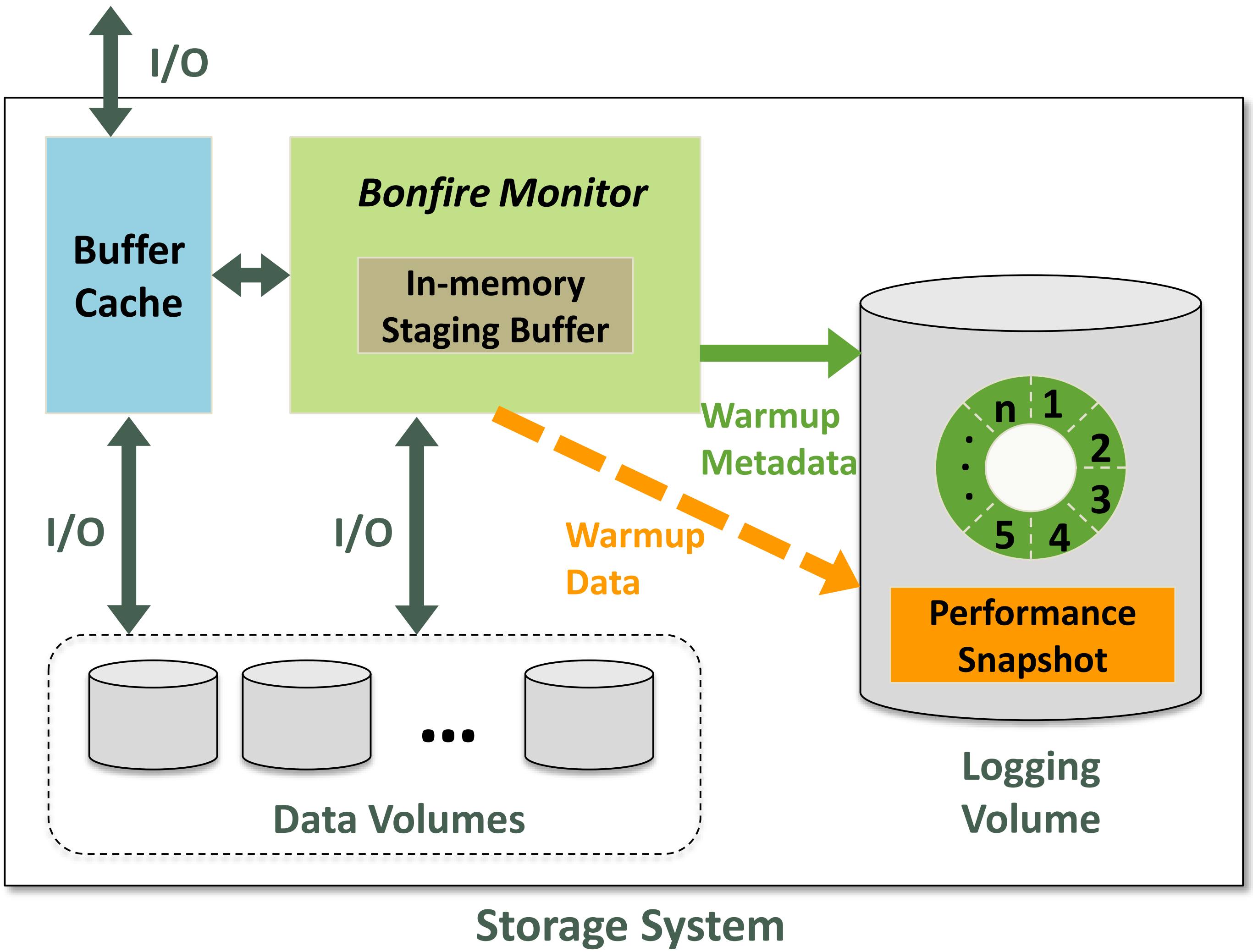
- Introduction
- Trace analysis for cache warmup
- Cache warmup algorithm study with simulation
- Bonfire architecture
- Evaluation results
- Conclusion

Bonfire Design

- Design principles
  - Low overhead monitoring and logging (efficient)
  - Bulk loading useful warmup data (effective and fast)
  - General design applicable to a range of scenarios

- Techniques
  - Last-K
  - Monitors I/O below the server buffer cache
  - Performance snapshot

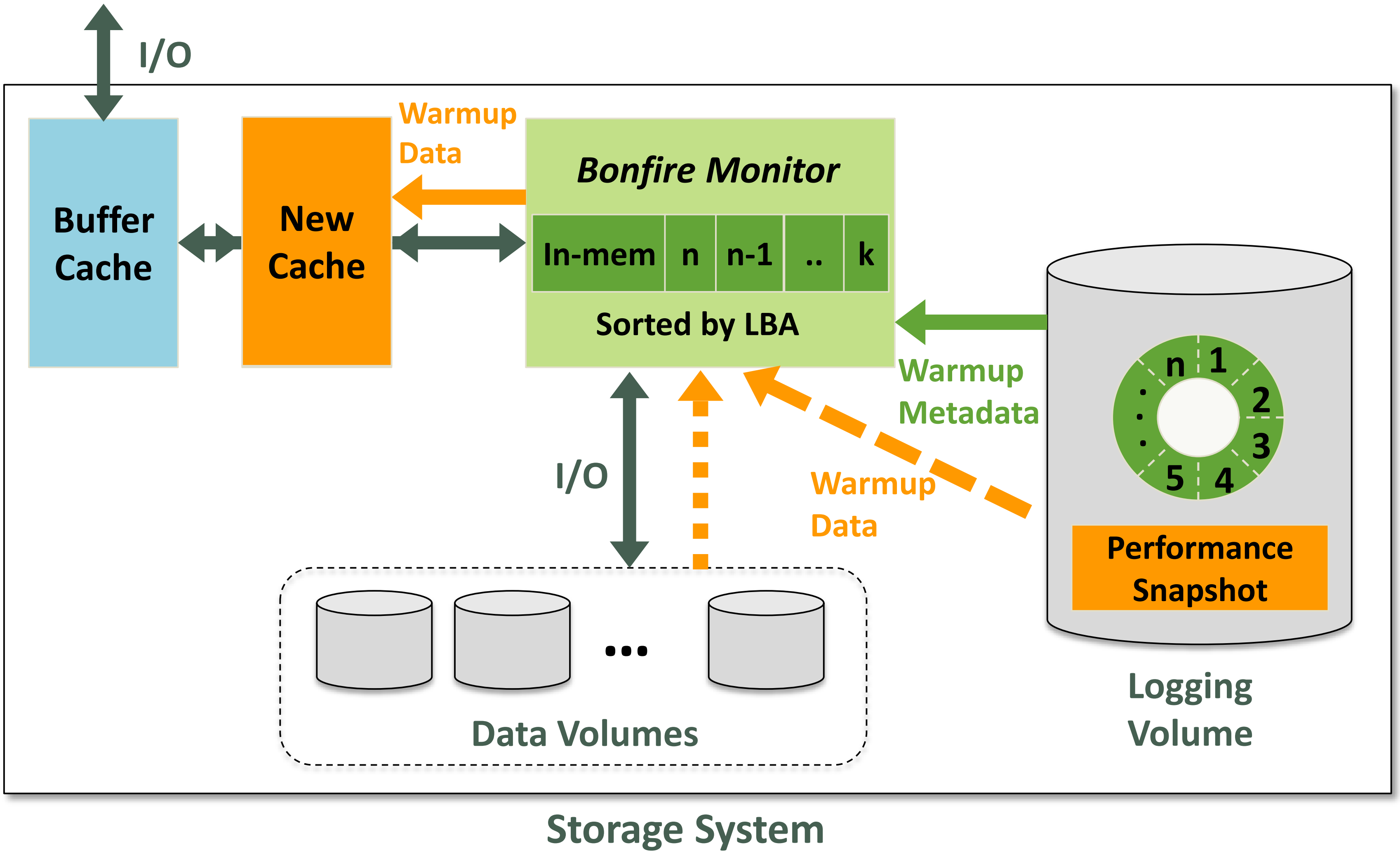
Bonfire Architecture: Monitoring



Only store warmup metadata: *metadata-only*

Store warmup metadata and data: *metadata+data*

Bonfire Architecture: Bulk Cache Warmup

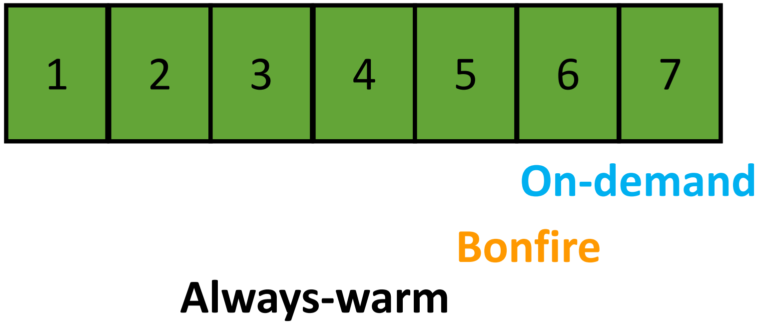


Outline

- Introduction
- Trace analysis for cache warmup
- Cache warmup algorithm study with simulation
- Bonfire architecture
- Evaluation results
- Conclusion

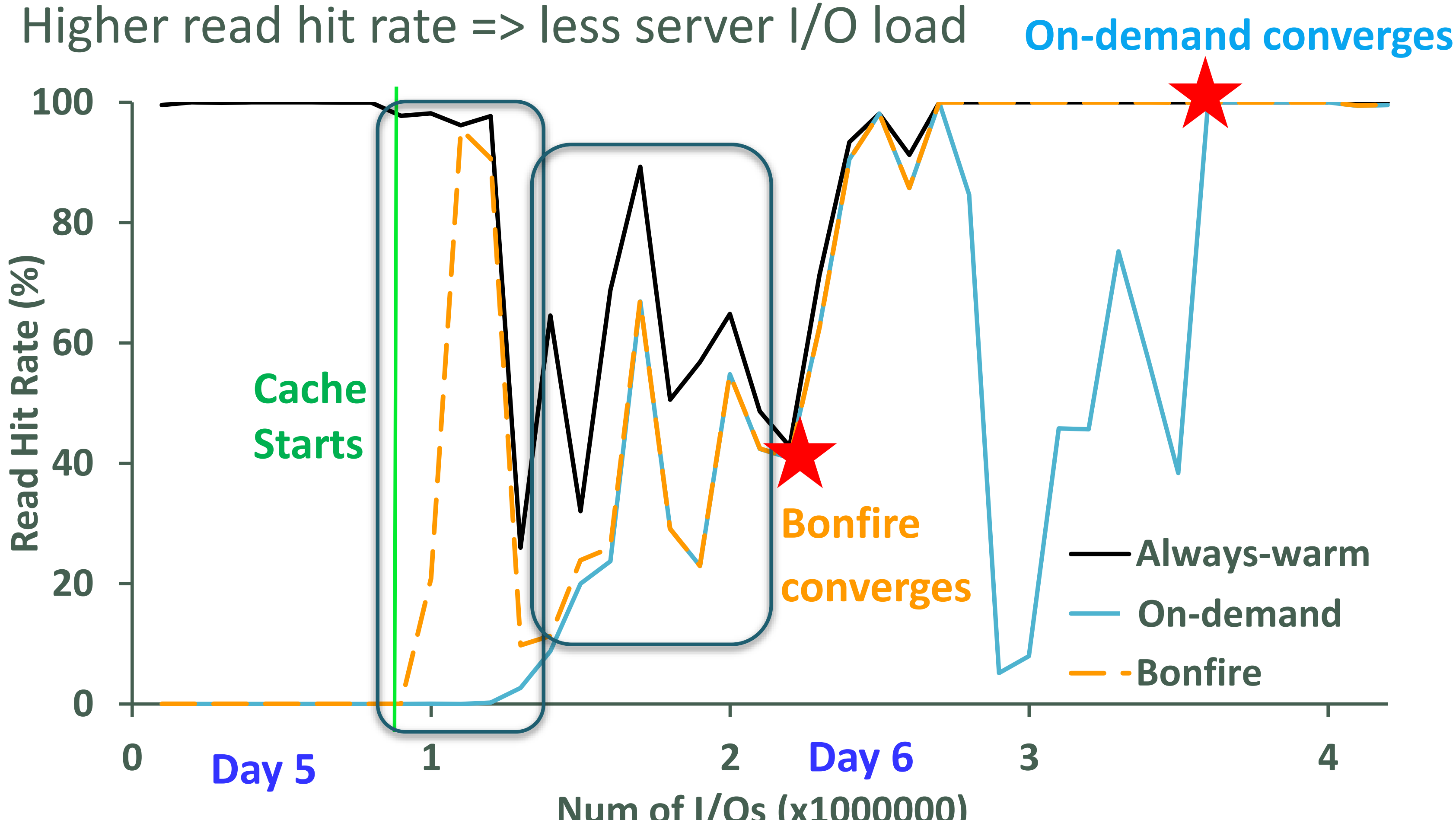
Evaluation Set Up

- Implemented Bonfire as a trace replayer
  - Always-warm, on-demand, and Bonfire
  - Metadata-only and metadata+data
  - Replay traces using sync I/Os
- Workloads
  - Synthetic workloads
  - MSR-Cambridge traces
- Metrics
  - Benefits and overheads



Benefit Results - Read Hit Rate of MSR Trace

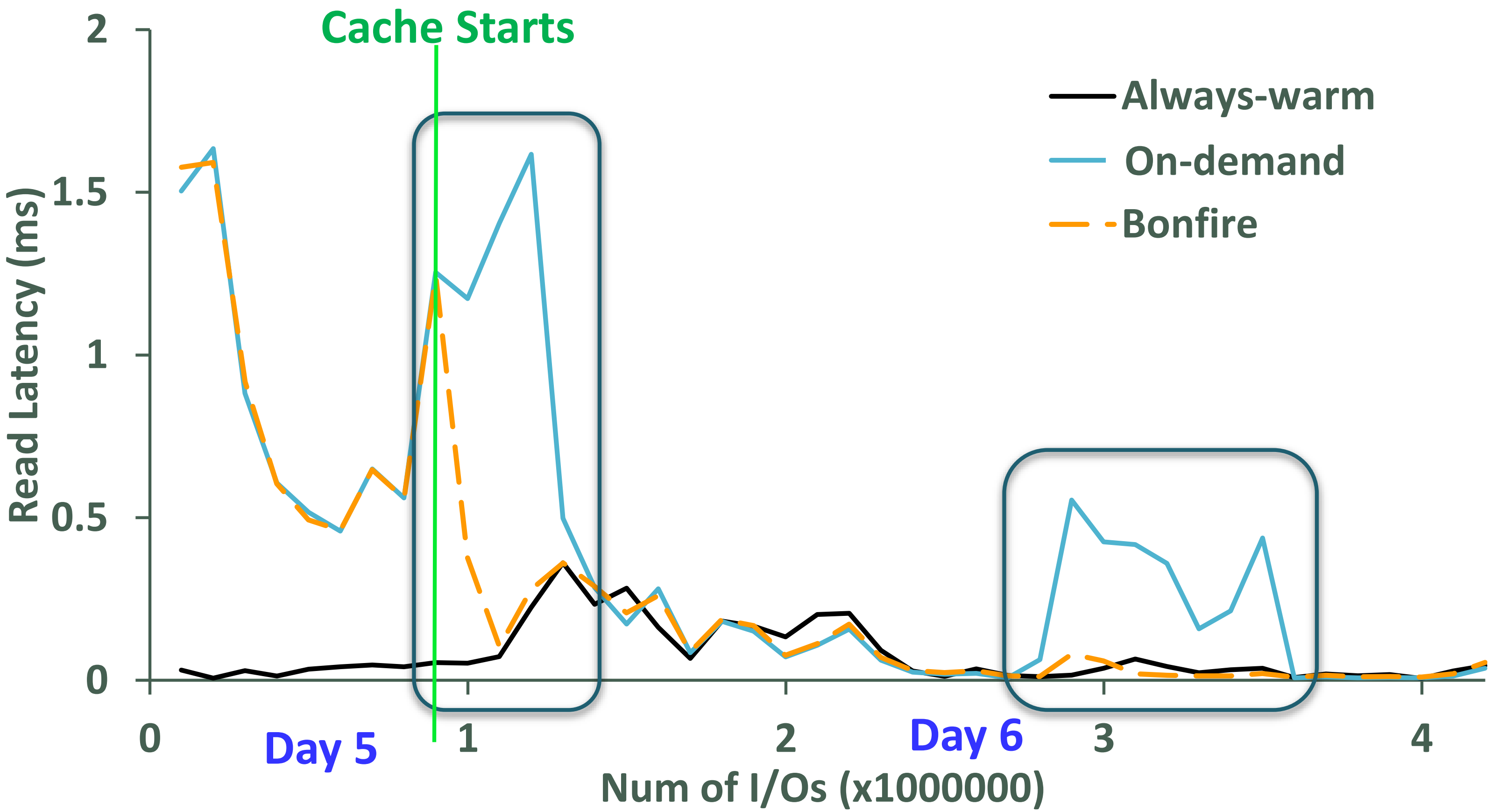
- Higher read hit rate => less server I/O load



* Results of a project server trace from MSR-Cambridge trace set

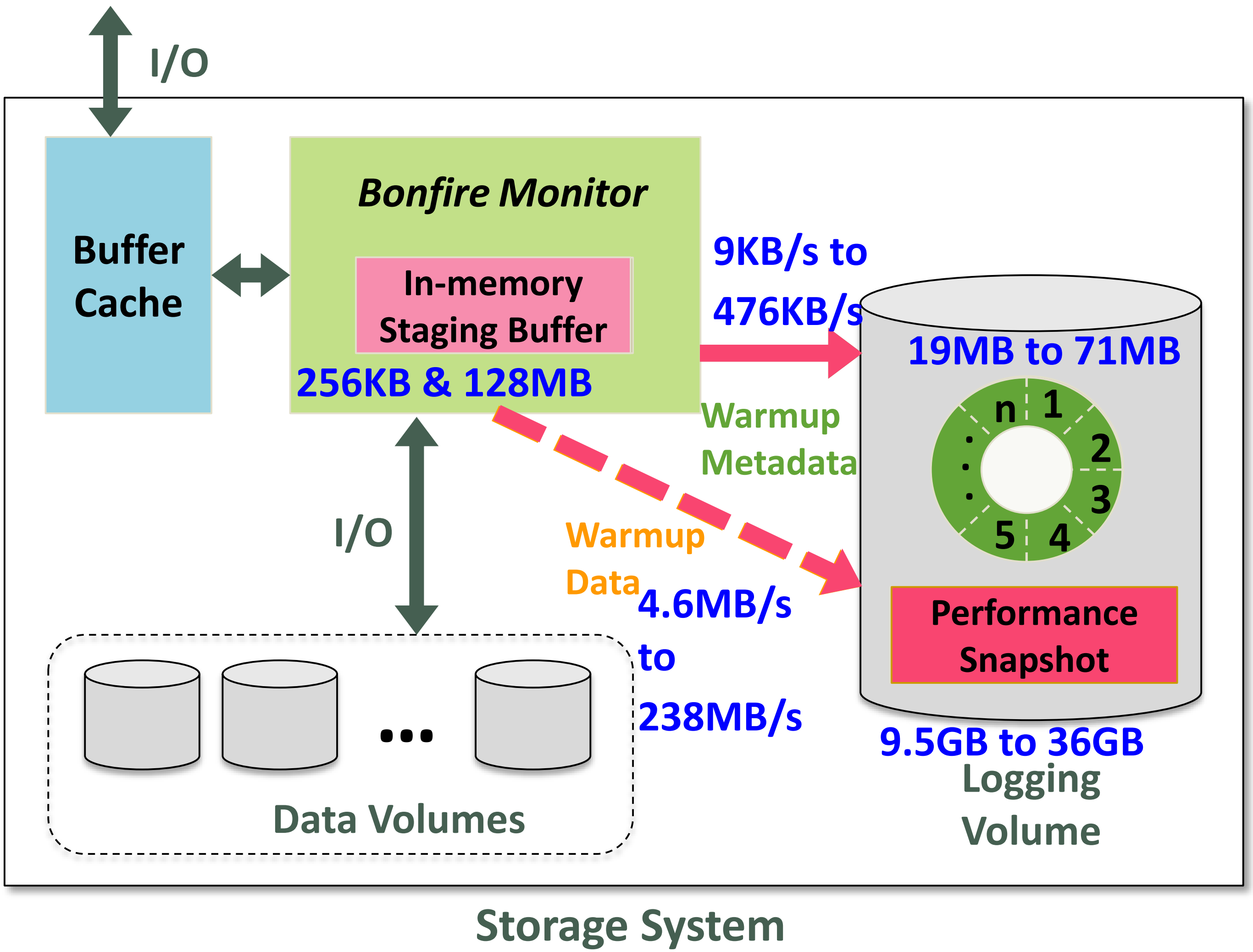
Benefit Results - Read Latency of MSR Trace

- Lower read latency => better application-perceived performance



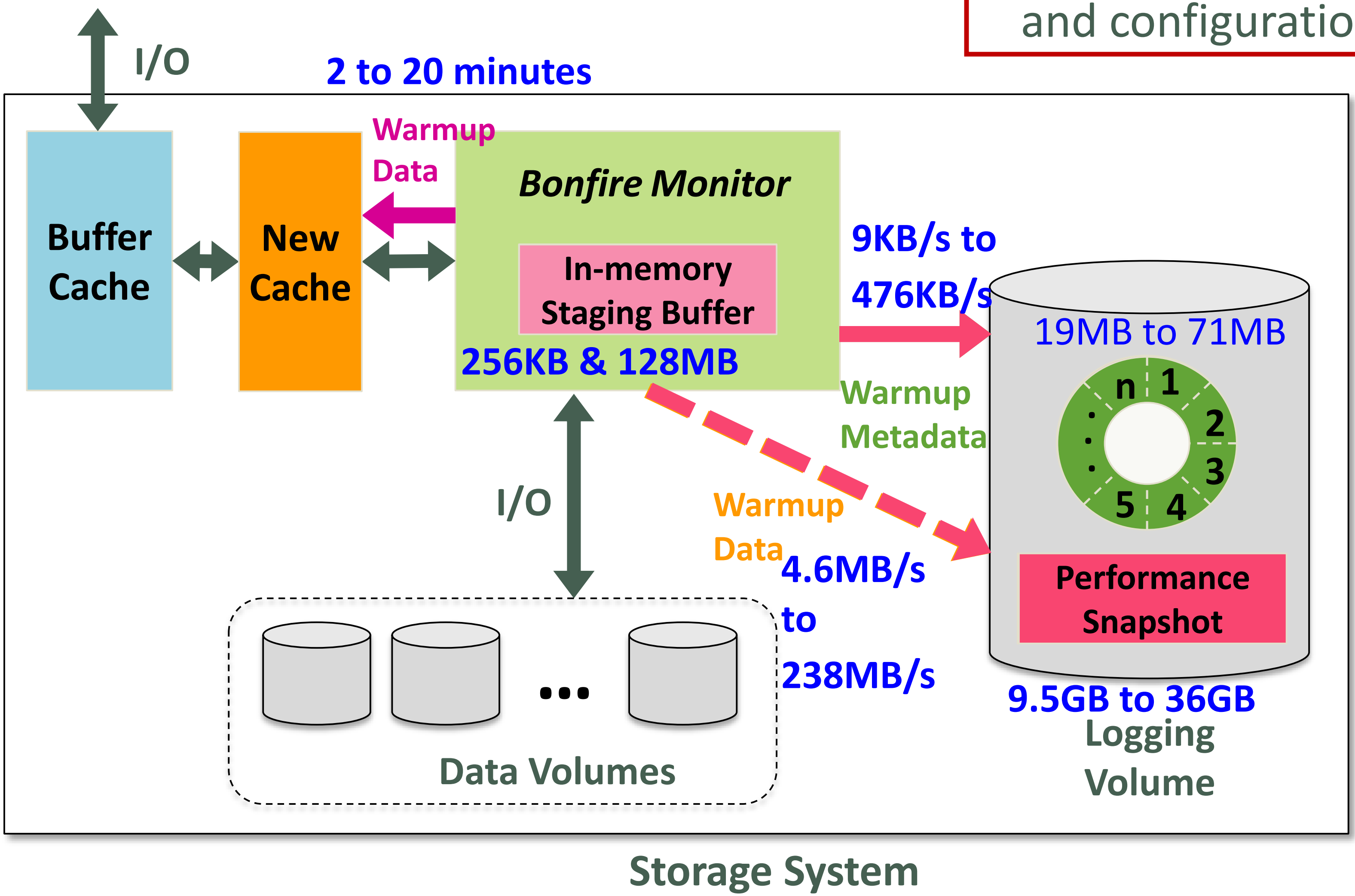
* Results of a project server trace from MSR-Cambridge trace set

Overhead Results



Overhead Results

Proper Bonfire scheme and configuration



Summary of Results

- Faster cache warmup
  - **59% to 100%** improvement over on-demand
- Less storage server I/O load
  - **38% to 200%** more reduction than on-demand
- Better application-perceived latency
  - Avg read latency **1/5 to 2/3** of on-demand
- Small controllable overhead

Conclusion

On-demand warmup doesn’t work anymore

- Warm up terabytes of caches take days

Bonfire and beyond

- Client-side cache warmup
- Application-aware warmup
- …

In need for more long big public traces !

NetApp®

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Thank You

Questions?

WISDOM http://wisdom.cs.wisc.edu/home

ADSL http://research.cs.wisc.edu/adsl