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
Does on-demand cache warmup still work?
Memory (Cache) Size (GB) vs. Year

- **1990**
  - < 10 GB

- **1995**
  - 16.0 GB

- **2000**
  - 256.0 GB

- **2005**
  - 4096.0 GB

- **2010**
  - 65536.0 GB

- **2015**
  - 10s of TBs

**Sequential: 2.5 hours**

**Random: 6 days + idle time**

**1 TB Cache**
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
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

<table>
<thead>
<tr>
<th>Server</th>
<th>Function</th>
<th>#volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mds</td>
<td>Media server</td>
<td>1</td>
</tr>
<tr>
<td>prn</td>
<td>Print server</td>
<td>1</td>
</tr>
<tr>
<td>proj</td>
<td>Project directories</td>
<td>3</td>
</tr>
<tr>
<td>src1</td>
<td>Source control</td>
<td>1</td>
</tr>
<tr>
<td>usr</td>
<td>User home directories</td>
<td>2</td>
</tr>
<tr>
<td>web</td>
<td>Web/SQL server</td>
<td>1</td>
</tr>
</tbody>
</table>

Reaccesses: Read After Reads and Read After Writes
# Questions for Trace Study

<table>
<thead>
<tr>
<th>Time</th>
<th>Space</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relative</strong></td>
<td>Q1: What’s the temporal distance?</td>
</tr>
<tr>
<td></td>
<td>Q3: What’s the spatial distance?</td>
</tr>
<tr>
<td></td>
<td>Any clustering of reaccesses?</td>
</tr>
<tr>
<td><strong>Absolute</strong></td>
<td>Q2: When do reaccesses happen (in terms of wall clock time)?</td>
</tr>
<tr>
<td></td>
<td>Q4: Where do reaccesses happen (in terms of LBA)?</td>
</tr>
</tbody>
</table>

![Graph showing temporal and spatial distances](image-url)
Q1: What is the Temporal Distance?

Time b/w Reaccesses for All Traces

Within an hour: Hourly

23-25 hours: Daily
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 reaccesses, 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

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

- **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

![Graph showing warmup period and hit-rate convergence time for Always-warm cache and New cache.](image)
Metrics: Server I/O Reduction

- Storage server I/O load reduction
  \[
  \frac{\text{Amount of I/Os going to cache}}{\text{Total I/Os}} \quad \text{during convergence time}
  \]

- Improvement in server I/O load reduction
  \[
  \frac{\text{Server I/O load reduction of Bonfire}}{\text{Server I/O load reduction of On-demand}}
  \]
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

- **Buffer Cache**
  - In-memory Staging Buffer
  - Warmup Metadata
  - Logging Volume
  - Performance Snapshot

- **Storage System**
  - Data Volumes
  - I/O

- Only store warmup metadata: *metadata-only*
- Store warmup metadata and data: *metadata+data*
Bonfire Architecture: Bulk Cache Warmup

Buffer Cache  New Cache  Bonfire Monitor  Performance Snapshot

I/O  Warmup Data  In-mem  n  n-1  ..  k  Sorted by LBA  Warmup Metadata  Warmup Data

I/O

Data Volumes  Storage System
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

Buffer Cache

Bonfire Monitor

In-memory Staging Buffer

256KB & 128MB

I/O

I/O

Data Volumes

Storage System

Performance Snapshot

19MB to 71MB

4.6MB/s to 238MB/s

9KB/s to 476KB/s

Warmup Metadata

Warmup Data

9.5GB to 36GB Logging Volume
Overhead Results

Proper Bonfire scheme and configuration

Storage System

Buffer Cache → New Cache

Bonfire Monitor

In-memory Staging Buffer

256KB & 128MB

Data Volumes

I/O

2 to 20 minutes

Warmup

Data

9KB/s to 476KB/s

9.5GB to 36GB Logging Volume

I/O

Logging

Volume

9MB to 71MB

Performance Snapshot

4.6MB/s to 238MB/s

In-memory Staging Buffer

Data In-memory Staging Buffer

2 to 20 minutes

Warmup Data

Data Volumes

Bonfire Monitor

Performance Snapshot

Proper Bonfire scheme and configuration

Buffer Cache → New Cache

I/O

Warmup Metadata

19MB to 71MB

2 to 20 minutes

Warmup Data

4.6MB/s to 238MB/s

9KB/s to 476KB/s

9.5GB to 36GB Logging Volume
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!
Thank You

Questions?

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

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