FusionRAID: Achieving Consistent Low Latency for Commodity SSD Arrays

Tianyang Jiang, Guangyan Zhang, Zican Huang, Xiaosong Ma, Junyu Wei, Zhiyue Li, Weimin Zheng

Tsinghua University
Qatar Computing Research Institute, HBKU
All-Flash Arrays (AFAs) On Rise

- Widely used in recent years

Banks

Datacenters

Clouds
All-Flash Arrays (AFAs) On Rise

- Widely used in recent years

- AFA market
  - Rapidly growing in past years
  - Growth projected to continue
  - Many products on market

[Graph showing growth from 2016 to 2023]

Data source: www.marketsandmarkets.com/Market-Reports/all-flash-array-market-41080938.html
All-Flash Arrays (AFAs) On Rise

- Widely used in recent years

- AFA market
  - Rapidly growing in past years
  - Growth projected to continue
  - Many products on market

Data source: www.marketsandmarkets.com/Market-Reports/all-flash-array-market-41080938.html
All-Flash Arrays (AFAs) On Rise

• Widely used in recent years

• AFA market
  • Rapidly growing in past years
  • Growth projected to continue
  • Many products on market

Data source: www.marketsandmarkets.com/Market-Reports/all-flash-array-market-41080938.html
Severe SSD RAID Performance Problems

- Higher latency variability compared to HDD RAID
  - Tail deviate more from norm
Severe SSD RAID Performance Problems

- Higher latency variability compared to HDD RAID
  - Tail deviate more from norm

![Chart showing variance factor comparison between clean and aged SSD and HDD arrays]
Severe SSD RAID Performance Problems

- Higher latency variability compared to HDD RAID
  - Tail deviate more from norm
Severe SSD RAID Performance Problems

• Higher latency variability compared to HDD RAID
  • Tail deviate more from norm

![Graph showing P99 latency and Median latency for HDD and SSD arrays in clean and aged states.](image_url)
Severe SSD RAID Performance Problems

- Higher latency variability compared to HDD RAID
  - Tail deviate more from norm
  - Further agitated by disk aging
Observations from Empirical Study
Observations from Empirical Study

1. Workloads usually irregular, with interleaving bursts
   - All-for-all model better than physically partitioning
Observations from Empirical Study

1. Workloads usually irregular, with interleaving bursts
   - All-for-all model better than physically partitioning

2. SSD RAID writes suffer significant software overhead
   - Much higher relative overhead than with HDD, and higher absolute overhead than with RAM
   - Mainly caused by synchronization
   - **Shorter write path desirable**

Bandwidth consumption in 4-workload mix

RAID write latency breakdown
Observations from Empirical Study

1. Workloads usually irregular, with interleaving bursts
   • All-for-all model better than physically partitioning

2. SSD RAID writes suffer significant software overhead
   • Much higher relative overhead than w. HDD, and higher absolute overhead than w. RAM
   • Mainly caused by synchronization
   • **Shorter write path desirable**
Observations from Empirical Study

1. Workloads usually irregular, with interleaving bursts
   • All-for-all model better than physically partitioning

2. SSD RAID writes suffer significant software overhead
   • Much higher relative overhead than w. HDD, and higher absolute overhead than w. RAM
   • Mainly caused by synchronization
   • Shorter write path desirable

Bandwidth consumption in 4-workload mix

RAID write latency breakdown
Observations from Empirical Study

1. Workloads usually irregular, with interleaving bursts
   • All-for-all model better than physically partitioning

2. SSD RAID writes suffer significant software overhead
   • Much higher relative overhead than w. HDD, and higher absolute overhead than w. RAM
   • Mainly caused by synchronization
   • Shorter write path desirable

3. SSD performance anomaly common, w. significant magnitude and duration
   • Found in all 6 SSD models tested, both consumer and DC
   • Latency spikes *tall and lasting enough* to be identified and sidestepped at runtime

---

**Bandwidth consumption in 4-workload mix**

- **Exchange**
- **VirtualDesktop**
- **Proxy**
- **Tensorflow**

---

**RAID write latency breakdown**

- **HDD**: 133.11ms
- **SSD**: 14.29ms
- **RAM**: 0.14ms

---

**Datacenter SSDs with random writes**
FusionRAID Overview

• New RAID design for AFAs
  • Reduces both average- and worst-case latencies
  • Works on commodity SSDs
  • Consolidates solutions motivated by individual observations
FusionRAID Overview

• New RAID design for AFAs
  • Reduces both average- and worst-case latencies
  • Works on commodity SSDs
  • Consolidates solutions motivated by individual observations

I/O requests

FusionRAID

Shared storage pool
FusionRAID Overview

- New RAID design for AFAs
  - Reduces both average- and worst-case latencies
  - Works on commodity SSDs
  - Consolidates solutions motivated by individual observations

Two-phase writes
**FusionRAID Overview**

- New RAID design for AFAs
  - Reduces both average- and worst-case latencies
  - Works on commodity SSDs
  - Consolidates solutions motivated by individual observations

---

**Diagram:**

- **RocksDB**
  - 4+1 RAID5 volume
- **Tensorflow**
  - 5+2 RAID6 volume

**I/O request processing**

- Allocation request
  - Read
  - Small writes
  - Spike detection & request redirection

- Replicated area

- SSD spike detection

- SSD pool
  - SSD\(_0\), SSD\(_1\), SSD\(_2\), SSD\(_3\), ..., SSD\(_n\)
Shared Storage Pool
Shared Storage Pool

1st-level mapping, w. dynamic block mapping table

User logical address space

Fusion logical address space

User perceived 4+1 RAID5 volume
Shared Storage Pool

User logical address space

Fusion logical address space

1\textsuperscript{st}-level mapping, w. dynamic block mapping table

2\textsuperscript{nd}-level mapping, w. static mapping function

Declustering-based stripe allocation

SSD ID + Per-SSD logical address space

User perceived 4+1 RAID5 volume

1st-level mapping, w. dynamic block mapping table

2nd-level mapping, w. static mapping function

SSD pool

MOLS
Shared Storage Pool

User logical address space

Fusion logical address space

1\textsuperscript{st}-level mapping, w. dynamic block mapping table

2\textsuperscript{nd}-level mapping, w. static mapping function

Declustering-based stripe allocation

SSD ID + Per-SSD logical address space

User perceived 4+1 RAID5 volume

RAID+ [FAST'18]
FusionRAID Optimized Writes
FusionRAID Optimized Writes

Large write request (Direct RAID write)

RAID storage

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request (Direct RAID write)

RAID storage

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request
(Direct RAID write)

RAID storage
5 10 7 14 21

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request
(Direct RAID write)

Small write request
(2-phase write)

RAID storage

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request
(Direct RAID write)

Small write request
(2-phase write)

RAID storage

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request
(Direct RAID write)

Small write request
(2-phase write)

RAID storage

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request
(Direct RAID write)

Small write request
(2-phase write)

RAID storage

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request (Direct RAID write)

RAID storage

5 10 7 14 21

Replicated storage

3 15 26 11 18

Small write request (2-phase write)

SSD pool

FusionRAID Optimized Writes
FusionRAID Optimized Writes

Large write request
(Direct RAID write)

Small write request
(2-phase write)

RAID storage

Replicated storage

SSD pool
FusionRAID Optimized Writes

Large write request
(Direct RAID write)

RAID storage

3 5 10 7 14 21

Replicated storage

0 2 12 9 28

Small write request
(2-phase write)

SSD pool
FusionRAID Optimized Writes

- Large write request (Direct RAID write)
  - RAID storage
    - In-position conversion w/o data movement
  - Replicated storage
    - Small write request (2-phase write)

SSD pool

- FusionRAID Optimized Writes
FusionRAID Optimized Writes

- **Large write request**
  - (Direct RAID write)
  - RAID storage:
    - 5 10 7 14 21
    - 3 15 26 11 18
  - In-position conversion w/o data movement

- **Small write request**
  - (2-phase write)
  - Replicated storage:
    - 0 2 12 9 28
  - Stripe reclaimed

- SSD pool:
  - Various states of SSD blocks

- RAID array:
  - Schematic representation of RAID storage and replication.
Spike Detection and Request Redirection

SSD pool
Spike Detection and Request Redirection

Per-drive monitor for light-weight, runtime spike detection
Spike Detection and Request Redirection

Per-drive monitor for lightweight, runtime spike detection

SSD pool

Straggler counter

Sliding-window based, per-SSD spike detection
Spike Detection and Request Redirection

Per-drive monitor for lightweight, runtime spike detection

SSD pool

Reactive request redirection

Straggler counter

Sliding-window based, per-SSD spike detection
Spike Detection and Request Redirection

Per-drive monitor for lightweight, runtime spike detection

SSD pool

Straggler counter

Sliding-window based, per-SSD spike detection

Reactive request redirection

5 12 24 9 28
Spike Detection and Request Redirection

Per-drive monitor for lightweight, runtime spike detection

Reactive request redirection

SSD pool

Straggler counter

Sliding-window based, per-SSD spike detection

5 12 24 9 28

3 23 11 17 5
Evaluation Overview
Evaluation Overview

- Testbed

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2 Intel Xeon E5-2650 V4</td>
</tr>
<tr>
<td>DRAM</td>
<td>128 GB</td>
</tr>
<tr>
<td>SSD</td>
<td>30 Intel D3-S4510</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu 16.04, Linux kernel 4.15.0</td>
</tr>
</tbody>
</table>
Evaluation Overview

• Testbed

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2 Intel Xeon E5-2650 V4</td>
</tr>
<tr>
<td>DRAM</td>
<td>128 GB</td>
</tr>
<tr>
<td>SSD</td>
<td>30 Intel D3-S4510</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu 16.04, Linux kernel 4.15.0</td>
</tr>
</tbody>
</table>

• Benchmark
• **Trace-driven** workloads
• **Real application** ( YCSB + RocksDB )
Evaluation Overview

• Testbed

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>2 Intel Xeon E5-2650 V4</td>
</tr>
<tr>
<td>DRAM</td>
<td>128 GB</td>
</tr>
<tr>
<td>SSD</td>
<td>30 Intel D3-S4510</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu 16.04, Linux kernel 4.15.0</td>
</tr>
</tbody>
</table>

• Benchmark
  • **Trace-driven** workloads
  • **Real application** (YCSB + RocksDB)

• Systems
  • **Commercial RAID**: 4-RAID5, RAID50
  • **Latest RAID in paper**: ToleRAID (FAST’16), LogRAID (SYSTOR’14, ATC’19)
Evaluation: Trace-driven Workloads
Evaluation: Trace-driven Workloads

- Running 4-workload mixes on compared RAID systems
- Randomly selected 20 mixes from 8 storage workloads
Evaluation: Trace-driven Workloads

- Running 4-workload mixes on compared RAID systems
- Randomly selected 20 mixes from 8 storage workloads
Evaluation: Trace-driven Workloads

- Running 4-workload mixes on compared RAID systems
- Randomly selected 20 mixes from 8 storage workloads
Evaluation: Trace-driven Workloads

- Running 4-workload mixes on compared RAID systems
- Randomly selected 20 mixes from 8 storage workloads

FusionRAID reduces median latency by 45%~81% and P99 latency by $8.3 \times \sim 35 \times$!
Evaluation: Applications and FusionRAID Overhead
Evaluation: Applications and FusionRAID Overhead

• Real application results
  • Running RocksDB on FusionRAID and RAID50
  • FusionRAID reduces tail latency by $4.1 \times$
Evaluation: Applications and FusionRAID Overhead

• Real application results
  • Running RocksDB on FusionRAID and RAID50
  • FusionRAID reduces tail latency by $4.1 \times$

• Conversion only brings **18% increase** in tail latency
Evaluation: Applications and FusionRAID Overhead

- Real application results
  - Running RocksDB on FusionRAID and RAID50
  - FusionRAID reduces tail latency by $4.1 \times$
- Conversion only brings 18% increase in tail latency
- FusionRAID without conversion consumes $2 \times$ space within running, and decrease to $1.17 \times$ if conversion on
FusionRAID: Achieving Consistent Low Latency for Commodity SSD Arrays

Thank you!