Supercharging RAID Small Writes for Heterogeneous Hardware

Rosie Wacha
Michael Sevilla
Scott Brandt
John Bent
Carlos Maltzahn

SRL/ISSDM Symposium 2012
October 23, 2012
Big Data is a Huge Problem

- Facebook sees 2.5 billion updates daily, 500TB/day
- Large Hadron Collider produces 60TB/day
- Need fast access to data
- Availability and reliability requirements encourage techniques like RAID (Redundant Array of Independent Disks)
RAID Background

RAID0: striped data with no parity

RAID4: dedicated parity

RAID5: distributed parity
Parity Limits Small Write Performance

![Graph showing the performance of RAID0 and RAID5 under varying write sizes. The x-axis represents write size (KB) ranging from 1 to 1000, and the y-axis represents throughput (MB/s) ranging from 0.1 to 100. The graph compares RAID0 (dotted line) and RAID5 (solid line) through data points illustrating performance differences.](image-url)
Parity Limits Small Write Performance

![Parity Limits Small Write Performance Diagram]

- **Throughput (MB/s)**
- **Write Size (KB)**
- **RAID0**
- **RAID5**

*Raid Overhead*
Parity Limits Small Write Performance

![Graph showing throughput (MB/s) vs write size (KB) for RAID0 and RAID5]

Throughput (MB/s) vs Write Size (KB)

- RAID0
- RAID5
Flash Solid State Drives (SSDs)

- Storage device using non-volatile flash memory
- Disk-like interface
- Attractive alternative to magnetic disks
  - 5-10X lower power
  - 2X throughput
  - 10X faster random access to data
  - 3-10X more expensive than disks
Overview

• RAIDE
  • Workload imbalance of certain write sizes
  • Place SSDs at stripe edges

• RAIDH
  • Prioritize parity placement on faster devices
RAIDE: Heterogeneous Workloads Impact RAID

- Small writes create workload imbalance in RAID5
- RAIDE uses SSDs at edges of array
- Edge of array: physical boundary between stripes
Edge Devices Work Harder

- Iostat read data for random write workload

- Read edges more often for parity construction

Edges of array $[e\ a\ b\ c\ f]$
Edge Devices Work Harder

- Iostat read data for random write workload
- Read edges more often for parity construction

![Graph showing read size distribution](image.png)

- Edges of array [e a b c f]

320KB

960KB
RAID5 Unevenness

- No reads
- Read A, parity (D)
RAID5 Unevenness

- Read A
- Read rest of stripe (C,E)
RAID5 Unevenness

- Read rest of stripe (A,B)
- Read E
**RAID5 Unevenness**

- Read D, parity (E)
- No reads
RAIDE Design

- Place SSDs at the edge of array
- No changes to software RAID
Experimental Setup

• Linux software RAID (4+1 RAIDs 64KB chunks)

  • Intel X25-E 64GB SSDs

  • Western Digital Caviar Black 640GB 7200 RPM 32MB Cache SATA 3.0Gb/s 3.5" HDDs

• Random write experiments
RAIDE Throughput Beats Naive Approach

![RAIDE Throughput Graph](image1)

![Throughput Normalized to RAID5N Graph](image2)
RAIDE Throughput Beats Naive Approach

Throughput (MB/s) vs. Write Size (KB)

Throughput Normalized to RAID5N vs. Write Size (KB)

320KB and 960KB write sizes are highlighted, showing improvements in throughput for RAIDE compared to RAID5N.
RAIDE Summary

• RAIDE up to 1.2X faster than RAID5N

• Hardware location is important

• Heterogeneous hardware impacts RAID

• How can we improve RAID algorithm to handle heterogeneity?
Heterogeneous Hardware with RAIDH

• Heterogeneous hardware
  • New arrays with SSDs and HDDs
    • Failed devices replaced with newer, faster devices
  • Weight devices based on relative speeds
    • Store more parity on faster devices
Weighted Parity Algorithm

- Input: array of weights, \( W \)
  - non-negative integers, sum to weight-sum

- Output: array of parity locations, \( P \)
  - \( \text{size}(P) \) is weight-sum, with values within array size

- e.g. \( W = \{0, 0, 0, 0, 4\} \rightarrow P = \{4, 4, 4, 4\} \rightarrow \text{RAID4} \)

- e.g. \( W = \{2, 0, 0, 0, 2\} \rightarrow P = \{0, 4, 0, 4\} \rightarrow \text{parity on edges} \)
Implementing Weighted Parity

- Edit file: src/drivers/md/raid5.c

- In `raid5_compute_sector()`

- Compute parity location using array geometry

  - `box-size = weight-sum*data-disks`
  
  - `offset = x-index%box-size`

  - `parity-index = offset/weight-sum`
RAIDH vs. RAID5 with Two SSDs + Three HDDs

- 1.2X improved throughput
Increased Unevenness: iostat Read Data

• RAIDH

• RAID5SS
Financial Trace Replay (Summary)

- Storage Performance Council traces
- Trace from one large financial institution
- Replayed 21 of 24 devices in the trace
RAIDH: 2.5X Faster Than RAID5

![Graph showing IOPS and normalized IOPS for RAIDH, RAID5SS, RAID4S, and RAID5. The graphs compare ASU Device ID on the x-axis and IOPS and normalized IOPS on the y-axis.]
RAIDH: Additional Heterogeneity

• Heterogeneous RAID adds flexibility

• RAIDH can be used in other hardware configurations
  • Larger arrays
  • More heterogeneous arrays (slow HDD+fast HDD+fast SSD, etc)
Conclusions

- RAIDE: 1.2X faster than RAID5N (same hardware)
  - No software changes
  - Exploits workload heterogeneity
- RAIDH: 2.5X faster than RAID5
  - No more complicated RAID5
  - Exploits hardware heterogeneity