The Pitfalls of Deploying Solid-State Drive RAIDs

Nikolaus Jeremic$^1$, Gero Mühl$^1$, Anselm Busse$^2$ and Jan Richling$^2$

Architecture of Application Systems Group$^1$
Faculty of Computer Science and Electrical Engineering
University of Rostock, Germany

Communication and Operating Systems Group$^2$
School IV - Electrical Engineering and Computer Sciences
Berlin University of Technology, Germany
Motivation

> Solid-state drives (SSDs) have potential to replace hard disk drives (HDDs) in performance-critical applications

> Why do we need SSD RAIDs, if SSDs are that much superior to HDDs?
  > even higher performance and reliability for applications such as Cloud Computing, OLTP
  > larger storage capacity

> Goal: Deploy fast and reliable SSD RAIDs for performance-critical applications

> Problem: Deployment of SSD RAIDs reveals pitfalls
Outline of the Talk

> Pitfalls of deploying SSD RAIDs
> Random write performance of SSDs
> Parity-less SSD RAIDs
> Parity-based SSD RAIDs
> Conclusions
Pitfall – I/O Topology and Bottlenecks

> Problem
> Most systems can handle throughput of single SSD
> Throughput of multiple SSDs sums up in SSD RAID
> Bottlenecks can occur along the path between processor cores and SSDs in a RAID

> Solution
> Evaluate the I/O topology of the used machine and eliminate bottlenecks (e.g., distribute SSDs to multiple controllers)
Pitfall – RAID Implementation

> Problem
  > Current hardware RAID controllers seem to be designed for HDDs
  > Even enterprise-class hardware RAID controllers can only handle total number of IOPS delivered by few SSDs

> Solution
  > Use software RAID to overcome the performance limitations of hardware RAID
  > Bottlenecks in software RAID implementations can still occur though at higher performance level
Pitfall – Asymmetry between Read/Write Speed

> Problem
  > Reading flash pages faster than writing
  > Writes in parity-based RAIDs slower than reads due to Read-Modify-Write operations
  > Effects can accumulate:
    Even faster reads and slower writes

> No simple solution
  > Property of flash memory

Source: Winbond
Pitfall – Synchronous SSD Aging

> Problem
  > SSDs have limited number of erase cycles
  > Lifespan of SSD depends on write workload
  > In RAIDs writes often distributed equally to all drives
    → Multiple drives may wear out at the same time

> Potential solution
  > Distribute parity unevenly to available drives
    (e.g., use dedicated parity drives)
  > Usefulness unclear as drives can actually fail after higher or lower number of erase cycles than expected/guaranteed
Pitfall – Workload History Dependency

> Problem

> Flash memory requires out-of-place updates and block-wise erasure leading to fragmentation if spare capacity is rare

> Fragmentation degrades performance especially for sustained random writes

> Additionally random write dominated workloads maximize fragmentation

> Fragmented drive contains mainly blocks with many occupied pages

> Less fragmented drive contains several blocks that have only few occupied pages
Pitfall – Workload History Dependency

> Solution

> Increase spare capacity to ensure that enough free flash blocks will be available anytime
> Garbage collector has to reclaim less flash blocks lowering the number of writes to move valid data
> Write amplification decreases substantially providing a much higher random write performance
Random Write Performance of SSDs (I)

> Visualization of SSD fragmentation
Random Write Performance of SSDs (II)

> Higher spare capacity improves sustained random write speed of a single SSD up to **19x**
> Throughput increases from **600 IOPS** to **11,500 IOPS**
> Will the situation be different for SSD RAIDs?

Source: Intel
Parity-less SSD RAIDs: Random Write Performance (I)

- Speedup factor for parity-less RAIDs (0,1,10) should be same as for single SSD because writes are evenly distributed
- Random write performance increases up to 15x - 19x
- Speedup factor slightly lower with more drives especially when combined with high spare capacity
Random write throughput increases up to **43,000 IOPS**

Throughput scales almost straight proportional with the number of drives except for 5 drives.
Parity-based SSD RAIDs: Random writes

> Prediction of random write performance more complex because random writes to RAID device incur reads
  > Speedup factor depends on number of write and read operations required to serve write request to RAID device
  > But spare capacity affects only write throughput
    → Speedup factor will be different from single SSD

> Goals
  > Predict speedup factor without measurements for considered SSD RAID setups
  > Approximate speedup factor for RAID device based on speedup factor of single SSD
Parity-based SSD RAIDs: Model

> Speedup factor for RAID device derived from speedup factor of single SSD and scalability factor

\[
\frac{w_2}{w_1} = S_f \cdot \frac{w_{\text{max},2}}{w_{\text{max},1}}
\]

Scalability factor for RAID device for RAID 5 with 4+ and RAID 6 with 6+ drives

\[
S_f = \frac{1+f_1}{1+f_2}, f_i = \frac{w_{\text{max},i}}{r_{\text{max}}}
\]

> Scalability factor for RAIDs with less drives detailed in paper
Parity-based SSD RAIDs: Evaluation (I)

- Random write performance increases up to **9x - 14x**
- Measured speedup factor 20% lower than expected
- Scaling problem for 4+ drives with high spare capacity
Parity-based SSD RAIDs: Evaluation (II)

Random write performance increases up to 11,000 IOPS
Scaling problems with high spare capacity
Conclusions

> Several pitfalls can prevent SSD RAIDs from exploiting their full potential

> Spare capacity increase improves sustained random write performance in SSD RAIDs significantly

> Parity-less SSD RAIDs superior to parity-based

> Scalability issues can arise for higher number of drives in combination with large spare capacity
Future Work

> Improve our performance prediction model
  > Consider interaction between reads and writes
  > Extend model to
    > predict sequential write performance
    > predict read performance

> Investigate performance issues

> Explore properties of hybrid RAID setups (SSD & HDD)
Discussion

Thank you for your attention!

Nikolaus Jeremic
nikolaus.jeremic@uni-rostock.de
http://www.ava.informatik.uni-rostock.de