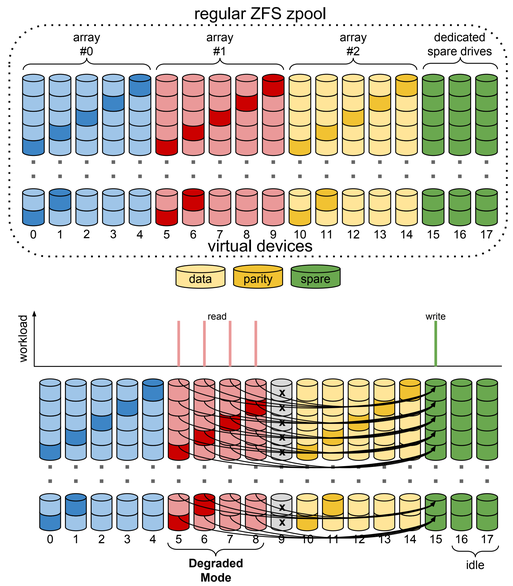
# Exploring Declustered RAID in ZFS for improved Storage Reliability and Availability

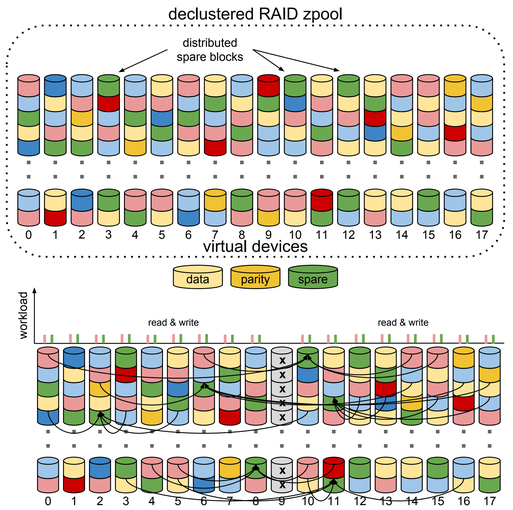
Zhi (George) Qiao [1] Song Fu [1] Hsing-bung Chen [2] Bradley Wade Settlemyer [2]

[1]University of North Texas, [2]Los Alamos National Lab

## Declustered RAID




In declustered RAID, all of the drives participated in recovery when disk failure onset.




Declustered RAID have following benefits:

1. Shorter rebuild time, less risk of data loss.
2. No dedicated spare drive, higher aggregated I/O.
3. Evenly distributed recovery workload.

## Performance Evaluation

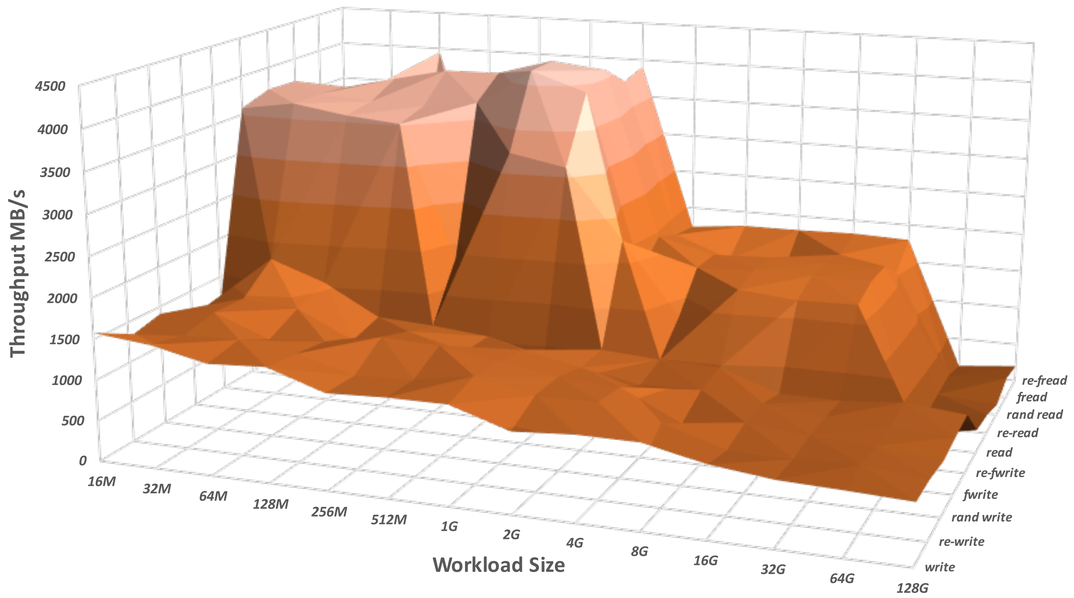



Figure 1: Performance profiling of dRAID3 using 8:3 parity ratio, total of 36 HDD including 3 spare disks at 85TB capacity.

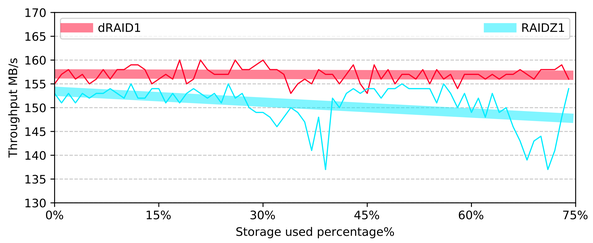



Figure 2: dRAID has marginal advantage for application I/O, since there is no dedicated spare drive.

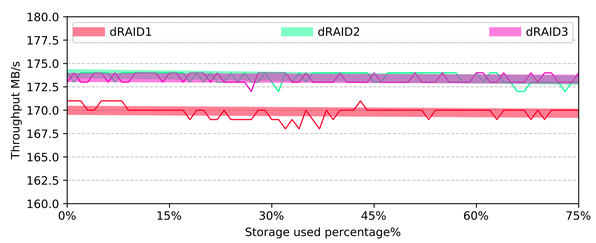



Figure 3: For the same parity ratio(4:1 as shown), the performance of draid1, draid2, and draid3 is very close.

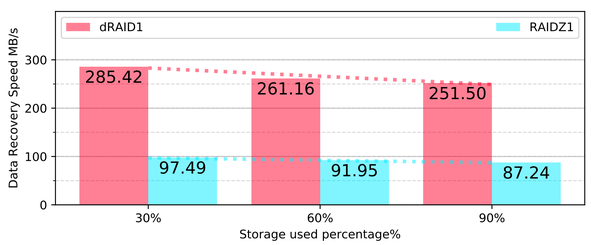



Figure 4: dRAID significantly outperforms regular ZFS resilvering, and the recovery performance degrades when available space decreases. (cfg=(1:1)x3+1)

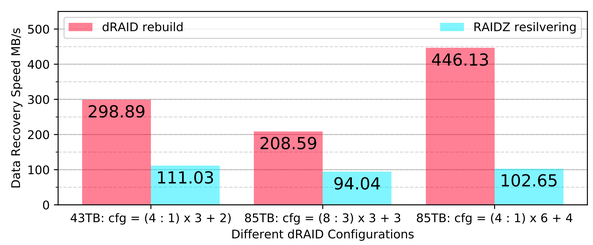



Figure 5: The number of drives participating in data reconstruction determines the recovery performance. The declustered RAID recovery speed improvement is scalable and has low overhead.

## Term & Methodology

The dRAID source code is available at `https://github.com/thegreatgazoo/zfs`.

| RAID Level | ZFS eqv. | dRAID eqv. |
|---|---|---|
| RAID 1 | mirroring | mirroring |
| RAID 5 | raidz1 | draid1 |
| RAID 6 | raidz2 | draid2 |
| RAID 7 | raidz3 | draid3 |

Table 1: RAID levels in ZFS and dRAID.

We use $cfg = (d : p)$ x $k + s$ to represent the configuration, where $d$ = data, $p$ = parities, $k$ = number of redundant arrays, and $s$ = spare. The following model is used to analyze declustered RAID speed-up ratio $\Phi$ during recovery. $f$ = number of failure drives.

$$\Phi = H \times \frac{k \times (d+p) + s - f}{d + p - f} \qquad (1)$$

$H$ indicates the dRAID overhead where $0 < H < 1$. When recovery speed $\frac{S_{empirical}}{S_{analytical}} \rightarrow 1$, the system overhead is low.

## Acknowledgements

Special thanks to LANL scientist Parks Fields and Scott White who provide insight and expertise that greatly assisted the research. We would also like to show gratitude to our colleagues from USRC during the course of the research. This publication has been assigned an LANL identifier LA-UR-18-26964.

## References

[1] Thomas JE Schwarz, Jesse Steinberg, and Walter A Burkhard. Permutation development data layout (pddl). In *High-Performance Computer Architecture, 1999. Proceedings. Fifth International Symposium On*, pages 214–217. IEEE, 1999.

[2] INTEL FEDERAL LLC. *dRAID: Declustered RAID for ZFS.* 2017.