CRITIQUE - 8 [Lecture - 9 Redundancy and Deduplication]

The main topic of this lecture is redundancy and deduplication. We discussed cost, performance and reliability aspects of redundancy. Some of the recent trends and techniques to recover the silent and noisy data corruption. We discussed the need for traditional volume managers, mirroring, error correcting codes (ECC), end-to-end data integrity and self-healing data in ZFS. we talked about the flaws in RAID-4 and RAID-5 and how ZFS addressed these issues in RAID-Z. Later we dove into the discussion on deduplication. we talked about different where we can leverage deduplication, various levels of deduplication - file level, block level, and byte level.

The lecture started with providing the experimental results of measurements done at CERN. The results report around 150 silent data corruptions over a span of 3 weeks. Hardware detects only noisy errors. So we need a good end to end verification to catch software errors. We can checksum the data to detect the silent errors and recover by mirroring the data in multiple drives or using RAID technology. The demand for more and more storage space rose and people started adding more disk. This broke the traditional file system(FS) since it was designed to manage only one disk at a time. To overcome this problem and avoid rewriting of the FS, a shim layer called "Volume manager" was added. Volume managers apart from supporting multiple disks, also provide mirroring of the data. Victor did a good job explaining the history, need for redundancy and volume manager.

Next, we talked about data redundancy through mirroring. Mirroring can be done with hardware or software. A volume manager is responsible for mirroring the data. The diagrams and explanation given for the flow of reading or writing of the data from the disks were lucid. I learned how a bad block could be passed back to the application that issued the read due to undetected silent errors. However, if the volume manager detects an error it will issue that read to different disks and correct the bad block on the other disk using majority voting. The discussion on how to leverage the performance of mirroring was interesting and I learned that with P2P we can have a better performance and reliability but with increased cost.

The discussion on error correction codes (ECC) was a bit hard to follow and felt it could have been explained better and in depth. The end-to-end data integrity in ZFS is done through checksum. The side by side comparison of disk checksum and ZFS checksum helped me to understand how checksum works in ZFS, how it resolves the trivial problem that was inherent in the traditional checksum. Storing the checksum in parent pointers than in the data block allows us to validate the data in an efficient manner as

compared to validating a data block that contains a checksum internally. Checksums hierarchy in ZFS forms a “Self-validating Merkle tree”.

The traditional RAID-4 and RAID-5 suffers from write hole and partial write stripes. ZFS addresses this using the RAID-Z technique by distributing the parity along with data (dynamic striping with variable block size) and can lose one drive before RAID failure. Taking a snapshot of the file system in ZFS is cheap and constant time operation. I learned that in the traditional resilvering technique the entire disk contents are copied to the new mirror even if an almost entire disk is free. Meta-data driven resilvering in ZFS copies the storage pool's block tree from the root down. It first copies all the important blocks. ZFS resilvers only the required data by tracking the modified blocks through “dirty region” logging in case of a short outage.

In the last half hour, we discussed deduplication. Deduplication is a technique of eliminating the duplicate data on the disk. It can be done as inline(when data is sent to the storage system), post-process or during data backup. Deduplication can be performed at various levels like file, block and byte level. The key to efficient deduplication is to compute a unique hash key for every object (file, block or byte). In file level deduplication entire file is used to compute the hash id, whereas in block level each file block is individually deduplicated. Block level takes advantage of the fact that larger files with only small differences can produce lots of duplication. Block level duplication is ineffective if the blocks don’t align. Thus we use byte level duplication with a sliding window concept to finding the common bytes of data.

Overall, I felt that the lecture was slow paced and dull. The part I liked most was that the presentation slides had enough pictures that made the concepts like COW(ROW), ZFS snapshot, RAID-Z, deduplication and other topics easy to grasp.

Reference:

- Redundancy in storage systems - Lecture slides by Victor Latushkin
- Data Deduplication - Lecture slides by Dr. Mark Maybee.
- http://searchdatabackup.techtarget.com/tip/The-pros-and-cons-of-file-level-vs-block-level-data-deduplication-technology