CRITIQUE - 9 [Lecture - 10 ZFS Intent log and File Protections]

The main topic of today's lecture is ZFS intent log (ZIL) and File protection (ACL). we discussed the need for ZFS intent log, journaling overview, ZIL working version, internal log management, ZIL commits, replay and parallel replay. We talked about the performance aspects of ZIL under which circumstances it can be good and bad. And the future performance ideas to boost the ZIL. In the next section, we discussed the file protections - protecting file system data from other users by setting simple file permission, privilege vs permissions, special security attributes in ZFS like immutability, read-only, no-unlink etc. Later topics included how access control list (ACL) is implemented in ZFS, components of ZFS, ZFS ACL architecture.

The lecture started with giving an introduction to synchronous and asynchronous file system requests like read, write, rename file etc. Low latency and high performance are critical for applications like SMB, iSCSI, DB etc. Synchronous behavior can be requested using O_SYNC or O_DSYNC while opening a file. From https://lwn.net/Articles/350219/ I learned that O_DSYNC is just like O_SYNC but the application need not wait until ancillary information(metadata like file modification date time etc) has been flushed to the disk. The need for journaling file system, UFS, and the benefits are straightforward and clearly explained.

In the next section, we discussed the ZFS intent logging(ZIL). System calls are saved in the memory as intent log transaction. It exists to keep track of in-progress synchronous write operations so that they can be completed or rolled back when system crashes or on a power failure. ZIL is not a write cache to boost performance but to keep the integrity of data. I got to know that ZIL can be configured on separate dedicated disks called SLOG(separate intent log). These disks are generally SSD, which are typically 10x faster than rotating disks.

The allocation of pool blocks on ZIL is done in a Round robin fashion from one device to the next in batches. Various concepts like the internal data structures used in ZIL, thundering herd, replay and parallel replay were precisely explained.

In this section, I learned about how to guard the file system data against misuse through file protection. Protecting files is one of the critical functionality of any file system. To protect the files in a shared or nonshared environment the concept of file ownership was introduced. File protection policies are integrated into the core of the operating system. Every file should be tied to a single user, unless it's a shared file in windows operating system where a file(s) can be owned by a group. The side by side

comparison of how file attributes behave in Solaris, Linux and windows were really impressive and beneficial.

File system data is protected by setting permission on the files and providing file privilege to the users. Unix defines three permission modes - read, write and execute and three class - owner, group and others. The distinction between permission and privilege was clearly explained along with examples. ZFS supports special attributes like immutability - cannot modify contents, read only - ability to modify only the metadata but not the contents, append-only - can append content to the EOF only, quarantined - these are virus infected files and can only be written.

Access control list (ACL) provide fine grain control over access to files. ZFS ACL is based on the specification of NFSv4 ACL, which provides richer and more fine-grained permission control. I learned about the various components of ACL. ACL is made up ACE (access control entry), inheritance control, group designation and access flags. ACE has the following components - Allow, Audit, Deny, and Alarm.

Both the guest lectures did a good job by providing a brief over of the ZIL and Access control lists in an interactive way. Mark shellenbaum gave a better picture of file permissions by comparing side by side the implementations of file permissions in Linux like OS and Windows. It would have been helpful to see a live demo on how to change file permission, recover it and various other operations. Overall, the lecture was really good.

Reference:

1. ZFS Intent log - Lecture slides by Neil Perrin.
2. File protection - Lecture slides by Mark Shellenbaum.
3. https://lwn.net/Articles/350219/