



ZFS

Ohad Rodeh



Introduction

- This lecture is based on **The Zetta-byte File-System (ZFS)**
SUNTM labs, 2004
- Built by a team at Sun led by Jeff Bonwick.



ZFS and WAFL

- ZFS uses copy-on-write in much the same way as WAFL
 - Network-Appliance is now suing SUN for using their WAFL patents without paying
- The interesting new features in ZFS are checksums and Z-RAID



Checksums

- Each indirect block contains pointers
- Each pointer includes a 256-bit checksum of the block it is pointing to
- Allows catching disk errors where the disk head misses and returns data from the wrong sector



Z-RAID

- Normally, RAID and the file-system are separated
- In ZFS they are integrated
- For example:
 - The RAID algorithm is Reed-Solomon 4+2P
 - There are 16KB of data to write
 - Compute the RAID code for an additional 8KB of data
 - Split the data into $6 \times 4KB$ parts
 - Allocate a 4KB block on each of disk
 - Write to each disk



Wider pointers

- Since the allocations on each disk are independent, we need to keep six pointers in order to locate the data
- This means that the pointers in the file-system are wider
 - Take up more space
 - Reduces the fanout



Recovery

- Recovery cannot be performed at the disk level alone
- A recursive traversal of the file-system is required
 - More efficient than RAID recovery if the file-system is small
 - Less efficient if the file-system takes up most of the disk



Notes

- ZFS uses the ARC caching algorithm
- Apple said it will use ZFS in its operating-system
- Generally speaking, ZFS has received good reviews



Summary

- ZFS is a new file-system that uses copy-on-write built by SUNTM
- ZFS has been open-sourced (license is *not* GPL)
- It verifies data through checksums
- Mixes the RAID and file-system levels