opensolaris™

ZFS

Marcin Kula
kulam@ee.pw.edu.pl
http://www.ee.pw.edu.pl/~kulam

ZFS Overview

- Provable data integrity
  - Detects and corrects silent data corruption
- Immense capacity
  - 128-bit filesystem
- Simple administration
- Fast

Trouble With Existing Filesystems

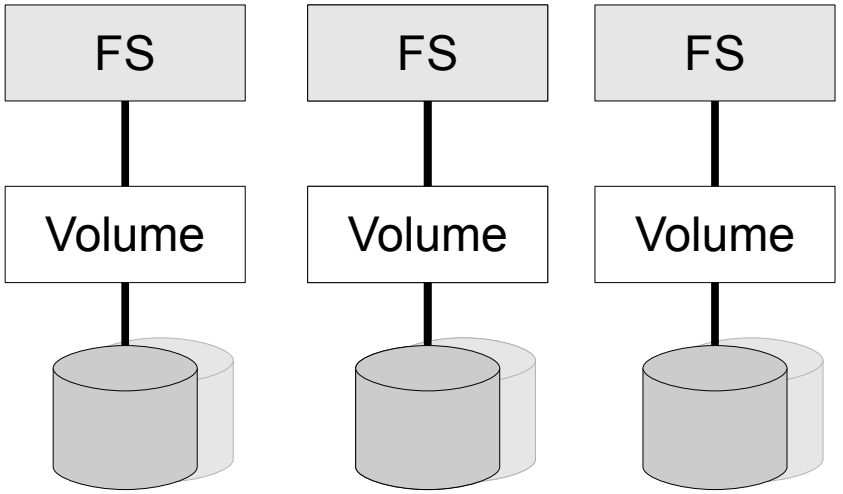
- **No defense against silent data corruption**
  - Any defect in disk, controller, cable, driver, or firmware
- **Brutal to manage**
  - Labels, partitions, volumes, provisioning, grow/shrink, /etc/vfstab...
  - Lots of limits:
    - filesystem/volume size, file size, number of files,
    - files per directory, number of snapshots, ...
  - Not portable between platforms (e.g. x86 to/from SPARC)
- **Slow**
  - Linear-time create, fat locks, fixed block size, native prefetch,
  - slow random writes, dirty region logging

ZFS Design Principles

- Pooled storage
  - Completely eliminates the antique notion of volumes
  - Does for storage what VM did for memory
- End-to-end data integrity
  - Historically considered “too expensive”
  - Turns out, no it isn't
  - And the alternative is unacceptable
- Transactional operation
  - Keeps things always consistent on disk
  - Removes almost all constraints on I/O order
  - Allows us to get huge performance wins

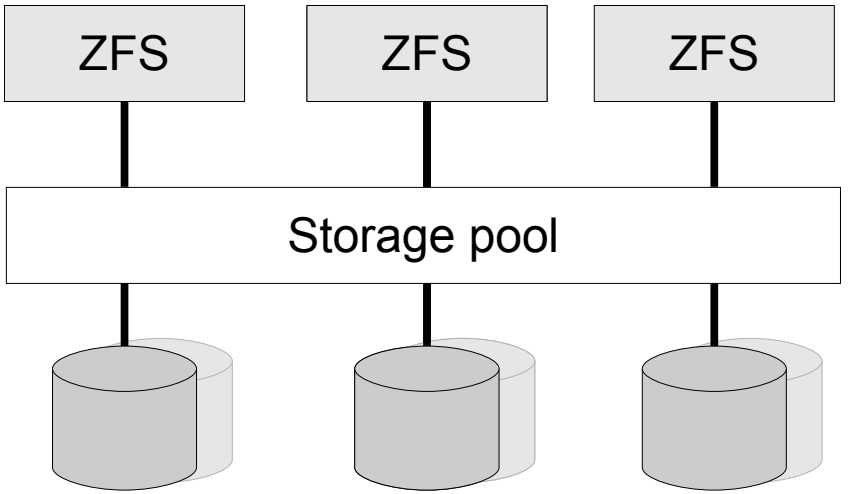
Volumes Versus ZFS

Traditional Volumes



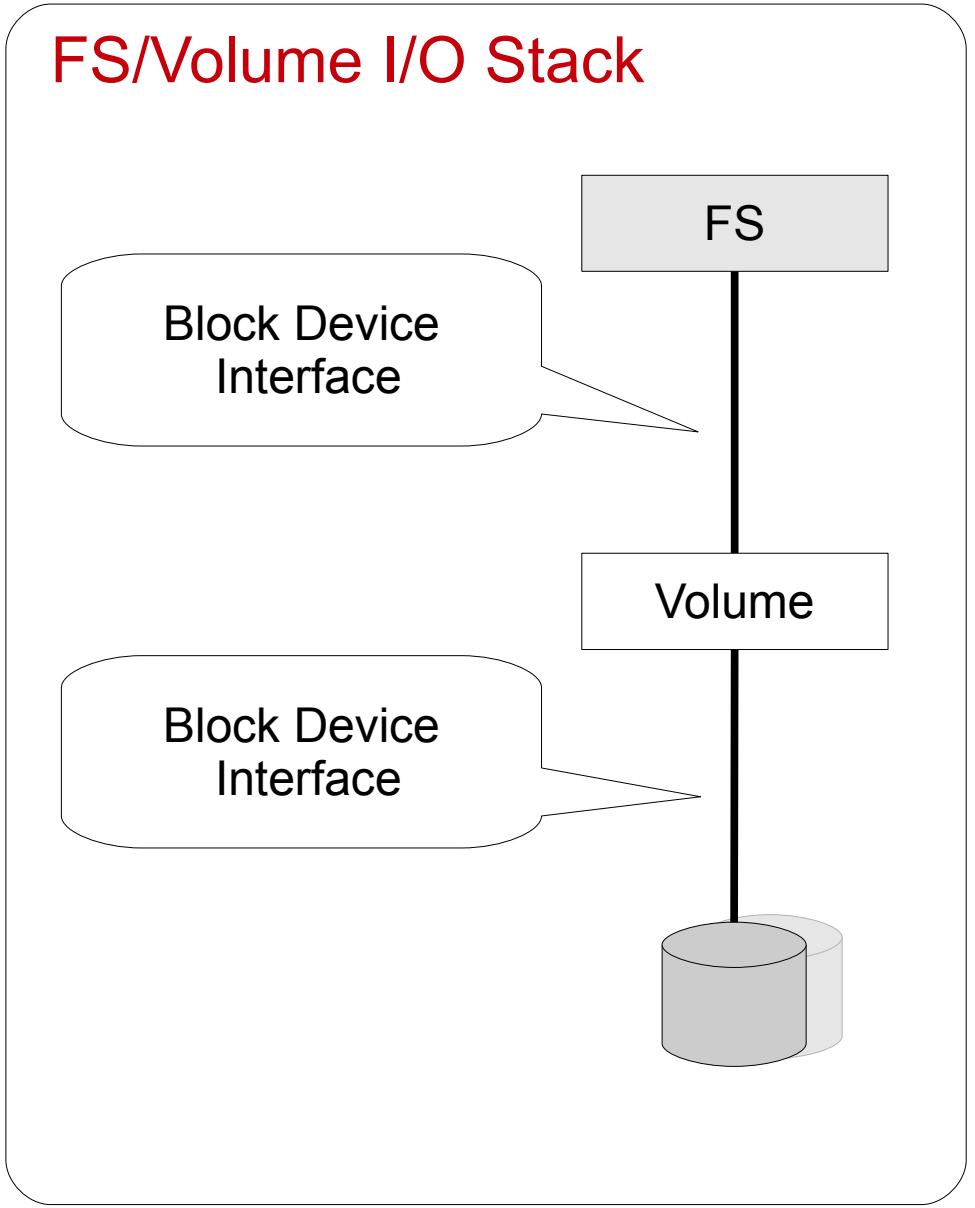
- Abstraction: virtual disk
- Partition/volume for each FS
- Grow/shrink by hand
- Each FS has limited bandwidth
- Storage is fragmented, stranded

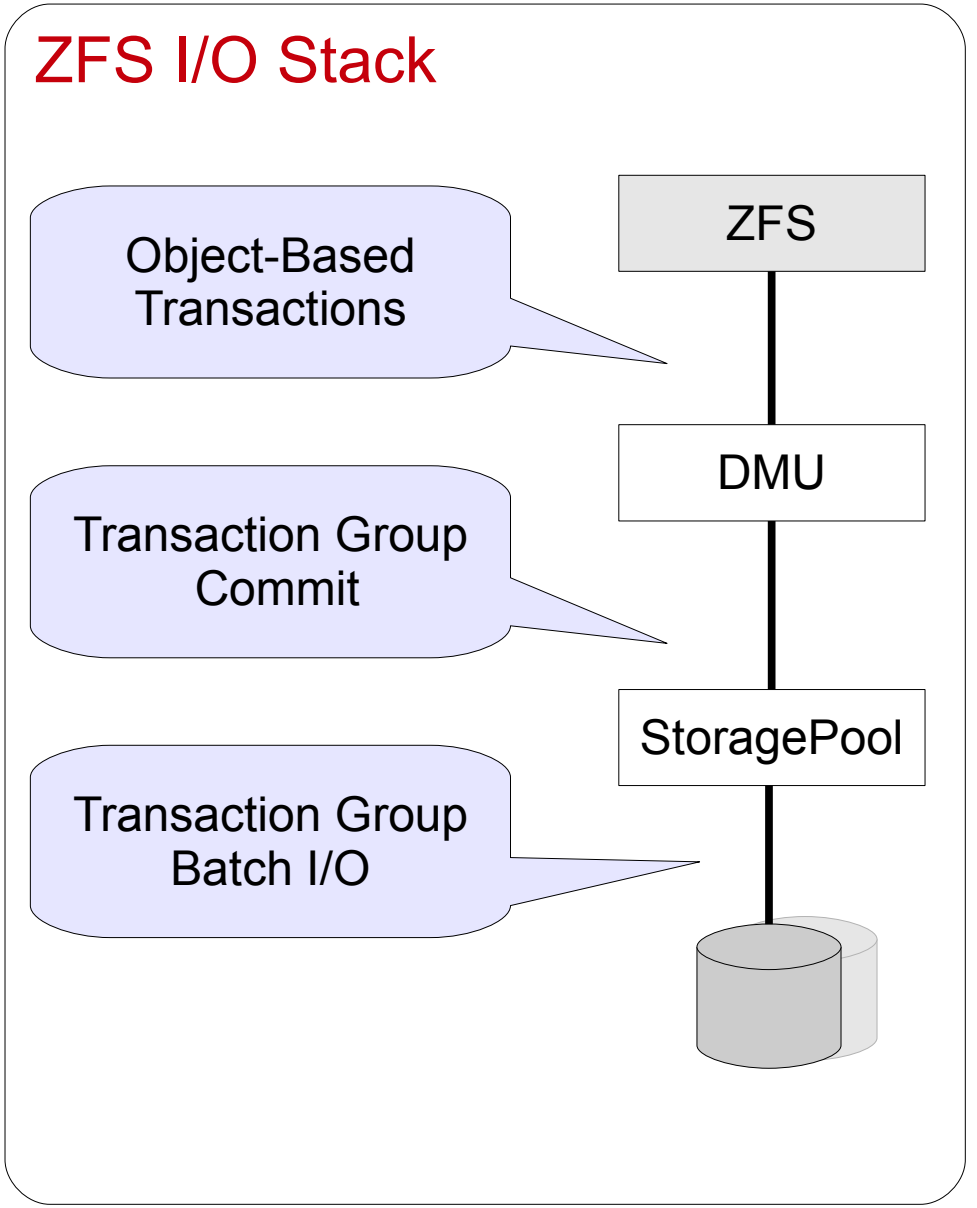
ZFS Pooled Storage



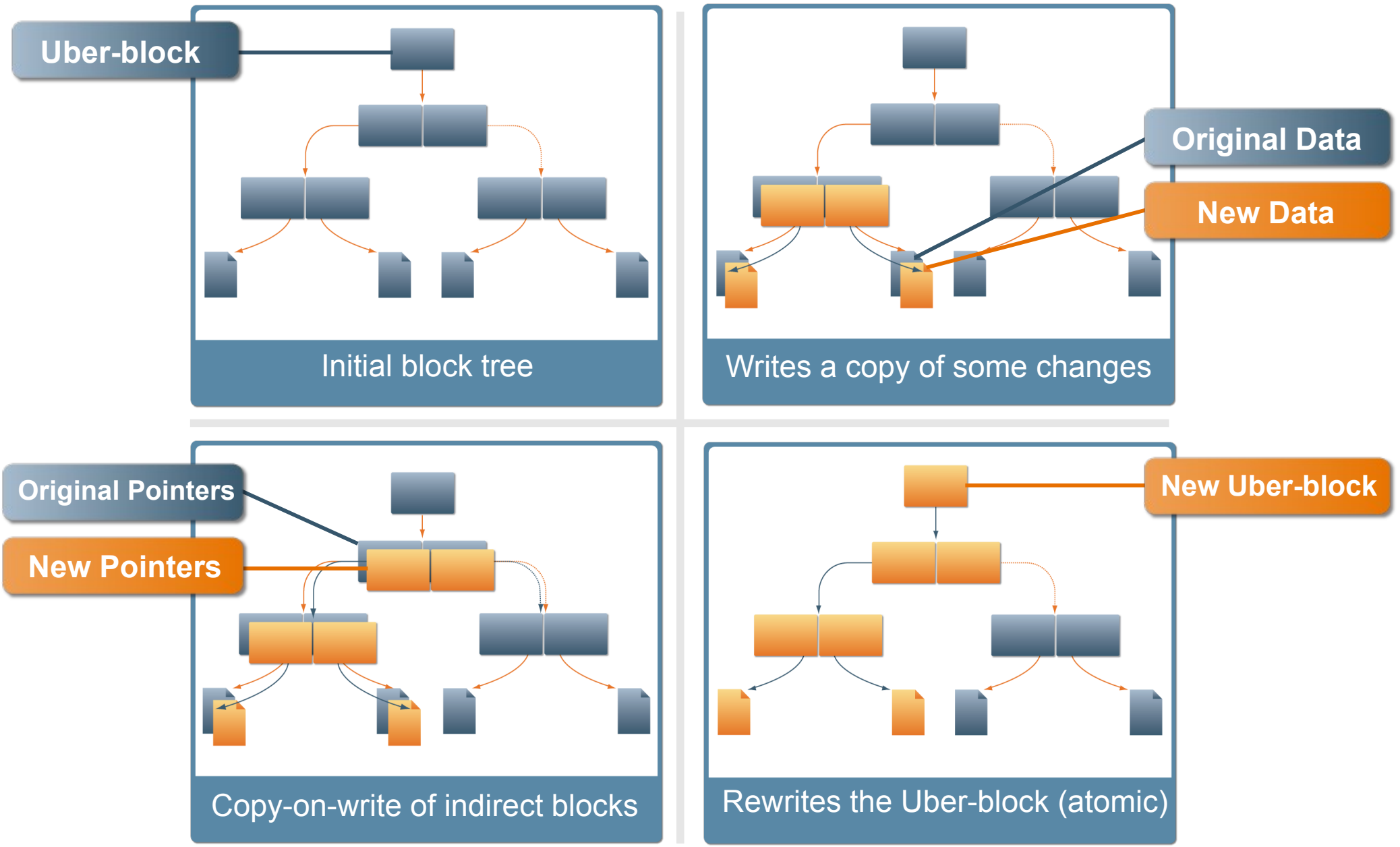
- Abstraction: malloc/free
- No partitions to manage
- Grow/shrink automatically
- All bandwidth always available
- Pool allows space to be shared

FS/Volume Model vs. ZFS



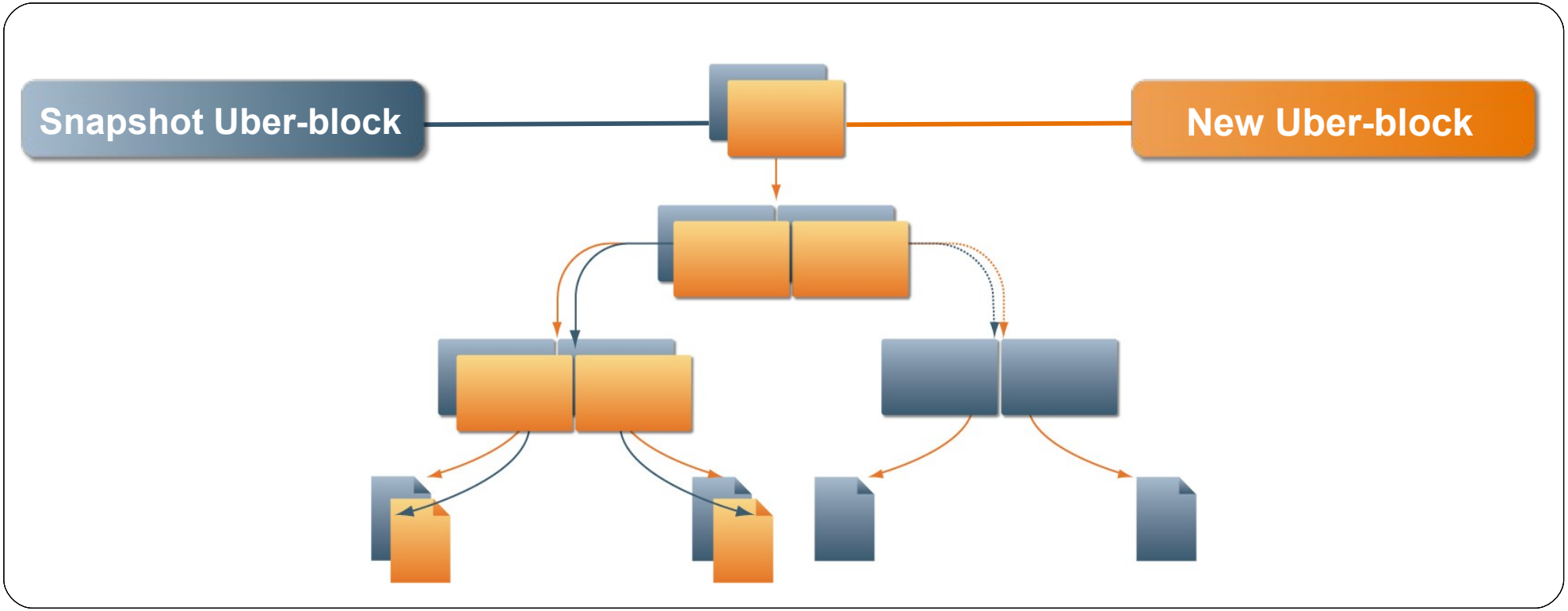


Copy-On-Write Transactions



Bonus: Constant-Time Snapshots

- At end of TX group, don't free COWed blocks
  - Actually cheaper to take a snapshot than not!

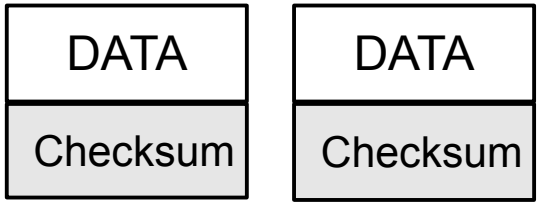


Snapshots & Clones

- Snapshots
  - Read-only point-in-time copy of file system
  - Very space efficient (Copy-on-write)
  - And instantaneous
- Clones
  - **Writable copy of a snapshot**
  - Clones Promotion
  - Ideal for storing many private copies of shared data:
    - Software installations
    - Workspaces
    - Diskless clients

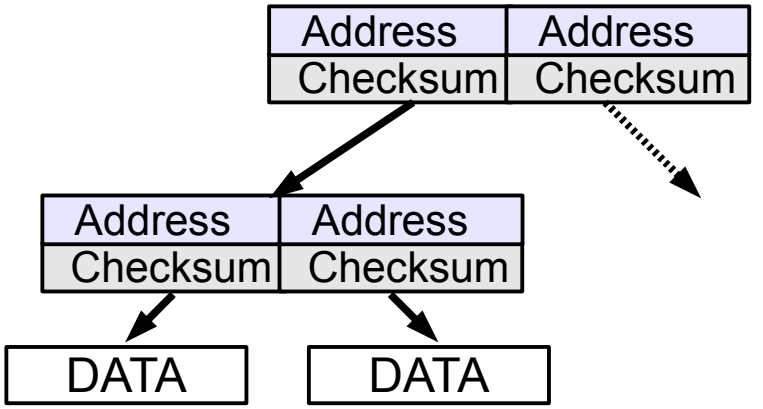
End-to-End Checksums

Disk Block Checksums



- ✔ Bit rot
- ✗ Phantom writes
- ✗ Misdirected reads and writes
- ✗ DMA parity errors
- ✗ Driver bugs
- ✗ Accidental overwrite

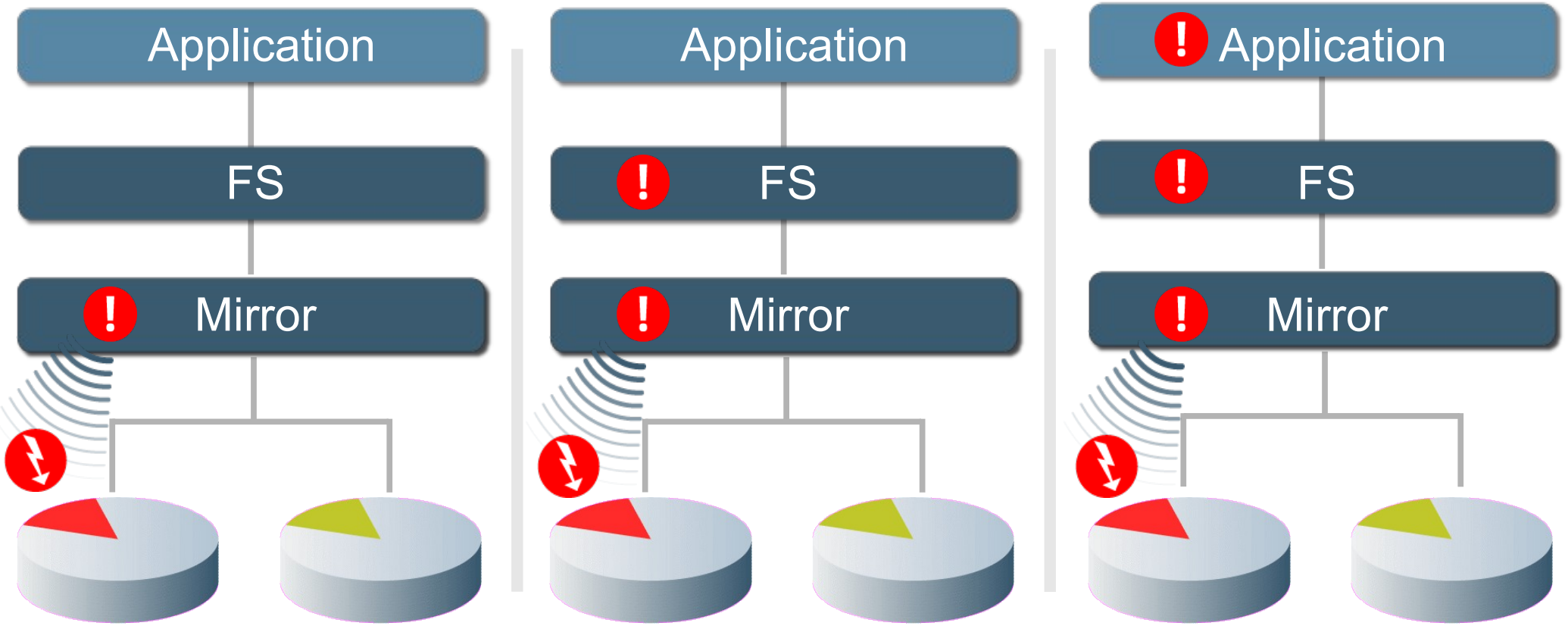
ZFS Checksum Trees



- ✔ Bit rot
- ✔ Phantom writes
- ✔ Misdirected reads and writes
- ✔ DMA parity errors
- ✔ Driver bugs
- ✔ Accidental overwrite

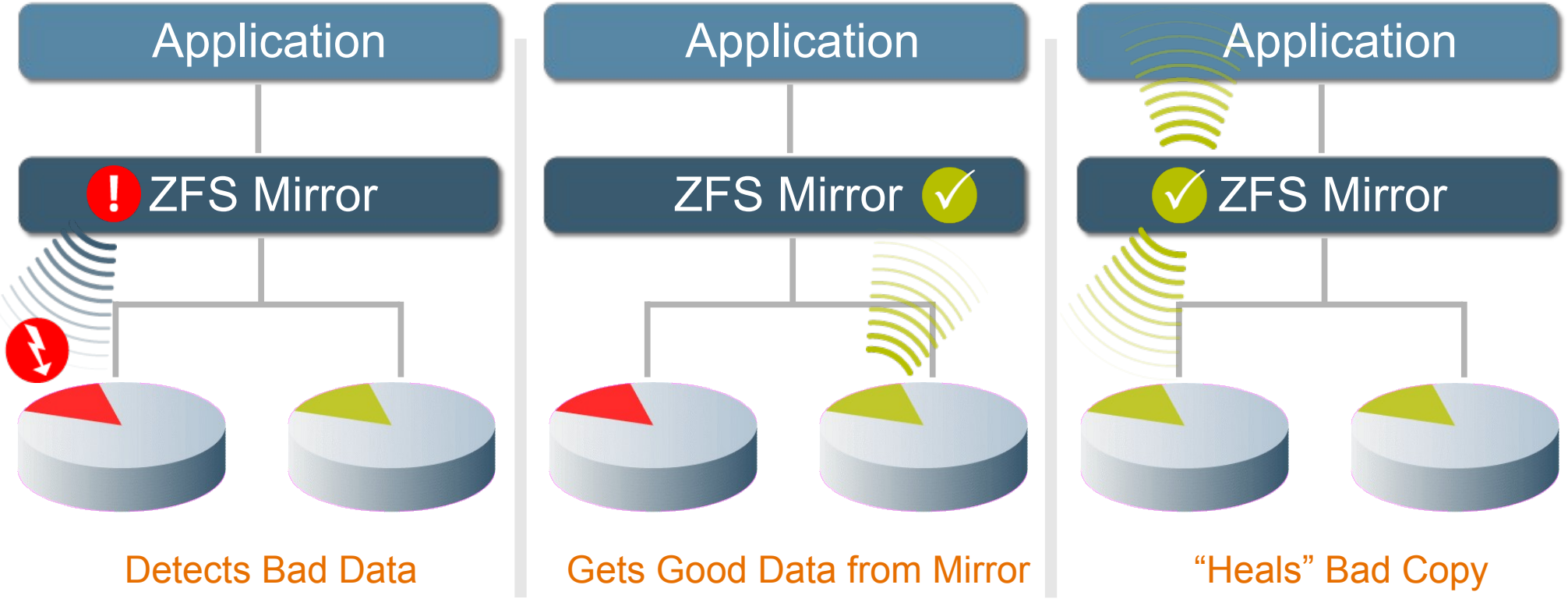
Traditional Mirroring

- Application issues a read
  - First disk has a corrupt block
    - **Application receives a corrupt data !**



Self-Healing Data

- Detect bad data using checksums
  - “heal” the data using its mirrored copy.



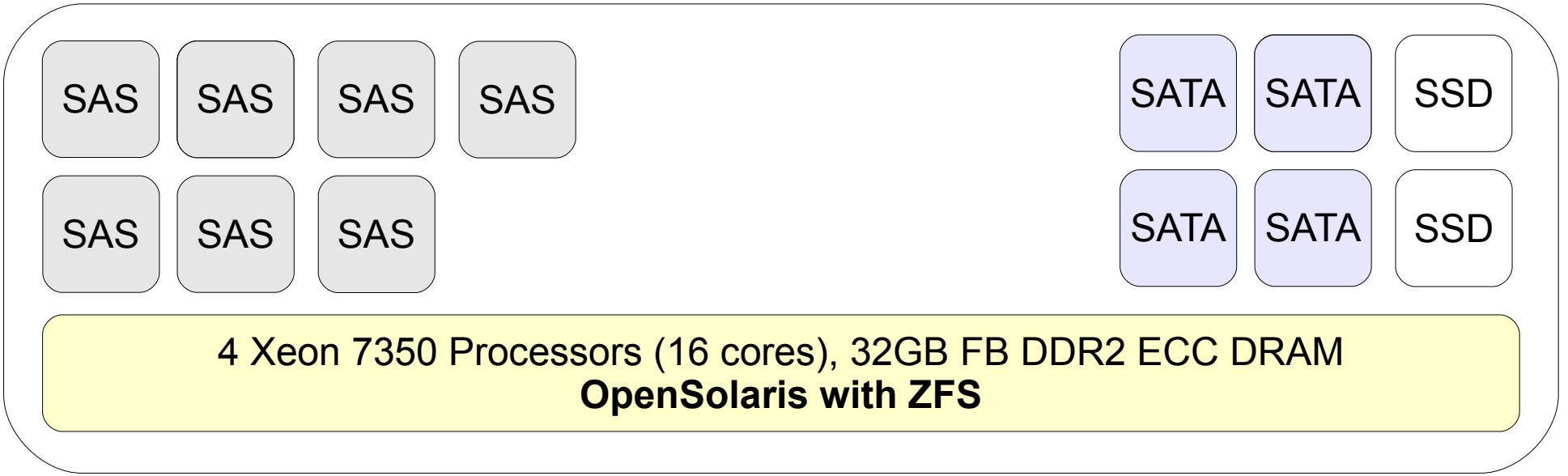
Hybrid Storage Pool (HSP)

- Adapted ZFS to integrate flash
- ZFS intent-log (ZIL) device
  - Accelerate small, synchronous writes
- Second Level Adaptive Replacement Cache (L2ARC)
  - Larger caching tier than ARC (DRAM)
  - “Evict-ahed” cache
  - Accelerate reads
- Example:

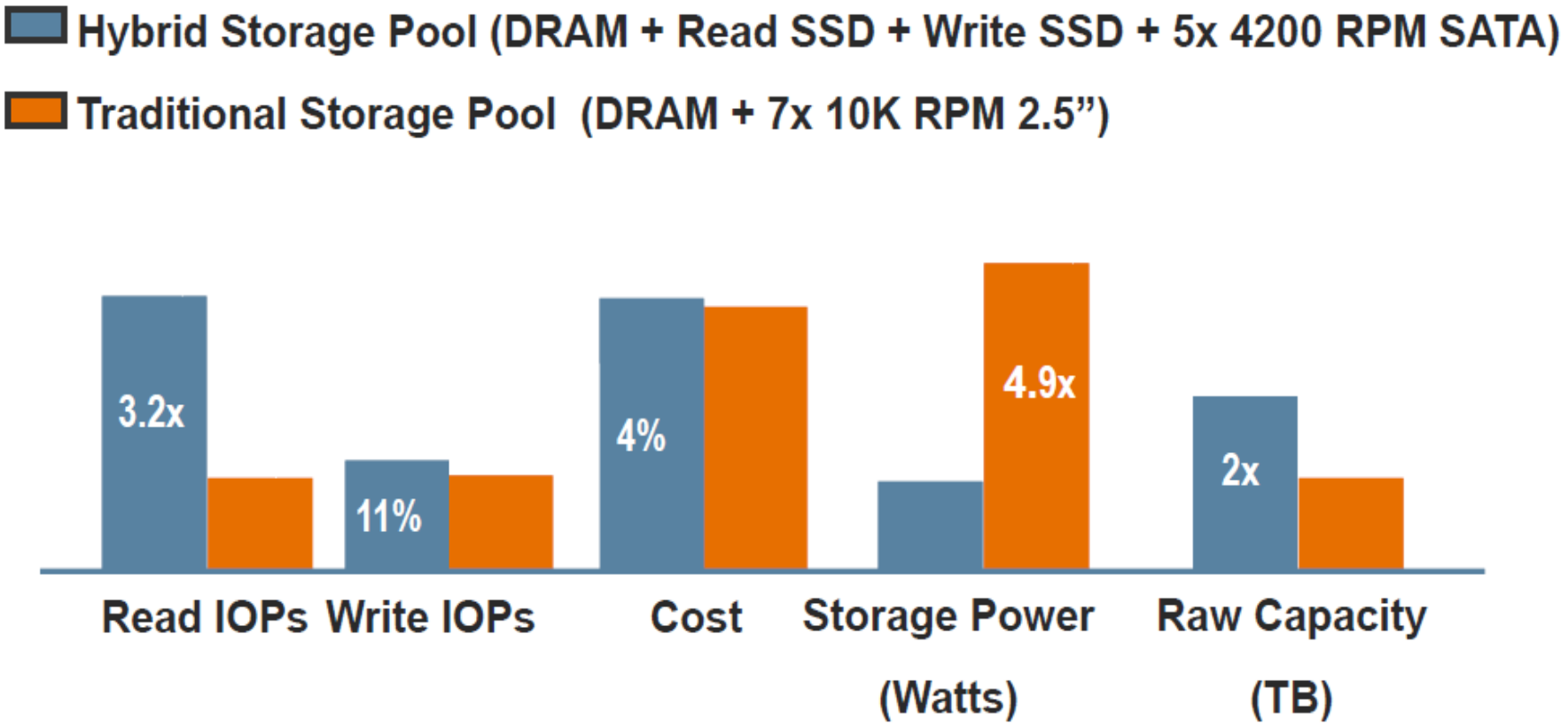
```
zpool create tank mirror c0d0 c1d0 log c2d0 cache c2d0
```

Example configuration

- **Configuration A**
  - SAS Drives
    - 10000 RPM
    - 146 GB / drive
- **Configuration B**
  - SATA Drives
    - 4200 RPM
    - 400 GB / drive
  - SSD Drives
    - 80 GB (Cache Device)
    - 32 GB (ZIL Device)



HSP Results


Hybrid Storage Pool (DRAM + Read SSD + Write SSD + 5x 4200 RPM SATA)
Traditional Storage Pool (DRAM + 7x 10K RPM 2.5")
3.2x
11%
4%
4.9x
2x
Read IOPs
Write IOPs
Cost
Storage Power (Watts)
Raw Capacity (TB)

ZFS Root

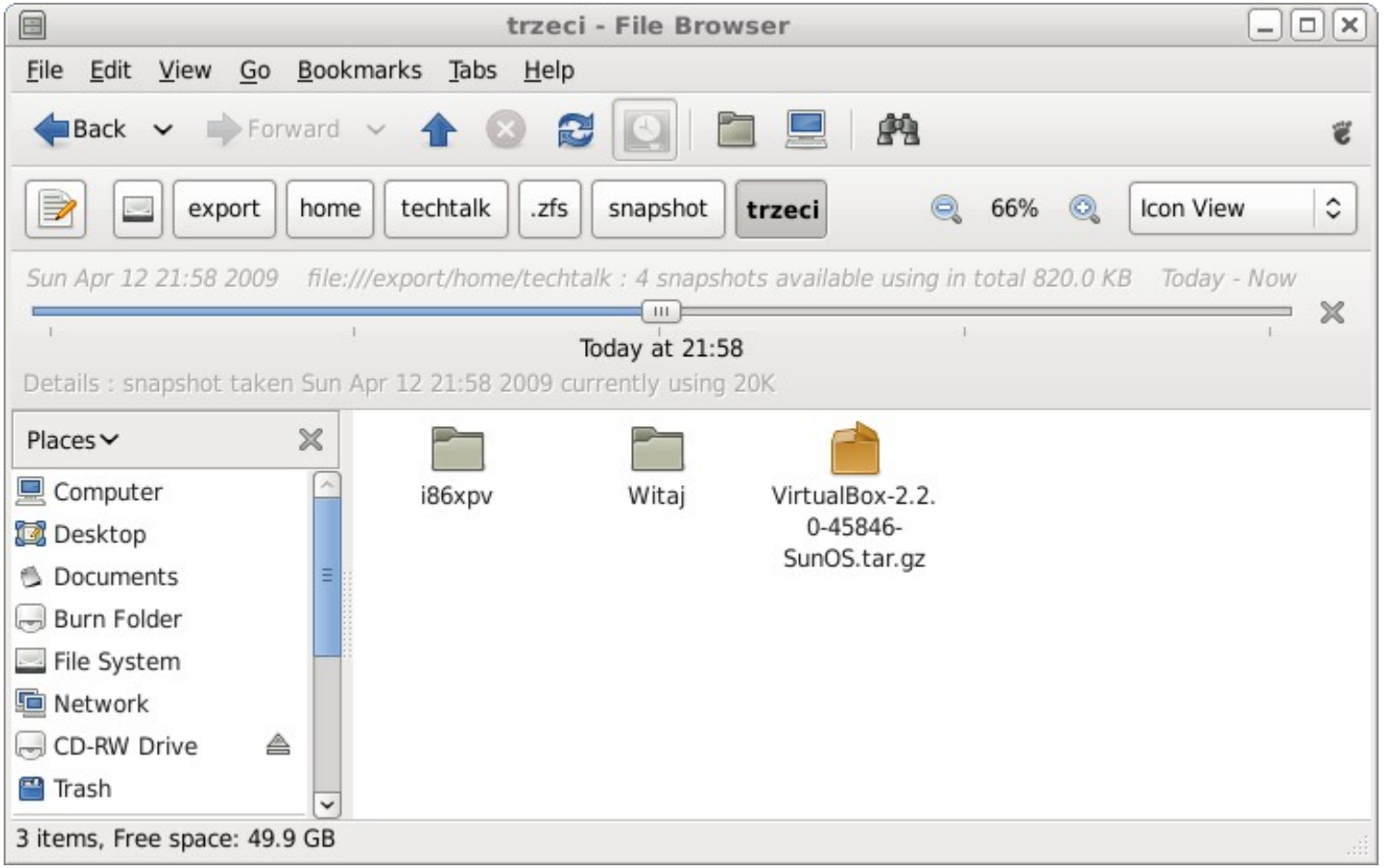
- Brings all the ZFS goodness to /
  - Checksums, compression, replication, snapshots, clones
  - Boot from any dataset
- Patching becomes safe
  - Take snapshot, apply patch... rollback if there's a problem
- Live upgrade becomes fast
  - Create clone (instant), upgrade, boot from clone
  - No “extra partition”
- Based on new Solaris boot architecture
  - ZFS can easily create multiple boot environments
  - GRUB can easily manage them

Beadm

- Manages boot environments
- BE automatically created on upgrade
  - You can reboot into BE before upgrade
- BE automatically created when package installation fails
  - You can reboot into BE before installation
- Commands
  - `$ beadm list`
  - `$ beadm create / destroy`
  - `$ beadm activate`
  - `$ beadm mount / unmount`

GUI Integration

- Time Slider
  - System -> Administration -> Time Slider Setup
  - Nautilus



ZFS in Numbers

- 2^48
  - snapshots in any file system
  - files in any individual file system
  - attributes for a file
  - files in a directory
  - 16 exabyte file systems
- 2^64
  - devices in a storage pool
  - storage pools per system
  - file systems per storage pool
- 16 exabyte files
- 16 exabyte attributes
- 3x1023 petabyte storage pools

OpenSolaris

Pool and Filesystem Management

- Create a storage pool named “data”
  - zpool create data mirror c0t0d0 c1t0d0
- Create home directory filesystem, mounted at /export/home
  - zfs create data/home
  - zfs set mountpoint=/export/home data/home
- Create home directories for several users

  Note: automatically mounted at /export/home/{ann,bob,sue} thanks to inheritance
  - zfs create data/home/ann
  - zfs create data/home/bob
  - zfs create data/home/sue
- Later, add more space to the pool
  - zpool add data mirror c2t0d0 c3t0d0

Setting Properties

- Automatically NFS-export all home directories
  - zfs set sharenfs=rw tank/home
- Turn on compression for everything in the pool
  - zfs set compression=on tank
- Limit Eric to a quota of 10g
  - zfs set quota=10g tank/home/eric
- Guarantee John a reservation of 20g
  - zfs set reservation=20g tank/home/john

ZFS Snapshots

- Read-only point-in-time copy of a filesystem
  - Instantaneous creation, unlimited number
  - No additional space used – blocks copied only when they change
  - Accessible through .zfs/snapshot in root of each filesystem

- Take a snapshot of Mark's home directory
  - zfs snapshot tank/home/marks@tuesday
- Roll back to a previous snapshot
  - zfs rollback tank/home/perrin@monday
- Take a look at Wednesday's version of foo.c
  - cat ~maybee/.zfs/snapshot/wednesday/foo.c

ZFS Clones

- Writable copy of a snapshot
  - Instantaneous creation, unlimited number
  - Ideal for storing many private copies of mostly-shared data
    - Software installations
    - Workspaces
    - Diskless clients
- Create a clone of your OpenSolaris source code
  - zfs clone tank/solaris@monday tank/ws/lori/fix

- ZFS Send / Receive (Backup / Restore)
  - Powered by snapshots
  - Full backup: any snapshot
  - Incremental backup: any snapshot delta
    - Very fast – cost proportional to data changed
  - So efficient it can be used for remote replication
- Generate a full backup

```
zfs send tank/fs@A >/backup/A
```

- Generate an incremental backup

```
zfs send -i tank/fs@A tank/fs@B >/backup/B-A
```

- Remote replication: send incremental

```
zfs send -i tank/fs@11:31 tank/fs@11:32 | ssh host zfs receive -d /tank/fs
```

Conclusion – ZFS is

- Simple
  - Concisely expresses the user's intent
- Powerful
  - Pooled storage, snapshots, clones, compression, RAID-Z, universal storage, ...
- Safe
  - Detects and corrects silent data corruption
- Fast
  - Dynamic striping, intelligent prefetch, pipelined I/O
- Open
  - http://www.opensolaris.org/os/community/zfs
- Free

Conclusion – ZFS is

- Simple
  - Concisely expresses the user's intent
- Powerful
  - Pooled storage, snapshots, clones, compression, RAID-Z, universal storage, ...
- Safe
  - Detects and corrects silent data corruption
- Fast
  - Dynamic striping, intelligent prefetch, pipelined I/O
- Open
  - http://www.opensolaris.org/os/community/zfs
- Free

Next Steps

- Visit www.opensolaris.com

Get It	Experience It	Install It
		
Available everywhere Smaller faster download	LiveCD Risk free	Easy Graphical Runs in virtualization environments

Next Steps

- **Join the Student Community**
  - http://osum.sun.com/group/pw
- Community
  - http://opensolaris.org/os/community/zfs
- Administration Guide
  - http://docs.sun.com/app/docs/doc/819-5461
- Resources in BigAdmin
  - http://www.sun.com/bigadmin/topics/zfs/
- ZFS vs. Linux Raid&LVM
  - http://www.unixconsult.org/zfs_vs_lvm.html

Questions

opensolaris™

Thank You.

Marcin Kula
kulam@ee.pw.edu.pl
http://www.ee.pw.edu.pl/~kulam