

# ZFS Overview

**Dominic Kay**  
**Sun Microsystems Ltd.**



# Trouble with Existing File Systems?

Good for the time they were designed, but...

No Defense  
Against Silent  
Data Corruption

Any defect in  
datapath can  
corrupt data...  
**undetected**

Difficult to  
Administer—Need  
a Volume Manager

Volumes,  
labels, partitions,  
provisioning  
and lots of limits

Older/Slower  
Data Management  
Techniques

Fixed  
block size,  
dirty region  
logging

# What is ZFS?

**A new way to manage data**

## End-to End Data Integrity

With check-summing and  
copy-on-write transactions

## Easier Administration

A pooled storage model –  
no volume manager



## Immense Data Capacity

The world's  
first 128-bit  
file system

## Data Services

Snapshots  
Clones  
Replication

# What is ZFS?

**A new way to manage data**

## End-to End Data Integrity

With check-summing and  
copy-on-write transactions

## Easier Administration

A pooled storage model –  
no volume manager



## Immense Data Capacity

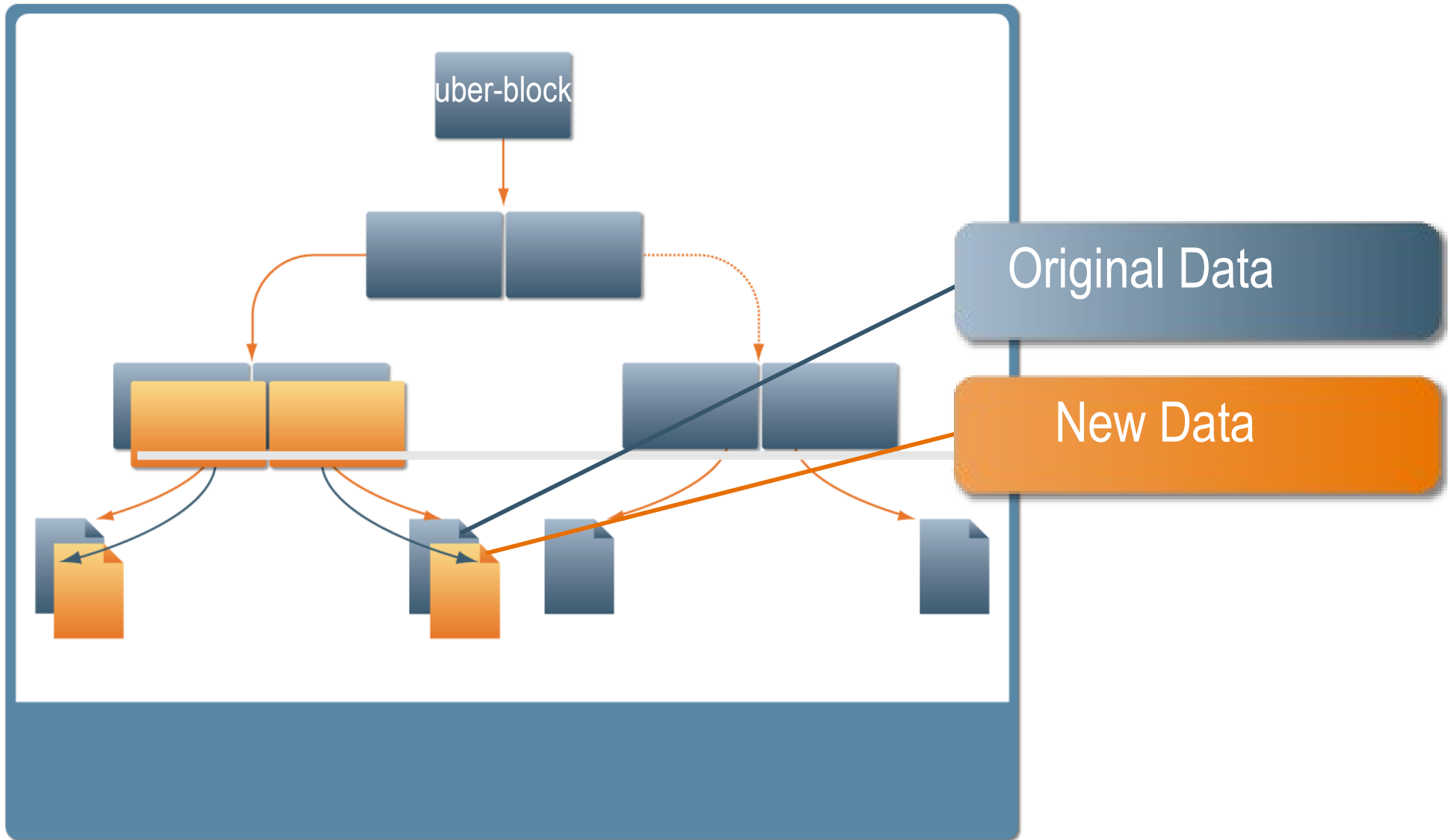
The world's  
first 128-bit  
file system

## Integrated Data Services

Snapshots  
Clones  
Replication  
Compression

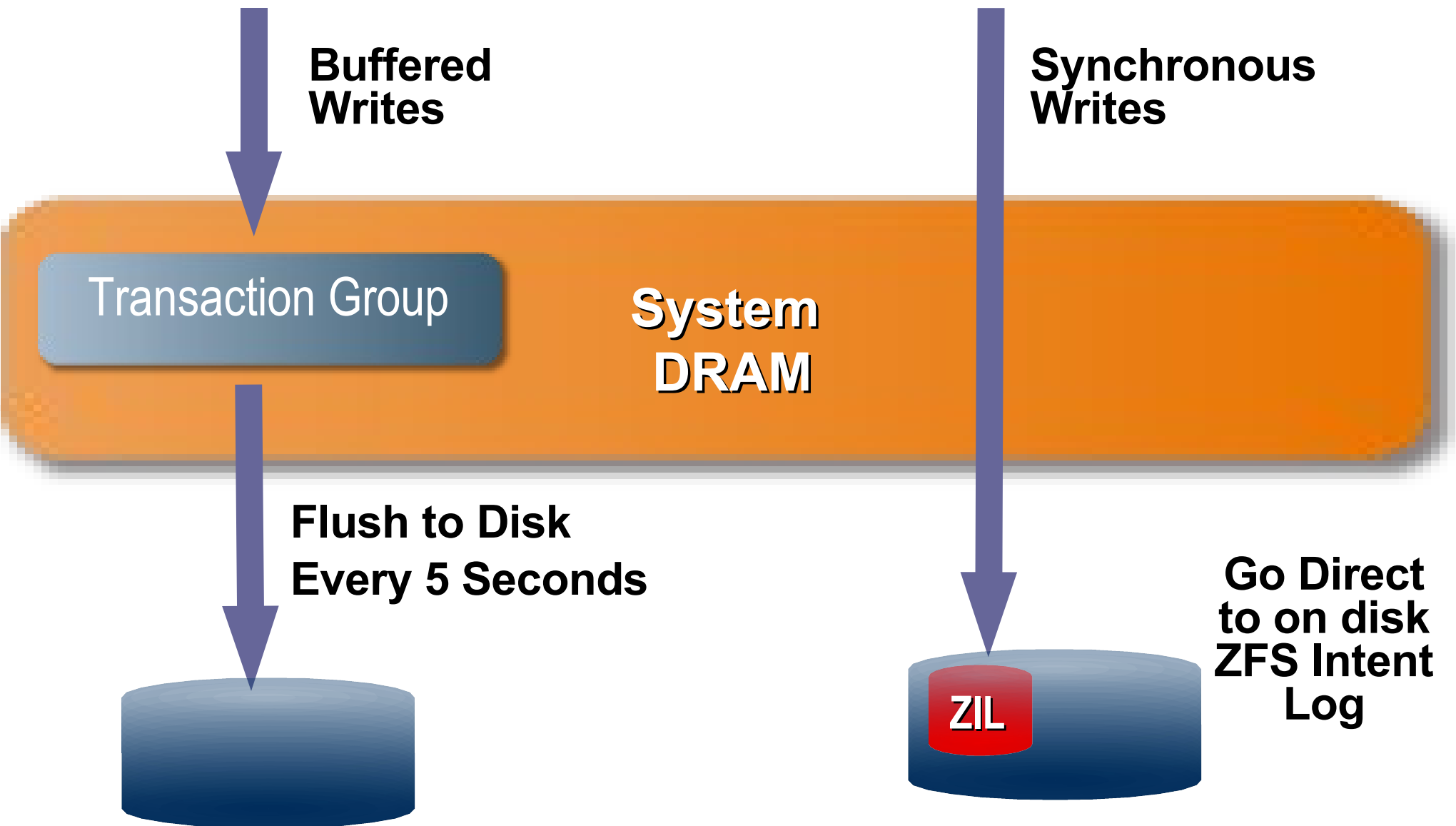
# Copy-on-Write

## Never Overwrite Existing Data



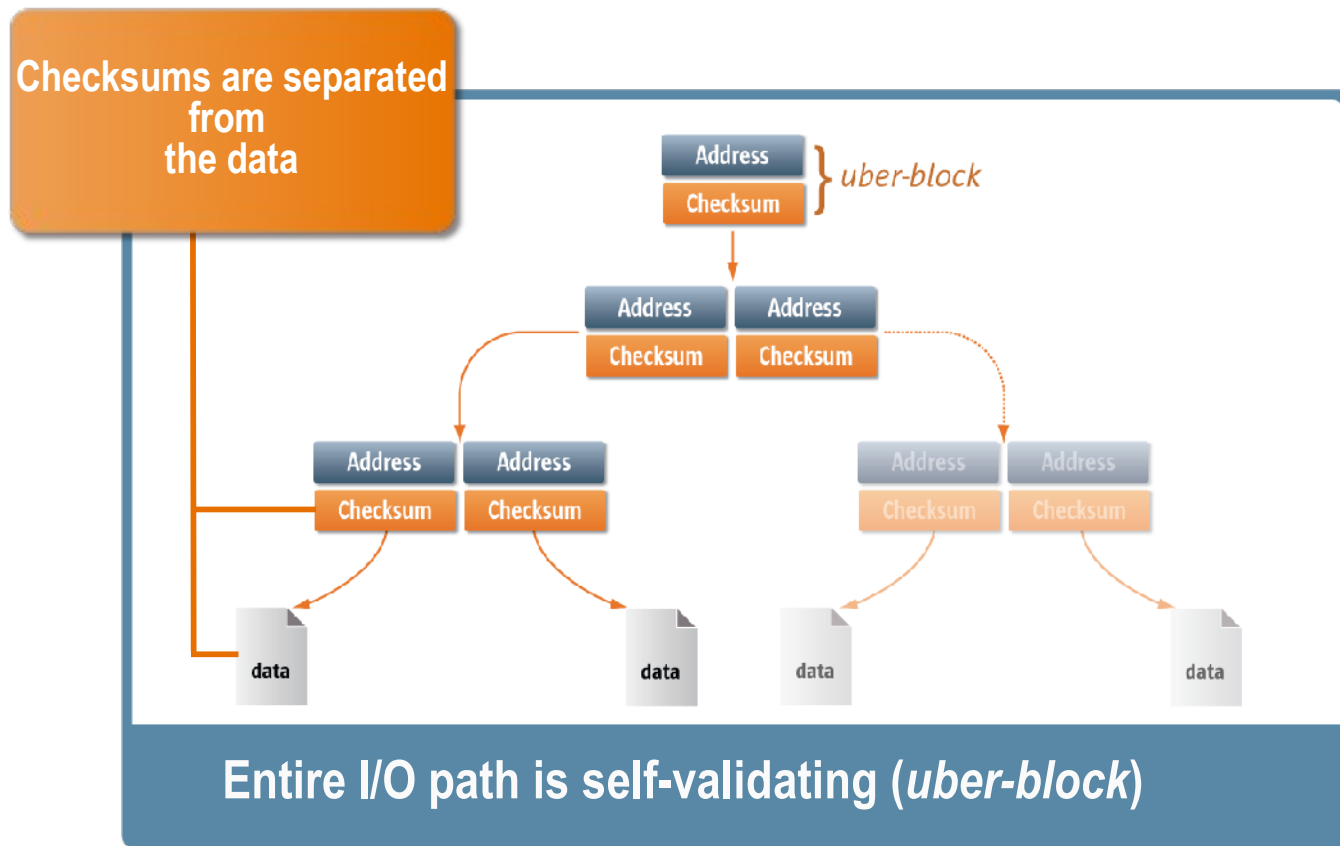
# Transactional

**Always Consistent On Disk**



# How Do We Know What We Just Read Was What We Wrote ?

## End-to-End Checksums



### Prevents:

- > Silent data corruption
- > Panics from corrupted metadata
- > Phantom writes
- > Misdirected reads and writes
- > DMA parity errors
- > Errors from driver bugs
- > Accidental overwrites



# Redundant Copies of Data

## Software RAID Protection

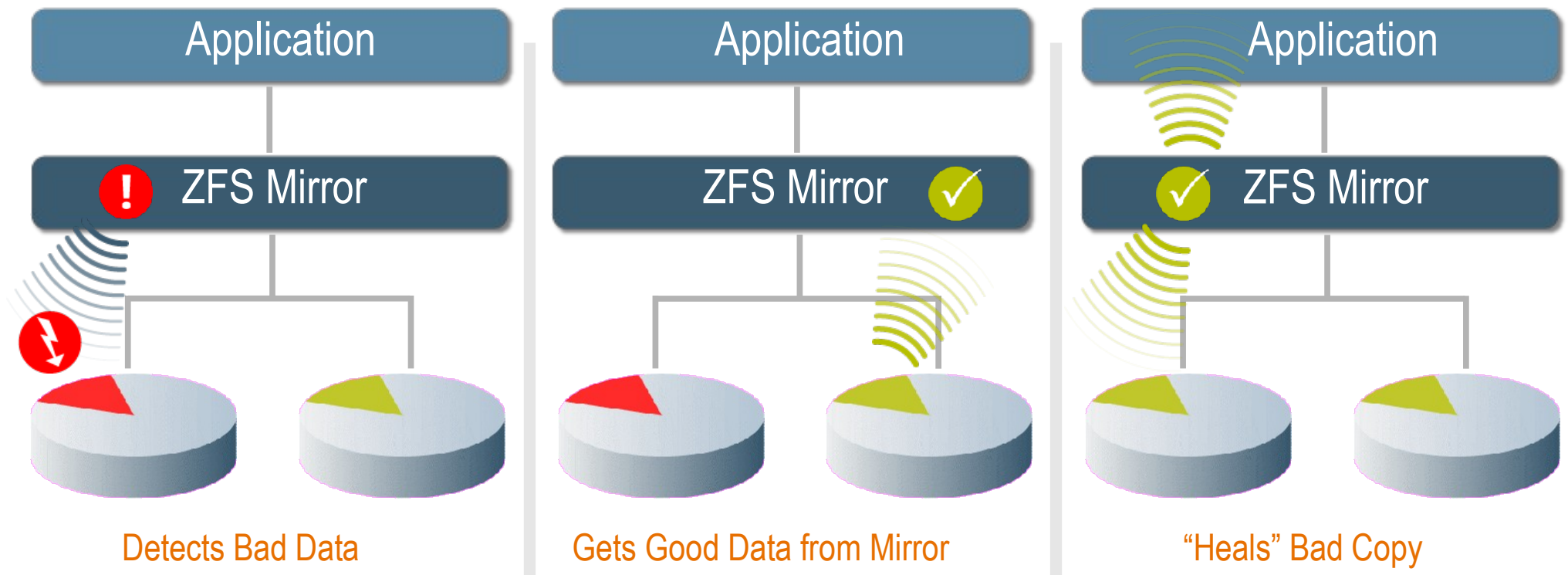
- RAID=Redundant Array of *Inexpensive* Disks
- ZFS supports
  - > Stripes (RAID-0)
  - > Mirroring (RAID-1)
  - > RAID-Z (Similar to RAID-5)
  - > RAID-Z2 (Double parity, similar to RAID-6)
- ZFS Transactional Design means no “RAID-5 write hole” when using RAID-Z





# Self-Healing Data

- Checksums are used to validate blocks
- If a Bad Block is found ZFS can repair it so long as it has another copy
- RAID-1 - ZFS can “heal” bad data blocks using the mirrored copy
- RAID-Z/Z2 - ZFS can “heal” bad data blocks using parity



# What is ZFS?

**A new way to manage data**

## End-to End Data Integrity

With check-summing and  
copy-on-write transactions

## Easier Administration

A pooled storage model –  
no volume manager



## Immense Data Capacity

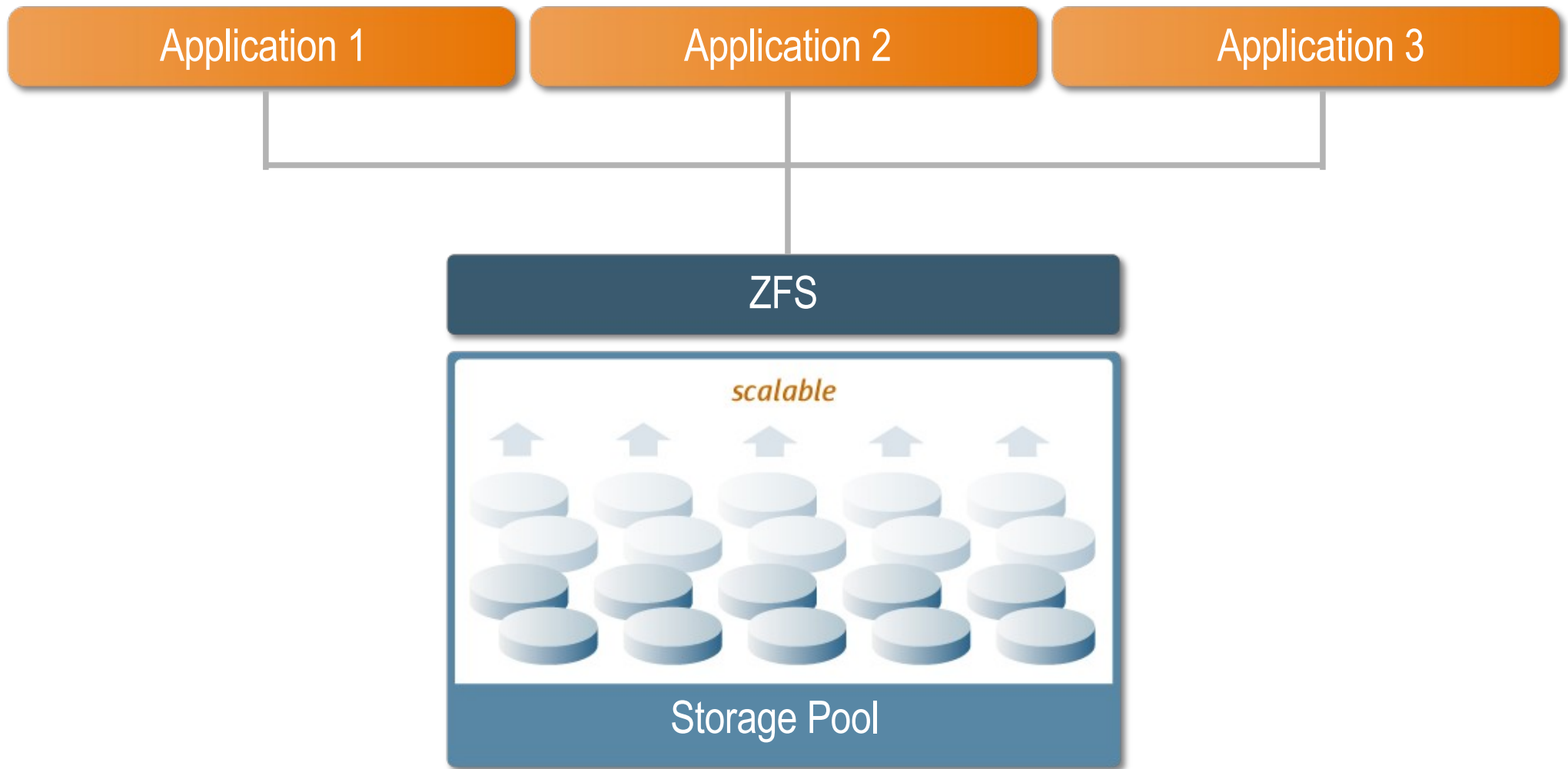
The world's  
first 128-bit  
file system

## Integrated Data Services

Snapshots  
Clones  
Replication

# No More Volume Manager!

Automatically add capacity to shared storage pool



# Volume Management Made Simple

## Solaris Volume Manager + UFS

Partition disks using format

Set up state replicas

```
# metadb -a -f c1t0d0s6 c2t0d0s6 ...
```

Initialise the disks

```
# metainit -f d1 1 2 c1t0d0s6 c1t1d0s6
```

```
# metainit -f d2 1 2 c2t0d0s6 c2t1d0s6
```

Construct a mirror

```
# metainit d0 -m d1
```

```
# metattach d0 d2
```

Create soft partitions

```
# metainit d10 -p d0 5g
```

```
# metainit d11 -p d0 5g
```

Create the file systems

```
# newfs /dev/md/dsk/d10
```

```
# newfs /dev/md/dsk/d11
```

Mount them

```
# mkdir /u01 /u02
```

```
# mount /dev/md/dsk/d10 /u01
```

```
# mount /dev/md/dsk/d11 /u02
```

Persist the mount points in /etc/vfstab

## Solaris ZFS

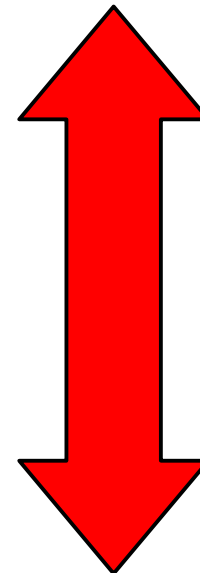
Create a zpool

```
# zpool create tank mirror c1t0d0 c2t0d0 mirror c1t1d0 c2t1d0
```

Create and mount the file systems

```
# zfs create zpool/u01 -o quota=5g
```

```
# zfs create zpool/u02 -o quota=5g
```



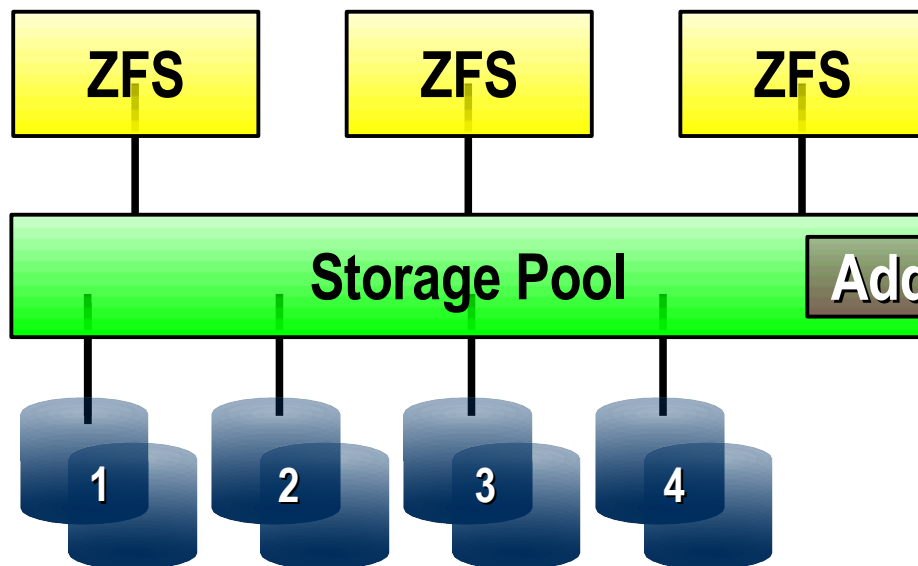
With just 2 disks  
SVM would take  
~15 minutes, ZFS  
~2 minutes

With dozens of  
disks the time  
saving would be  
even greater...

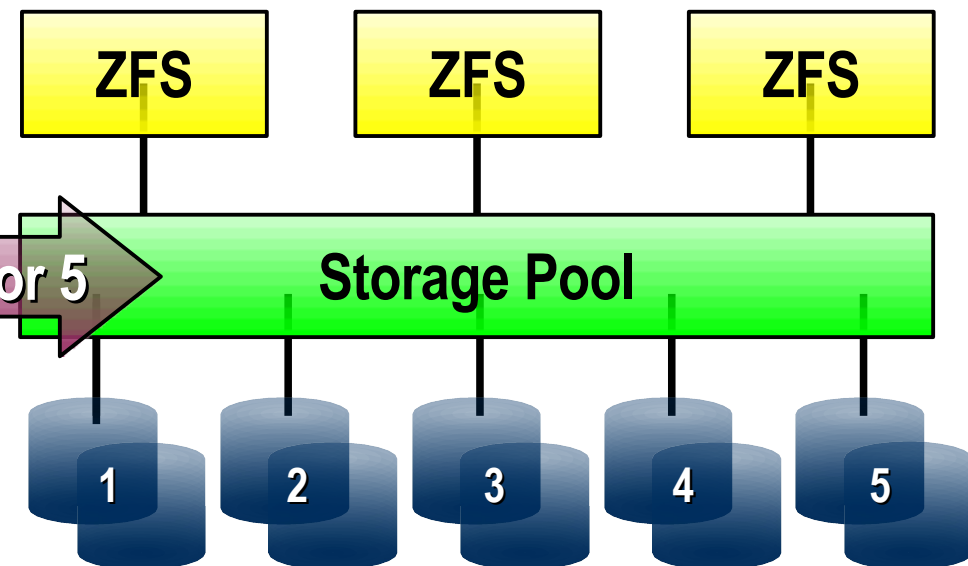
# Dynamic Striping

## Automatically Distributes Load Across All Device

- Writes: striped across all four mirrors
- Reads: wherever the data was written
- Block allocation policy considers:
  - > Capacity
  - > Performance (latency, BW)
  - > Health (degraded mirrors)



- Writes: striped across all five mirrors
- Reads: wherever the data was written
- No need to migrate existing data
  - > Old data striped across 1-4
  - > New data striped across 1-5
  - > COW gently reallocates old data



## ZFS File Systems Security

- ZFS ACLs allow fine grain access control
- File systems become control points
- File system properties are inherited
- Inheritance makes administration a snap
- Manage logically related file systems as a group
- Responsibilities can be delegated



### **Config Data is Stored within the Data**

- When the data moves, so does its config info

### **Pools can be Exported and Imported**

- Allows pools to be moved between systems
- Pools persist between OS upgrades

### **“Adaptive Endian-ness”**

- Hosts always write in their native “endian-ness”

### **Opposite “Endian” Systems**

- Write and copy operations will eventually byte swap all data!

# **Storage Pool Migration**



# Easier Administration

- Pooled Storage Design makes for Easier Administration
- Straightforward Commands and a GUI



# Easier Administration

## Web Based GUI

APPLICATIONS


VERSION

LOG OUT

HELP

User: root Server: isv-x4500b

ZFS Administration



Sun Microsystems, Inc.

Hide panel ☐

System Summary/Tasks

Storage Pools (1)

File Systems (10)

Volumes (0)

Snapshots (2)

Device Hierarchy

fis8

Virtual Devices

8k

backup1

new\_oradata

new\_oralog

new\_postgres

oraarchive

oradata

oralog

postgres

Common Tasks

First Time Use

Create a Storage Pool

Import a Storage Pool

File Systems

Create a File System

Delete a File System

Roll Back a File System to a Previous Snapshot

Volumes

Create a Volume

Delete a Volume

Roll Back a Volume to a Previous Snapshot

Storage Pools

Create a Storage Pool

Delete a Storage Pool

Add Capacity to a Storage Pool

Import a Storage Pool

Export a Storage Pool

Display History of a Storage Pool

Upgrade a Storage Pool

Replace a Device in a Storage Pool

Snapshots

Create a Snapshot

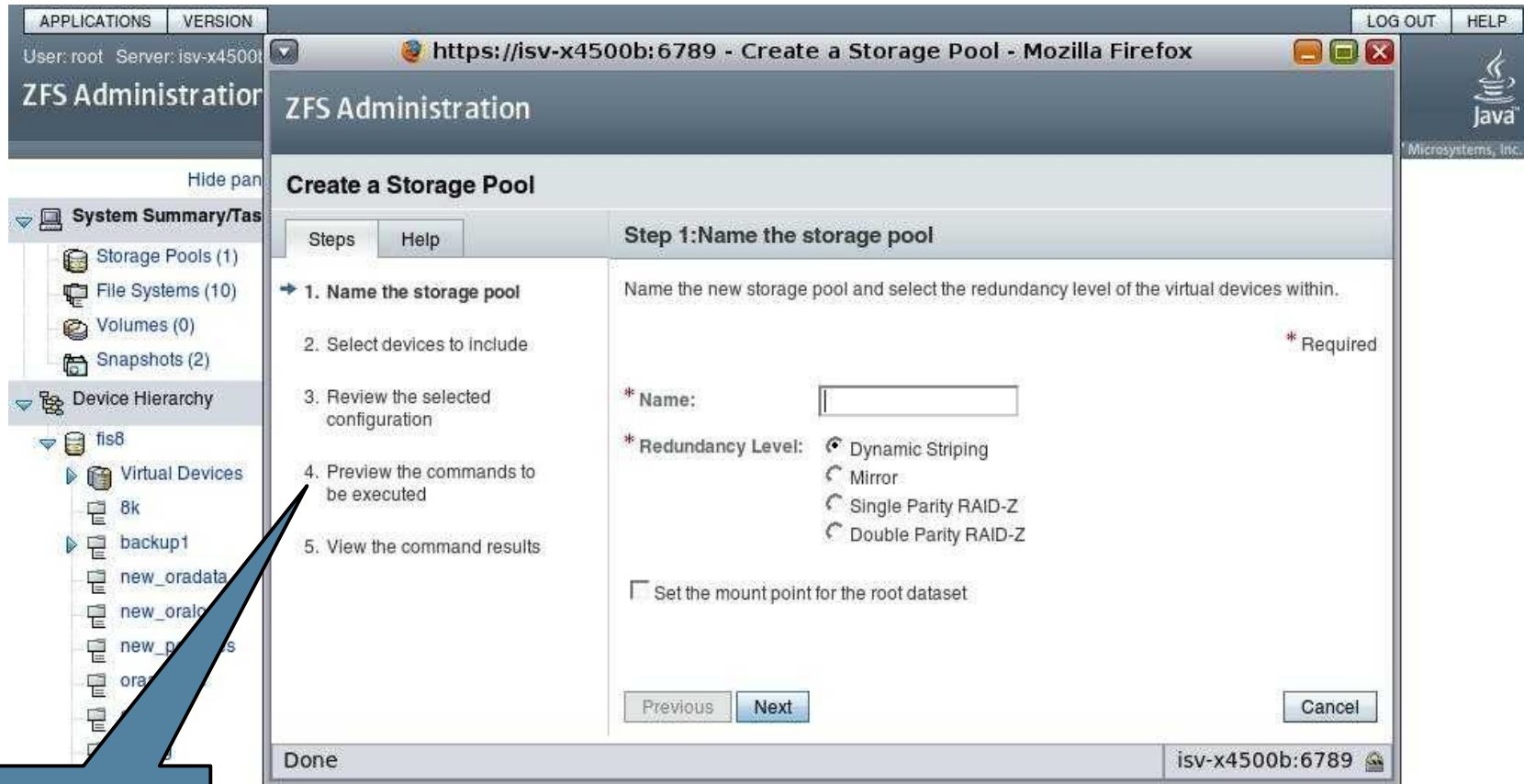
Delete a Snapshot

Clone a File System Snapshot

Clone a Volume Snapshot

# Easier Administration

## Pool Creation Wizard



APPLICATIONS VERSION

User: root Server: isv-x4500b

ZFS Administration

Hide panel

System Summary/Tasks

- Storage Pools (1)
- File Systems (10)
- Volumes (0)
- Snapshots (2)

Device Hierarchy

- fis8
  - Virtual Devices
    - 8k
    - backup1
    - new\_oradata
    - new\_oralc
    - new\_p
    - ora

Steps Help

Step 1: Name the storage pool

Name the new storage pool and select the redundancy level of the virtual devices within.

\* Required

\* Name:

\* Redundancy Level:
 

- ☒ Dynamic Striping
- ☐ Mirror
- ☐ Single Parity RAID-Z
- ☐ Double Parity RAID-Z

☐ Set the mount point for the root dataset

Previous Next Cancel

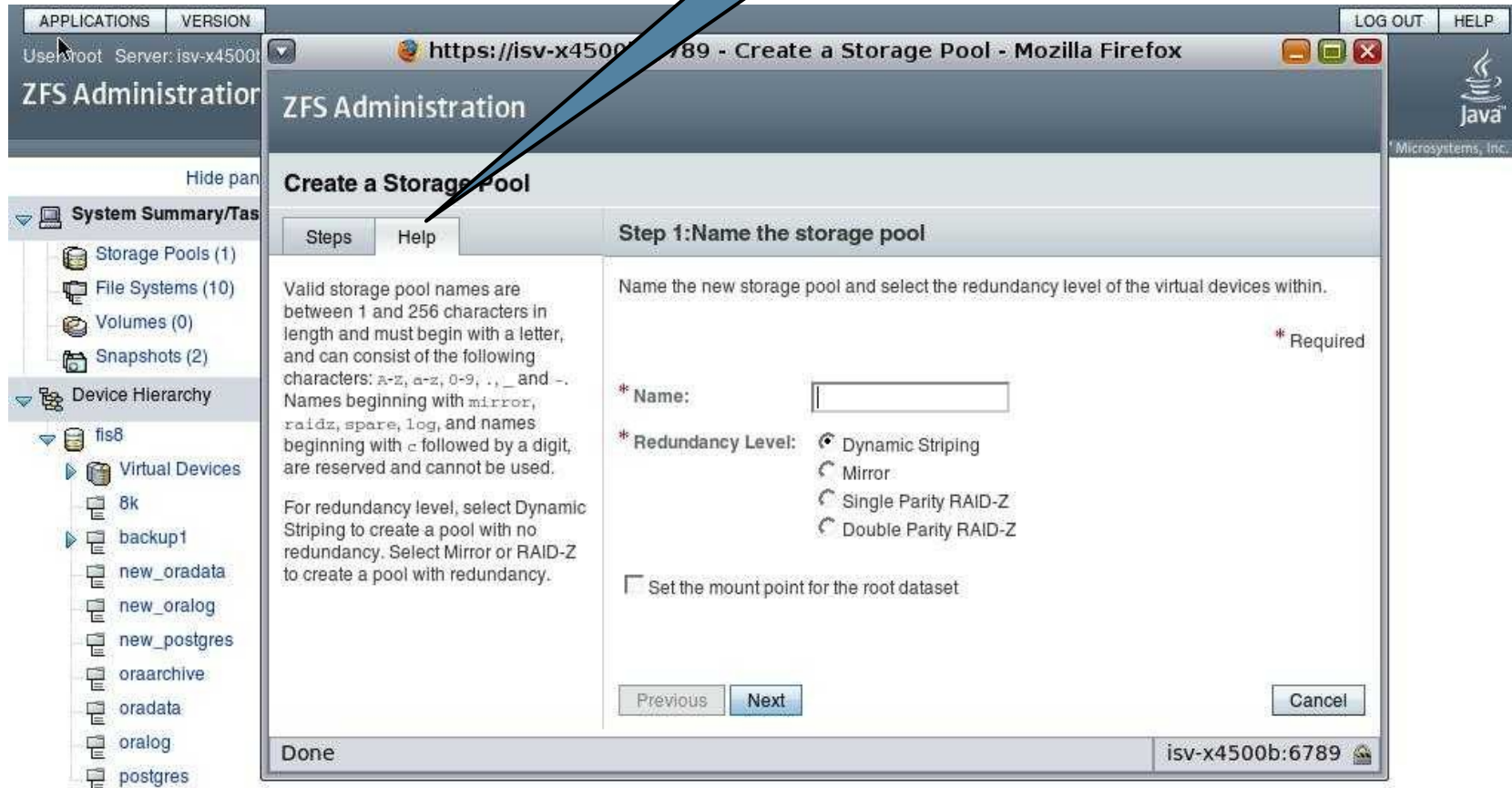
Done isv-x4500b:6789

Can Save  
Commands as a  
Shell Script

# Easier Administration

## Pool Creation Wizard

Help Tab



The screenshot shows the ZFS Administration web interface in a Mozilla Firefox browser window. The address bar shows the URL `https://isv-x4500b:6789 - Create a Storage Pool - Mozilla Firefox`. The page title is "ZFS Administration".

On the left, there is a sidebar with a "System Summary/Tasks" section and a "Device Hierarchy" section. The "Device Hierarchy" shows a tree structure starting with "fis8", which contains "Virtual Devices" (8k, backup1, new\_oradata, new\_oralog, new\_postgres, oraarchive, oradata, oralog, postgres).

The main content area is titled "Create a Storage Pool". It has a navigation bar with "Steps" and "Help" tabs. The "Help" tab is selected, and a blue arrow points to it from the "Help Tab" label above. The "Steps" tab shows "Step 1: Name the storage pool".

The "Help" tab content includes:
 

- Valid storage pool names are between 1 and 256 characters in length and must begin with a letter, and can consist of the following characters: A-Z, a-z, 0-9, ., \_ and -. Names beginning with mirror, raidz, spare, log, and names beginning with c followed by a digit, are reserved and cannot be used.
- For redundancy level, select Dynamic Striping to create a pool with no redundancy. Select Mirror or RAID-Z to create a pool with redundancy.

The "Steps" tab content includes:
 

- Name the new storage pool and select the redundancy level of the virtual devices within.
- \* Required
- \* Name:
- \* Redundancy Level:
  - ☒ Dynamic Striping
  - ☐ Mirror
  - ☐ Single Parity RAID-Z
  - ☐ Double Parity RAID-Z
- ☐ Set the mount point for the root dataset
- Buttons: Previous, Next, Cancel

The bottom status bar shows "Done" and the session ID "isv-x4500b:6789".

# Easier Administration

## File System Creation Wizard

APPLICATIONS VERSION

User: root Server: isv-x4500b

ZFS Administration

LOG OUT HELP

Java

Microsystems, Inc.

Hide panel

System Summary/Tasks

Storage Pools (1)

File Systems (10)

Volumes (0)

Snapshots (2)

Device Hierarchy

fis8

Virtual Devices

8k

backup1

new\_oradata

new\_oralog

new\_postgres

oraarchive

oradata

oralog

postgres

Steps Help

Step 1: Define the file system

1. Define the file system

2. Review the selected configuration

3. Preview the commands to be executed

4. View the command results

Name the new file system and select the parent file system. Optionally, select an existing snapshot on which to base the new file system.

\* Required

\* Parent File System:

Browse...

\* Name:

Snapshot to Clone:

Browse...

☐ Configure the mount point and properties of the new file system

Previous

Next

Cancel

Done

isv-x4500b:6789

# What is ZFS?

**A new way to manage data**

## End-to End Data Integrity

With check-summing and  
copy-on-write transactions

## Easier Administration

A pooled storage model –  
no volume manager



## Immense Data Capacity

The world's  
first 128-bit  
file system

## Integrated Data Services

Snapshots  
Clones  
Replication





128-bit File System

No Practical Limitations  
on File Size, Directory  
Entries, etc.

Concurrent Everything

Uses capacity efficiently

**Immense Data Capacity**



# What is ZFS?

**A new way to manage data**

## End-to End Data Integrity

With check-summing and  
copy-on-write transactions

## Easier Administration

A pooled storage model –  
no volume manager



## Immense Data Capacity

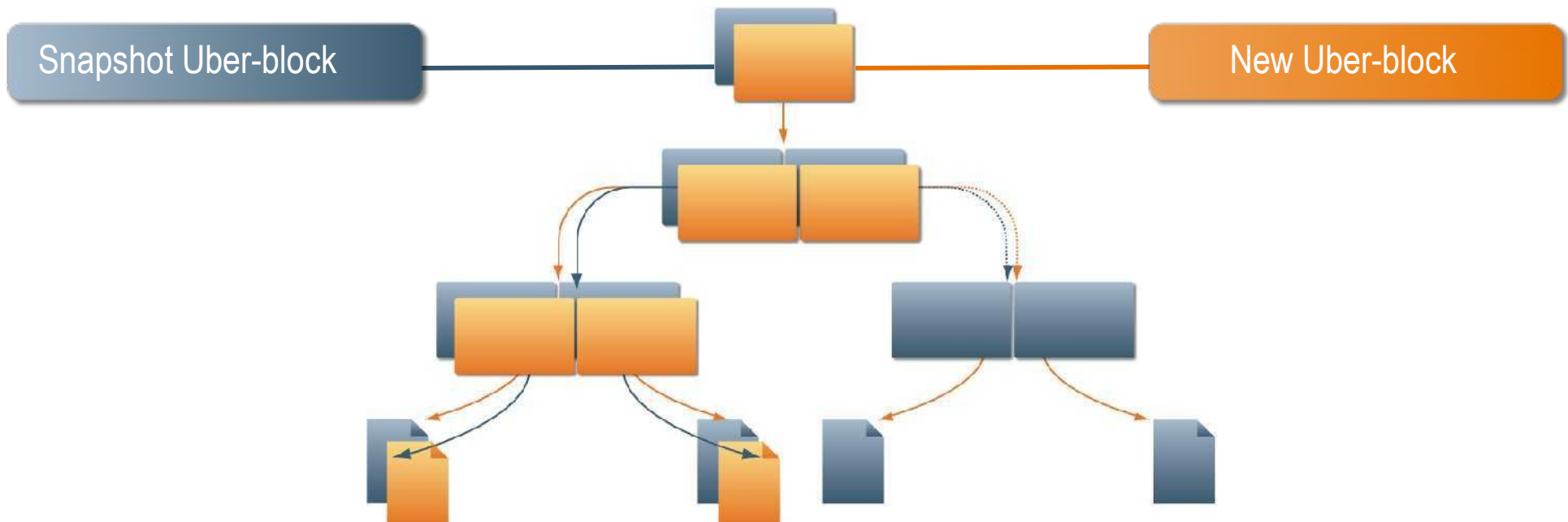
The world's  
first 128-bit  
file system

## Integrated Data Services

Snapshots  
Clones  
Replication

# ZFS Snapshots

- Provide a read-only point-in-time copy of file system
- Copy-on-write makes them essentially “free”
- Very space efficient – only changes are tracked
- And instantaneous – just doesn't delete the copy



# Easier Administration

## Snapshot Creation Wizard



# ZFS Clones

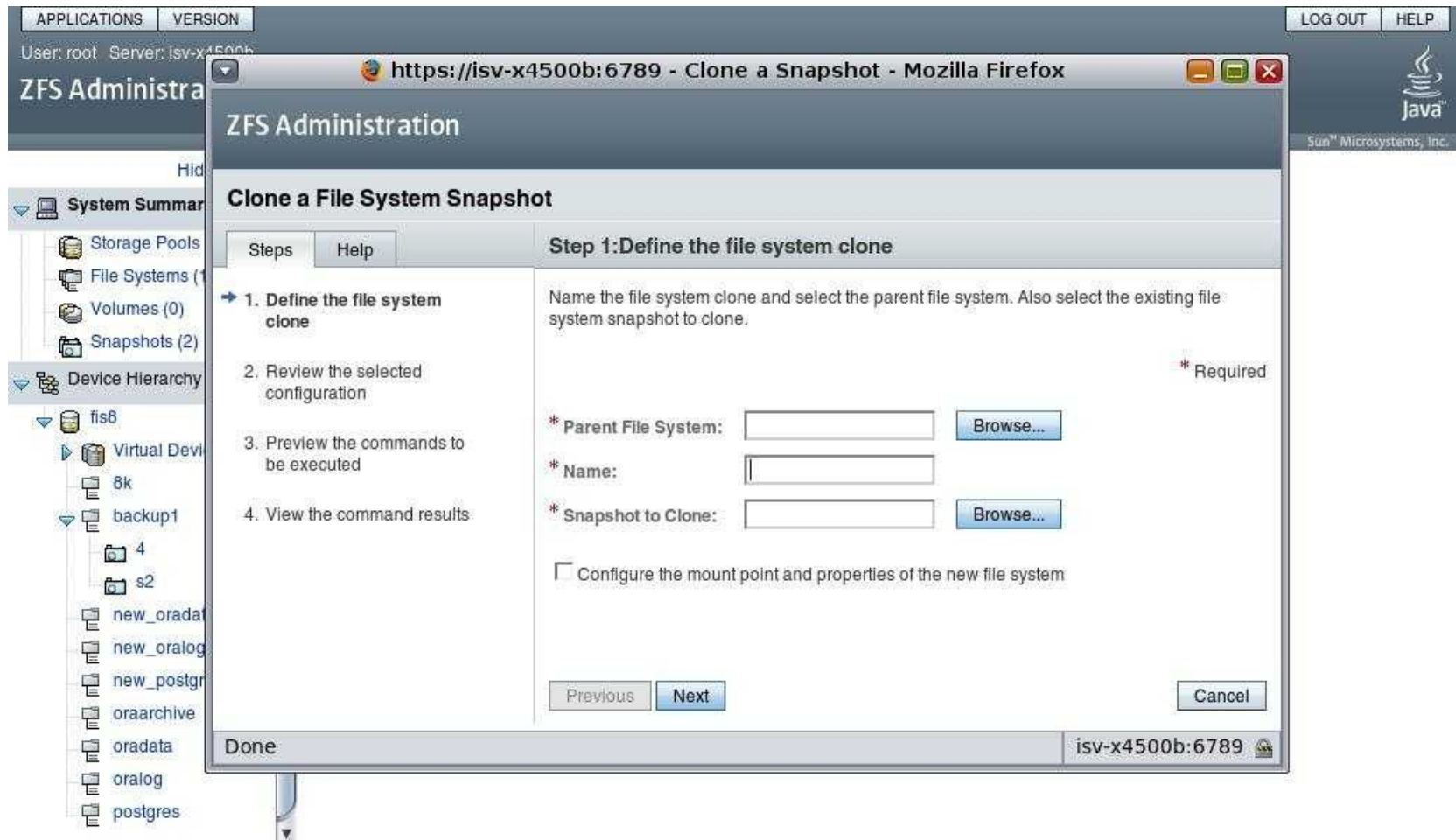
- A Clone is a **Writable** copy of a snapshot

```
# zfs clone tank/myfiles@monday tank/myclone
```

- Near instant creation
- Uses zero space until changes are made to the Clone
- Can safely make available multiple writable copies of the same data to multiple users using clones without making full physical copies
- You can make snapshots of Clones
- Clones can be promoted to replace original file system

# Easier Administration

## Clone Creation Wizard



# Replication

## ZFS Send/Receive

- Powered by snapshots
  - > Full backup: any snapshot
  - > Incremental backup: any snapshot delta
- Generate a full backup to another system

```
# zfs send tank/fs@A | ssh host2 zfs recv newtank/fs
```

- Generate an incremental backup

```
# zfs send -i tank/fs@A tank/fs@B | ssh host2 zfs recv newtank/fs
```

# Solid State Disks, ZFS & Hybrid Storage Pools





# Why Applications Don't Perform

## Waiting for DATA – HDDs can't keep up



- Today's **Multi-Core, Multi-Socket** application server design are increasingly held back by slow storage
- When requesting data, the server spends **most of it's time waiting for storage**
- Application performance **remain sluggish** regardless of the Server CPU horsepower
- The traditional remedy of adding more expensive **DRAM may no longer suffice** as data sets double every 2 years

# New Server Memory Hierarchy



# Solid State Drives (SSD)

## Enterprise advantage from commodity FLASH

- SSD has three major parts:
  - > A) Controller
  - > B) DRAM
  - > C) FLASH bank
- Controller also performs
  - > Wear leveling
  - > CRC
  - > Bad block mapping
- Controller provides the host interface such as SATA, SAS or FC



# Cost Effective Performance

SSDs are 70X more cost effective



- Enterprise HDD
  - > \$5 GB
  - > 180 Write IOPS
  - > 320 Read IOPS
  - > 300 GB
  - > ~18W
- \$ per IOPS: 2.43



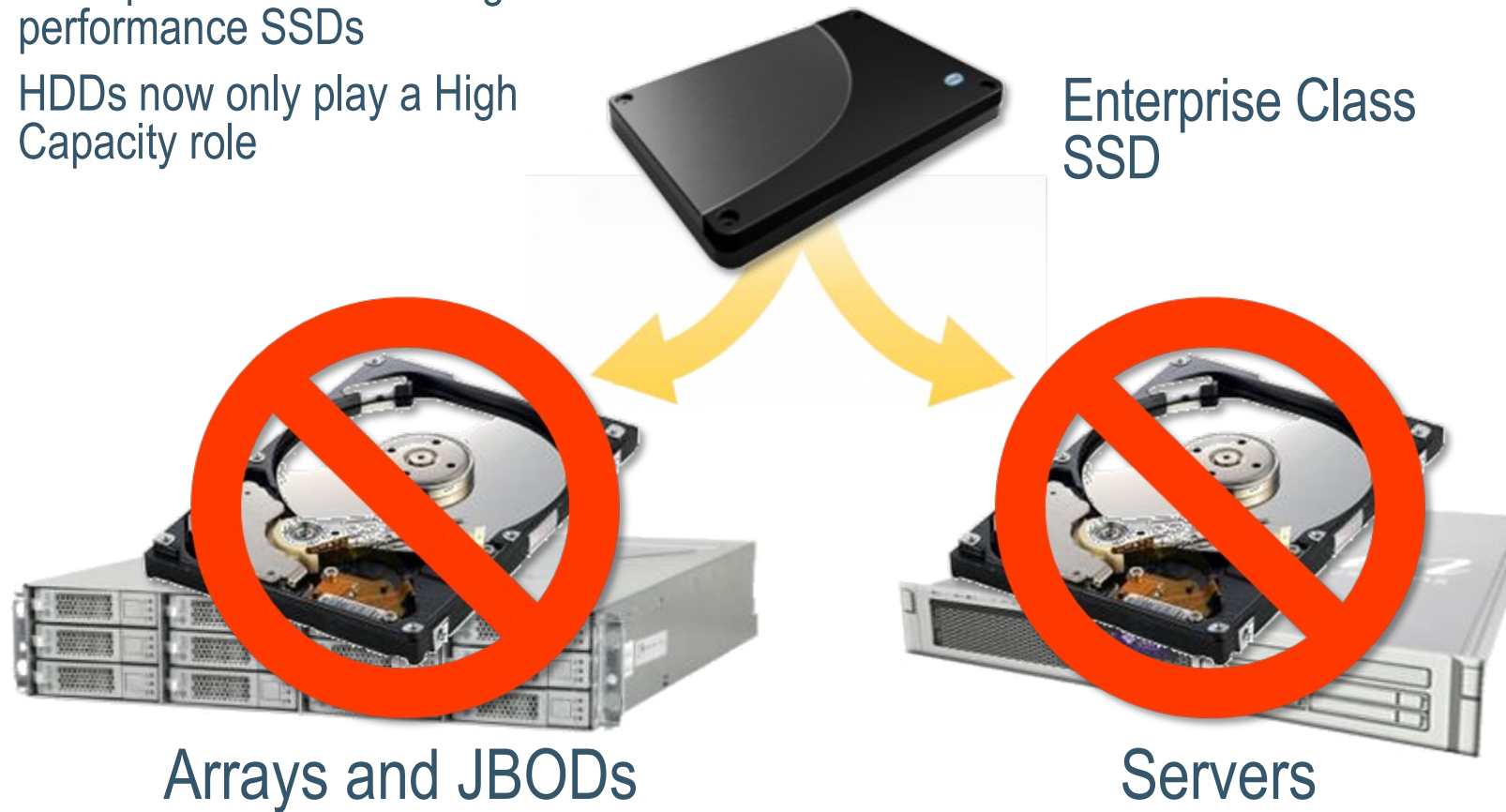
- Enterprise SSD
  - > \$35 GB
  - > 7,000 Write IOPS
  - > 35,000 Read IOPS
  - > 32GB
  - > ~3W
- \$ per IOPS: 0.04



# Sun SSD Strategy

## HDD Replacement

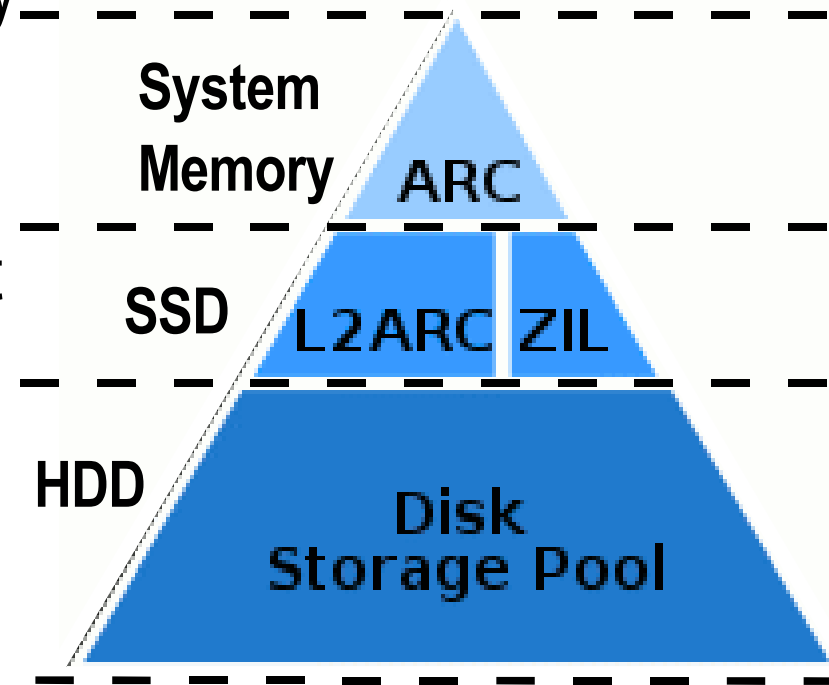
- High performance 15K HDDs are replaced with 150X higher performance SSDs
- HDDs now only play a High Capacity role



# ZFS Hybrid Storage Pools

## SSDs Accelerate Synchronous Writes

- ZFS caches blocks in system main memory
- The Cache is called the ZFS Adaptive Replacement Cache (ARC)
- All synchronous writes go to the ZFS Intent Log (ZIL) before they can complete
- ZFS can separate the ZIL onto separate devices (ZFS log devices or slogs)
- Putting the ZIL on SSDs will improve the response times of synchronous writes

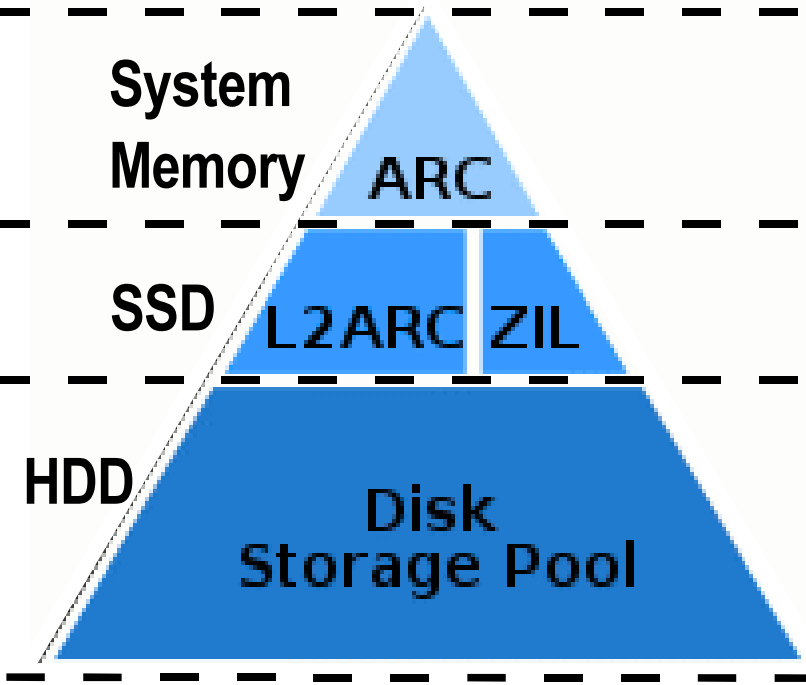


*Read Neil Perrin's blog on [blogs.sun.com](http://blogs.sun.com) for more on ZFS log devices*

# ZFS Hybrid Storage Pools

## SSDs Accelerate Reads – ARC & L2ARC

- Older/least frequently accessed blocks are evicted from ARC by newer data or due to application demands for memory
- Blocks evicted from the ARC are written into L2ARC
- On reads, if we miss the ARC we go to the L2ARC
- If the reads miss the L2ARC we go to disk
- Pre-fetched data does not go into L2ARC
- SSDs used as L2ARC gave a 730% performance improvement over just 7200RPM drives in a recent benchmark

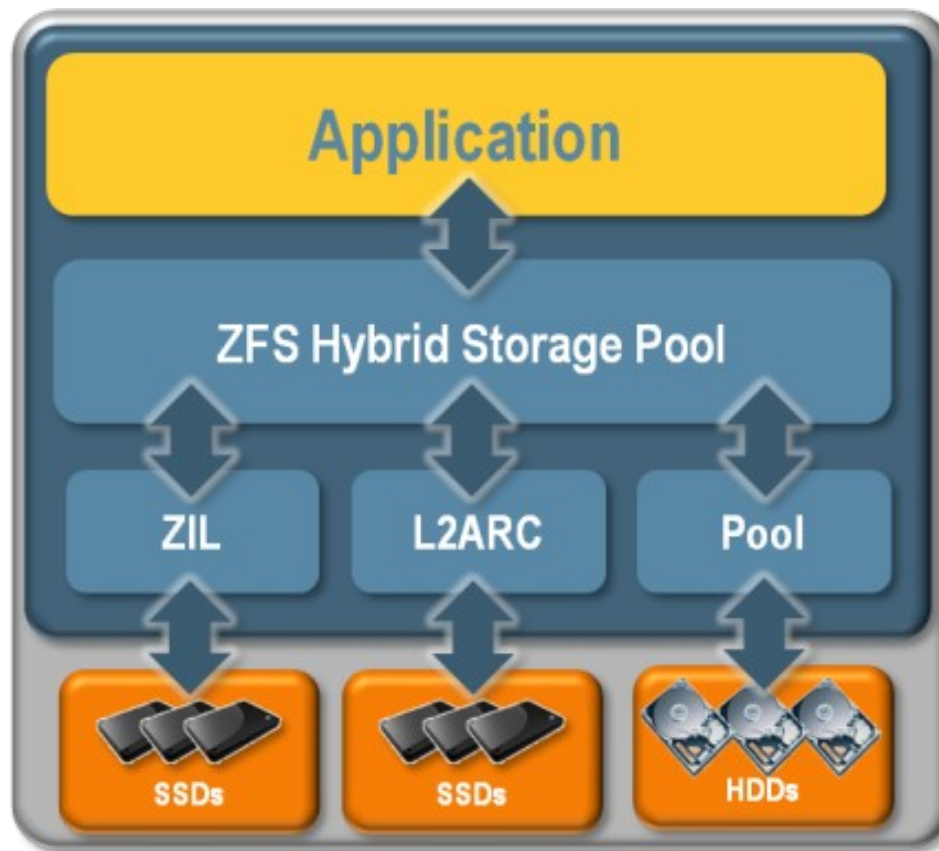


**For more see Brendan Greg's blog: <http://blogs.sun.com/brendan/entry/test>**



# ZFS Turbo Charges Applications

## Hybrid Storage Pool Data Management

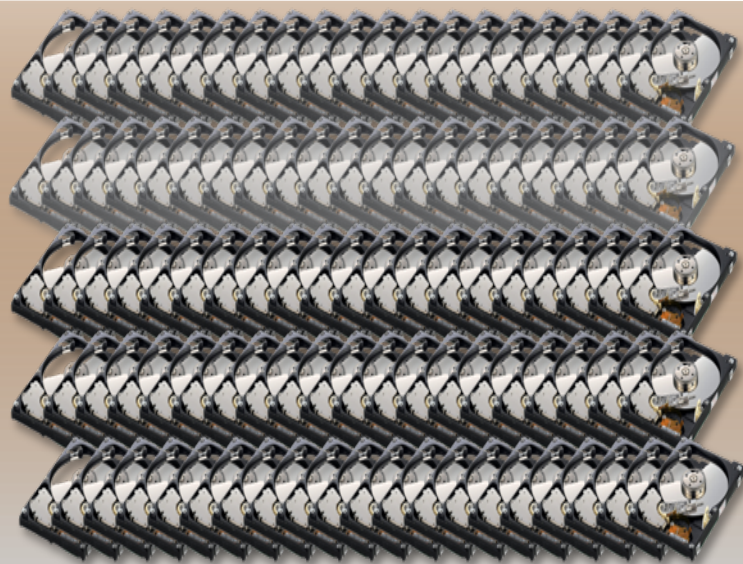


- Latency Sensitive Writes & Reads handled by SSDs
- Bulk transfers handled by HDDs

# ZFS Hybrid Storage Pools

**Faster, Cheaper, Less Power**

Enterprise HDDs



More IOPS  
Lower \$/GB  
Lower Power Consumption  
Less Rack Space

Hybrid Storage Pool



**For more on HSPs, see Adam Leventhal's article in the Communications ACM Magazine <http://mags.acm.org/communications/200807/>**

# ZFS Overview

**Dominic Kay**

**[blogs.sun.com/dom](http://blogs.sun.com/dom)**

**[dominic.kay@sun.com](mailto:dominic.kay@sun.com)**

