



ZFS & DTrace

Vítězslav Bátrla
Solaris RPE
Sun Microsystems

ZFS

ZFS file-system

ZFS

First 128-bit file-system

Designed to have simple user interface

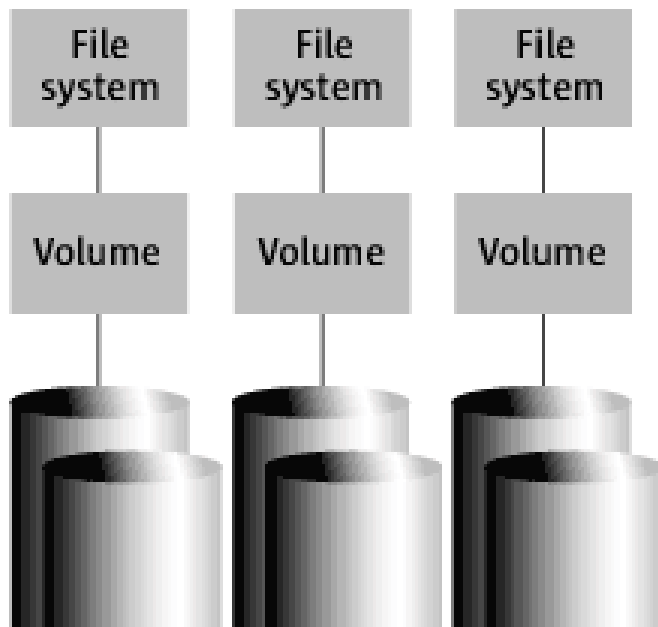
Combines volume manager and file-system

Available in Solaris 10, OpenSolaris

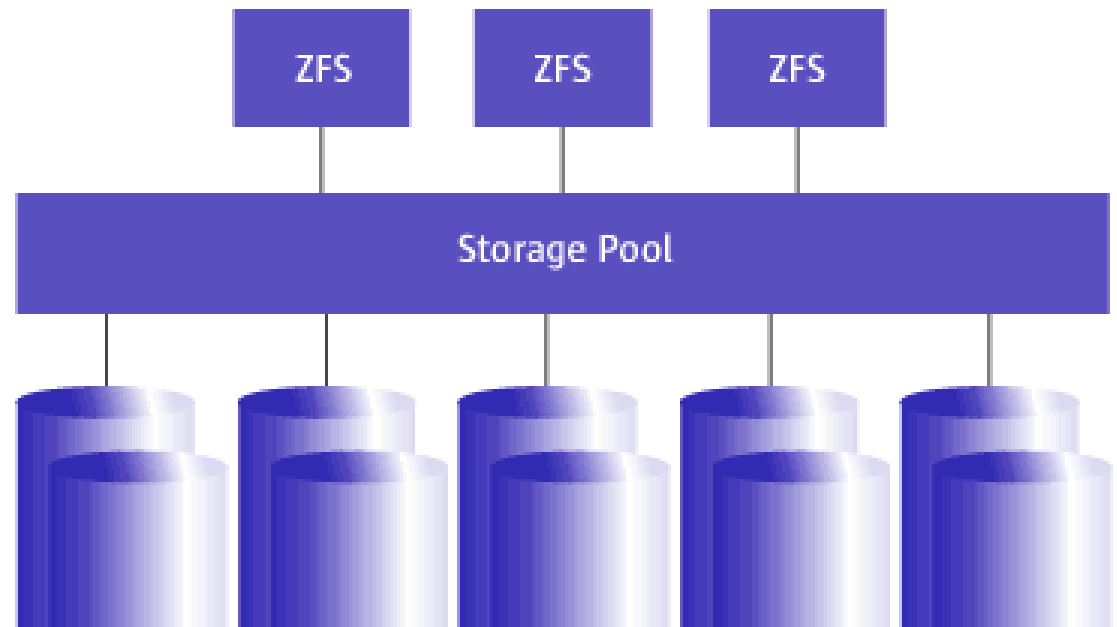
ported to BSD, Mac OS X, Linux?

ZFS

Traditional Volumes



ZFS Pooled Storage



Storage pools (zpools)

Similar to volume manager layer

With advantage that all pool capacity is available to all file-systems

No need to resize partitions (just add a disk to pool)

Consists of vdevs (virtual devices)

Whole disk, disk partition, plain file

Add vdev to pool on the fly

Vdev configurations

RAID 0, RAID 1, RAID-Z, RAID-Z2

(similar to RAID 5/6)

ZFS filesystems

Created on top of storage pool

- Instant creation of filesystem

Simple like directories

Supports reservations, quota

- reservation – guaranteed capacity (minimum)

- quota – maximum allowed capacity

Each filesystem has many options

- Transparent compression (lzjb, gzip)

- Case sensitivity

- Delegated administration

Copy-on-write

Copy-on-write transactional model

- Guarantees consistency

- Allows instant snapshots and clones

Snapshot

- Read-only point in time copy of filesystem

- Useful for backups

Clone

- “Writable snapshot”

- Avoids duplication of data on disk

 - Write data once, take snapshot, create multiple clones

 - Multiple boot environments

Other features

FS level checksum

Detects silent data corruption

Adaptive endianness

Use the same disk on different platforms

Dynamic striping

Striping the load to multiple devices

Scrubbing and resilver operations

Support for SSD disks

SSD in zpool are fast, ZFS uses them as cache

ZFS Demo

Demo 1

Storage pool creation – RAID 1

Snapshots

Checksum

Accidental overwrite

Demo 2

Cloning Linux zones (CentOS) on OpenSolaris

Saves disk space

Saves time

Quick deployment of new zone

ZFS Q&A

Questions?

DTrace

Dynamic Tracing

What's DTrace?

Dynamic tracing framework

Designed for production systems

Unlike running debugger on live system

Tracing user-level application and kernel

No need to recompile the binary

(doesn't require debug builds)

No performance impact on the system when off

Supports C, Java and now scripting languages

Javascript, Python, Ruby, PHP ...

D programming language

A new programming language

Subset of “C” language with tracing support

Designed for safety

- No dangerous “loops” or “if” constructs

D scripts

- Like shell scripts, the interpreter is DTrace

Tracing is controlled by D programs

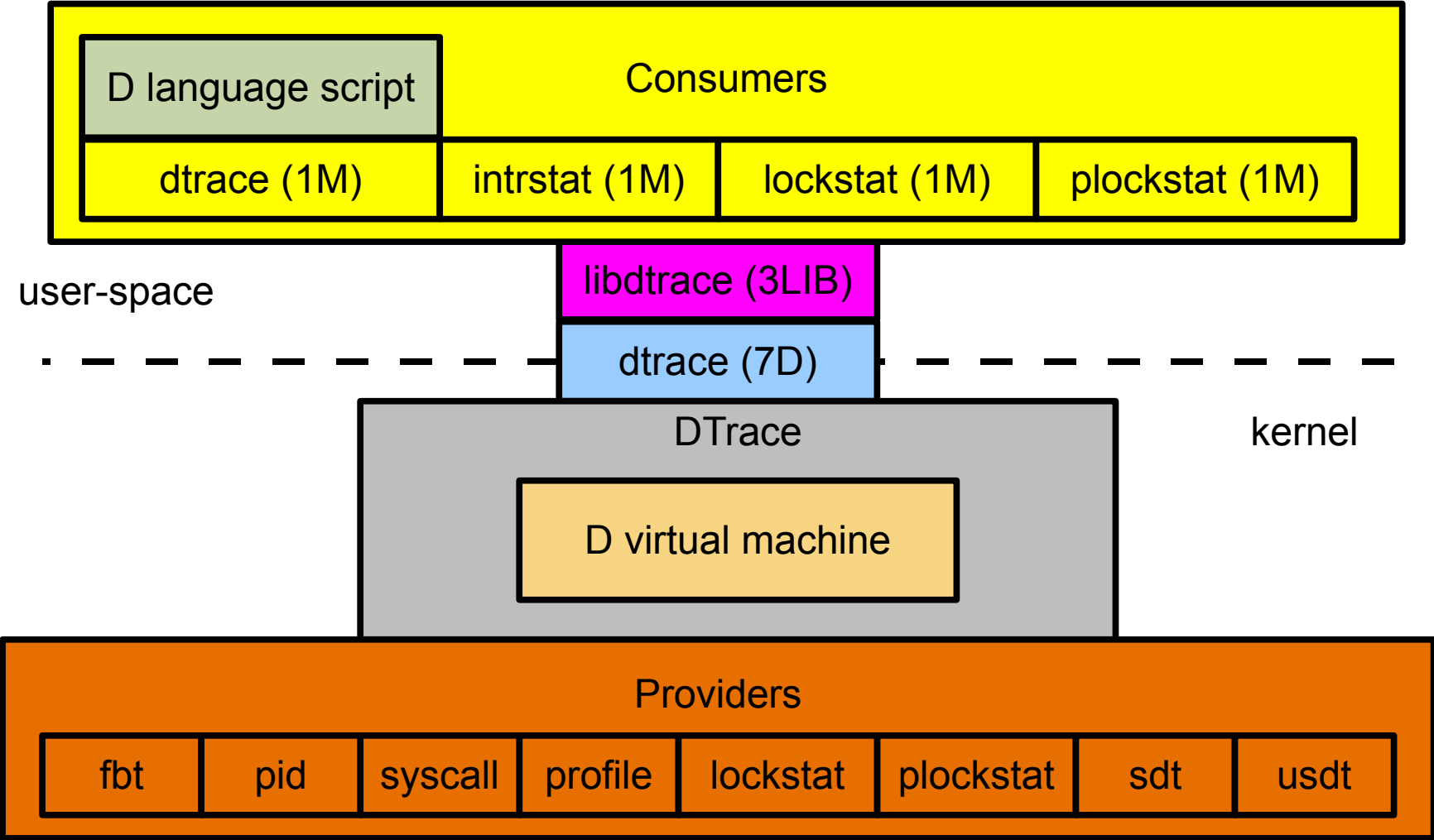
- Probes specify what instrument

 - Function entry, exit points

- Actions describe what is done when probe fires

 - Printing arguments, return value, dump memory and much more...

Architecture



DTrace demo

Demo 1

Observing C program

Demo 2

Syscall

Demo 3

Aggregations

Demo 4

iostat

DTrace Q&A

Questions?



ZFS & DTrace

Vítězslav Bátrla
Solaris RPE
Sun Microsystems