



MSC Malaysia

OPEN SOURCE CONFERENCE **2009**

Satyajit Tripathi and Sivakumar S.
ISV-Engineering, Sun Microsystems

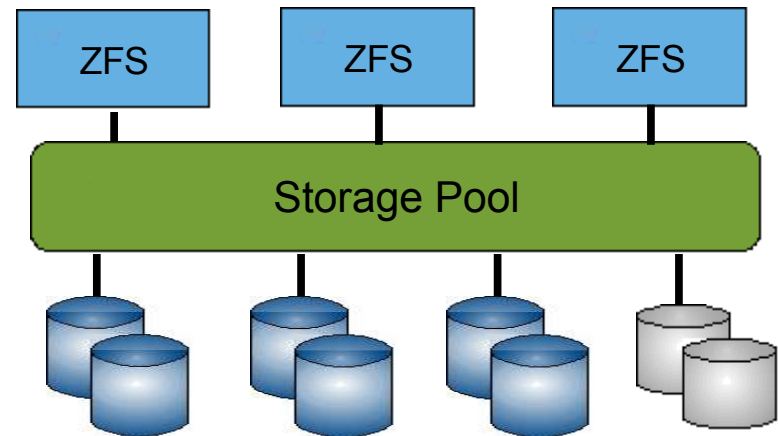
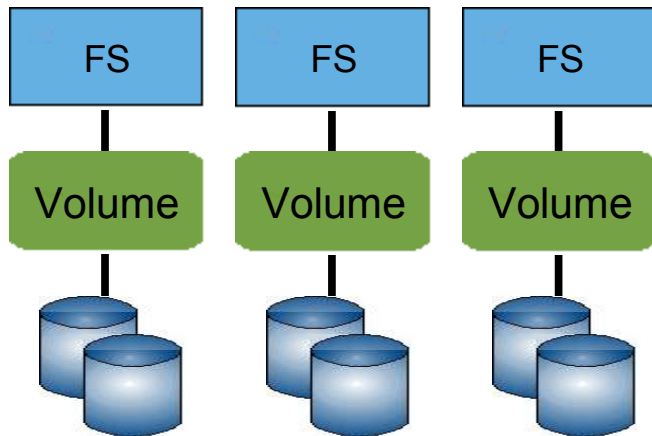
OpenSolaris Advanced Technologies

- ZFS (Zettabyte File System)
- SMF (Service Management Facility)
- Zones and Containers
- DTrace
- And more ...

Zettabyte File System (Zeta = 10^{21})

- Better, Safer way to Manage Data
- First 128-bit File System
 - Capacity 1 billion TB. Virtually unlimited.
- Fast and High Performance
- No Silent Data Corruption
 - End-to-end integrity
 - Aggressive 256-bit checksums
- Instantaneous Snapshots
 - Almost Zero Overhead and Easy Rollback
- Application Compatibility
- Easy Administration. No Volume Manager
- Efficient Mirroring and Remote Replication

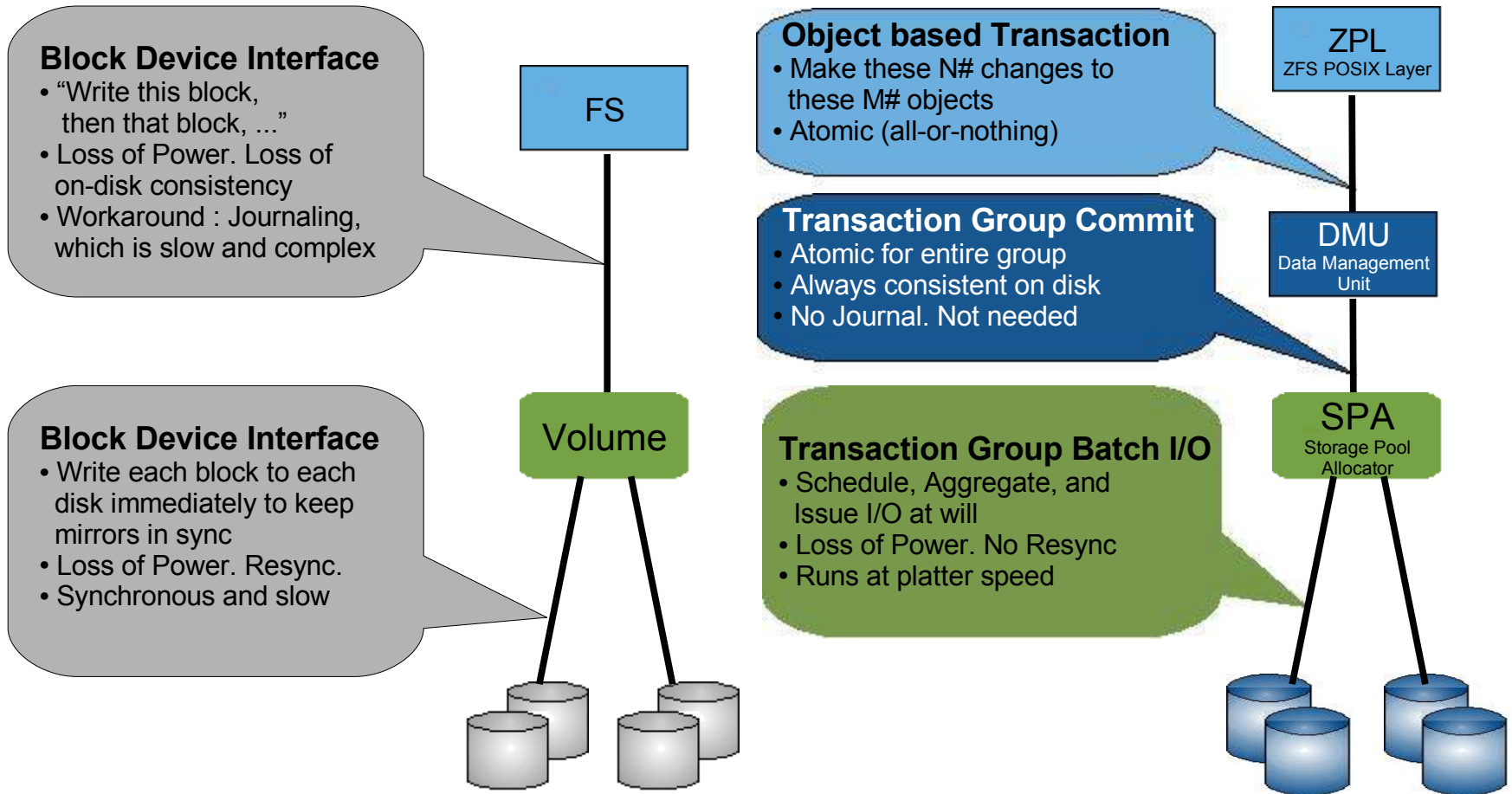
ZFS Virtually Unlimited



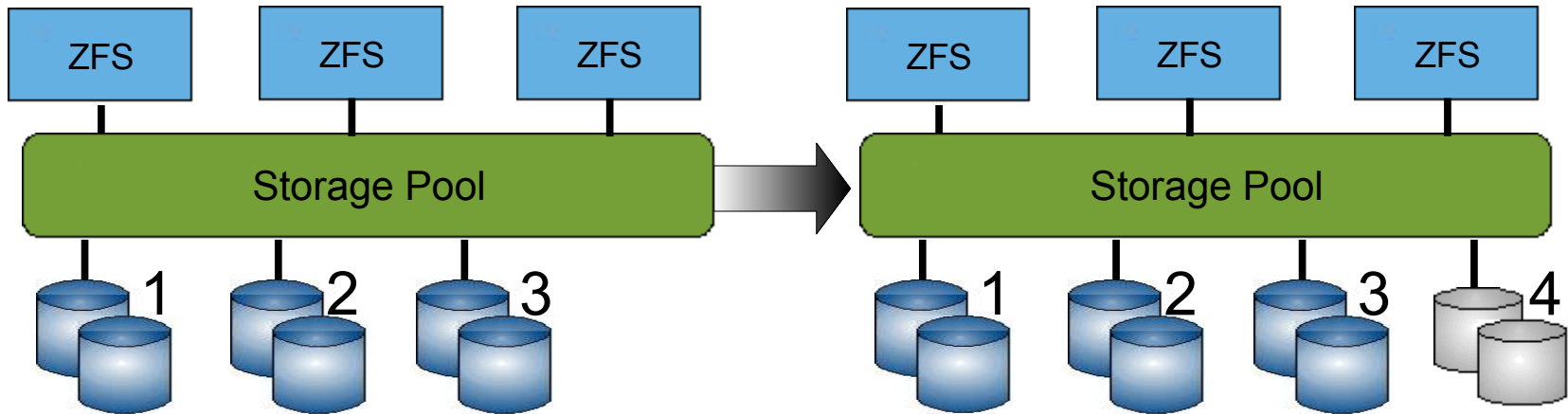
- Traditional Volumes
- Abstraction : Virtual disk
- Partition for each FS
- Grow or shrink manually
- Storage is fragmented, stranded

- ZFS Pooled Storage
- Abstraction : malloc, free
- No partition to manage
- Grow or shrink automatic
- Storage in the Pool is shared

ZFS I/O Stack



ZFS Dynamic Stripping



BEFORE

- Writes : striped across all 3 mirrors
- Reads : wherever data was written
- Block Allocation Policy Considers
 - Capacity
 - Performance (latency, BW)
 - Health (degraded mirrors)

AFTER

- Writes : striped across all 4 mirrors
- Reads : wherever data was written
- No need to migrate existing data
 - Old data striped across 1-3
 - New data striped across 1-4
 - Copy-On-Write (COW) gently reallocates old data

ZFS Additional Capabilities

- Intelligent Prefetch
 - Multiple Independent Pre-fetch Streams
 - Very useful for Streaming Service Provider
 - Automatic Length and Stride Detection
 - Very useful for HPC Applications
- Variable Block Size
- Built-in Compression. Built-in Encryption
- Per-user and Per-group Quota Support
- Common Internet FS (CIFS) Support
- Integrated with Zones and FMA
- And more ...

Service Management Facility (SMF)

- Simplified Service Management
 - Services are Objects easily managed with few simple commands
- Automated Restart of Failed Services
 - SMF Monitors services and Pro actively Restart on Failure
- Persistent Service Configuration
 - Service Definition and Configuration persist across reboot
- Explicit Dependencies
- Easier Debugging
- Faster Boot/Shutdown Processes
- Delegated Service Administration
 - Administrative Role Delegation to non Root users

SMF Service Components

○ SMF Service Manifest

- Defines Default Service Properties in the Repository
- System Manifest files under `/var/svc/manifest`. Imported at boot

○ Service Start Method(s)

- Defines the Interaction between Service Restarter and the Service

○ Service Executable(s)

- Executable called by Method(s) to Implement the Service

○ Service Log

- Records the output of the Service. Useful in Debugging Failure
- `/var/svc/log/milestone-multi-user-server:default.log`

○ Service Fault Management Resource Identifier (FMRI)

- Identifies specific Service Instance
Example : `svc:/milestone/multi-user-server:default`

Zones and Container

- Virtualization instrumented within OS Kernel
 - Resource Management
 - Controlled and Dynamic Resource Allocation
 - Prioritize Applications
 - Name space, Security, and Fault Isolation
- Light Weight and No Overhead
 - Theoretical limit is 8192
- Easy to Clone and Migrate
 - Rapidly create and multiply OS environment
- Increase System Utilization
 - Scales with OS
- Single Point Administration at Global Zone

Zone States

○ Configured

- Configuration completed and committed

○ Installed

- Packages installed successfully

○ Ready

- Virtual Platform established

○ Running

- Zone booted successfully and running

○ Shutting Down

- Temporary state in the process of shutting down

○ Down

- Temporary state completed shutting down, to go to Installed state

Zones Usefulness for Developers

- Virtual OS Environment Partitioning
 - One large system can be utilized by multiple users non-intrusively
- Rapidly proliferate similar Development Environment
 - Clone within the same System
 - Migrate to the new System
- Simulate Distributed Application on multiple zones
- Apply latest OS patch Once on the Global Zone
 - Available to all Zones
- Preserve the Pre-configured Development or Deployment Environment with simple commands

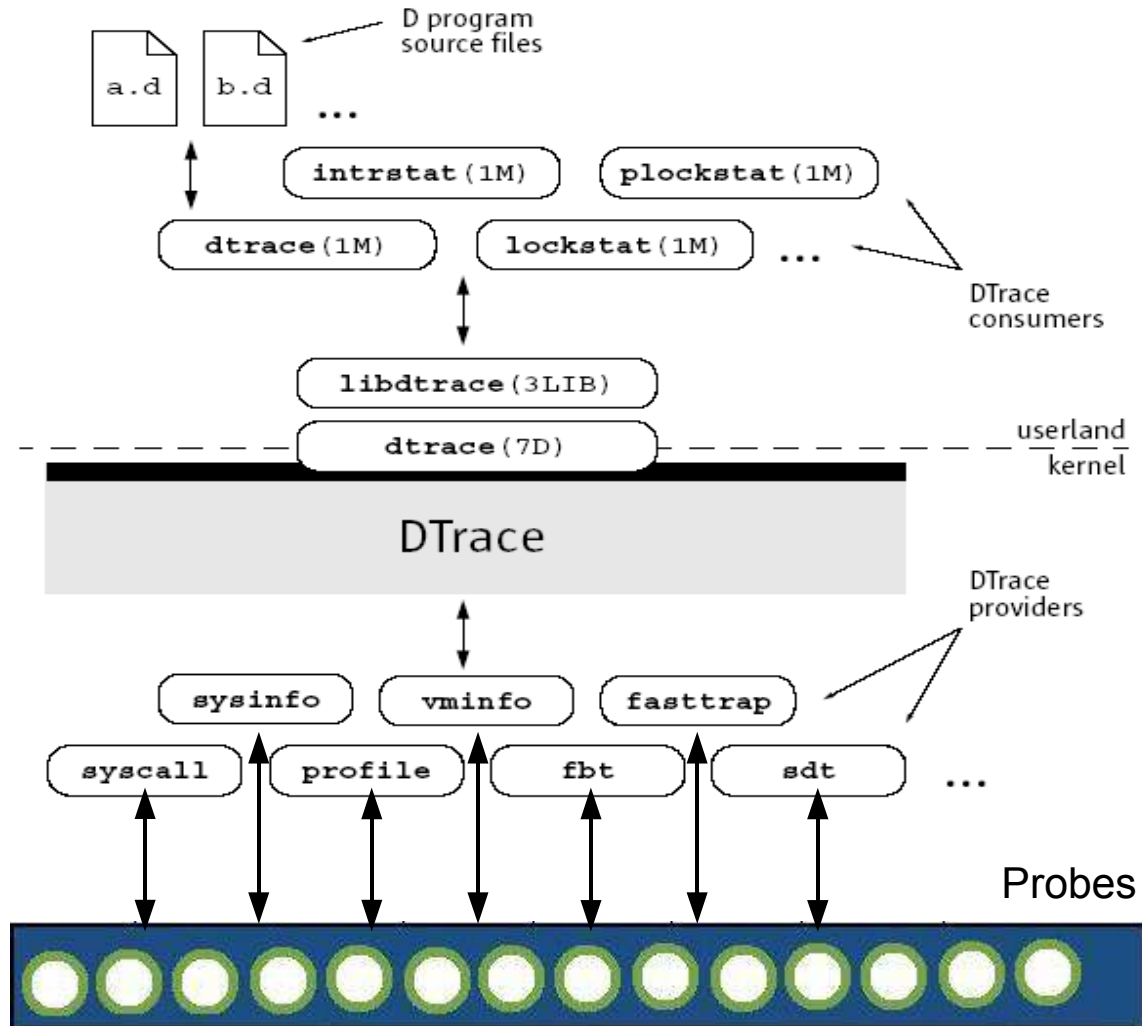
Branded Zone (BrandZ)

- BrandZ extends Solaris Zones Infrastructure
 - Brand Attribute set at Zone Create time
 - Branded Installation Routine for Arbitrary Collection of Software
 - Pre/Post-boot scripts for boot-time setup and configuration
 - Simple commands zoneadm and zonecfg
- Brand Ix to run Linux binary application unmodified within a Zone with Solaris Kernel
- Brand Ix runs on x86/x64 booted with 32/64-bit Kernel

Dynamic Tracing (D Trace)

- DTrace, A Comprehensive Framework for Solaris™ Operating Environment
 - Implement New DTrace Providers
 - Implement fully Configurable DTrace Probes
 - Implement New DTrace Consumers and Data Display
- Observability using DTrace
 - Aggregate Arbitrary Behavior of the OS and User Programs
 - Dynamically Enable and Manage Probes
 - Dynamically Associate Predicates and Actions with Probes
 - Dynamically Manage Trace Buffers and Probe Overhead
 - Examine Live Production System or a Crash Dump

DTrace World



D Language

- 'D' Language, like 'C' and constructs similar to awk
- Complete Access to Kernel C types
- Complete Access to Statics and Globals
- Rich built-in Variable set
- Complete Support for ANSI-C Operators
- Example : Trace the pid of every process named date with syscall open(2)

```
#!/usr/sbin/dtrace -s  
  
syscall::open:entry  
/execname == "date"/  
{  
    trace(pid);  
}
```

Probe Name
o Entry point of open

Predicate
o Process is named "date"

Action
o Print the process ID

DTrace Probe

- Point of Instrumentation within OS Kernel
- Has a **Name**
- Identifies the **Module** and **Function** it Instruments
- Accessible through **Provider**
- 4 Attributes, Name, Module, Function & Provider, defines a tuple to uniquely identify a Probe

```
probe description (provider:module:function:name)
/ predicate /
{
    action statements
}
```

- Each Probe is Assigned an Integer Identifier

DTrace Provider

- Methodology for Instrumenting Probes
- Provides Probe Access to DTrace Framework
- Informed by DTrace to Enable a Probe
- Transfers Control of the Probe to DTrace when Enabled

DTrace Consumer

- Process that interacts with DTrace
- Concurrent Consumers unlimited. DTrace handles Multiplexing
- dtrace(1M) is DTrace Consumer. Generic front-end to the DTrace Facility

Write an Application Probe

○ Define Provider

```
provider myserv {  
    probe query_receive(string, string);  
    probe query_respond();  
}
```

○ Include sdt.h

○ Add Probe code to the Application

```
query = wait_for_new_query();  
DTRACE_PROBE2(myserv, query_receive, query->clientname, query->msg);  
process_query(query)
```

○ Compile the Application

```
dtrace -G -32 -s myserv.d src1.o src2.o ...  
cc -o myserv myserv.o src1.o src2.o ...
```

Performance Tools

○ Process Stats

cpustrack : per-processor hardware counter
pargs : process arguments
pflags : process flags
pcred : process credentials
pldd : process library dependency
psig : process signal disposition
pstack : process stack dump
pmap : process memory map
pfiles : open files and names
prstat : process statistics
ptree : process tree
ptime : process micro-state times
pwdx : process working directory

○ Process Control

pgrep : grep for processes
pkill : kill processes list
pstop : stop processes
prun : start processes
prctl : view/set process resources
pwait : wait for process
preap : reap a zombie process

○ Process Tracing & Debugging

abitrace : trace ABI interfaces
dtrace : trace the world
mdb : debug/control processes
truss : trace functions, system calls

○ Kernel Tracing & Debugging

dtrace : trace and monitor kernel
lockstat : monitor locking statistics
lockstat -k : profile kernel
mdb : debug live and kernel cores

○ System Stats

acctcom : process accounting
busstat : Bus hardware counters
cpustat : CPU hardware counters
iostat : IO & NFS statistics
kstat : display kernel statistics
mpstat : processor statistics
netstat : network statistics
nfsstat : nfs server stats
sar : kitchen sink utility
vmstat : virtual memory stats

Thank You! for Participating



Track back URL

<http://blogs.sun.com/stripathi>

Track back URL

<http://blogs.sun.com/stripathi>