



Becoming an OpenSolaris Power User

Dave Miner and Nicholas Solter
Authors of "OpenSolaris Bible"

Learn how to harness the power of advanced OpenSolaris features - ZFS, Networking, DTrace, and more!

goal

Agenda

- Overview of OpenSolaris Power Features
- ZFS
- Networking
- DTrace

Agenda

- Overview of OpenSolaris Power Features
- ZFS
- Networking
- DTrace

OpenSolaris Power Features

Software Management

- LiveCD iso
- Simple interactive installer
- Automated Installer
- Image Packaging System
 - Network package repositories
 - Easily add software with graphical or command line tools
 - Software updates with graphical or command line tools
- Boot environments
- Distribution Constructor

OpenSolaris Power Features

File Systems

- ZFS
 - Much more in a few minutes...
- UFS and SVM
 - Legacy file system and volume manager
- NFS and CIFS
 - File-share with Windows and Linux
- NIS and LDAP
 - Manage directory services

OpenSolaris Power Features

Networking

- Network Auto-Magic
- Multicast DNS
- Inetd integrated with Service Management Facility
- Firewall capability with ipfilter
- Virtualization and Availability - Details in a few minutes...

OpenSolaris Power Features

Security

- Secure by Default
 - Most network services off
- Role-based access control (RBAC)
 - Similar to sudo
- IPsec
- Logs and auditing
- Kerberos
- Trusted Extensions

OpenSolaris Power Features

Reliability and Availability

- **Fault Management (FMA)**
 - Unified infrastructure for predictive self-healing
 - Both hardware and software
- **Service Management (SMF)**
 - Consistent administration model for managing system services
 - Replaces rc scripts
- **High-Availability Clustering**
 - Increase the availability of your system as a whole using hardware redundancy

OpenSolaris Power Features

Monitoring and Observability

- Many useful observability tools
 - prstat, vmstat, mpstat, iostat, cpustat
 - proc-tools
 - And many more
- Dynamic Tracing (DTrace)
 - More in a few minutes...

OpenSolaris Power Features

Virtualization

- Provide illusion of exclusive access to shared system resources
- Resource Management
 - Manage workloads running on the same OS instance
- Zones
 - Operating-system level virtualization
 - Single kernel, but secure and isolated environments
- xVM Hypervisor and Logical Domains (Ldoms)
 - Full hypervisor based virtualization supporting guest domains
 - x86 and SPARC (sun4v) respectively
- VirtualBox
 - Runs on guest operating system

OpenSolaris Power Features

Software Development

- Webstack integration
 - Apache, Apache Tomcat, MySQL, PostgreSQL, GlassFish,...
- Java Development
- C and C++ Development
 - GCC or SunStudio
- Perl, Python, Ruby, PHP, ...
- NetBeans IDE
- Mercurial and Subversion SCM
- Sophisticated debugging support
 - dbx, GDB, mdb, libumem.,,,

Agenda

- Overview of OpenSolaris Power Features
- ZFS
- Networking
- DTrace

ZFS

High-end storage for everyone

- Efficiency via pooled model
- High reliability
- Superb scalability – both up *and* down
- Superior ease-of-use
- Flexible architecture evolving with storage technology

ZFS Pools

- Managed using `zpool` command
- Pool Types
 - Simple – Single-device (disk or partition)
 - Concatenation – Multiple devices without redundancy
 - Mirror – Full duplication across devices aka RAID1
 - RAIDZ – ZFS version of RAID5
- Performance and Availability Capabilities
 - Spare Devices
 - Log Devices
 - Cache Devices
- Operational history - `zpool history`
- Simple performance monitoring – `zpool iostat`
- Fewer pools for better efficiency

ZFS Datasets

- Managed using `zfs` command
- Three types
 - File system
 - Volume
 - Snapshot – point-in-time copy of file system or volume
- Easy delegated management – `zfs allow`
- Use file systems liberally, they're the primary management boundary!

ZFS Dataset Properties

- Sharing – `sharenfs`, `sharesmb`, `shareiscsi`
- Reliability and space – `copies`, `compression`
- Security – `aclinherit`, `aclmode`, `devices`, `exec`, `setuid`
- And many more...
- Plus user properties!

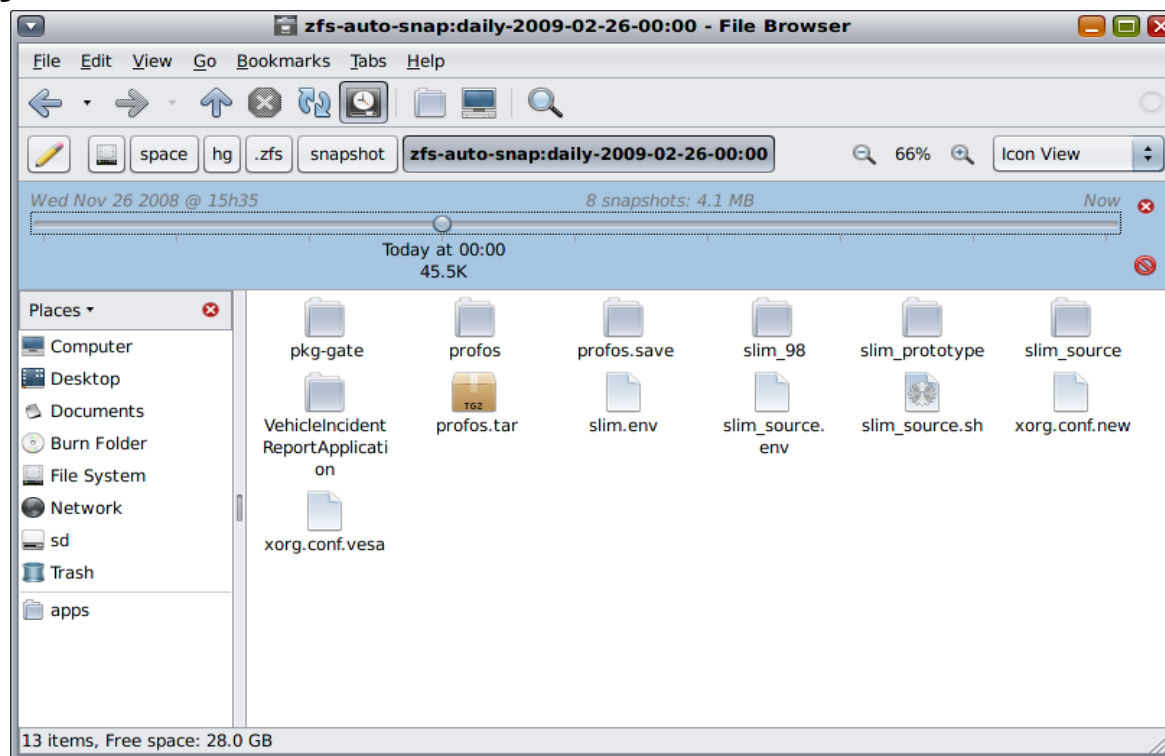
ZFS Demonstration

demo

Time Slider

Integrating ZFS with the GNOME desktop

- Automatic snapshots scheduled via cron
- Nautilus file browser extended to browse snapshot history



Time Slider Demonstration

demo

Agenda

- Overview of OpenSolaris Power Features
- ZFS
- Networking
- DTrace

Network Virtualization and Availability

- Virtual Links (VLANs)
- Link Aggregation
- IP Multipathing
- Virtual NICs

VLANs

- Standardized as IEEE 802.1Q
- Used to create logical network topology atop the physical topology
- Each VLAN is identified by a numeric tag added to the link-layer packet

VLAN Example

```
# dladm create-vlan -l e1000g0 -v 2 red0
```

```
# dladm show-link
```

LINK	CLASS	MTU	STATE	OVER
e1000g0	phys	1500	up	--
red0	vlan	1500	down	e1000g0

```
# dladm show-vlan
```

LINK	VID	OVER	FLAGS
red0	2	e1000g0	-----

```
# ifconfig red0 plumb
```

```
# ifconfig red0 dhcp start
```

```
# ifconfig red0
```

```
red0:
```

```
flags=201004803<UP, BROADCAST, MULTICAST, DHCP, IPv4, CoS> mtu
```

```
1500 index 4
```

```
inet 10.1.252.174 netmask ffff0000
```

```
ether 0:15:b7:28:91:62
```

Link Aggregation

- Standardized as IEEE 802.3ad
- Aggregate multiple physical links for higher performance and reliability
- Requires switch support
 - Switch is still a single point of failure
- Passive failure detection using link state, active using LACP

Link Aggregation Example

```
# dladm create-aggr -l e1000g0 -l e1000g1 aggr0
# dladm show-link
LINK      CLASS  MTU   STATE  OVER
e1000g0  phys   1500  up     --
e1000g1  phys   1500  up     --
aggr0    aggr   1500  up     e1000g0 e1000g1
# dladm show-aggr
LINK  POLICY  ADDRPOLICY  LACPACTIVITY  LACPTIMER  FLAGS
aggr0 L4      auto        off           short      -----
# ifconfig aggr0 plumb
# ifconfig aggr0 dhcp start
# ifconfig aggr0
aggr0:
flags=201004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4,
CoS> mtu 1500 index 3
    inet 129.147.50.23 netmask fffffe00 broadcast
129.147.51.255
    ether 8:0:27:7a:72:c8
```

IP Multipathing (IPMP)

- Solaris-specific feature
- Aggregate multiple IP interfaces for higher reliability
- No switch support required, recommend multiple switches for better redundancy
- Allows for standby interfaces a la storage hot spares
- Passive failure detection using link state, active using ICMP probes
 - Active requires additional IP addresses
- Use `ifconfig` to create a group and assign interfaces
- Use `ipmpstat` to monitor group

IPMP Example (Active-Active with Probes)

```
# ifconfig ipmp0 ipmp
# ifconfig e1000g0 group ipmp0 -failover up
# ifconfig e1000g1 group ipmp0 -failover up
# ifconfig ipmp0 dhcp start
# ifconfig ipmp0:1 plumb
# ifconfig ipmp0:1 dhcp start
# ifconfig e1000g0 dhcp start
# ifconfig e1000g1 dhcp start
# ifconfig e1000g0
e1000g0:
flags=209044843<UP, BROADCAST, RUNNING, MULTICAST, DHCP, DEPRE
CATED, IPv4, NOFAILOVER, CoS> mtu 1500 index 5
  inet 10.2.1.3 netmask ffff0000 broadcast 10.2.255.255
  groupname ipmp0
```

IPMP Example (continued)

```
# ifconfig ipmp0
```

```
ipmp0:
```

```
flags=8201004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4  
,CoS,IPMP> mtu 1500 index 4
```

```
inet 10.2.1.1 netmask ffff0000 broadcast 10.2.255.255  
groupname ipmp0
```

```
# ifconfig ipmp0:1
```

```
ipmp0:1:
```

```
flags=8201004843<UP,BROADCAST,RUNNING,MULTICAST,DHCP,IPv4  
,CoS,IPMP> mtu 1500 index 4
```

```
inet 10.2.1.2 netmask ffff0000 broadcast 10.2.255.255
```

Virtual NICs

- Carve up a single physical NIC into multiple virtual NICs
 - `dladm create-vnic`
- Assign virtual NICs to Zones or VM's
- “Stub” Ethernet device for machine-internal network without hardware
 - `dladm create-etherstub`
- Flows for network traffic management - `flowadm`
- Link properties for associating CPU resources with network links
 - Integrates network resource management with system resource management
 - `dladm set-linkprop`
- See Project Crossbow for examples,
<http://www.opensolaris.org/os/project/crossbow>

Agenda

- Overview of OpenSolaris Power Features
- ZFS
- Networking
- DTrace

What is DTrace?

- Tool to observe and understand dynamic behavior of entire software system
- Fundamentally different from previous tools
 - Software does not have to be instrumented or explicitly built for monitoring
 - Safe for use on live production systems
 - No overhead until *probes* are enabled, and then only minimal overhead
- Includes programming language for expressing variety of tracing behavior

Who Can Benefit from DTrace?

➤ Administrators

- Understand the system in order to tune and set policies
- Debug live problems

➤ Developers

- Optimize applications

➤ Everyone else

What Can You Do with DTrace?

- Determine which process is modifying a file
- Measure the amount of time spent in each function by an application
- Determine which application is spending the most time on the cpu
- Keystroke logging (not necessarily recommended...)
- In summary, answer pretty much any question you have about anything on your system

Simple Example

Trace all system calls

Probe specification

`syscall:::entry`

Predicate

`/pid != $pid/`

{

`printf("%d %s %s\n", pid, probefunc, execname);`

}

Pre-defined
variables

Action

Run syscall Example

```
# dtrace -q -s syscall.d
718 write firefox-bin
718 lwp_park firefox-bin
718 read firefox-bin
718 ioctl firefox-bin
718 pollsys firefox-bin
718 ioctl firefox-bin
718 pollsys firefox-bin
718 ioctl firefox-bin
718 pollsys firefox-bin
589 ioctl gnome-volume-man
589 pollsys gnome-volume-man
624 portfs clock-applet
624 ioctl clock-applet
624 pollsys clock-applet
585 portfs nautilus
585 ioctl nautilus
...
```

More Complex Example

Determine time each process is spending on CPU

```
sched:::on-cpu
/pid != $pid/
{
  self->t = timestamp;
}
```

Thread-specific variable



```
sched:::off-cpu
/self->t/
{
  @cputime[execname] = sum(timestamp - self->t);
  self->t = 0;
}
```

Associative array



Aggregation



Run CPU-time Example

```
# dtrace -q -s cputime.d
```

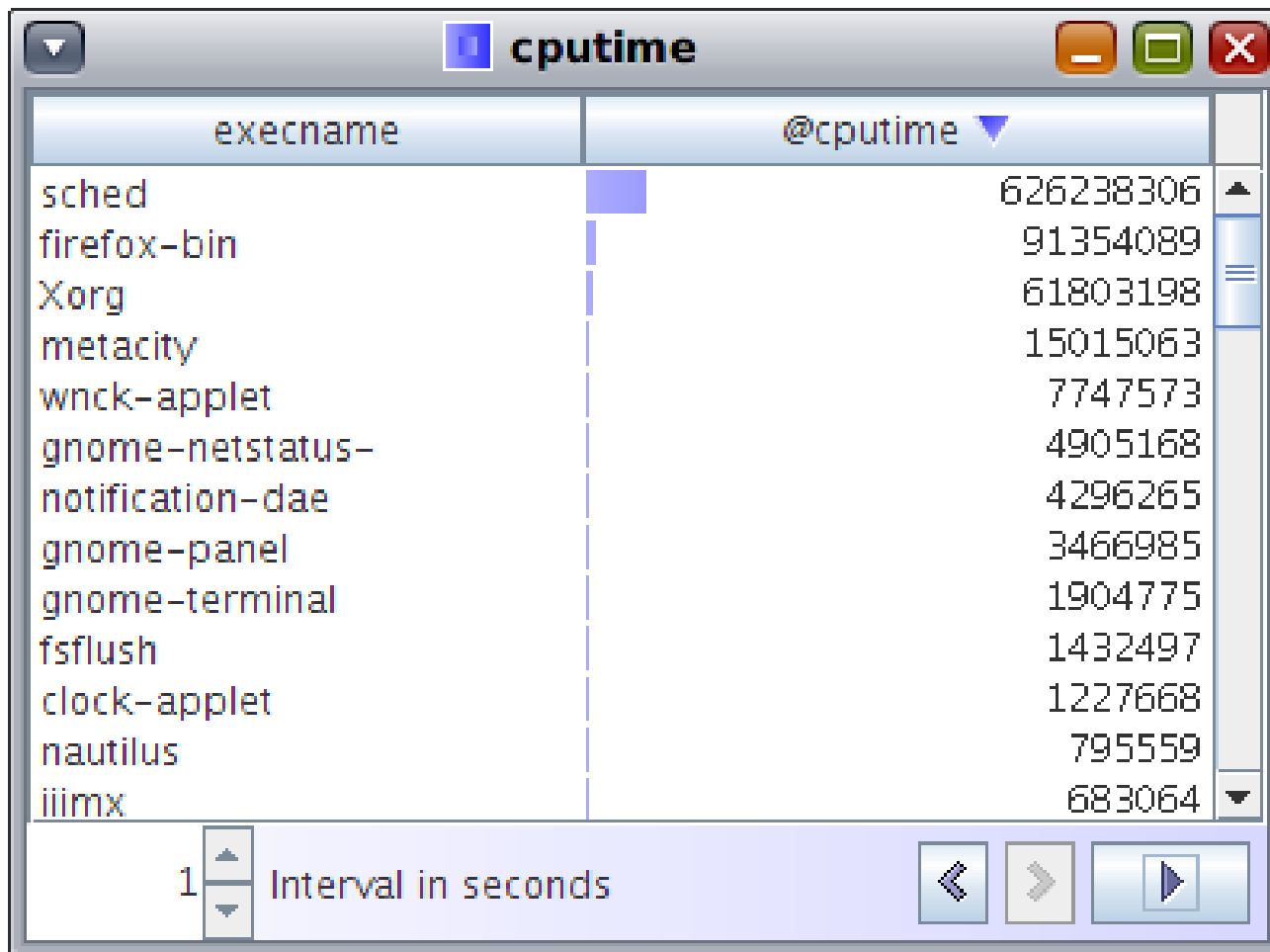
```
^C
```

metacity	98
devfsadm	168109
mdnsd	433063
dhcpcagent	550435
sendmail	767811
dlnmgmtd	1455941
gam_server	2398009
xscreensaver	2578289
gnome-settings-d	3728513
updatemanagernot	4146873
...	
nautilus	7418007
fsflush	7938174
Xorg	9267514
gnome-terminal	12227762
gnome-panel	13701543
gnome-netstatus-	19195334
firefox-bin	69696567
sched	2720718306

The DTrace Toolkit

- Many useful DTrace scripts provided for you
 - dtruss
 - xvmstat
 - opensnoop
 - And many, many more...
- In /opt/DTT/
 - /opt/DTT/Bin has symlinks to all of them

Chime GUI for visualising DTrace output



Chime Installation

- Download package from
<http://opensolaris.org/os/project/dtrace-chime/install/>
- Install Package
 - # gunzip osol0chime-i386-1.4.pkg.gz
 - # pkgadd -d osol0chime-i386-1.4.pkg
- Launch
 - # /opt/OSOL0chime/bin/chime
- Use with DTrace Toolkit built-ins or add your own displays
 - Intuitive and easy to use

DTrace in Action

Chime and the DTT

demo

NetBeans DTrace Plugin

- Install from Tools->Plugins
- Develop and run DTrace scripts from NetBeans
- Includes Chime

NetBeans IDE 6.5

File Edit View Navigate Source Refactor Run Debug Profile Versioning Tools Window Help

Search (Ctrl+I)

Projects Files Services DT... x

ToolKit Chime

Category
TopMostUseful New Script

Scripts
iosnoop
lwpstime.d
execsnoop
opensnoop
iotop
dtruss
hotuser
dvmstat
functime.d
errinfo
rwtop
memleak.d
iopattern

Configuration
DTrace config
Pid
Script Args
Executable
Executable Args

```

/*
 * Print header
 */
dtrace::BEGIN
{
    last_event[""] = 0;

    /* starting values */
    counts = COUNTER;
    secs = INTERVAL;
    disk_r = 0;
    disk_w = 0;

    printf("Tracing... Please wait.\n");
}

/*
 * Check event is being traced
 */
io:genunix::start,
io:genunix::done
{
    /* default is to trace unless filtering, */
    this->ok = FILTER ? 0 : 1;

    /* check each filter, */
    (OPT_device == 1 && DEVICE == args[1]->dev_statname)? this->ok = 1 : 1;
    (OPT_file == 1 && FILENAME == args[2]->fi_pathname)? this->ok = 1 : 1;
}

```

1:1 INS

Output - DTrace - iotop

```

▶ 0 1194 1171 java          cmdk0 102 0 R    1412096
  0  937  934 pkg           cmdk0 102 0 R    2097152
  0 1171 1080 java          cmdk0 102 0 R    4130304
  0   0   0 sched             cmdk0 102 0 R    11886592
  0  2   0 pageout          zfs2  182 2 W    21549056
  0  0   0 sched             cmdk0 102 0 W    28019712

```

□[H[2]2009 Feb 25 15:42:14, load: 4.50, disk_r: 6296 KB, disk_w: 25411 KB

UID	PID	PPID	CMD	DEVICE	MAJ	MIN	D	BYTES
0	1194	1171	java	cmdk0	102	0	R	1412096
0	937	934	pkg	cmdk0	102	0	R	2097152
0	1171	1080	java	cmdk0	102	0	R	4130304
0	0	0	sched	cmdk0	102	0	R	11886592
0	2	0	pageout	zfs2	182	2	W	21549056
0	0	0	sched	cmdk0	102	0	W	28019712

Run

Summary

- OpenSolaris contains powerful features in many areas
 - Software Management, File Systems, Networking, Security, Reliability and Availability, Virtualization, Monitoring and Observability, Software Development, and much more
- ZFS
 - Including Time Slider
- Networking
 - VLANs, VNICs, IPMP, and more
- DTrace
 - Including Chime GUI

For More Information

- Download OpenSolaris and try it out!
 - <http://www.opensolaris.com/>
- Run it in VirtualBox if you don't have hardware available
 - <http://www.virtualbox.org/>
- Join the OpenSolaris community
 - <http://www.opensolaris.org/>
- Books
 - *OpenSolaris Bible* (Solter, Jelinek, and Miner; Wiley, 2009)
 - *Solaris Internals* (McDougall and Mauro; Prentice Hall, 2007)
 - *Solaris Performance and Tools* (McDougall, Mauro, and Gregg; Prentice Hall, 2007)
 - *Solaris Application Programming* (Gove, Prentice Hall, 2008)



Becoming an OpenSolaris Power User

thank you

Dave Miner and Nicholas Solter
{dave.miner,nicholas.solter}@sun.com