

Introduction to the OpenSolaris Operating System

Ben Rockwood
Director of Systems
Joyent

History

- Sept '04: Tonic Project opens “Pilot”, Invite Only
- Jan '05: CDDL License OSI Approved and Made Public, DTrace code release
- June '05: OpenSolaris opens to the public
- Mar '07: Constitution Ratified, OGB Elected

Seriously, It's Open

- Bug Database: bugs.opensolaris.org
- OpenGrok: cvs.opensolaris.org
- Nightly Builds: genunix.org/mirror/
- Hg: hg.opensolaris.org/hg/onnv/onnv-gate
- etc....

Governing Board

- Consists of 7 persons, elected by the community, not appointed by Sun
 - James Carlson
 - Alan Coopersmith
 - Casper Dik
 - Glynn Foster
 - Stephen Lau
 - Rich Teer
 - Keith Wesolowski

Contribution

- Contributors are paired with a sponsor
- Code is reviewed and processed by Sun's existing standards (today)
- No degradation in Solaris quality

Nifty... So what?

OpenSolaris Features

- Service Management Facility (SMF)
- DTrace: Dynamic Tracing
- Zones: Light-weight OS Virtualization
- ZFS: Worlds most advanced filesystem
- Resource Controls
- Fault Management Architecture (FMA)
- GRUB
- iSCSI (Target & Initiator)
- And much, much more...

Slow-aris is dead.

Service Management Facility (SMF)

Service Management Facility (SMF)

- Init scripts suck (still available as legacy)
- Services defined in an XML Manifest
- Services are intelligent, aware of dependancies and service state
- Dependancy awareness allows for parallel execution
- Services restart if processes fail

Example Manifest

```
<?xml version='1.0'?>
<!DOCTYPE service_bundle SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type='manifest' name='export'>

  <service name='network/svnserve' type='service' version='1'>

    <create_default_instance enabled='true'/>
    <single_instance/>

    <dependency name='loopback' grouping='require_all' restart_on='error' type='service'>
      <service_fmri value='svc:/network/loopback:default'/>
    </dependency>
    <dependency name='physical' grouping='optional_all' restart_on='error' type='service'>
      <service_fmri value='svc:/network/physical:default'/>
    </dependency>

    <exec_method name='start' type='method' exec='/opt/csw/bin/svnserve -d' timeout_seconds='60'>
      <method_context/>
    </exec_method>
    <exec_method name='stop' type='method' exec=':kill' timeout_seconds='60'>
      <method_context/>
    </exec_method>
    <exec_method name='refresh' type='method' exec=':kill && /opt/csw/bin/svnserve -d' timeout_seconds='60'>
      <method_context/>
    </exec_method>

    <stability value='Unstable'/>

    <template>
      <common_name>
        <loctext xml:lang='C'>Subversion Standalone Server</loctext>
      </common_name>
      <documentation>
        <doc_link name='Version Control with Subversion' uri='http://svnbook.red-bean.com/'/>
      </documentation>
    </template>

  </service>
</service_bundle>
```

SMF is aware...

```
$ svcs ssh
```

```
STATE      STIME    FMRI
online     May_05   svc:/network/ssh:default
```

```
$ svcs -p ssh
```

```
STATE      STIME    FMRI
online     May_05   svc:/network/ssh:default
           May_05   100431 sshd
```

```
$ svcs -l ssh
```

```
fmri       svc:/network/ssh:default
name       SSH server
enabled    true
state      online
next_state none
state_time Sat May 05 03:41:39 2007
logfile    /var/svc/log/network-ssh:default.log
restarter  svc:/system/svc/restarter:default
contract_id 60
dependency require_all/none svc:/system/filesystem/local (online)
dependency optional_all/none svc:/system/filesystem/autofs (disabled)
dependency require_all/none svc:/network/loopback (online)
dependency require_all/none svc:/network/physical (online)
dependency require_all/none svc:/system/cryptosvc (online)
dependency require_all/none svc:/system/utmp (online)
dependency require_all/restart file:///localhost/etc/ssh/sshd_config (online)
```

SMF is responsible...

```
$ svcs -p sshd  
svcs: Pattern 'sshd' doesn't match any instances
```

```
STATE      STIME    FMRI
```

```
$ svcs -p ssh
```

```
STATE      STIME    FMRI
```

```
online     May_05   svc:/network/ssh:default  
           May_05   100431 sshd
```

```
$ kill -9 100431
```

```
$ svcs -p ssh
```

```
STATE      STIME    FMRI
```

```
online     4:17:53  svc:/network/ssh:default  
           4:17:53  282175 sshd
```

```
$ tail /var/svc/log/network-ssh:default.log
```

```
...
```

```
[ May 6 04:17:51 Stopping because all processes in service exited. ]
```

```
[ May 6 04:17:52 Executing stop method (:kill) ]
```

```
[ May 6 04:17:53 Executing start method ("/lib/svc/method/sshd start") ]
```

```
[ May 6 04:17:53 Method "start" exited with status 0 ]
```

What isn't running and why?

```
$ svcs -vx
```

```
svc:/network/nfs/nlockmgr:default (NFS lock manager)
```

```
State: disabled since Sat May 05 03:40:49 2007
```

```
Reason: Disabled by an administrator.
```

```
See: http://sun.com/msg/SMF-8000-05
```

```
See: man -M /usr/share/man -s 1M lockd
```

```
Impact: 1 dependent service is not running:
```

```
svc:/network/nfs/client:default
```

Pamper yourself

- Restart_on dependancies can restart a service if a dependancy changes state
- exclude_all dependancy can start a service if all its dependancies have stopped
- Meta-data and Help information stored with service
- Application configuration can be stored within SMF.

ZFS

ZFS

- 128-bit filesystem
- End-to-end checksum integrity (sha256, fletcher2, fletcher4)
- Object based storage with plugin architecture
- File-to-disk management; ie: Volume Manager Included!
- RAID0, RAID1, RAIDZ, RAIDZ2 supported

ZFS Toys (Cont)

- Integrated Compression (lzjb, gzip)
- NFS and iSCSI sharing with a single command
- NFSv4 style ACL's
- Unlimited snapshots
- Snapshot clone, promote, rollback
- ... and more!

Thinking ZFS

- ZFS changes the meaning of “file system”
- Disks are grouped into “Zpools” (volumes)
- Datasets are created on the Zpool (filesystems)
- Datasets have properties that can be set or inherited

Simplicity Defined

```
$ zpool create Users mirror /vdev/disk01 /vdev/disk02
```

```
$ zpool status Users
```

```
pool: Users
```

```
state: ONLINE
```

```
scrub: none requested
```

```
config:
```

NAME	STATE	READ	WRITE	CKSUM
Users	ONLINE	0	0	0
mirror	ONLINE	0	0	0
/vdev/disk01	ONLINE	0	0	0
/vdev/disk02	ONLINE	0	0	0

```
errors: No known data errors
```

```
$ df -h /Users
```

Filesystem	size	used	avail	capacity	Mounted on
Users	984M	24K	984M	1%	/Users

```
$ zfs create Users/benr
```

```
$ zfs create Users/mark
```

```
$ df -h | grep Users
```

Users	984M	27K	984M	1%	/Users
Users/benr	984M	24K	984M	1%	/Users/benr
Users/mark	984M	24K	984M	1%	/Users/mark

ZPool's

```
# zpool status
pool: pool
state: ONLINE
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
pool	ONLINE	0	0	0
raidz2	ONLINE	0	0	0
c2d0	ONLINE	0	0	0
c3d0	ONLINE	0	0	0
c4d0	ONLINE	0	0	0
c5d0	ONLINE	0	0	0
c6d0	ONLINE	0	0	0
c7d0	ONLINE	0	0	0
c8d0	ONLINE	0	0	0

errors: No known data errors

```
$ zpool status -v
pool: local
state: ONLINE
status: One or more devices has experienced an error resulting in data
corruption. Applications may be affected.
action: Restore the file in question if possible. Otherwise restore the
entire pool from backup.
see: http://www.sun.com/msg/ZFS-8000-8A
scrub: none requested
config:
```

NAME	STATE	READ	WRITE	CKSUM
local	ONLINE	0	0	0
cl d0s3	ONLINE	0	0	0

errors: Permanent errors have been detected in the following files:

local/benr:<0x50a81>

Dataset Properties

- Quota's
- Reservations (pre-allocated disk space)
- Compression
- Mount point
- NFS/iSCSI Sharing
- Snapshot Visibility
- atime, setuid, readonly, etc.

Control With Ease

```
$ zfs get all Users/benr
NAME      PROPERTY      VALUE      SOURCE
Users/benr type          filesystem  -
Users/benr creation     Sun May 6 4:13 2007 -
Users/benr used        24.5K      -
Users/benr available   984M      -
Users/benr referenced  24.5K     -
Users/benr compressratio 1.00x     -
Users/benr mounted     yes       -
Users/benr quota       none      default
Users/benr reservation none      default
Users/benr recordsize  128K     default
Users/benr mountpoint  /Users/benr default
Users/benr sharenfs    off       default
Users/benr shareiscsi  off       default
Users/benr checksum    on        default
Users/benr compression off       default
Users/benr atime       on        default
Users/benr devices    on        default
Users/benr exec        on        default
Users/benr setuid      on        default
Users/benr readonly   off       default
Users/benr zoned      off       default
Users/benr snapdir    hidden    default
Users/benr aclmode    groupmask default
Users/benr aclinherit secure    default
Users/benr canmount   on        default
Users/benr xattr     on        default
```

```
$ zfs set quota=10g Users/benr
```

```
$ df -h /Users/benr
```

```
Filesystem      size  used  avail capacity Mounted on
Users/benr      10G   24K   984M    1%    /Users/benr
```

ZFS Snapshots

```
$ zfs snapshot Users/benr@050507
```

```
$ zfs list Users/benr
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
Users/benr	24.5K	984M	24.5K	/Users/benr

```
$ zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
Users	146K	984M	27.5K	/Users
Users/benr	24.5K	984M	24.5K	/Users/benr
Users/benr@050507	0	-	24.5K	-

```
...
```

```
$ cd /Users/benr/.zfs/snapshot/050507/
```

```
$ zfs clone Users/benr@050507 Users/tamr
```

```
$ zfs list
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
Users	184K	984M	28.5K	/Users
Users/benr	46K	984M	24.5K	/Users/benr
Users/benr@050507	21.5K	-	24.5K	-
Users/tamr	0	984M	24.5K	/Users/tamr

```
...
```

Fire Your Storage Guy

- ZFS makes enterprise grade storage toys available to everyone
- Works on iPod and EMC Symmetrix
- Remove complexity to focus on requirements using snapshots, simplified administration, and fine-grained control

Thumper & ZFS

- 48 SATA disks
- 24TB Raw Storage
- RAIDZ2 with 3 Spares, 18TB Raw (w/o compression)
- Try using LinuxRAID or LVM on one...

Solaris Zones

Zones

- Light-weight OS Virtualization
- Only one kernel
- Virtual Networking within Zone
- Can be “sparse” (loopback global filesystems) or “whole root” (independent filesystems)
- Completely isolated from other zones
- 5 second reboots

Easy to Create

```
$ zonecfg -z javadev01
javadev01: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:javadev01> create
zonecfg:javadev01> info
zonename: javadev01
zonename: javadev01
zonepath:
brand: native
autoboot: false
bootargs:
pool:
limitpriv:
scheduling-class:
inherit-pkg-dir:
  dir: /lib
inherit-pkg-dir:
  dir: /platform
inherit-pkg-dir:
  dir: /sbin
inherit-pkg-dir:
  dir: /usr
zonecfg:javadev01> set zonepath=/Users/javadev01
zonecfg:javadev01> commit
zonecfg:javadev01> exit
$ zoneadm list -vc
ID NAME          STATUS    PATH                BRAND
0 global        running  /                   native
- javadev01    configured /Users/javadev01   native
```

Resource Controls

Defining Workloads

- Solaris resource controls (rctl) can be applied to “workloads” such as:
 - Per User or Group
 - Per Task (newtask)
 - Per Process
 - Per Zone

Soft Limits

- Memory Capping (rcapd)
- Swap Capping (max-swap)
- CPU Capping (cpu-cap)
- Thread Limits (max-lwps)
- Locked Memory (max-locked-memory)
- SHM/Semaphore Limits

Hard Limits

- Dynamic Resource Pools can be used to “hard” partition CPU resources (today) and Memory (future)
- CPU resources are allocated into “psets” and projects or Zones bind to these pools

Fair Share Scheduler

- The FSS schedule ensures fairness (go figure)
- Workloads are defined CPU Shares and prioritized based on available shares
- Workload resources are adjusted over time to ensure everyone gets what they are due
- Originally created for ISP's
- Works hand-in-hand with other resource controls to give fine grain control

Soft Limits & Zones

```
$ zoncfg -z myzone
zoncfg:myzone> create
zoncfg:myzone> set zonpath=/zones/myzone
zoncfg:myzone> set scheduling-class=FSS
zoncfg:myzone> set cpu-shares=10
zoncfg:myzone> set max-lwps=5000

zoncfg:myzone> add capped-cpu
zoncfg:myzone:capped-cpu> set ncpus=1
zoncfg:myzone:capped-cpu> end

zoncfg:myzone> add capped-memory
zoncfg:myzone:capped-memory> set physical=1g
zoncfg:myzone:capped-memory> set swap=1g
zoncfg:myzone:capped-memory> end

zoncfg:myzone> add net
zoncfg:myzone:net> set address=8.12.33.101
zoncfg:myzone:net> set physical=e1000g0
zoncfg:myzone:net> end
```

A Layered Approach

- All resource controls are intended to be mixed and matched together to meet requirements
- Designed with 1 CPU up to beyond 256 CPU in mind
- Example: Partition CPU's with pset's, sub-allocate with CPU Capping, and balance usage with FSS

Zones + Resource Control = Container

Pulling It All Together

Web Deployments

- Problem: Need multiple environments (production, staging, development) in a limited environment
- Solution: Multiple Zones on each system using CPU/Memory Capping and FSS to ensure they don't interfere with each other

Web Deployments

- Problem: Application deployments don't always go according to plan; need a fast backout method
- Solution: ZFS Snapshots. Snapshot before deploying, if something goes wrong just rollback and restart

Web Deployments

- Problem: Code issue may require immediate fix but further analysis
- Solution: Snapshot the ZFS filesystem for the Zone, deploy fix, then clone the zone snapshot and bring it up for further analysis

Web Deployments

- Problem: Database restart may impact application stack
- Solution: SMF dependancies; in the event of a database restart, restart every component in the stack to restore proper state.

Conclusions

- Solaris is known for stability
- Solaris is faster than ever and has world class support for X86 and leads the pack for 64bit applications
- Ground breaking innovation in Solaris 10 & OpenSolaris are compelling, practical, and most importantly, easy to use
- Solaris is now Open Source with a thriving community

Resources

- Solaris 10: FREE! sun.com/solaris
- OpenSolaris: opensolaris.org
- Solaris Express: Developer Edition
 - Early Access Release with Support (\$250 per year or \$49 per incident)
- Solaris Express: Community Edition
 - Cutting Edge Release without Support

Thank You

Ben Rockwood
Director of Systems
Joyent

<http://cuddletech.com>
<http://joyent.com>