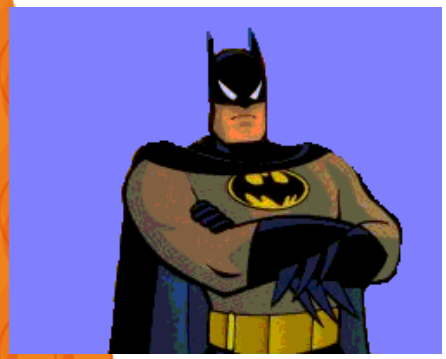
SAS 9 on Solaris 10 – Calling SuperHeroes to Fight SuperVillians



Maureen Chew
maureen.chew@sun.com
Staff Engineer
Sun Miicrosystems
SUGI 31, Paper 222-31
San Francisco, 28Mar06

Agenda

- Standard Solaris Tools
- Solaris Containers
- Service Management Framework
- DTrace
- Bonus: Lasso of Truth

Standard Solaris Tools

- *Get me help – STAT!*
- Most take a time interval (ie: 5)
- prstat (Process)
- mpstat (processor)
- vmstat (virtual memory)
- netstat (network)
- iostat (i/o)

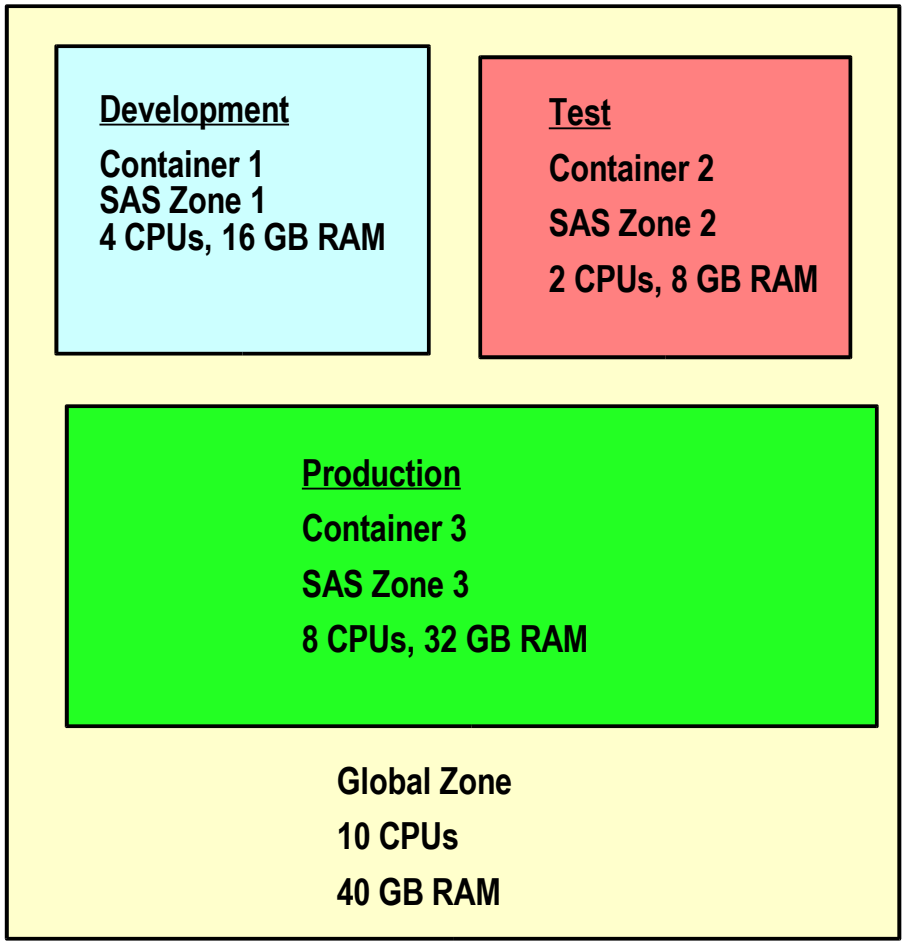
Solaris Containers

Rationale

- "web" and "containments
- Consolidate trusted and non trusted apps
- Dev, test, prod
- Service developers and users who need root

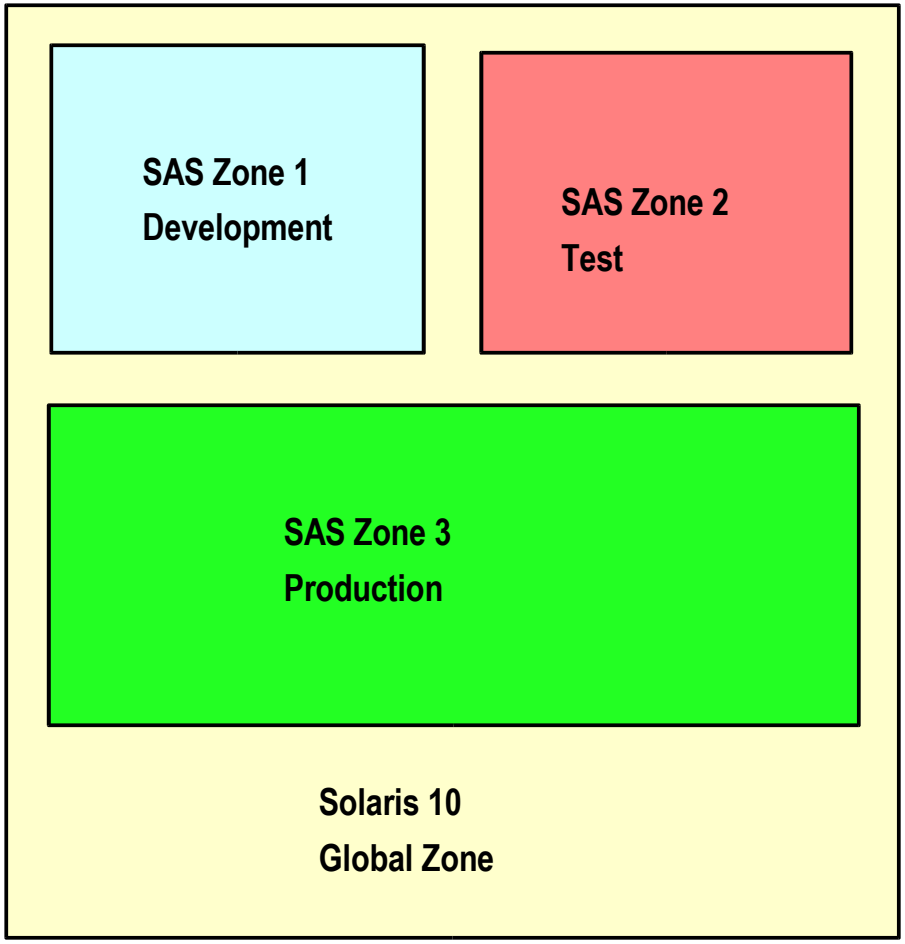
Containers vs. Zones

- Container = Zone + resource management
- Dynamically move CPU & memory resources between zones



Solaris Containers

- Virtualization of the OS as Independent Zones
- Appears as a different instance of Solaris
- Each container can be rebooted w/o affecting other containers
- Each can bring up the Metadata server of port 8561



Predictive, Self-Healing

- Fault Management Architecture
  - > Automatically and silently detects and diagnoses underlying problem
- Service Management Framework (SMF)
  - > Standardized control mechanisms for application services

Service Management Framework

- New and improved way to start/stop services
- Major component of fault management architecture
- Replaces init.d startup scripts (but legacy support remains)
- Can automatically restart a failed service

Service Management Framework

- Create a service manifest file, and shell script file to define the start, stop, restart methods for the service
- Validate and import the service manifest using **svccfg(**1M)
- Enable or start the service using **svcadm**(1M)
- Verify service is running using **svcs**(1)

DTrace – Batman Utility Belt

- Batarangs, Bat cuffs, communications device, 1st aid kit, forensic kit, gas mask, grappling hook, infrared goggles, kryptonite ring, laser torch, line gun, lock picks, camera, smoke goggles, mini toolkit, money, rebreather, Batmobile remote control, grenades, thermite...

DTrace does it all and MORE

DTrace

- Dynamic Tracing framework
- Used to observe, debug, tune
- Enables one to ask the system arbitrary questions about the behavior of the system or applications
- Scripts are written in 'D' - (similar to a mix of C/perl/awk)
- DTrace expertise == resume standout

DTrace

- Many useful examples included in /usr/demo/dtrace
- Best starter - Google dtrace, ***I'm feeling lucky***
- Sample scripts – show me....
  - All failed open(2) system calls and their failure code
  - All signals by sender and recipient
  - Count calls into libc(3)
  - write(2) times by process name

DTrace - Example

- Show total I/O by process name, along with write stats

```
bash-2.05b$ cat iototal.d
#pragma D option quiet

io:::start
{
   @[args[1]->dev_statname, execname,pid ] = sum(args[0]->b_bcount);
}

syscall::write:entry
{
   @writes[probefunc] = count();
   @sizes["write Sizes"] = quantize(arg2);
}

END
{
   printf("%10s %20s %10s %15s\n", "DEVICE", "APP", "PID", "BYTES");
   printa("%10s %20s %10d %15@d\n", @);
}
```

DTrace – example results

```
bash-2.05b$ dtrace -s iototal.d

    DEVICE                 APP        PID           BYTES

       sd1               fsflush          3            5120
       sd0                    vi      21042           11776
       sd0              cpudiagd        493           24576
       sd1                 sched          0           54784
       sd0               fsflush          3          369152
       sd0                 sched          0          698368
       ssd102              sched          0         1370624
       ssd102            fsflush          3         1936896
      sd1                    sas      21054       811515904
       sd0                   sas      21054       811735040
       ssd102                sas      21054       880518144


  write                                                  99095

  write Sizes

          value  ------------- Distribution ------------- count

              0 |                                         0
              1 |                                         38
              2 |                                         3
              4 |                                         4
              8 |                                         10
             16 |                                         8
             32 |                                         0
             64 |                                         4
            128 |                                         0
            256 |                                         0
            512 |                                         11
           1024 |                                         0
           2048 |                                         0
           4096 |                                         1
           8192 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 99016
          16384 |                                         0
```

99K writes issued, all length of 8K

Solaris 10

- Many other reasons to upgrade
  - > Performance
  - > TCP/IP stack rewrite
  - > Fault Management Architecture
  - > Security enhancements – least privileges, trusted Solaris
  - > Enterprise grade OS
  - > COST is FREE RTU
  - > Open Source

Bonus – Lasso of Truth

SAS 9 Install & Deploy Tips and Tricks

- Skill set survey
  - general Solaris admin basics
  - DBA for SAS/ACCESS
  - App Server/Web Container
  - Xythos WebDav
  - Platform LSF for Grid
  - SPDS
  - Careful planning, patience
  - in SAS we truss(1M)

Bonus: Lasso of Truth

- Very large, I/O intensive jobs – use larger BUFSIZE(SAS option-8K -> 64K) UBUFSIZE=128K for data set views
  - >15 hr vs 13 hr : dmreg, neural, decision tree (views)
- Set CPUCOUNT (especially on large systems)
- US IV+ performance vs. US IV rocks

Bonus: Lasso of Truth

- Metadata repository should be in high performance dir
- Installation defaults or state stored in 3 places
  - $HOME/vpd.properties, .sasprefs, Solaris package registry
- Can't share “sas” user via NFS home directory if supporting multiple installs or havoc can ensue
- CD library – execute permission on all scripts
- Uninstall is not automated
  - Don't 'rm -rf' else stale package info will remain

```
# pwd
/d0/apps/sas/sas9-1201/SASManagementConsole/9.1/_uninst
# ls
UninstSASMC     uninstall.dat  uninstall.jar
```

Bonus: Lasso of Truth

- Java Client
  - > Data Integration Studio, Cube Studio, OLAP Monitor, Map Studio, etc...
  - > It's Java after all


```
root@ctclava # pwd
/net/ctcsun8/sharefs1/SAS/913-SE21/client3cd/databuilder

root@ctclava # ls
media.inf     setup.exe     setup.ini     setup.jar     setup_s64.sh
```

- Solaris 10

Bonus: Lasso of Truth

- High end SAS ETL results on Sun Fire E25K, Sun StorEdge 3510 with high performance shared QFS file system
- SAS Data Integration Studio
  - Bulk load : 4.22 TB input, < 80 min
    - 3.22 TB sustained throughput
  - World record...
- And last but not least ...

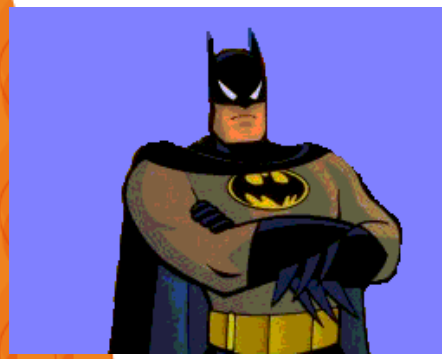
Bonus: Lasso of Truth

- Coming soon to a CD near you ...

SAS 9.1.3 SP4 on Solaris 10 / Opteron*

*And other 64bit x86 architectures

SAS 9 on Solaris 10 – Calling SuperHeroes to Fight SuperVillians



Maureen Chew
maureen.chew@sun.com
Staff Engineer
Sun Miicrosystems
SUGI 31, Paper 222-31
San Francisco, 28Mar06