Joyent

smart computing

SmartOS Additions/ Modifications for Illumos

Max Bruning

Title

Phone Number

Email

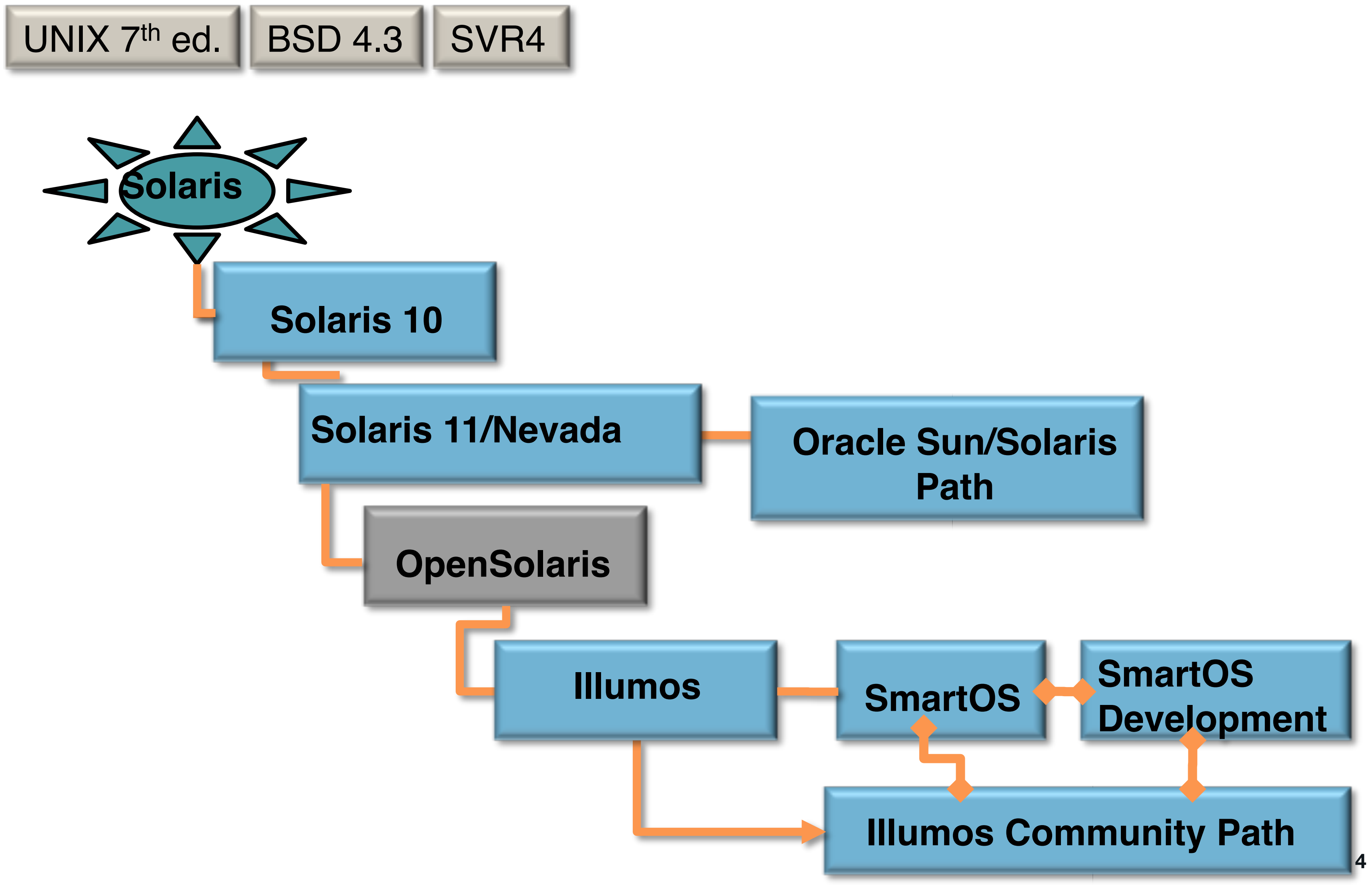
Topics

- Joyent and SmartOS Introduction
- Zones
- ZFS
- DTrace/mdb
- KVM

Joyent Introduction

- **Joyent uses SmartOS for 2 products**
  - SmartDataCenter (SDC)
    - Orchestration software that allows users to build their own public/private clouds
  - Joyent Public Cloud
    - Joyent-provided data centers running SDC that allow users to provision “machines”
    - “Machines” can be running SmartOS, Linux, or Windows
  - “Machines” run in zones
    - Minimal virtualization overhead for native zones

SmartOS Lineage



Zones

- **Joyent uses zones in SmartOS for:**
  - Provisioning machines in the “cloud”
    - “Smart Machines” in a Joyent branded zone
    - “Virtual Machines” in a kvm branded zone
  - The “usual” reasons
    - Isolation
    - Resource control
    - Privileges
  - Many of Joyent’s SDC services run in their own zones

Scalable Zone Memory Capping

- **Replaced rcapd**
  - rcapd is a single-threaded process to handle all zones on the system
  - zone’s memory cap is not an rctl
  - Monitoring only by rcapstat
- **SmartOS uses a thread in zoneadmd for memory capping**
  - 1 zoneadmd process per zone
  - Cap is implemented as an rctl
  - Statistics are maintained as kstats
  - rcapd service is disabled by default
    - Still needed for memory capping projects
- **Per-zone load averages**

Memory Capping Commands

- **New rctl - zone.max-physical-memory**

```
# prctl -n zone.max-physical-memory -i zone 87b5509f-7c02-47e2-8b31-b26eadfc0746

zone: 895: 87b5509f-7c02-47e2-8b31-b26eadfc0746
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
zone.max-physical-memory
        usage            142MB
        privileged      1.00GB      -   deny                                 -
        system          16.0EB    max   deny                                 -
```

New kstat for Memory Capping

```
# kstat -n 87b5509f-7c02-47e2-8b31-b26ead -c zone_memory_cap
module: memory_cap                        instance: 895
name:   87b5509f-7c02-47e2-8b31-b26ead  class:    zone_memory_cap
  anon_alloc_fail                        0
  anonpgin                               0
  crtime                                 8748783.89571479
  execpgin                               0
  fspgin                                 51
  nover                                  0
  pagedout                               0
  pgpgin                                 51
  physcap                                1073741824
  rss                                    256745472
  snaptime                               8757319.36290128
  swap                                   483696640
  swapcap                                2147483648
  zonename                               87b5509f-7c02-47e2-8b31-b26eadfc0746

# kstat -n physicalmem_zone_895
module: caps                              instance: 895
name:   physicalmem_zone_895              class:    zone_caps
  crtime                                 8748783.89569346
  snaptime                               8757751.8876096
  usage                                  256749568
  value                                  1073741824
  zonename                               87b5509f-7c02-47e2-8b31-b26eadfc0746
```

zonememstat

- **Replacement for rcapstat**

```
# zonememstat
                                ZONE      RSS      CAP  NOVER     POUT
                              global    381MB        -      -        -
4a75b605-9613-4c5b-9834-b8d3073d0021     33MB    512MB      0      0MB
0c544c75-922a-4f32-895b-01517fd05e53     30MB    512MB      0      0MB
b860b3fd-813d-4c20-8dfa-7a74e50fa461     55MB    128MB      0      0MB
2da3ff7e-a5f9-4353-aa83-9b3e33c15575    163MB   1152MB      0      0MB
0e4b11d0-e2bc-4fc2-bdb9-bb472fb28223    202MB    256MB      1     67MB
7662d71d-e398-4675-b547-f9a795c77c64     53MB    128MB      0      0MB
ab72fc58-6237-41fb-8d7b-c99e94cc4d76   2082MB   3072MB      0      0MB
857030a1-de54-4977-96d3-557de3e08fc3    113MB   4096MB      0      0MB
0508071f-3379-46e3-82c5-8174fc1ce43d     42MB    128MB      0      0MB
b40464a3-1a9a-4c0e-9c2e-9788a37c0084    284MB   1280MB      0      0MB
4ec7a215-e314-46ff-8429-d4fe95230760    215MB    256MB      0      0MB
1697ac4c-f2b9-4ad0-97cf-d3def655627c     41MB    512MB      0      0MB
87b5509f-7c02-47e2-8b31-b26eadfc0746    244MB   1024MB      0      0MB
```

RSS Calculation

- **rcapd and prstat used (undocumented) getvmusage() system call to determine RSS**
- **For large processes, this causes noticeable latency as the process is stopped during calculation**
- **zonememstat, the memory capping thread in zoneadmd, and prstat now use an approximation**
    - Avoids calling getvmusage()
    - May over-estimate RSS (shared pages may be counted multiple times)
    - If approximation shows over memory cap, revert to getvmusage()

svcs/svcadm Command Enhancements

```
# svcs -z 87b5509f-7c02-47e2-8b31-b26eadfc0746
STATE          STIME    FMRI
legacy_run      8:02:38 lrc:/etc/rc2_d/S20sysetup
legacy_run      8:02:38 lrc:/etc/rc2_d/S72autoinstall
legacy_run      8:02:38 lrc:/etc/rc2_d/S89PRESERVE
legacy_run      8:02:38 lrc:/etc/rc2_d/S98deallocate
online          8:02:36 svc:/system/svc/restarter:default
online          8:02:36 svc:/system/early-manifest-import:default
...
```

```
# svcs -Z -xv
#  (shows all zones)
# svcs -Z -L svc:/network/ssh:default
/var/svc/log/network-ssh:default.log
/zones/4a75b605-9613-4c5b-9834-b8d3073d0021/root/var/svc/log/network-ssh:default.log
/zones/0c544c75-922a-4f32-895b-01517fd05e53/root/var/svc/log/network-ssh:default.log
...
```

```
# svcadm -z 87b5509f-7c02-47e2-8b31-b26eadfc0746 enable rlogin
```

```
# svcprop -z 87b5509f-7c02-47e2-8b31-b26eadfc0746 rlogin
firewall_config/apply_to astring ""
firewall_config/exceptions astring ""
firewall_config/policy astring use_global
firewall_config/value_authorization astring
solaris.smf.value.firewall.config
inetd/name astring login
...
```

Additional Zone Enhancements

Joyent

- **Joyent Branded Zone**
  - Uses a “sparse root” model
  - /usr is mounted read-only
- **KVM Branded Zone**
  - Runs a single process - qemu
  - Limited privileges
- **Many bug fixes**
  - Specifically for zone shutdown hangs
- **Zone-aware `wall(1M)`**

CPU Bursting

- **Mechanism that allows a zone to go over a *baseline* cpu usage.**
- **Controlled by:**
  - rctls
    - `zone.cpu-burst-time`
      - **Number of seconds a zone can run at zone.cpu-cap before capped at cpu-baseline value. Once capped at baseline, wait cpu-burst-time seconds before restoring cpu-cap. 0 means burst indefinitely. "Spiky" zones remember time accrued.**
    - `zone.cpu-baseline`
      - **"Normal" level of cpu usage. Over this is "bursting". May be set, but by default is calculated base on zone memory and amount of "provisionable memory" (~83% of total memory).**
    - `zone.cpu-cap`
      - **Hard cap on cpu usage. 100% of 1 cpu value is 100.**

CPU Bursting Example

```
# prctl -i zone 87b5509f-7c02-47e2-8b31-b26eadfc0746
...
zone.cpu-burst-time
        usage                0
        system          2.15G     max   none                                  -
zone.cpu-baseline
        usage               40
        privileged          40       -   none                                  -
        system          4.29G     max   none                                  -
zone.cpu-cap
        usage                0
        privileged         350       -   deny                                  -
        system          4.29G     inf   deny                                  -
zone.cpu-shares
        usage           1.02K
        privileged      1.02K       -   none                                  -
        system          65.5K     max   none                                  -
```

CPU Bursting kstats

```
# prctl -t privileged -n zone.cpu-burst-time -v 10 -i zone 87b5509f-7c02-47e2-8b31-b26eadfc0746
# prctl -i zone 87b5509f-7c02-47e2-8b31-b26eadfc0746
zone: 895: 87b5509f-7c02-47e2-8b31-b26eadfc0746
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
...
zone.cpu-burst-time
        usage              10
        privileged         10       -   none                                -
        system          2.15G     max   none                                -
...
```

Run some compute bound stuff for a bit, then...

```
# kstat -n cpucaps_zone_895
module: caps                            instance: 895
name:   cpucaps_zone_895                class:    zone_caps
      above_base_sec                  243038
      above_sec                       23
      baseline                        40
      below_sec                       691880
      burst_limit_sec                 10
      bursting_sec                    3
      crtime                          8748783.89582975
      effective                       40
      maxusage                        356
      nwait                           5
      snaptime                        9440687.28752677
      usage                           40
      value                           350
      zonename                        87b5509f-7c02-47e2-8b31-b26eadfc0746
```

ZFS Additions

- Per-zone ZFS I/O throttle
- ZFS Dump to a Raid-Z Pool
- vfstat(1M)
- ziostat(1M)

Per-Zone ZFS I/O Throttle

Joyent

- **Problem**
  - One zone issuing many I/O requests can impact other zones
  - Flushing a ZFS transaction group (TXG) can cause all zones I/O to be impacted
- **Solution is to implement per-zone ZFS I/O throttles**
- **Reference**
  - http://dtrace.org/blogs/wdp/2011/03/our-zfs-io-throttle/

ZFS I/O Throttle Requirements

- **Ensure consistent and predictable I/O latency across zones**
- **Sequential vs. Random I/O patterns have very different characteristics (orders of magnitude)**
  - Don’t track IOPS or throughput
- **If no contention, a zone should be able to use the entire disk bandwidth**

ZFS I/O Throttle Components

Joyent

- **Per-Zone I/O utilization metric**
  - (# read syscalls)*(average read latency) + (# write syscalls)*(average write latency)
- **Mechanism to throttle zone I/O requests when zone use is over its fair share**
- **Throttle compares I/O utilization across all zones**
  - If a zone uses more than an “average” I/O utilitization, read/write syscalls from that zone are delayed by up to 100 microseconds
- **Each zone has a (relative) ZFS I/O *priority*, set via prctl or zonecfg**

ZFS I/O Throttle Example

```
# prctl -n zone.zfs-io-priority -i zone aec68fba-2906-47d7-bf12-c74afc54d1e4
zone: 34: aec68fba-2906-47d7-bf12-c74afc54d1e4
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
zone.zfs-io-priority
        usage               1
        privileged          1       -   none                                -
        system          1.02K     max   none                                -
# prctl -n zone.zfs-io-priority -i zone b204cbe5-476a-4d5d-96bf-1bfb86546bed
zone: 35: b204cbe5-476a-4d5d-96bf-1bfb86546bed
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
zone.zfs-io-priority
        usage               1
        privileged          1       -   none                                -
        system          1.02K     max   none                                -
# ziostat -Z 5 | egrep 'zone|aec68fba|b204cbe5'
...
    r/s   kr/s   actv wsvc_t asvc_t  %b zone
  524.1 66580.2    0.9    0.0    1.7   87 aec68fba (36)
  547.7 69956.7    0.9    0.0    1.6   86 b204cbe5 (37)
# vfsstat -Z 5 | egrep 'b204cbe5|aec68fba|zone'
...
  r/s   w/s  kr/s  kw/s ractv wactv read_t writ_t  %r  %w   d/s  del_t zone
9383.4   0.0 75067.5   0.0   1.0   0.0    0.1    0.0  98   0 591.1   99.7 aec68fba
(36)
8167.4   0.0 65324.7   0.0   1.0   0.0    0.1    0.0  98   0 513.1   97.4 b204cbe5
(37)
#
```

ZFS I/O Throttle Example (Continued)

```
# prctl -n zone.zfs-io-priority -t privileged -r -v 10 -i zone
b204cbe5-476a-4d5d-96bf-1bfb86546bed
# prctl -n zone.zfs-io-priority -i zone b204cbe5-476a-4d5d-96bf-1bfb86546bed
zone: 37: b204cbe5-476a-4d5d-96bf-1bfb86546bed
NAME    PRIVILEGE       VALUE    FLAG   ACTION                       RECIPIENT
zone.zfs-io-priority
        usage              10
        privileged         10       -   none                                 -
        system          1.02K     max   none                                 -
# ziostat -Z 5 | egrep 'zone|aec68fba|b204cbe5'
...
    r/s   kr/s   actv wsvc_t asvc_t  %b zone
  486.6 62280.3    0.9    0.0    1.8  86 aec68fba (36)
  604.3 77351.5    0.9    0.0    1.5  89 b204cbe5 (37)
# vfsstat -Z 5 | egrep 'b204cbe5|aec68fba|zone'
...
  r/s   w/s  kr/s   kw/s ractv wactv read_t writ_t  %r  %w   d/s  del_t zone
13215.5   0.0 105724.3   0.0   1.0   0.0    0.1    0.0  97   0 826.2  100.0 aec68fba
(36)
6847.6   0.0 54781.2   0.0   1.0   0.0    0.1    0.0  99   0   0.0    0.0 b204cbe5
(37)
```

ZFS Enhancements

- ZFS I/O Throttle
- Ability to Dump to a RAID-Z Pool
- Each SmartMachine has its own ZFS file system
- KVM virtual machines use ZFS volumes for disk
- New kstats

Live Image Support

- **SmartOS runs as a “live image” from a USB key**
- **Root file system uses ramdisk and is mounted read-only**
  - Better Security
  - Disk space is used for zones
  - /opt, /var, /etc/zones are mounted in the “zones” zpool
- **Persistent services can be added via /opt/custom/smf/*xxx*.xml and /opt/custom/*method***

VNIC Enhancements

- Dynamic VNICs are created/destroyed when zones boot/halt
- Each zone can have a vnic created by the global zone with a “friendly” name (net0, for instance)
- Enhanced dladm, dlstat, and flowadm commands with zone support

coreadm Enhancements

- Support to limit the number of core dumps
- Added %Z corefile name pattern for zonepath

DTrace Enhancements

- **llquantize**
  - Log/Linear Quantization (see http://dtrace.org/blogs/bmc/2011/02/08/llquantize/)
- **vmregs[]** - Retrieve Intel VT-x registers
- **tracemem() action takes a dynamic size argument**
- **toupper()/tolower() subroutines**
- **lltostr() D subroutine takes an optional base**
- **SDT probes for zvol_read and zvol_write**
- **dtrace_helper_actions_max set to 1024**
- **Unregister of defunct provider probes**

mdb Enhancements

- Disassembler support for Intel VT-x instructions
- 16-bit disassembler support
- ::printf
- tab completion
- ::ugrep and ::kgrep for sizes less than 4
- ::scalehrtime
- ::findjsobjects
- mdb_v8
- ::walk jsframe and jstack
- mdb API function for iterating object symbols

ipdadm

- Utility for simulating pathological networks by induce packet drops, delays, and corruption
- Can be used on specific zones (with exclusive IP stacks)
- For global zone, also effects zones with shared IP stacks
- Allows testing of “real-world” environments
- See ipdadm(1M)

```
# ipdadm corrupt 10
Corrupted MAC on input.
Disconnecting: Packet corrupt
```

Driver and Module Updates

- ixgbe updated
- igb updated
- Incorporated March 2012 acpica code from Intel
- Ported open ipmi driver from Freebsd

Other Stuff

Joyent

- Many fixes for handling error cases preventing zone shutdown
- FSS fixes to prevent zone starvation
- Crontab works from both /etc and /var for user-defined cron jobs
- Fixed IP DCE scaling issue
- Reduced SMF RSS (critical when there are lots of zones)
- Lots of miscellaneous bug fixes
- And... KVM

KVM on SmartOS

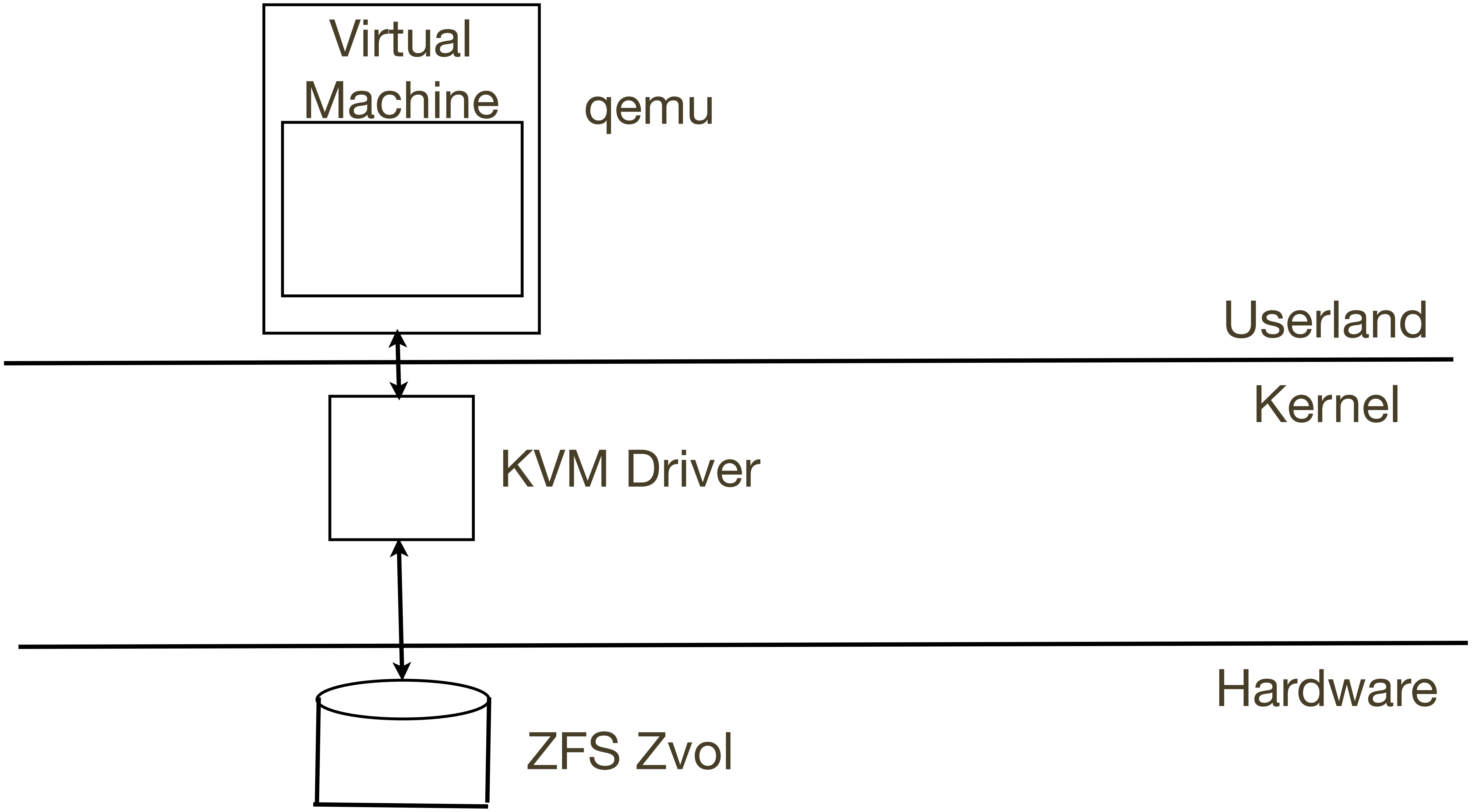
Joyent

- **SmartOS is used by Joyent as a hypervisor.**
- **We create SmartOS zones, but can also create zones running Linux, Windows, FreeBSD, etc. using KVM**
- **This section of the talk will discuss KVM on SmartOS**
  - KVM was ported from Linux and released August, 2011
  - Topics
    - Overview
    - Some Implementation Details
    - Added commands

SmartOS KVM Overview

- KVM runs as a device driver in the SmartOS kernel
- User level is qemu (mostly unchanged from Linux)
- qemu runs in a “kvm” branded zone
- Uses zfs Volumes for disk space
- Uses virtio drivers in the guest OS for disk and network I/O
- No changes were made to the illumos kernel for KVM
- Added kstats and DTrace probes

SmartOS KVM Architecture



KVM vs. Xen vs. VirtualBox

- **Xen requires changes to the host**
  - Xen code is scattered in various places in the kernel of the host (+100k lines of code)
- **VirtualBox**
  - Performance is an issue
- **KVM**
  - No changes to host or guest OS

SmartOS KVM Implementation

- **Driver has 29889 lines of C code (with some assembler)**
- **Much of the porting effort was translation from Linux to SmartOS**
  - Some things are not easily translated
    - Low level CPU-specific registers, memory management code, some ioctl handling, scheduling, etc.
- **Code is organized similar to Linux**
  - We track changes to Linux kvm
- **Qemu (user level) is 661359, and mostly unchanged from Linux**

Some Restrictions

- **Virtual machine memory is locked (i.e., no overprovisioning allowed)**
- **Intel with VT-x and EPT support only**
- **No shared memory/page tables between VMs**
  - No Kernel Samepage Sharing (ksm)

KVM - How it works

```
*       Userland (qemu)                   Kernel    (see kvm.c in source)
*                         |
*    |-----------|        |
*    | VCPU_RUN  |--------|-----------------|
*    | ioctl(2)  |        |                 |
*    |-----------|        |                \|/
*          ^              |            |---------|
*          |              |            | Run CPU |
*          |              |       |--->| for the |
*          |              |       |    |  guest  |
*          |              |       |    |---------|
*          |              |       |         |
*          |              |       |         |
*          |              |       |         |
*          |              |       |         | Stop execution of
*          |              |       |         | guest
*          |              |       |         |------------|
*          |              |  |---------|                 |
*          |              |  |  Handle |                 |
*          |              |  |  guest  |                \|/
*          |              |  |  exit   |               /   \
*     |---------|         |  |---------|             /       \
*     |  Handle |         |       ^                /  Can the  \
*     |  guest  |         |       |--------------/ Kernel handle \
*     |  exit   |         |           Yes        \   the exit    /
*     |---------|         |                        \  reason?  /
*          ^              |                          \       /
*          |              |                            \   /
*          |              |                              |
*          |              |                              | No
*          |--------------|------------------------------|
```

Creating a KVM Virtual Machine

```
# echo "https://datasets.joyent.com/datasets" > /var/db/imgadm/sources.list
# imgadm update
updating local images database...
Get https://datasets.joyent.com/datasets...
done
# imgadm avail
UUID                                 OS      PUBLISHED  URN
c952c558-b640-11e1-b30c-0376247e919c smartos 2012-06-14 sdc:sdc:percona:1.5.1
beabff26-b405-11e1-8281-4fd3c943ceb5 smartos 2012-06-11 sdc:sdc:percona:1.5.0
bbab652a-af5d-11e1-bb80-23b0b09d0df2 smartos 2012-06-05 sdc:sdc:smartosstandard64:1.0
...
e4cd7b9e-4330-11e1-81cf-3bb50a972bda linux   2012-04-04 sdc:jpc:centos-6:1.0.1
...
# imgadm import e4cd7b9e-4330-11e1-81cf-3bb50a972bda
e4cd7b9e-4330-11e1-81cf-3bb50a972bda doesnt exist. continuing with install
e4cd7b9e-4330-11e1-81cf-3bb50a972bda successfully installed
image e4cd7b9e-4330-11e1-81cf-3bb50a972bda successfully imported
#
```

Creating a KVM Virtual Machine (Continued)

```
# imgadm info e4cd7b9e-4330-11e1-81cf-3bb50a972bda
{
  "volume": {
    "creation": "2012-01-20T06:00:58.000Z",
    "name": "zones/e4cd7b9e-4330-11e1-81cf-3bb50a972bda@dataset",
    "type": "snapshot",
    "used": 0
  },
  "children": {
    "snapshots": [],
    "clones": []
  },
  "manifest": {
    "name": "centos-6",
    "version": "1.0.1",
    "type": "zvol",
    "cpu_type": "qemu64",
    "description": "Centos 6 VM 1.0.1",
    "created_at": "2012-04-04T02:51:46.994Z",
    "updated_at": "2012-04-04T02:51:46.994Z",
    "os": "linux",
    "image_size": "10240",
    "files": [
      {
        "path": "centos-6-1.0.0.zvol.bz2",
        "sha1": "79b374d14930a9449d5427118a6dcefacd073a26",
        "size": 662329985,
        "url": "https://datasets.joyent.com/datasets/e4cd7b9e-4330-11e1-81cf-3bb50a972bda/
centos-6-1.0.0.zvol.bz2"
      }
```

Creating a KVM Virtual Machine (Continued)

```
# imgadm list
UUID                                 OS      PUBLISHED  URN
e4cd7b9e-4330-11e1-81cf-3bb50a972bda linux   2012-04-04 sdc:jpc:centos-6:1.0.1
# cat /var/tmp/zonedef
{
  "brand": "kvm",
  "resolvers": [
 "8.8.8.8",
    "8.8.4.4"
  ],
  "default-gateway": "10.88.88.1",
  "ram": "512",
  "vcpus": "2",
  "nics": [
    {
      "nic_tag": "admin",
      "ip": "10.88.88.33",
      "netmask": "255.255.255.0",
      "gateway": "10.88.88.1",
      "model": "virtio",
      "primary": true
    }
  ],
  "disks": [
    {
      "image_uuid": "e4cd7b9e-4330-11e1-81cf-3bb50a972bda",
      "boot": true,
      "model": "virtio",
      "size": 10240
    }
  ]
}
# vmadm create -f /var/tmp/zonedef
Successfully created aa92d91b-cf31-4bd6-91db-86c226de11f8
# vmadm list -v
UUID                                  TYPE  RAM      STATE             ALIAS
aa92d91b-cf31-4bd6-91db-86c226de11f8  KVM   512      running           -
#
```

kvmstat

```
# kvmstat 5
   pid vcpu |   exits :   haltx   irqx  irqwx    iox  mmiox |   irqs   emul   eptv
 71832    0 |      14 :       2      0      0      1      0 |      2     10      0
 71832    1 |      11 :       2      0      0      1      0 |      2      9      0
 71832    2 |       7 :       1      0      0      0      0 |      1      5      0
 71832    3 |       9 :       1      0      0      0      0 |      1      6      0
   pid vcpu |   exits :   haltx   irqx  irqwx    iox  mmiox |   irqs   emul   eptv
 71832    0 |      10 :       1      0      0      0      0 |      1      8      0
 71832    1 |       8 :       1      0      0      0      0 |      1      6      0
 71832    2 |      10 :       2      0      0      1      0 |      2      7      0
 71832    3 |      13 :       2      0      0      1      0 |      2     10      0
...
(now running some “stuff” on the VM)
   pid vcpu |   exits :   haltx   irqx  irqwx    iox  mmiox |   irqs   emul   eptv
 71832    0 |    3236 :     417     12      4    323      5 |    425    541   1303
 71832    1 |    3189 :     418      5      6    324      6 |    426    539   1257
 71832    2 |    3224 :     417      7      8    327      5 |    427    539   1287
 71832    3 |    3212 :     415     11      6    325      4 |    424    539   1278
   pid vcpu |   exits :   haltx   irqx  irqwx    iox  mmiox |   irqs   emul   eptv
 71832    0 |    3340 :     334     32      5    117      7 |    344    530   2057
 71832    1 |    3070 :     198      6      3    153      8 |    206    417   2101
 71832    2 |    3352 :     200     18      5    380     12 |    210    503   2047
 71832    3 |    3377 :     192     39      3    110     10 |    203    636   2207
...
```

KVM DTrace Probes

- **66 SDT probes**
  - Entry and exit from virtual machine
  - Emulation
  - “XXX” probes
- **Access to VT-x registers**
- **And, of course, 1478 fbt entry/return probes**

Example KVM DTrace Probes

Joyent

```
dtrace -n 'vmx_vcpu_run:kvm-vrun{self->ts = timestamp;} vmx_handle_exit:kvm-vexit/self-
>ts/{@ = quantize(timestamp-self->ts); self->ts = 0;} tick-10sec{printa("%@ld", @);}'
dtrace: description 'vmx_vcpu_run:kvm-vrun' matched 3 probes
CPU     ID                    FUNCTION:NAME
  5  70195                      :tick-10sec
           value  ------------- Distribution ------------- count
            1024 |                                         0
            2048 |@@@@@@@@@@@@@@@@@                        158
            4096 |@@@@@@@@@                                79
            8192 |@@@@@@@@@@@@                             109
           16384 |@@                                       18
           32768 |                                         1
           65536 |                                         0

   5  70195                      :tick-10sec
           value  ------------- Distribution ------------- count
             512 |                                         0
            1024 |                                         660
            2048 |@@@@@@@@                                 37575
            4096 |@@@@@@@@@@@@@@@@@@@                      91110
            8192 |@@@@@@@                                  33789
           16384 |@@                                       10849
           32768 |@@                                       11073
           65536 |@                                        4431
          131072 |                                         784
          262144 |                                         326
          524288 |                                         217
         1048576 |                                         94
         2097152 |                                         41
         4194304 |                                         2
         8388608 |                                         0
```

More DTrace

```
# cat vmtime.d
#pragma D option quiet

        kvm-guest-entry
        {
                self->entry = timestamp;
        }

        kvm-guest-exit
        /self->entry/
        {
                @[pid, vmregs[VMX_VIRTUAL_PROCESSOR_ID]] =
                    quantize(timestamp - self->entry);
        }

        END
        {
                printa("pid %d, vcpu %d: %@d\n", @);
        }
```

And Another DTrace Example

```
# dtrace -n 'kvm-guest-entry{@[probefunc, vmregs[VMX_GUEST_RIP]] = count();} kvm-guest-
exit{@[probefunc, vmregs[VMX_GUEST_RIP]] = count();} tick-10sec{printa("%s: %lx = %@d\n", @);}'
dtrace: description 'kvm-guest-entry' matched 3 probes
CPU     ID                    FUNCTION:NAME
  3  70195                      :tick-10sec
kvm_guest_enter: ffffffff81008d71 = 1
kvm_guest_enter: ffffffff81008d76 = 1
kvm_guest_enter: ffffffff8101eb4c = 1
kvm_guest_enter: ffffffff81062111 = 1
kvm_guest_exit: ffffffff81008d71 = 1
kvm_guest_exit: ffffffff81008d76 = 1
kvm_guest_exit: ffffffff8101eb4c = 1
kvm_guest_exit: ffffffff81062111 = 1
kvm_guest_enter: ffffffff8157ed80 = 5
kvm_guest_exit: ffffffff8157ed80 = 5
kvm_guest_enter: ffffffff8101ef5a = 20
kvm_guest_enter: ffffffff8101ef6a = 20
kvm_guest_enter: ffffffff8101ef7f = 20
kvm_guest_enter: ffffffff81434a1a = 20
kvm_guest_exit: ffffffff8101ef5a = 20
kvm_guest_exit: ffffffff8101ef6a = 20
kvm_guest_exit: ffffffff8101ef7f = 20
kvm_guest_exit: ffffffff81434a1a = 20
kvm_guest_enter: ffffffff81009e23 = 46
kvm_guest_exit: ffffffff81009e23 = 46
kvm_guest_enter: ffffffff8101eb46 = 231
kvm_guest_exit: ffffffff8101eb46 = 231
```

KVM kstats

```
# kstat -m kvm
module: kvm                             instance: 0
name:   vcpu-0                          class:    misc
   crtime                           392690.871534584
        exits                           2626442
 fpu-reload                       27733
   halt-exits                       41392
   halt-wakeup                      21847
   host-state-reload                982601
  hypercalls                       0
       insn-emulation                   1141865
 inst-emulation-fail              0
       invlpg                           0
       io-exits                         944877
  irq-exits                        130358
  irq-injections                   42956
   irq-window-exits                 991
     mmio-exits                       6387
    nmi-injections                   0
       nmi-window-exits                 0
       pf-fixed                         120903
  pf-guest                         0
       pid                              71832
   request-irq-exits                0
       signal-exits                     1
       snaptime                         408375.594455416
        zonename                         4fdd7a1e-6eea-4208-ad24-de1ef603d789

 ...
```

KVM mdb Additions

```
> ::walkers !grep kvm
kvm                      - walk all the kvm structures
kvm_mem_alias            - walk kvm_mem_alias structures for a given kvm
kvm_memory_slot          - walk kvm_memory_slot structures for a given kvm
kvm_mmu_page_header      - walk the kvm_mmu_page_header cache
kvm_pte_chain            - walk the kvm_pte_chain cache
kvm_ringbuf_entry        - given a kvm_ringbuf_t, walk its entries
kvm_rmap_desc            - walk the kvm_rmap_desc cache
kvm_vcpu                 - walk the kvm_vcpu cache
>
> ::dcmds !grep kvm
kvm_gpa2qva              - translate a guest physical to a QEMU virtual
address
kvm_gsiroutes            - print out the global system interrupt (GSI) routing
table
kvm_ringbuf_entry        - print out a kvm ring buffer entry
>
```

KVM ZFS Usage

```
# zfs list
NAME                                               USED  AVAIL  REFER  MOUNTPOINT
zones                                             40.0G  2.10T   483K  /zones
zones/47e6af92-daf0-11e0-ac11-473ca1173ab0         177M  2.10T   177M  /zones/47e6af92-
daf0-11e0-ac11-473ca1173ab0
zones/4fdd7a1e-6eea-4208-ad24-de1ef603d789         112K  10.0G    81K  /zones/
4fdd7a1e-6eea-4208-ad24-de1ef603d789                                         <- zone files
zones/4fdd7a1e-6eea-4208-ad24-de1ef603d789-disk0  29.5M  2.10T  2.78G  -     <- boot disk
zones/4fdd7a1e-6eea-4208-ad24-de1ef603d789-disk1   646M  2.10T   646M  -     <- data disk
zones/4fdd7a1e-6eea-4208-ad24-de1ef603d789/cores    31K  4.25G    31K  /zones/
4fdd7a1e-6eea-4208-ad24-de1ef603d789/cores
zones/77e42823-4b9d-4ddf-8a11-1f0434c20eb9        4.47M  5.00G   177M  /zones/
77e42823-4b9d-4ddf-8a11-1f0434c20eb9
zones/77e42823-4b9d-4ddf-8a11-1f0434c20eb9/cores    31K   500M    31K  /zones/
77e42823-4b9d-4ddf-8a11-1f0434c20eb9/cores
zones/config                                        53K  2.10T    53K  legacy
zones/cores                                         31K  10.0G    31K  /zones/global/cores
zones/dump                                        24.0G  2.10T  24.0G  -
zones/e4cd7b9e-4330-11e1-81cf-3bb50a972bda        2.78G  2.10T  2.78G  -
zones/opt                                          325M  2.10T   325M  legacy
zones/swap                                          12G  2.11T    16K  -
zones/var                                         63.0M  2.10T  63.0M  legacy
```

KVM Log Files

```
# svcs -L vmadmd
/var/svc/log/smartdc-vmadmd:default.log
# cat /zones/4fdd7a1e-6eea-4208-ad24-de1ef603d789/root/tmp/vm.log
+ exec /smartdc/bin/qemu-system-x86_64 -m 2048 -name 4fdd7a1e-6eea-4208-ad24-de1ef603d789 -uuid
4fdd7a1e-6eea-4208-ad24-de1ef603d789 -cpu qemu64 -smp 4 -drive file=/dev/zvol/rdsk/zones/
4fdd7a1e-6eea-4208-ad24-de1ef603d789-disk0,if=virtio,index=0,media=disk,boot=on -drive file=/
dev/zvol/rdsk/zones/4fdd7a1e-6eea-4208-ad24-de1ef603d789-disk1,if=virtio,index=1,media=disk -
boot order=cd -device virtio-net-pci,mac=90:b8:d0:8a:60:d3,tx=timer,x-txtimer=200000,x-
txburst=128,vlan=0 -net
vnic,name=net0,vlan=0,ifname=net0,ip=151.1.224.73,netmask=255.255.255.192,gateway_ip=151.1.224.6
5,hostname=maxtest-centos,dns_ip0=8.8.8.8,dns_ip1=8.8.4.4 -smbios
'type=1,manufacturer=Joyent,product=SmartDC HVM,version=6.1,serial=4fdd7a1e-6eea-4208-ad24-
de1ef603d789,uuid=4fdd7a1e-6eea-4208-ad24-de1ef603d789,sku=001,family=Virtual Machine' -
pidfile /tmp/vm.pid -chardev socket,id=qmp,path=/tmp/vm.qmp,server,nowait -qmp chardev:qmp -
chardev socket,id=serial0,path=/tmp/vm.console,server,nowait -serial chardev:serial0 -chardev
socket,id=serial1,path=/tmp/vm.ttyb,server,nowait -serial chardev:serial1 -vnc unix:/tmp/vm.vnc
-parallel none -usb -usbdevice tablet -k en-us -vga cirrus
Could not open option rom 'extboot.bin': No such file or directory
Start bios (version 0.6.1.2-20110201_165504-titi)
Ram Size=0x80000000 (0x0000000000000000 high)
CPU Mhz=2394
...
Found 4 cpu(s) max supported 4 cpu(s)
...
Booting from Hard Disk...
Booting from 0000:7c00
#
```

Summary

- **SmartOS**
  - Zones, ZFS, KVM, DTrace, SMF
  - An OS for the Cloud, but also good for general purpose use
  - Open Source
  - Available at www.smartos.org

Try the Joyent Public Cloud

Joyent

- Go to http://my.joyent.com
- Signup
- For billing information, use coupon code “eujoyeur”
- Limited to first 50 signups
- Good for a 1GB SmartMachine