Using SmartOS as a Hypervisor

SCALE 10x

Robert Mustacchi
rm@joyent.com (@rmustacc)
Software Engineer
What is SmartOS?

- Solaris heritage
  - Zones - OS level virtualization
  - Crossbow - virtual NICs
  - ZFS - pooled storage, data integrity
  - DTrace - production safe Dynamic Tracing

- Hypervisor Focus
  - Core OS image booted from external media
  - Persist user data and minimal convenience config
  - Tools to simplify management

- KVM - Hardware Virtualization

- Open Source distribution of illumos

- illumos is the successor of OpenSolaris
Zones - OS Level Virtualization

- A zone is an entirely self-managed container
  - Configure own users, disks, networking, services
  - Feels like a standalone OS

- Isolation
  - Zones can’t see each other
  - Global zone can inspect local zones
  - Exclusive network stacks
  - Filesystem isolation

- Resource Controls
  - Memory, Disk and Network I/O
  - CPU Shares and Caps

- Privileges

- Zone Brands
  - Sparse
  - Legacy support - S10
OS Virtualization

- Minimal overhead - no hardware to emulate
- Share the same kernel - higher density

- Allows for services in the global zone to inspect the others e.g. DTrace
Crossbow - Virtual NICs

- Create virtual NICs and virtual switches

- Connect VNICS to:
  - Physical NICs
  - Virtual switch

- Antispoof
  - MAC address
  - IP addresses
  - DHCP

- Bandwidth controls

- Simple as `dladm create-vnic -l igb0 foo0`
ZFS - Using disks better

• ZFS is a copy on write filesystem
• Pooled Storage
  • Don’t have to guess your partition sizes
  • Managed in datasets
  • Quotas and reservations can be changed on the fly
  • zvols - virtual block devices
• Multiple RAID options
  • RAID 1 (Mirroring)
  • RAID Z1, Z2, Z3 (Single, double, and triple parity)
  • RAID 0 (Striping)
ZFS - Using data better

- Enterprise features built in
- 128-bit Checksums on everything
- Data Integrity
- Compression
- Deduplication
- Adaptive Replacement Cache
- Hybrid Storage
  - SLOG
  - L2ARC
ZFS Snapshots and Clones

- Snapshots are cheap to take
- You can clone a snapshot into a read/write copy
- Send/Recv incremental snapshots
DTrace

- Dynamic instrumentation of production systems
  - Originally released in 2003 for Solaris 10, open-sourced in 2005
  - Available on SmartOS, illumos, and all other Solaris-derived systems
  - Available on Mac OS X, FreeBSD, QNX, and WIP on Linux, NetBSD, Sony Vita

- Supports static and dynamic probes in both userland and the kernel with arbitrary actions and predicates

- Aggregates data in the kernel
  - Allows us to support high numbers of events per second

- Designed to be safe for production use from the get go
MySQL query latency can be measured with a (long) one-liner:

```
# dtrace -n 
mysql*:::query-start { self->start = timestamp; } mysql*:::query-done /self->start/ {
    @[“nanoseconds”] = quantize(timestamp - self->start);
    self->start = 0;
}
```

nanoseconds

<table>
<thead>
<tr>
<th>value</th>
<th>Distribution</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>2048</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>4096</td>
<td>@</td>
<td>93</td>
</tr>
<tr>
<td>8192</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>16384</td>
<td>@@@</td>
<td>232</td>
</tr>
<tr>
<td>32768</td>
<td>@@</td>
<td>172</td>
</tr>
<tr>
<td>65536</td>
<td>@@@@@</td>
<td>532</td>
</tr>
<tr>
<td>131072</td>
<td>@@@@@@@@@@@@@@</td>
<td>1513</td>
</tr>
<tr>
<td>262144</td>
<td>@@@@@@@@@@@@@@@@</td>
<td>428</td>
</tr>
<tr>
<td>524288</td>
<td>@@@@@@@@@@@@@@</td>
<td>258</td>
</tr>
<tr>
<td>1048576</td>
<td>@@@@</td>
<td>127</td>
</tr>
<tr>
<td>2097152</td>
<td>@@@@</td>
<td>47</td>
</tr>
<tr>
<td>4194304</td>
<td>@@@</td>
<td>20</td>
</tr>
<tr>
<td>8388608</td>
<td>@@</td>
<td>33</td>
</tr>
<tr>
<td>16777216</td>
<td>@@</td>
<td>9</td>
</tr>
<tr>
<td>33554432</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Porting KVM

- Why?
  - People need to virtualize existing build out
  - Give flexibility to run other OSes
  - Still need all the other technology we talked about

- Joyent started the port in Fall of 2010 and released it at KVM Forum in August 2011

- Actively used in production in Joyent’s Public Cloud

- Only Intel processors with EPT currently supported
  - Community working on AMD support (Josh Clulow, Rich Lowe, …)

- Porting gotchas
  - Didn’t find new bugs in KVM - just self inflicted wounds
  - Duplicate PITs
  - Not properly saving per-CPU GSBASE
  - Not properly resetting FPU state
Put QEMU in a Zone

- Each QEMU instance is init in its own KVM branded zone
- Only kvm branded zones get /dev/kvm by default
- Zone reduces QEMU Attack surface
- Leverages zones features for isolation and limited privileges
Arming QEMU with Crossbow

- Wrote a new QEMU network backend to use a VNIC
- Each NIC in the guest corresponds to a VNIC in the host
- VNIC backend has an optional DHCP server
- Antispoof is enabled by default
  - Portions of antispoof eliminated if not needed
- Enables insight into guest networking throughput
Back QEMU with ZFS

- Each disk in the guest is backed with a zvol (virtual block device)
- You can snapshot and rollback the zvols
- ARC can help with random reads, SLOG with synch writes
- Rapid provisioning through clones
  - Create a small basic golden install
  - Clone that for every provision
  - Create an empty data disk based on need
  - Less than one minute from provision to ping
  - This process is automated with vmadm(1M)
- Leverage ZFS send and receive for replication and backup
KVM meet DTrace

- As of QEMU 0.14, QEMU has DTrace probes — we lit those up on illumos
- Added a bevy of SDT probes to KVM itself
  - including all of the call-sites of the trace_*() routines
- Added vmregs[] variable that queries current VMCS
  - See guest registers
- Can all be enabled dynamically and safely, and aggregated on an arbitrary basis
  - per-VCPU, per-VM, per-CPU, etc.
Seeing DTrace - ext3 writes
Seeing DTrace - KVM IRQs

Virtual Machine: Interrupt Requests decomposed by subsecond offset and IRQ Vector

server hostname Equal Submit Query

SUBSECOND OFFSET

- Isolate selected
- Exclude selected

X-axis: Time, in 1 second increments

Displaying subsecond offset up to 1.02 s
Seeing DTrace - KVM Disk I/O Latency

Virtual Machine: disk I/O operations decomposed by latency and operation type

- **server hostname**: Equal
- **Latency**:
  - **Read**
  - **Write**

**X-axis**: Time, in 1 second increments
**Displaying latency up to 103 ms**
Seeing DTrace - KVM CPU Sampling
What is DTrace doing?

- We sample the CPUs at 99 hz (can do higher rates)
- We read the guest’s value of CR3 from the VMCS
- We aggregate with CR3 as the key
- The value is the distribution of when in the second

```
profile::profile-99hz
{
    @(l1tostr(vmregs[VMX_GUEST_CR3], 16))] = lquantize(((timestamp) % 1000000000) / 1000000000, 0, 1000000000, 0, 1000, 10);
}
```
Can we do better?

- Why can’t we DTrace into the guest?
- Get a little help from the guest - symbol table
- Add the knowledge of how to walk EPT
- What once were traps have to become VMEXITS
- It’s all program text, just in QEMU’s address space

Providers
- vfbt - Entry and return from function in the kernel
- vsyscall - Entry and return from system calls
- vpid?! - Trace guest userland processes
Wrapping up: SmartOS as a Hypervisor

- High-tenancy: SmartOS containers
- OS flexibility: KVM
- Highly observable with DTrace
- Strong Isolation and Protection
  - Zones and Crossbow
- Data is protected and easy to manage
  - Pooled storage and datasets
- Management tools - vmadm
Thank you

- SmartOS Resources
  - Download SmartOS - http://smartos.org
  - SmartOS Mailing List - http://smartos.org/smartos-mailing-list/
  - SmartOS Wiki - http://wiki.smartos.org
  - illumos - http://illumos.org
  - Contribute to SmartOS - http://github.com/joyent/smartos-live
  - Hop into #illumos on irc.freenode.net and say hello

- Thanks
  - Max Bruning and Bryan Cantrill for their work on KVM
  - Josh Wilsdon for vmadm
  - John Sonnenschein for driving all the SmartOS resources
  - Joyent and illumos community for their support
  - SCALE10x volunteers for a great conference

- rm@joyent.com, rmustacc on freenode/twitter