Joyent

Using SmartOS as a Hypervisor

SCALE 10x

Robert Mustacchi
rm@joyent.com (@rmustacc)
Software Engineer

What is SmartOS?

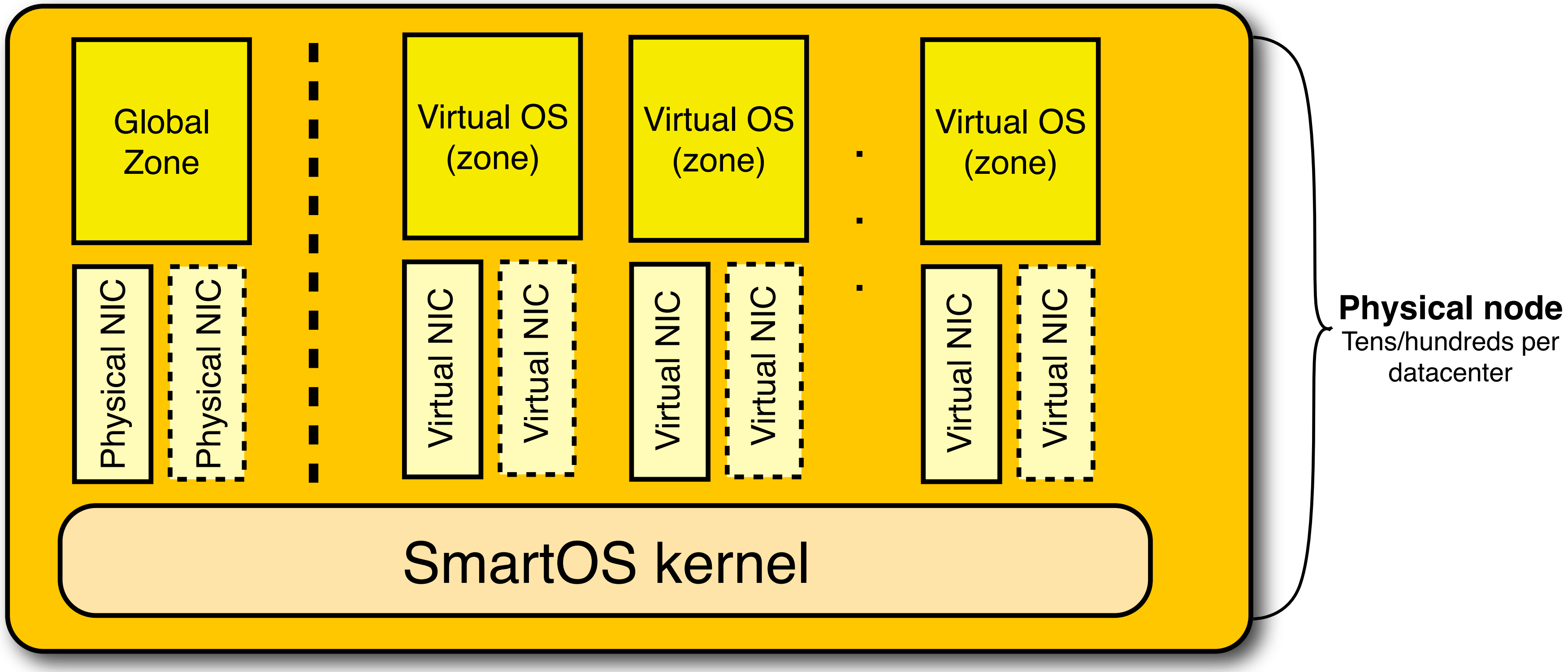
- Solaris heritage
  - Zones - OS level virtualization
  - Crossbow - virtual NICs
  - ZFS - pooled storage, data integrity
  - DTrace - production safe Dynamic Tracing
- Hypervisor Focus
  - Core OS image booted from external media
  - Persist user data and minimal convenience config
  - Tools to simplify 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

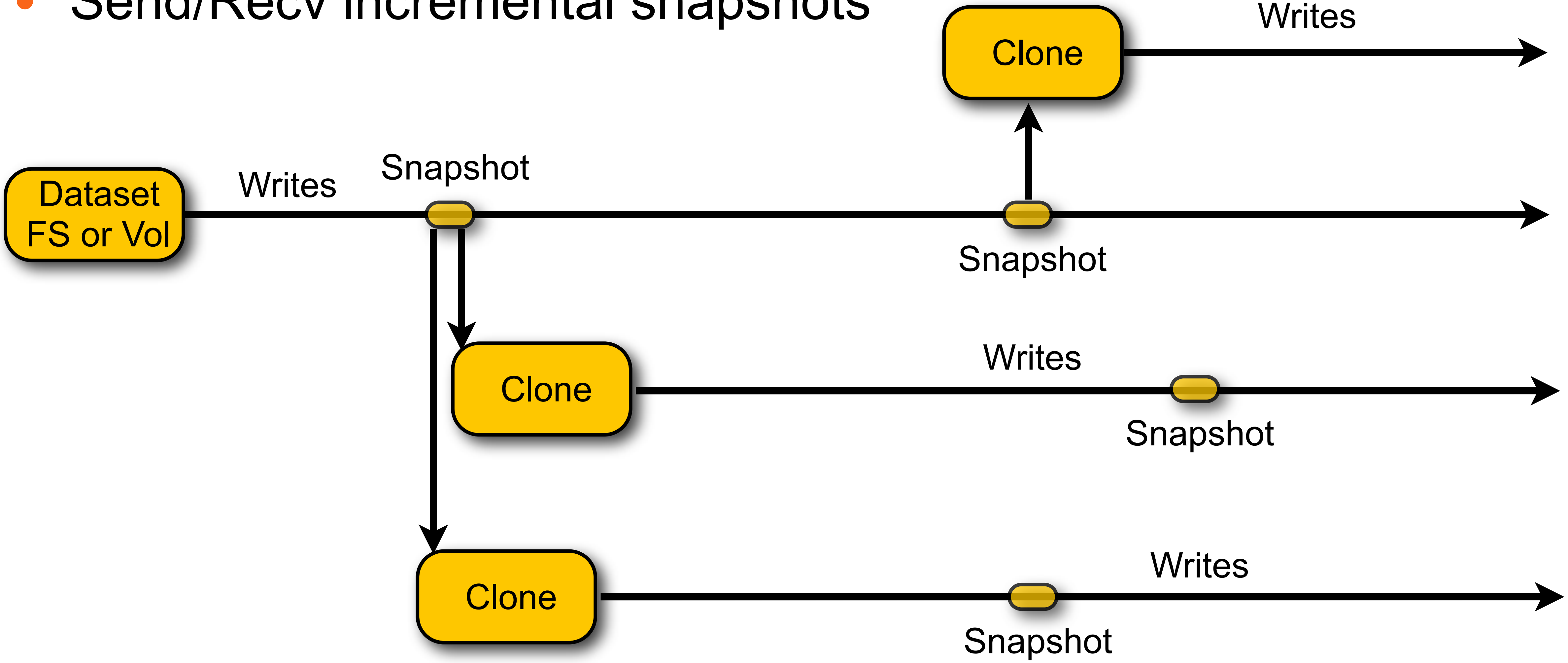
- ZFS is a copy on write filesystem
- Pooled Storage
    - Don’t have to guess your partition sizes
    - Managed in datasets
    - Quotas and reservations can be changed on the fly
    - zvols - virtual block devices
- Multiple RAID options
    - RAID 1 (Mirroring)
    - RAID Z1, Z2, Z3 (Single, double,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

DTrace Example: MySQL Query Latency

- 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
       value  ------------- Distribution ------------- count
        1024 |                                         0
        2048 |                                         16
        4096 |@                                        93
        8192 |                                         19
       16384 |@@@                                      232
       32768 |@@                                       172
       65536 |@@@@@@                                   532
      131072 |@@@@@@@@@@@@@@@@@                        1513
      262144 |@@@@@                                    428
      524288 |@@@                                      258
     1048576 |@                                        127
     2097152 |@                                        47
     4194304 |                                         20
     8388608 |                                         33
    16777216 |                                         9
    33554432 |                                         0
```

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

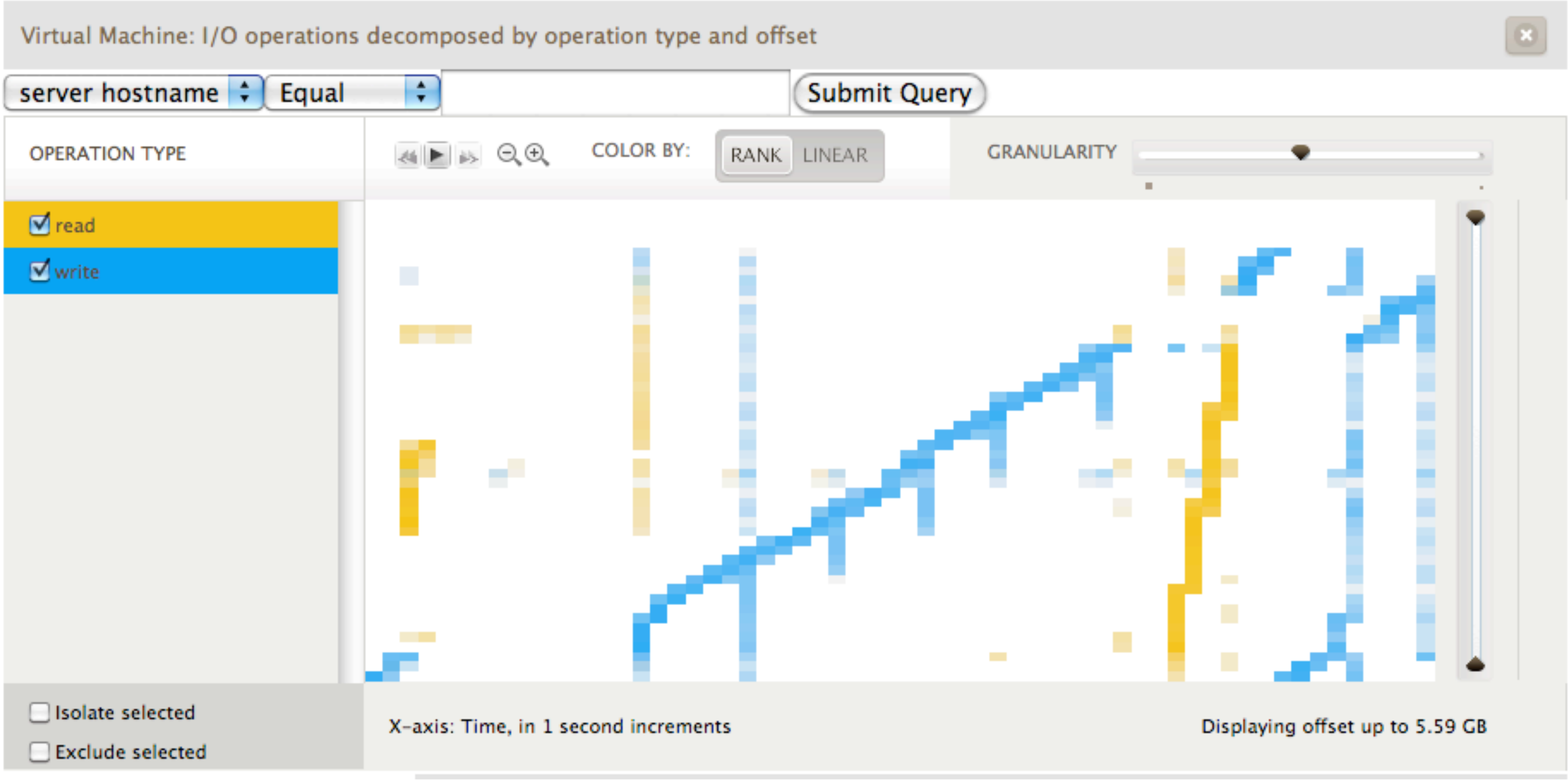
Back QEMU with ZFS

- 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



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

server hostname | Equal | Submit Query

OPERATION TYPE

☑ read

☑ write

COLOR BY: RANK LINEAR

GRANULARITY

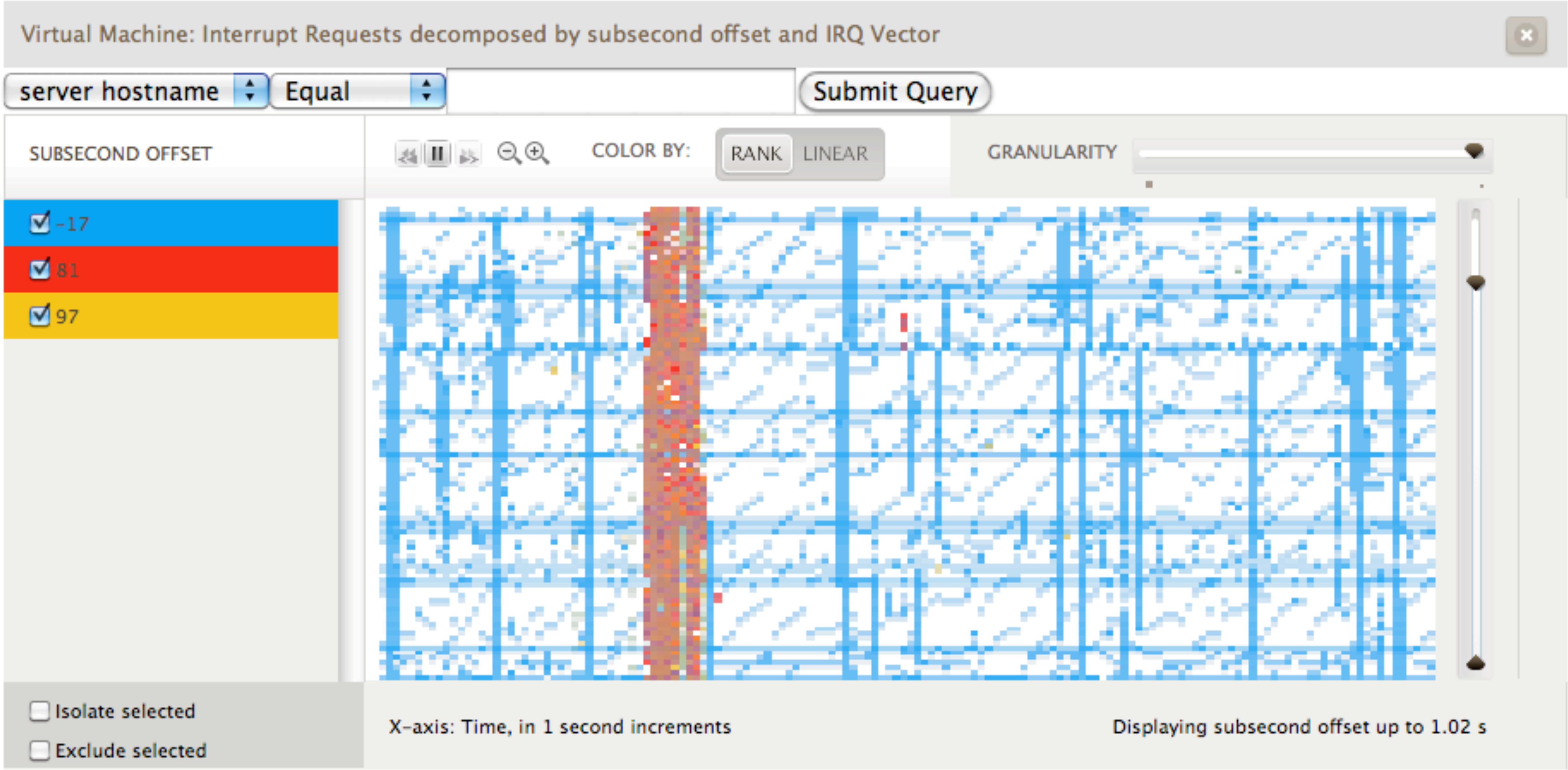
□ Isolate selected

□ Exclude selected

X-axis: Time, in 1 second increments

Displaying offset up to 5.59 GB

Seeing DTrace - KVM IRQs



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

server hostname | Equal | Submit Query

SUBSECOND OFFSET

☑ -17

☑ 81

☑ 97

□ Isolate selected

□ Exclude selected

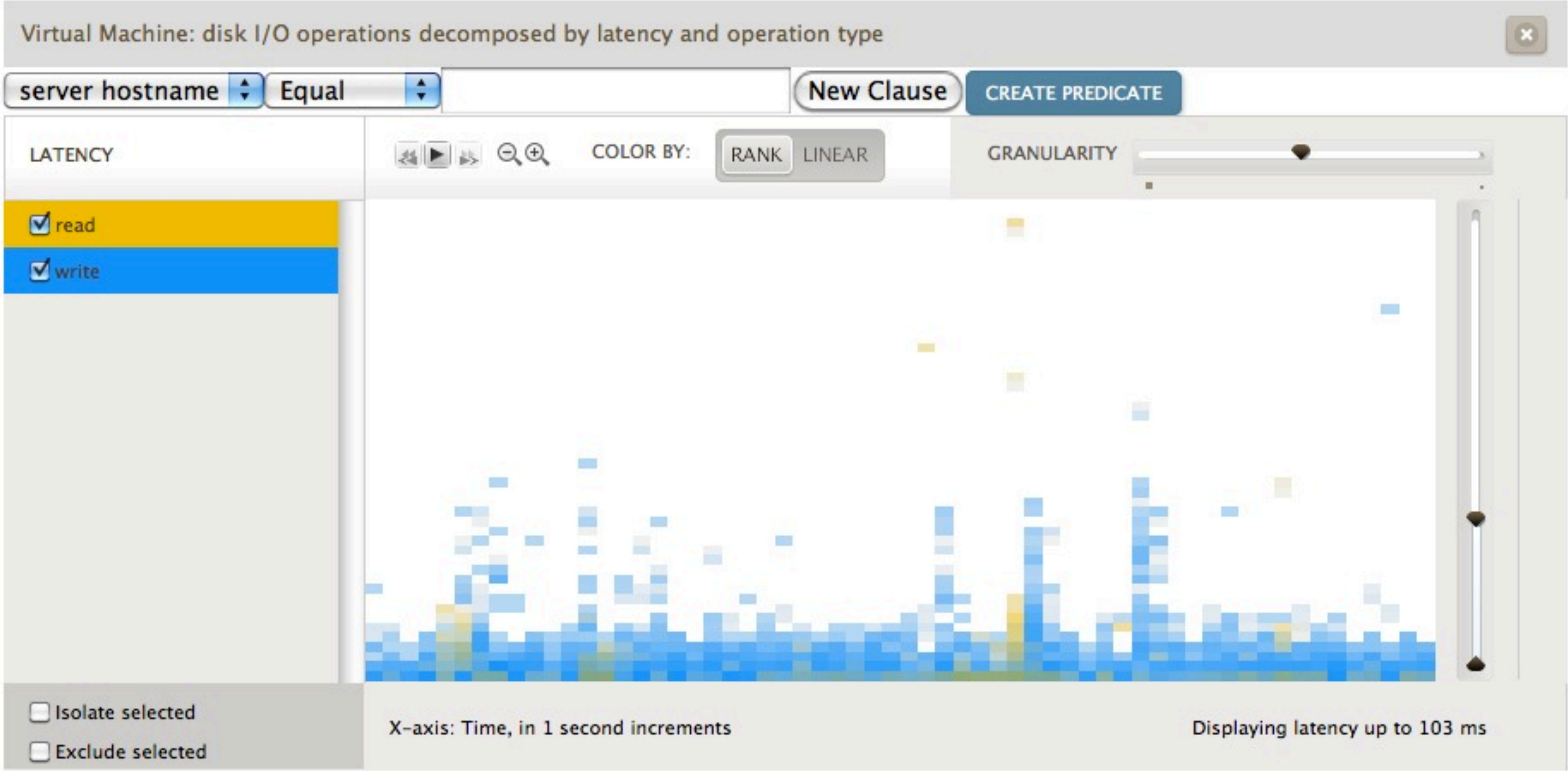
COLOR BY: RANK LINEAR

GRANULARITY

X-axis: Time, in 1 second increments

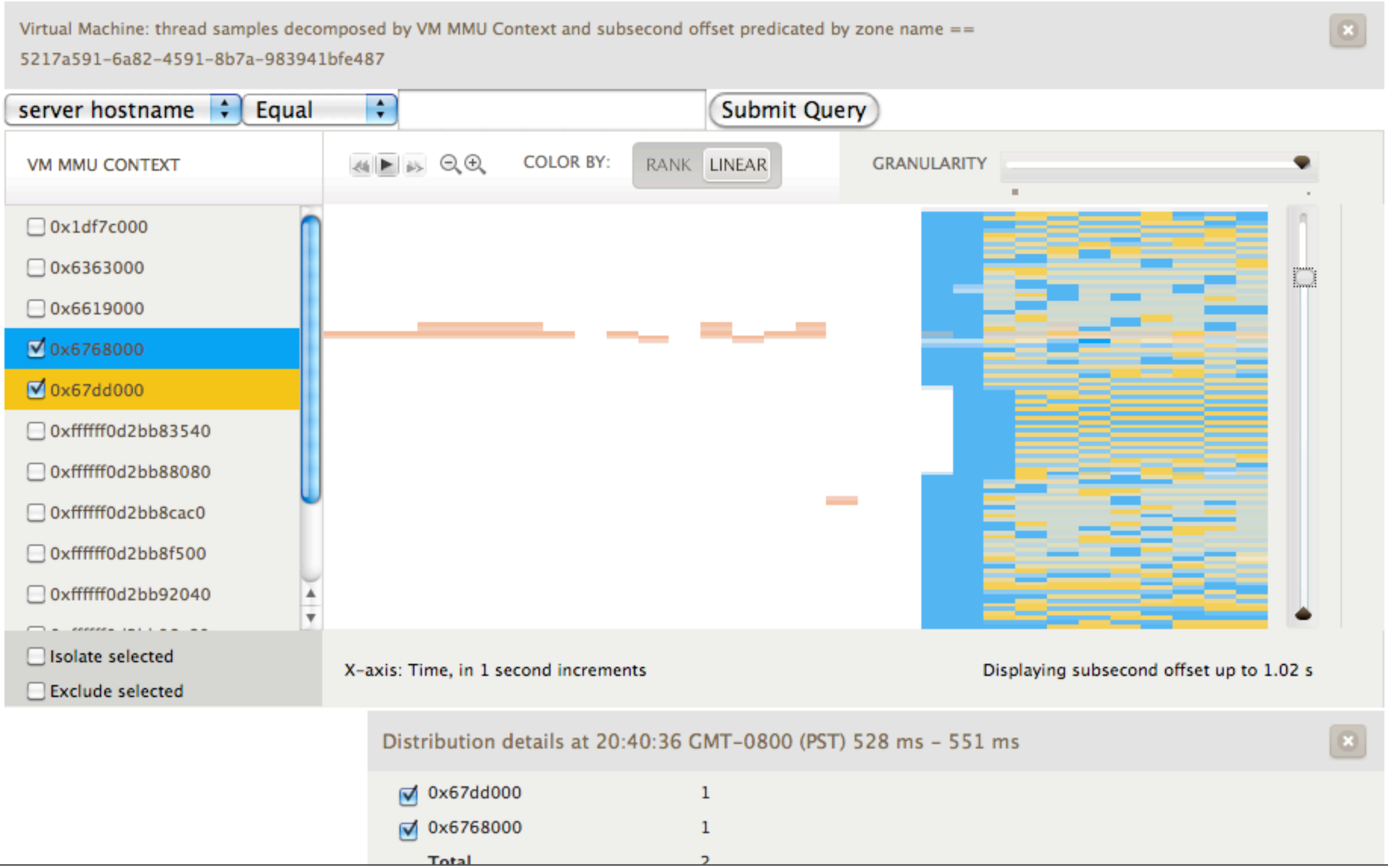
Displaying subsecond offset up to 1.02 s

Seeing DTrace - KVM Disk I/O Latency



Virtual Machine: disk I/O operations decomposed by latency and operation type
server hostname
Equal
New Clause
CREATE PREDICATE
LATENCY
COLOR BY:
RANK
LINEAR
GRANULARITY
☑ read
☑ write
□ Isolate selected
□ Exclude selected
X-axis: Time, in 1 second increments
Displaying latency up to 103 ms

Seeing DTrace - KVM CPU Sampling



Virtual Machine: thread samples decomposed by VM MMU Context and subsecond offset predicated by zone name == 5217a591-6a82-4591-8b7a-983941bfe487

server hostname | Equal | Submit Query

- □ 0x1df7c000
- □ 0x6363000
- □ 0x6619000
- ☑ 0x6768000
- ☑ 0x67dd000
- □ 0xffffff0d2bb83540
- □ 0xffffff0d2bb88080
- □ 0xffffff0d2bb8cac0
- □ 0xffffff0d2bb8f500
- □ 0xffffff0d2bb92040

□ Isolate selected

□ Exclude selected

X-axis: Time, in 1 second increments

Displaying subsecond offset up to 1.02 s

Distribution details at 20:40:36 GMT-0800 (PST) 528 ms - 551 ms

☑ 0x67dd000	1
☑ 0x6768000	1
Total	2

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
{
     @[(lltostr(vmregs[VMX_GUEST_CR3], 16))] =
         lquantize(((timestamp) % 1000000000) /
         1000000, 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