SmartOS: The Modern Operating System

Bryan Cantrill
VP, Engineering

bryan@joyent.com
@bcantrill

Speaker

Joyent

Bryan Cantrill
VP, Engineering
Joyent

Housekeeping

- This webinar will be recorded, and a link sent to you within 24 hours.
- Please use the GoToWebinar chat feature to submit questions.
- We'd love your feedback in the short survey at the end of the webinar.

Agenda

Joyent

- The challenges of the cloud and virtualization
- The solution
- A brief history of virtualization
- 2011: The race in cloud computing
- SmartOS and SmartDataCenter
- Q&A

Cloud: An operating systems problem

Joyent

- Cloud computing accentuates many technical challenges:
  - Must be able to *store data absolutely reliably* — but cannot rely on traditional centralized storage
  - Must allow for *efficient multi-tenancy* operation — even at high levels of tenancy
    - Must have *network virtualization* to allow for both flexibility and security in multi-tenant environments
    - Must have *complete observability* to allow the system to be understood when it doesn't perform
    - Must be able to *run entire legacy stacks* — including the operating system
- These problems are all *operating systems* problems

SmartOS: An operating systems solution

Joyent

- Joyent developed SmartOS, an operating system that incorporates essential modern OS technologies:
  - ZFS: Enterprise-class copy-on-write filesystem featuring constant time snapshots, writable clones, built-in compression, checksumming, volume management, etc.
  - OS-based virtualization (Zones): Entirely secure virtual OS instances offering hardware performance, high multi-tenancy
  - Network virtualization (Crossbow): Virtual NIC Infrastructure for easy bandwidth management and resource control
  - DTrace: Facility for dynamic, *ad hoc* instrumentation of production systems that supports *in situ* data aggregation, user-level instrumentation, etc. — and is absolutely safe
- But there was a missing piece...

SmartOS: The missing component

Joyent

- Despite its rich feature-set, SmartOS was missing an essential component: the hardware virtualization necessary to run entire legacy stacks
- But given that this feature is essential, how did Joyent make it this far?
- To understand how Joyent got here — and what we've done to rise to the challenge of hardware virtualization— one needs to understand the nature of virtualization in the cloud...

Cloud: The virtualization challenge

Joyent

- Operating a public cloud has significant technological and business challenges:
  - From a technological perspective, must deliver highly elastic infrastructure with acceptable quality of service across a broad class of users and applications
  - From a business perspective, must drive utilization as high as possible while still satisfying customer expectations for quality of service
- These aspirations are in tension: multi-tenancy can significantly degrade quality of service
- The key enabling technology for multi-tenancy is *virtualization* — but where in the stack to virtualize?

Hardware-level virtualization?

Joyent

- The historical answer — since the 1960s — has been to virtualize at the level of the hardware:
  - A virtual machine is presented upon which each tenant runs an operating system of their choosing
  - There are as many operating systems as tenants
- The historical motivation for hardware virtualization remains its advantage today: it can run entire legacy stacks unmodified
- However, hardware virtualization exacts a heavy tolls: operating systems are not designed to share resources like DRAM, CPU, I/O devices or the network
- *Hardware virtualization limits tenancy*

Platform-level virtualization?

Joyent

- Virtualizing at the application platform layer addresses the tenancy challenges of hardware virtualization…
- ...but at the cost of dictating abstraction to the developer
- This creates the “Google App Engine problem”: developers are in a straightjacket where toy programs are easy — but sophisticated apps are impossible
- Virtualizing at the application platform layer poses many other challenges:
  - Security, resource containment, language specificity, environment-specific engineering costs

Joyent's solution: OS-level virtualization

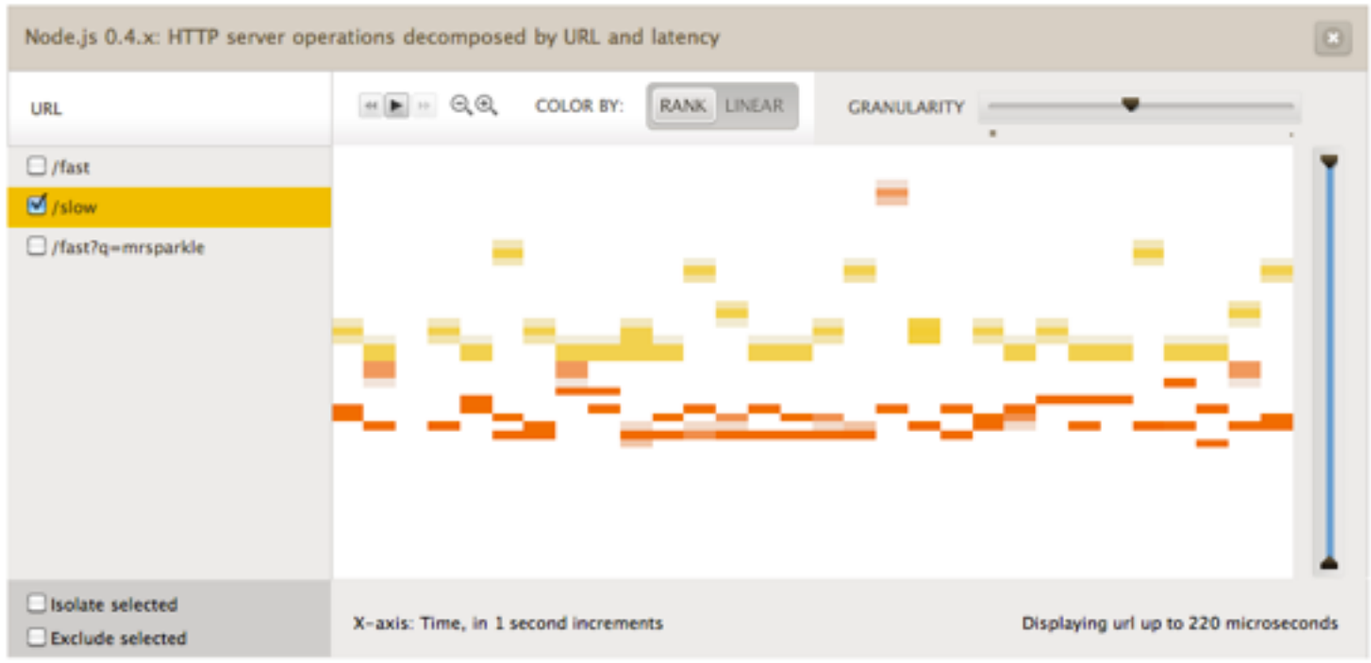
Joyent

- Virtualizing at the OS level hits the sweet spot:
  - Single OS (single kernel) allows for efficient use of hardware resources, and therefore allows load factors to be high
  - Disjoint instances are securely compartmentalized by the operating system
  - Gives customers what appears to be a virtual machine (albeit a very fast one) on which to run higher-level software
  - Gives customers PaaS when the abstractions work for them, IaaS when they need more generality
- OS-level virtualization allows for highest levels of tenancy without dictating abstraction or sacrificing efficiency
- Joyent is the only cloud software provider to offer integrated OS-level virtualization

The OS-level virtualization advantage

Joyent

- Only OS-level virtualization allows the operator *total visibility* into utilization and operation
- Joyent has built upon this to develop *cloud analytics*, a facility that allows operators to differentiate by offering a real-time visualization of cloud latency
- For example, PaaS HTTP latency decomposed by both latency and URI:



Revisiting hardware-level virtualization

Joyent

- Virtualizing at the OS level is the right answer for any PaaS offerings — and more generally for any new development…
- ...but what to do with legacy software or software that has its own OS dependency?
- Hardware-level virtualization still plays a role — and while that role will become diminished as offerings move from IaaS to PaaS and SaaS, it remains essential for cloud deployments

Aside: history of x86 virtualization

Joyent

- Unlike historical IBM architectures, x86 was not initially designed to be virtualized; x86 cannot be virtualized using traditional trap-and-emulate techniques
- For many years, x86 was thought to be “unvirtualizable”
- Through some crafty techniques borrowed from higher-level virtual machines, VMware implemented efficient x86 virtualization based on *binary translation*
- Though originally aimed at workstations, VMware found success virtualizing enterprise workloads, many of which were (largely idle) Windows-based apps running on dedicated machines

VMware: The accidental cloud

Joyent

- Through enterprise workload consolidation, VMware found itself developing offerings that looked increasingly like what we now call cloud
- This was not deliberate: VMware was architected for workload consolidation, not cloud computing
- The difference is around scale and economics: a cloud must be able to deliver data-intensive, real-time applications in a multi-tenant environment
- VMware's inextricable dependencies on hardware-level virtualization for compute and centralized storage have made this challenge acute

Meanwhile, elsewhere...

Joyent

- With VMware having proven it was possible, binary translation was implemented by Fabrice Bellard in the open source QEMU (“quick emulator”)
- Intel— and then AMD — added microprocessor support for efficient virtualization
  - The first generation of this support was barely competitive with VMware’s binary translation…
  - ...but the second generation blew its doors off
- During this time, a startup (Qumranet) developed the necessary layer for this hardware support to be used efficiently by QEMU on Linux
- This glue — dubbed “kernel virtual machine” (KVM) —is open source, and Qumranet was bought by Red Hat

State of cloud computing, circa 2010

Joyent

- VMware's hypervisor, ESX, was originally designed to be lean, but was having to get more and more feature rich to efficiently offer a multi-tenant cloud...
- ...but KVM-based solutions were not competitive in part because these features also did not exist in Linux:
  - No observability
  - No enterprise-grade storage stack
  - No history of optimizing for multi-tenancy load
- And Joyent had the highly-scalable, enterprise-grade operating system in SmartOS — but was missing the KVM layer to allow for running legacy loads

2011: The race in cloud computing

Joyent

- Could Joyent port KVM to SmartOS faster than VMware could turn ESX into a real, multi-tenancy OS?
- Could Joyent port KVM to SmartOS faster than Red Hat could add enterprise-grade cloud features to RHEL?
- Answer: Joyent completed the port of KVM and made it generally available on August 15, 2011
- With the addition of KVM, SmartOS is the leading OS for offering *both* OS-level and hardware-level virtualization on the same infrastructure
- And the hardware virtualization in SmartOS comes with industry leading observability, a highly scalable foundation, and an enterprise storage stack

SmartOS: The hybrid cloud foundation

Joyent

- The cloud will include both hardware-virtualization for legacy loads/IaaS, and OS-level virtualization for new developments/PaaS/SaaS
- The level at which one virtualizes is ultimately a business decision at the level of the app: how important is efficiency/scale vs. supporting a legacy environment?
- Operators can give themselves a tremendous competitive advantage by building a *hybrid* cloud in which *both* OS-level and hardware-virtualization are first class operations
- **Only SmartOS allows such a hybrid cloud** — and on a trusted, tested, highly scalable, highly observable, enterprise-grade system!

SmartDataCenter: A SmartOS-based cloud

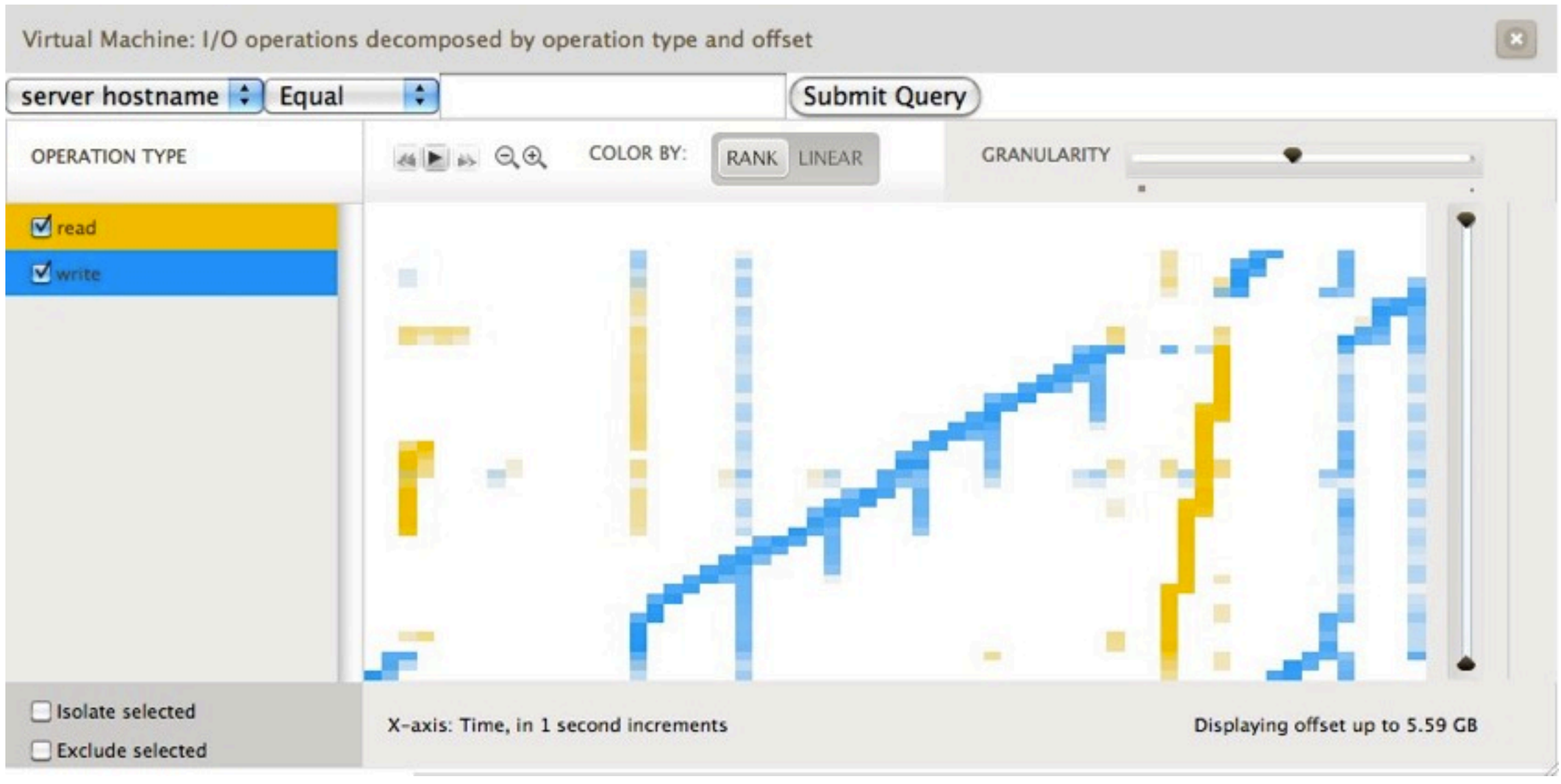
Joyent

- SmartDataCenter is Joyent's SmartOS-based product for operating a cloud — including orchestration, API-based provisioning, distributed observability, etc.
- Leverages key strengths in SmartOS, e.g.:
  - Uses KVM and Zones + Crossbow to allow for both OS- and hardware-based virtualization — and both together to securely contain hardware VMs!
  - Uses ZFS and its cloning capabilities for nearly instant provisioning of hardware VMs
  - Uses ZFS caching to allow better-than-hardware performance from hardware VMs
  - Uses DTrace as the foundation of real-time cloud analytics, allowing for new observability into VMs

SmartDataCenter: Screenshot or GTFO

Joyent

- For example, using SDC to observing Linux ext3 write offsets in a logical volume on a workload that creates and removes a 3 GB file:



SmartOS: A modern operating system

Joyent

- While the challenges of the cloud motivated our work on SmartOS, its use is not limited to the cloud!
- SmartOS is a best-of-breed across different OS families:
  - Zones, Crossbow, DTrace and ZFS from illumos
  - KVM from Linux
  - Packaging technology from NetBSD
  - Toolchain from GNU
- SmartOS is available now: http://smartos.org

Joyent

Questions?

Please use the chat function in
your GoToWebinar Console.

Thank you!

Joyent

- A recording will be sent to you within 24 hours
- Please fill out the brief survey as you sign out
- Please visit Joyent.com for more Joyent-related info
- Visit SmartOS.org for more open source info
- Twitter.com/joyent