# OpenSolaris in the Real World

or, How Joyent turned a cutting edge operating system into a cutting edge product.

Ben Rockwood
Director of Systems, Joyent

email: benr@joyent.com
blog: cuddletech.com/blog/

Joyent

# Agenda

- Overview of Joyent and our product line
- Our Architecture and Choices
- Old Problems, New Solutions

Joyent

# Joyent & TextDrive

- TextDrive started as a shared hosting provider for developers, by developers
- Joyent started as a Web 2.0 application company
- In 2005 the two companies merged

Joyent

# Differentiators

- TextDrive is known for its technical expertise, staff, value, and performance
- Ruby on Rails experts
- Incubator for innovation
- Transparency and strong community

Joyent

# Joyent Products

- Shared and Business Hosting
- Joyent Connector
- StrongSpace
- BingoDisk
- Accelerator Hosting (Containers)

Joyent

# Shared & Biz Hosting

- Traditional Hosting Model
- Deployed currently on Dell 2850 systems running FreeBSD
- Largely powered by Webmin & VirtualMin
- All customer services are local to each system

Joyent

# Joyent Connector

- Web 2.0 Collaboration Application
- Shared calendaring, email, contacts, files, etc.
- Implemented in Ruby on Rails
- Deployed on 5 Containers, on 4 Systems: 2 T1000's and 2 X4100's (Nothing in the global zones)
- Utilizing NFSv3 for backing storage

Joyent

# StrongSpace

- Secure “Global” (Internet) storage
- Accessible via SFTP/SCP or HTTPS file browser (Ruby on Rails)
- Well suited as backup and archiving solution
- Accounts from 5GB to 100GB
- rsync has native support for SCP
- Implemented in a Container on a Thumper

Joyent

# BingoDisk

- Collaborative Web Storage: WebDav
- WebDav native support in Windows, OS X, Konquorer, Nautilus (GNOME/JDS), etc.
- Also accessible via Web Interface (Ruby on Rails)
- Accounts from 25GB to 100GB
- Implemented in a Container on a Thumper

Joyent

# Accelerators

- VPS or "Dedicated Virtual Server" Hosting
- Standard Solaris Containers (Zones)
- Sold in "shares" of a Sun Fire X4100: 1/4th, 1/8th, 1/16... more coming soon.
- Low cost, "burst-able" solution
- Capacity on demand
- Implemented on Sun Fire X4100

Joyent

# Architecture

- 2 Data Centers: San Diego & Emeryville, both Level(3) Facilities
- Entirely Copper Gigabit Ethernet switching environment
- All new deployments on 3 standardized configurations:

Joyent

# Sun Fire T1000

- 32 Threads, 1Ghz, 8GB of memory
- ... Ruby threading sucks.
- Deployment without fore-thought is problematic
- Without extensive pre-planning X4100's are simply more economical.
- Plans for selling Niagara Containers on Hold

Joyent

# Sun Fire X4100

- "Max'ed Config": 4 2.4Ghz Operton Cores, 16GB of memory, Dual SAS Disks
- The perfect "General Purpose Server"

Joyent

# Sun Fire X4500 Galaxy

- Max'ed Configuration: 4 2.6Ghz Operton Cores, 16 GB of memory
- 48 500GB SATA Disks, Single Pool
- RAIDZ2, yielding 18TB usable

Joyent

# Why Galaxy?

- Lights Out Management superior to other major vendors (ALOM/ILOM)
- Superior Design (Serviceability)
- Excellent OnBoard Networking (Quad GigaEther and Dedicated Management)
- Single Hardware/Software Vendor

Joyent

# Networking

- All systems use standardized network configuration:
  - e1000g0: Public Network
  - e1000g1: Private Network
  - e1000g2 & 3: Aggregated Storage Network (dladm: aggr1)

Joyent

# Operating System

- Standardized on OpenSolaris (ON_NV 43)
- All systems JumpStarted in standardized configurations for the 3 different models
- Migrating everything to ON_NV 56 when SX:CR Releases (Duckhorn)

Joyent

# Why Solaris?

- Seamless 32/64bit still not as simple as it seems on other platforms
- DTrace, SMF, FMA, et al... For us, for our customers.
- Solaris defines “Carrier Grade”
- Quite simply the deployment OS of choice
- Integrated virtualization: Zones and ZFS, perfect for multihosting

Joyent

# Why ON_NV?

- Solaris 10 has played catchup to the hype
- ZFS/Zone Integration
- Zone Manageability (zone move, zone clone)
- Zone Migratability (zone import/export)
- iSCSI Target

Joyent

# Zones & Joyent Connector

- T1000's used for mail (Well Threaded)
- X4100's used for PostgreSQL & Rails application (Poorly Threaded)
- Each "Server" is in a Container for flexibility

Joyent

# Parallel Deployment with Zones

- Developers are tired of development and staging environments different from production
- Solution: Containerize deployment and clone!  All 3 environments on the same system, resource protected by FSS.

Joyent

# Zones & Consolidation

- BingoDisk and StrongSpace applications deployed on a single Thumper, each in a container, everything is local.
- ZFS Compression... ‘nuff said.
- Single ZFS Pool (RAIDZ2) in GlobalZone accessed by both containers via “dataset”
  - zfspool/bingo
  - zfspool/strongspace

Joyent

# Zones & Hosting

- X4100's store OS on first SAS disk, ZFS Pool on second
- Master "Template Zone" is copied into ZFS filesystem
- Zone is then cloned multiple times, changing only IP's and passwords, to create customer containers
- ZFS clones reduce disk usage, Quotas enforce "zone root" usage to 5GB, and allow for snapshots.

Joyent

# Thumper, ZFS, and NFS

- Shared storage (25GB) provided to each container via NFSv3 on ZFS
- Each customer gets a filesystem with quota, NFS security, and compression inherited
- Snapshot capability accessible to customer via .zfs/snapshots

Joyent

# ZFS & NFS: ZIL Kills

- ZFS Intent Log (ZIL) destroys NFS small file performance
- 10MB Tarfile can take 3 minutes to untar on NFSv3 (fully tuned) with ZIL enabled, although sequential access is 100MB/s or better
- Putting "set zfs:zil_disable=1) in /etc/system improved same untar to 5 seconds.

Joyent

# Why NFS Isn't Enough

- NFS is easy and accepted, but...
- Many OLTP databases have issues with NFS
- ZFS is under there, but hidden from the users control beyond snapshot access
- Users want access to ZFS from their containers
- Zones on NFS is a no-no.

Joyent

# iSCSI: The Future

- iSCSI allows:
  - Zones to be stored centrally, thereby abstracted from “the metal”. Can you say “HA”?
  - ZFS filesystems can be accessed within a zone as a dataset and managed by the user.

Joyent

# ZFS, Zones, iSCSI

- Virtualization is a double edged sword, consolidating 8 onto 1 means 8 die when 1 does. Maintenance is that more serious and costly.
- Solution: Each zone is created in a thinly provisioned ZVol on a Thumper, exported as an iSCSI Target, imported on a given system, and run.
- The New Problem: Managing hundreds of iSCSI Targets and ZVol's with containers in them from one box.

Joyent

# Resource Management

- Sun Container marketing is massively misleading
- Customers have been led to think:
  - They can “cap” CPU (“pay for idle”)
  - They can “cap” memory
  - They can do these from the global zone

Joyent

# RM Reality: Memory

- Memory can be “capped” by rcapd
- Properly defining limits by project, task or process can be confusing.
- rcapd runs and is managed within a zone
- The Problem: Customers can simply disable rcapd  (“The GridZones Problem.”)
- No protection for swap

Joyent

# RM Reality: CPU

- /etc/projects runs within the zone and is therefore useless.
- Existing RCTL's are insufficient
- No way to cap CPU usage
- FSS less effective when everything is equal
- FSS are NOT percentages!

Joyent

# A Partial RM Solution

- From the global zone we can use pools and psets, together with the FSS, to "balance" zones.
- This provides "burstability" but allows visibility ("Why is the load average 120?")
- Psets can limit this effect slightly but is undesirable (ie: on 1/8th server, bind a pair of zones to a single core)
- Without a ceiling to hit, upsell is almost impossible. 1/8th and 1/32nd aren't really any different.

Joyent

# Enter DuckHorn

- Project DuckHorn will provide:
  - New rctl: cpu-cap
  - rcapd running in globalzone
  - Management of RM from ‘zonecfg’ itself
  - Clear distinction between “hard” and “soft” resource management
  - ...and more.

Joyent

# Projects We're Watching Closely

- Duckhorn
- iSCSI Target & iSNS Projects
- Enhanced SMF Profiles
- Crossbow (Network Virtualization)
- Duckwater (Simplified Name Services)

Joyent

# The Sad Truth...

- Solaris is really powerful... but more complex than we admit
  - RM is very complex to both understand and implement
  - Security (RBAC, Auditing, etc) is non-intuitive and complex
  - As we simplify underlying components (ZFS) we open ourselves to larger problems of management (Thousands of FS)

# ZFS Gotcha's

- ZIL can kill small file performance
- Large numbers of filesystems can cause long boot times (reportedly fixed)
- 'zfs destroy' is evil. Typo's could result in killing a pool.
- ZFS Replication (send/receive) aren't recursive
- Snapshot management isn't integrated

Joyent

# ZFS as a Backup Solution

- Create a ZFS Pool with Compression Enabled
- Create a filesystem for each “job” (System)
- Rsync to the ZFS Filesystem
- Snapshot it nightly
- Incr Rsync nightly and destroy old snapshots as they age.

Joyent

# The Upgrade Problem

- Upgradability of Zones is a real concern
- iSCSI abstraction may help limit this, but to what extent is yet to be seen
- Uninstalling Zones simply isn't an option.
- LiveUpgrade doesn't support Zones (yet)

Joyent

# Questions

Joyent

# Contact Info:

- Joyent: www.joyent.com
- Joyent Blog: www.joyeur.com
- Cuddletech: www.cuddletech.com
- benr@joyent.com

Joyent