



CO-PRESENTED BY O'REILLY &



Exploring Virtual Clusters for Rails Development and Deployment

**Bradley Taylor
Rails Machine, LLC**



About Me

- Owner, Rails Machine
- Software Developer
- Savannah, GA



Scope of Session

- Overview of virtualization
- Rails considerations
- Examples
- Case Studies



Honorable Mentions

- Server hardware
- Hardware load balancers
- SAN
- Cluster file systems



Virtualization

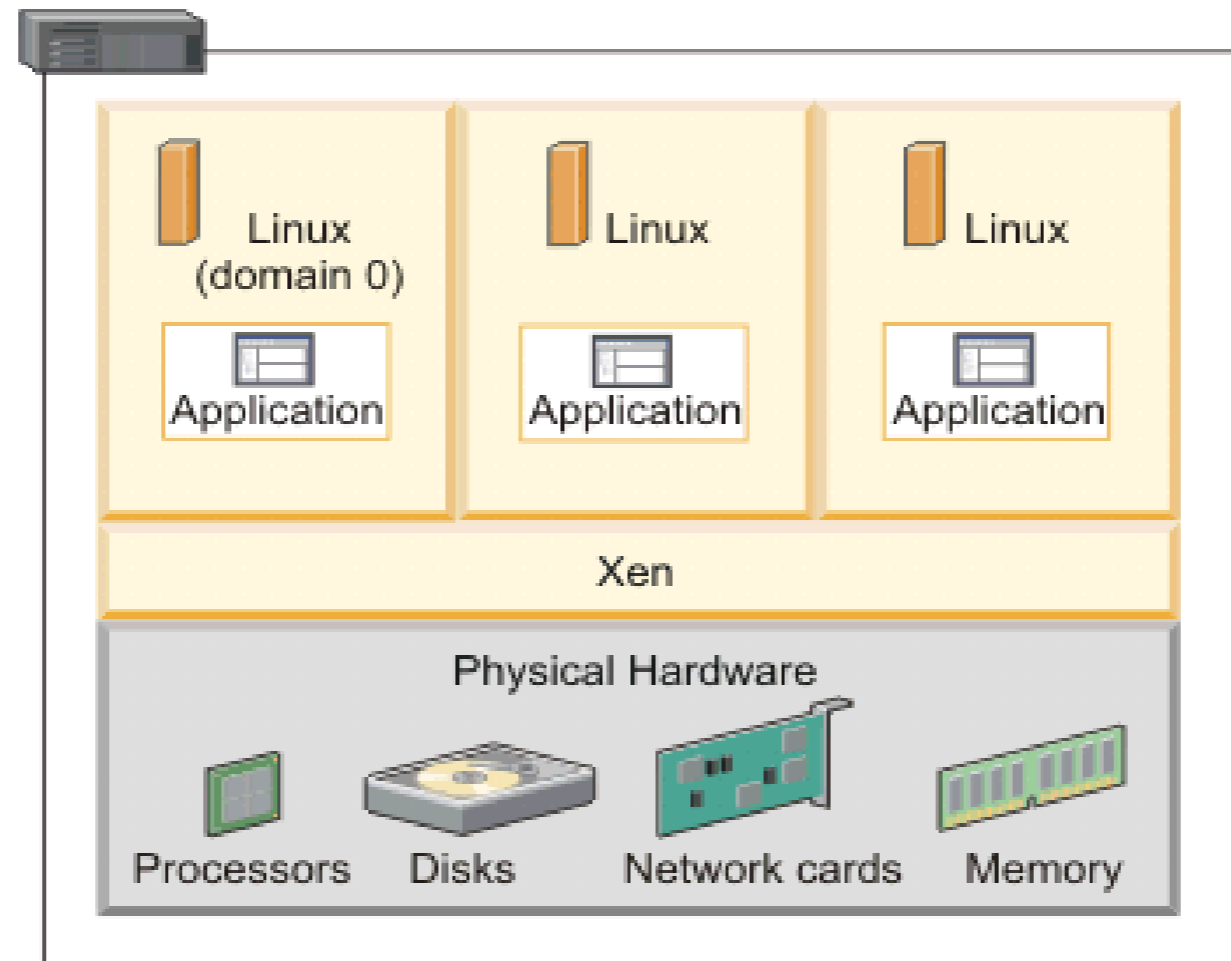
- Partitioning physical hardware
- Multiple servers
- Behaves the same



RAILSCONF

Technologies

- Xen
- Virtuozzo
- VMWare
- Solaris Containers (Zones)
- KVM



Borrowed from IBM

EICAZ516-0



Virtual Machine

- Behaves like a physical server
- Operating system
- Many per physical server
- Apartment complex
- AKA: vps, slice, container, accelerator, vm



Shared Hardware

- Processor (multi-core)
- Memory
- Storage (RAID)
- Network



Dedicated Resources

- Guaranteed memory
- Guaranteed storage
- Guaranteed CPU usage



RAILS CONF

Why

- Consolidate servers
- Isolate applications
- Replicate configurations
- Utilize all available hardware
- Tune resource allocation



Consolidation

- Less hardware
- Less data center infrastructure
- 3x 2GB servers -> 1x 8GB server
- Save money



Isolation

- Protect applications from each other
- Different operating systems
- Different stacks for different apps
- Avoid problems of shared hosting
- Contain intrusion



Replication

- Easily create from images
- Standardized deployment
- Automation



RAILS CONF

Utilization

- Fully utilize CPU
- Reclaim unused memory
- Reclaim unused storage
- Nooks and crannies



Resource Allocation

- Give a server exactly what it requires
- Grow/shrink “on demand”
- “Load balance” servers on hardware
- Guaranteed vs. burstable



Some Numbers

- 40+ host servers -> 400+ virtual machines
- 4 data center cabinets
- 256MB-2GB Memory
- 10GB-100GB Storage
- 30-40 virtual per 16GB physical
- 6-14 virtual per 4GB physical
- Appliance images (Rails, db, php, etc)



RAILSCONF

No Going Back

- Powerful
- Flexible
- Low cost
- Rapid
- Security



Virtual Clusters

- Multiple virtual servers working together
- Shared infrastructure
- Application and services layer
- Servers are almost “free”
- Expand/contract as needed



Many Versus One

- More than one application
- More than one service
- More than one customer
- More than one version
- More than one technology



Gains for Rails

- Development environments (stages)
- Memory isolation
- Resources to support different traffic
- Protect against PHP/Java
- Accessible multi-server scaling



Deployment Tools

- Subversion – common configuration
- Capistrano – automate updates
- Rake – stack assembly



Examples

- Many designs
- Business driven
- Application driven



Production/Staging

- Two virtual servers
- Released version
- Development version
- Test changes
- Test stack upgrades
- User acceptance



Web-App/DB/Staging

- Three virtual servers
- Shared database
- Ready to scale
- Resource tuning



Mixed Languages

- One virtual per technology: Rails, PHP, Java, etc
- Perfect stack
- Legacy support
- Shared services (db, mail, dns)
- No “dirty” feeling



High Availability

- Two or more physical servers
- Hardware redundancy
- Multiple application pairs
- Foundation
- Easier with shared storage
- Easier with load balancer



Multiple Applications

- One server per application
- Shared services
- Customer isolation
- Rapid deployment



Specialized Roles (Appliances)

- Each virtual has a job
- Standard: web, app, db
- Expanded: mail, ftp, dns, file
- Virtual data center: load balancer, firewall



Application Scaling

- Requires foundation
- More CPU
- More memory
- Parasite servers (nooks and crannies)



Case Study: EastMedia

- Multiple environments/customers
- Incubate customers
- Development (runs off trunk)
- Staging (runs off branch)
- Production (runs off release tag)
- Shared Subversion/Trac
- 2x 4GB physical → 8 virtual



Case Study: Boom Design Group

- Multiple stages, customers, applications
- Internal (development)
- Customer review (staging)
- Shared services (db, svn, monitoring)
- Low traffic apps (4 per server)
- Customer isolation (1 per server)
- 2GB physical memory -> 9 virtual (distributed)



Summary

- Application driven
- Service driven
- Customer/business driven
- Resource pools
- Infrastructure
- Less physical servers



Questions?



Contact

- railsmachine.com
- bradley@railsmachine.com