Highly Available Central Services
III
A Virtualization Approach

Thomas Finnern

DESY/IT

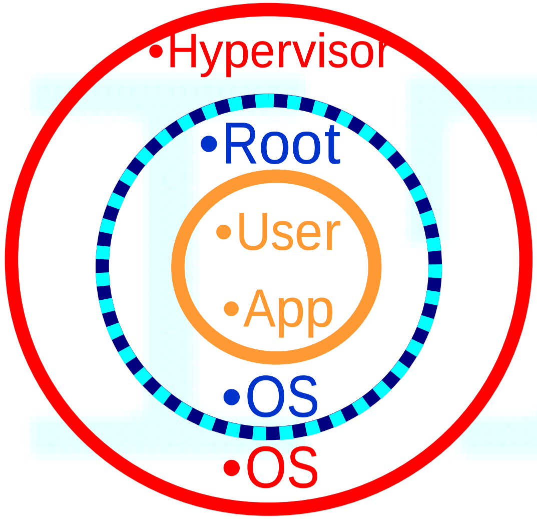
Overview

- Virtualization
- Why
- Features
- How
- Hypervisor
- Guests
- Use Cases
- Where to
- Conclusions

Virtualization

- ✶ **Definition**
  - ✪ Network Resources
  - ✪ Storage Resources
- ✶ **Used Virtualizations**
  - ✪ Clustering
    - ■ Sun Cluster
  - ✪ Content Based Routing
    - ■ f5 Layer 7 Routing
    - ■ Cisco ?
  - ✪ Host Virtualization
    - ■ Solaris Container
    - ■ XEN

Virtualization is the technique of managing and presenting storage devices and other resources functionally, regardless of their physical layout or location.



In computing, paravirtualization is a virtualization technique that presents the abstraction of virtual machines with a software interface that is similar but not identical to that of the underlying hardware.

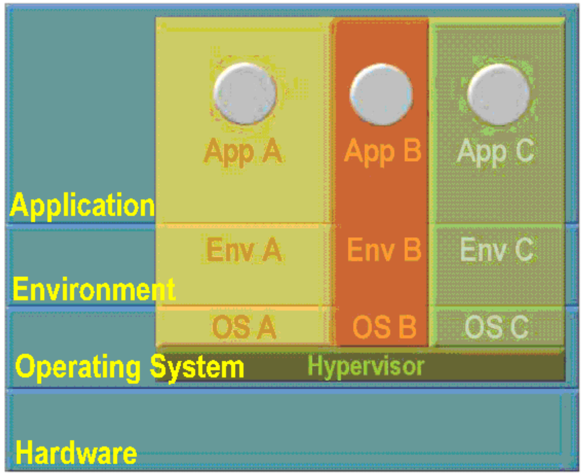
Why

- ✶ **Minimize Efforts**
  - ✪ Easy Provisioning
  - ✪ Easy Resource Control
  - ✪ Multi-OS Service Offer
  - ✪ Security / Service Separation
- ✶ **Getting Better**
  - ✪ Enhanced Load Distribution
  - ✪ Enhancing Fault Tolerance and Security
  - ✪ Separate Test, Developing and Production on same Hardware

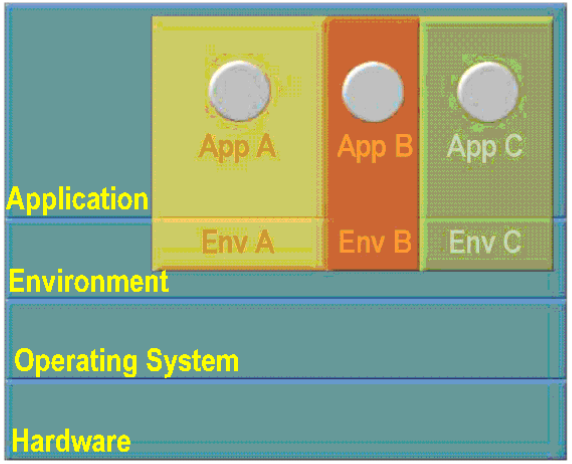
solaris opensolaris redhat suse ULTRASPARC AMD 64 Opteron

How

- **RedHat**
  - RHEL 4.4 with XEN 3 Hypervisor
  - Plan: RHEL 4.5+, 5.0+ with Built In Support
  - Logical Volume Manager
  - Hypervisor and Guests
  - 2 8-Core-4G-X4200 + 1 PC



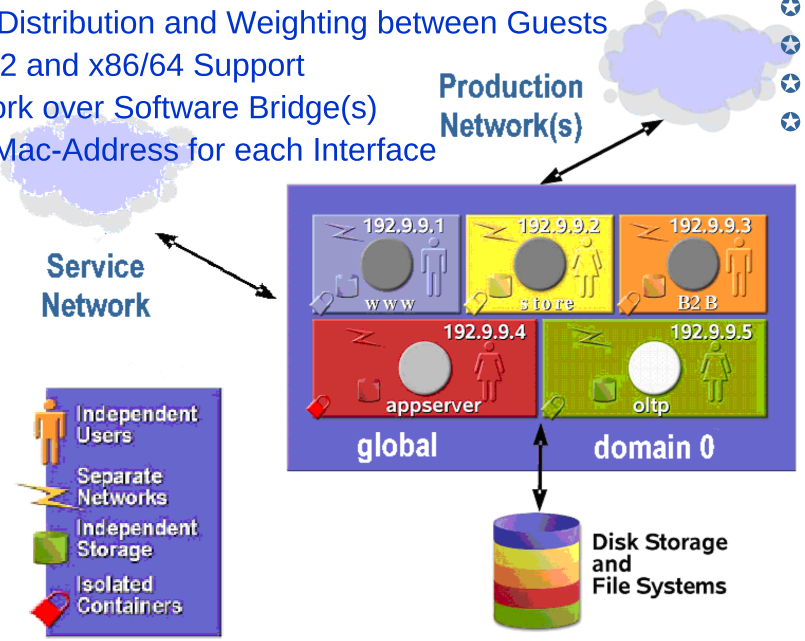
- **Sun**
  - Solaris 10 Zones
  - Plan: (Open)Solaris 11 with XEN
  - ZFS (Solaris 10 U3+)
  - Zone + SRM = Container
  - 1 8-Core-4G-X4200 + other

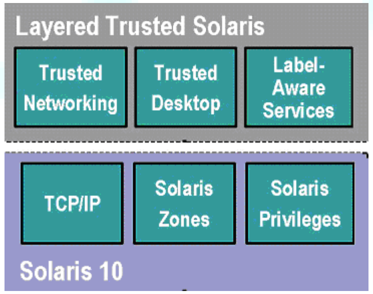


Key Features and Benefits

- **XEN:**
  - Open Source
  - Complete Separation between Guests
  - Paravirtualization
  - Intel-Vanderpool and AMD-Pacifica Supported for unmodified Guests (without Live Migration)
  - Virtual Machines near Native Speed
  - Live Migration of Guests
  - Up to 32 virtual CPU's per Guest
  - Load Distribution and Weighting between Guests
  - X86/32 and x86/64 Support
  - Network over Software Bridge(s)
  - Own Mac-Address for each Interface

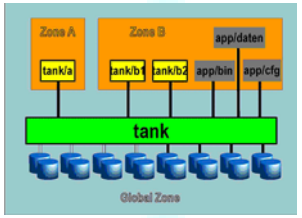
- **Solaris Container:**
  - Open Solaris
  - Separation (and Shares) configurable
  - One OS for All
  - Sparc and x86
  - Guests Machines at Native Speed
  - No Live Migration (?, cloning planned)
  - Cores per Zone not limited
  - Resource Management between Containers
  - Sparc and x86 with 32/64 bit Support
  - Network over selected Device(s)
  - No own Mac-Address for each Device (?)
  - Sparse-root and Whole-root Models





Hypervisor / Global Zone

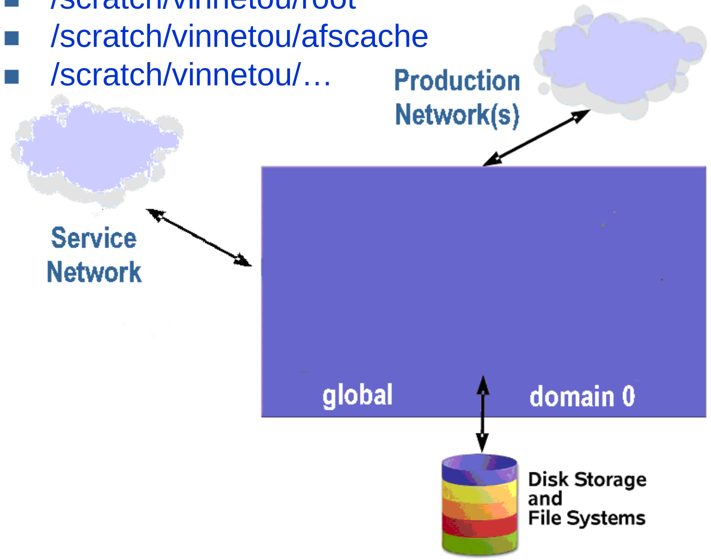
- **XEN Test:**
  - Defined one admin domain cross platform
  - Configuration: Add feature vms
  - Reconfiguration: New Install / Reboot
  - Hypervisor with Minimum Installation
  - Logical Volume Manager for FS-Handling
  - Separate Network for Service and Production
  - Guest Installations from Scratch
  - Unvisible Filesystems
    - /scratch/vsges/root
    - /scratch/vsges/sge
    - /scratch/vsges/…
  - Changing to Image Preparation



- **Customisation**
  - Image Preparation
    - Partitioning
    - OS and Data
    - Application
    - Networking
  - Deployment Methods
    - mk_image
    - mk_virtual
  - Live Cycle Management
    - As Usual …

- **Methods**
  - mk_image
    - Install File or
    - Clone/Copy
    - Pack
  - mk_virtual
    - Unpack
    - Networking
    - Partitioning
    - Glue

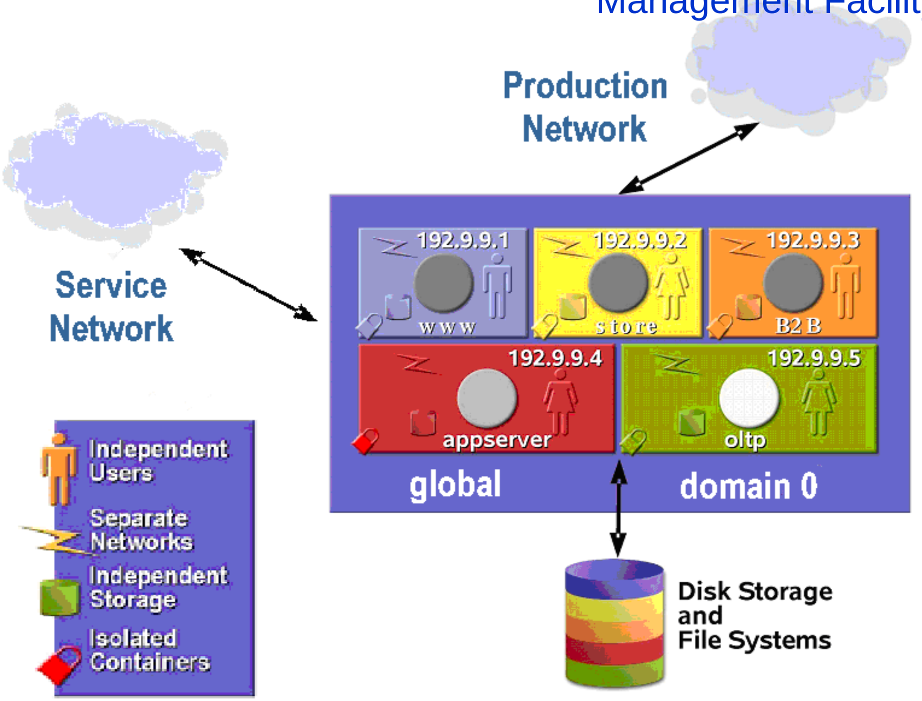
- **Solaris Container Test:**
  - Defined one admin domain cross platform
  - Configuration: Add feature vms
  - Reconfiguration: On the fly
  - Global Zone Installation will/can be inherited
  - ZFS for FS-Handling
  - Separate Network for Service and Production
  - Guest Installations from Scratch
  - Visible Filesysstems
    - /scratch/vinnetou/root
    - /scratch/vinnetou/afscache
    - /scratch/vinnetou/…



Guests / Zones and Pools

- **XEN Test:**
  - Configuration: Add feature vms-<SERVER>
  - Openafs kernel module must be compiled
  - XEN-Mac 0x00FFFFFF&IP|0x02000000

- **Solaris Container Test:**
  - Configuration: Add feature vms-<SERVER>
  - afs can be inherited (ro ?)
  - Next version: privilege grant for kernel module usage
  - Slight differences in SVC (Service Management Facility) for own sshd startup



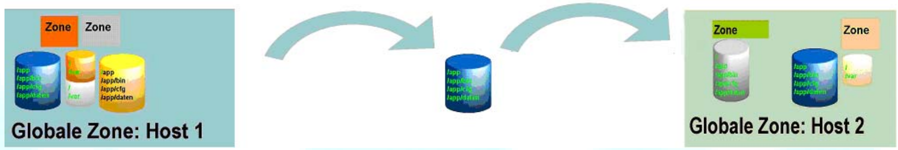
Expected Use Cases

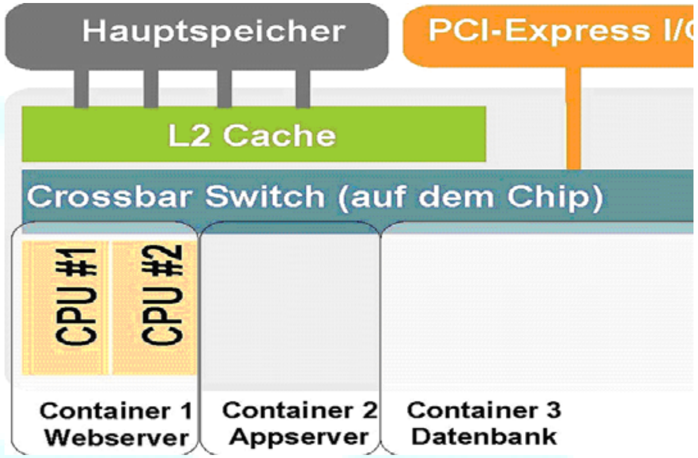
- **Separation of Applications and Services**
- **Providing Service IP Addresses**
- **Application Specific User Registry and System Settings**
- **License Cost Optimisation**
  - e.g. Oracle “Capped Containers”
- **Application Specific Resource Binding**
  - Storage
  - Network
  - CPU
  - Peak Load Elimination
- **Multi OS Offer for Customers**
  - Test Environments
  - Developing Environments
  - Migration Support
  - Multi Customer Pools e.g. SGE

- **Automatic Deployment of e.g. WEB-Services**
- **Educational Environments for Admins and Students**
- **Database Factory**
- **Security by Encapsulation**
- **New Service Environments**
  - Order and Delivery within 1 Hour
  - Central Image Handling over SAN
- **Ease Computer Adminstration**
  - New Operating Concepts
    - Installation
    - Testing
    - Monitoring
    - Backup
    - Patching
- **...**

Where to

- ✶ **Running Solaris and Linux on one Host**
- ✶ **Running 100-Core Hosts over Internal Crossbar as Farm or WEB Appliance**
- ✶ **Cross Cluster Hopping**
  - ✪ Live Migration
  - ✪ Image Moving

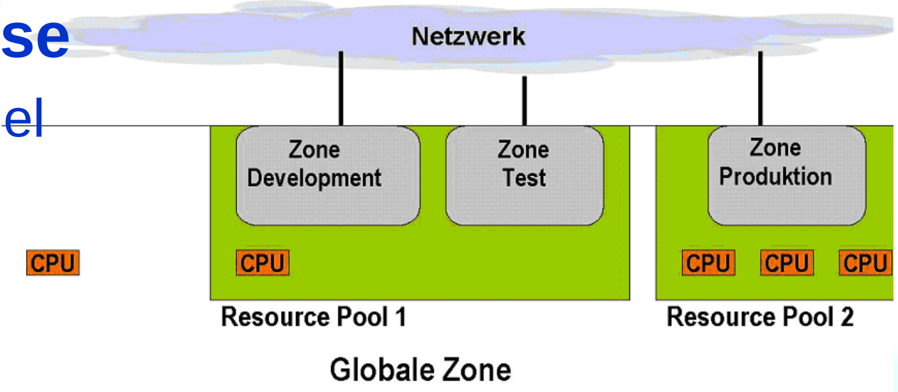




- ✶ **First Production Tests**
  - ✪ Planned for 2007

Conclusions

- **Virtualization Is Simple To Use**
  - Nice Operating and Support Model
  - Administrate only Big Boxes
  - Cross Platform Support



- **Future**
  - Real or Para Virtualization will be standard
  - XEN Hypervisor for Solaris and vice versa
  - Standard provisioning frame (kernel, partition, image, data, …)
- **Security**
  - Don’t let your system be virtualized by somebody else …