



Creating a Virtual Computing Lab

- Stephen Rondeau
- Institute of Technology
 - UW Tacoma
 - 27 Jan 2014

Agenda

- Origins of the Virtual Computing Lab (VCL)
- Rationale for a VCL
- VCL Architecture
- Experience with VCL
- Problems
- Future Directions
- Links
- Addenda

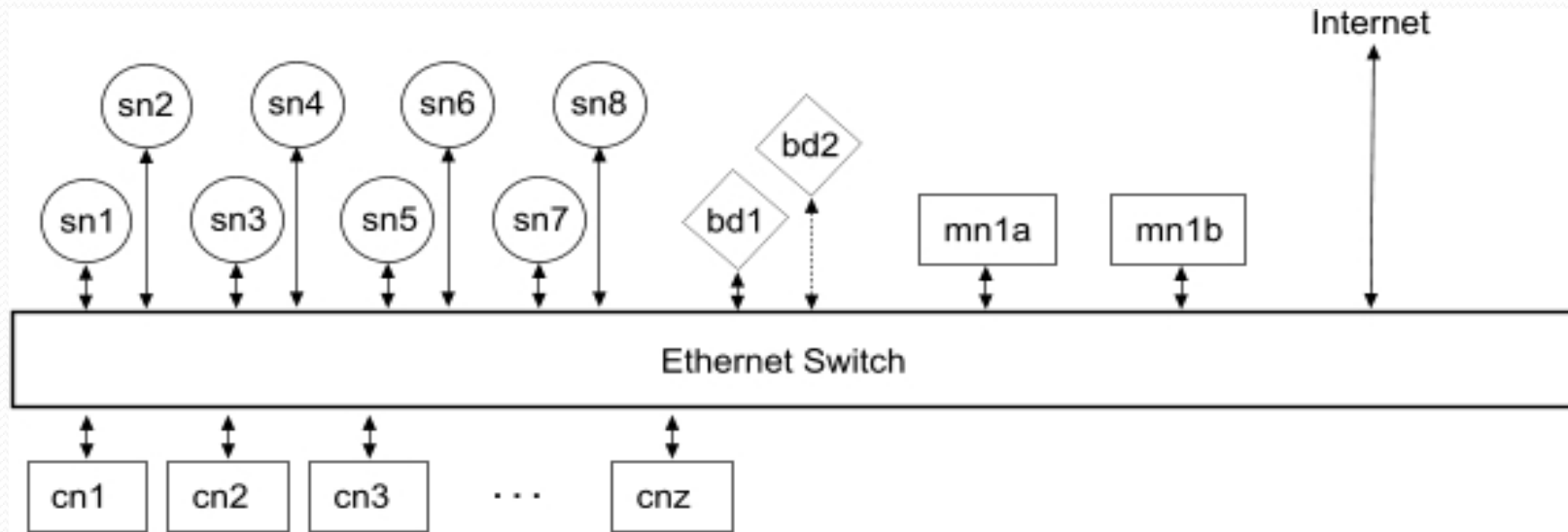
Origins of the VCL

- 2004 to present, NCSU/IBM project
 - Provide remote access to variety of computing resources, for teaching and research
- A “cloud” before the term was popularized
 - In 2006, by Amazon
- Infrastructure as a Service (IaaS)
 - Dedicated virtual or real machines
 - Only dedicated for a set time interval
- Open-sourced in 2008
 - Apache VCL (<http://vcl.apache.org/>)

Rationale for a VCL

- Provide remote access to restricted software
- Allow remote control of dedicated machines
- Utilize pool of available computers (Hadoop)
- Homogenize environment across diverse devices
- Extend classroom/lab computer capabilities
- Reduce lab workstation upgrade costs
- Support CS/IT distance-learning curricula

VCL Architecture



- Compute nodes: cn1..cnz ($z \leq 30$)
- Storage nodes: sn1..sny ($y \leq 12$)
- Backup disks: bd1..bd2
- Management nodes: mn1a/b
- Ethernet switch: 48-port Gb

VCL Compute Node Choices

- Many small hosts (4 cores, <10GB RAM, etc.):
 - **advantage:** performance
 - **disadvantages:**
 - physical environment
 - many IP addresses
 - high cost per core
- Few big hosts (scores of cores, >10GB RAM)
 - **advantages:**
 - physical environment
 - few IP addresses
 - low cost per core
 - **disadvantage:** performance

VCL Compute Node Specs

- CPU
 - 64-bit with virtualization support
 - At least 4 cores per CPU (our spec: 4&16 cores/CPU)
- Minimum 1GB RAM per core (our spec: 2GB/core)
- 50GB for host OS plus minimum 2GB per core
 - Our spec: 20GB/core, mainly due to Windows 7/2008 guests
- One NIC per CPU
 - New idea -- unsure if this is correct
 - Apache VCL says 2 NIC min.
 - Possibly more if using iSCSI to access virtual disks

VCL Control

- **VM administrator:** creates VM environment
 - User/team accounts
 - VM definitions and usually guest OSes
 - Virtual network (if needed)
- **User/Team**
 - Uses pre-defined VM(s) and network
 - Interacts with remote display of VM(s)
 - Controls VM start/poweroff
 - Can boot from pre-defined ISO
 - Optionally install to predefined virtual disk

VCL Control: User View



Experience with the VCL

- Too complicated without manage_vc GUI
 - Cryptic generated passwords
 - Long strings of IP:port numbers
 - Unclear information file
 - Too little control over VMs for Sys Admin class
- Statistics as of Aut2013
 - 442 student, 8 faculty/researcher touches (repeats)
 - 16 full courses (not internships, proj. or indep. study)
 - Several projects/internships/research
- Moodle LMS moved to VCL in Aut2013

Experience with the VCL (cont'd)

- Heavy use in computer/network security
 - Requirement: network isolation and control
 - Network control should be in instructor's hands
 - Cut off internet access after students download/upgrade
 - Must not serve dynamic IP addresses to real subnet
 - Led to:
 - Unsuccessful testing of OpenVswitch
 - Could not distribute VN over subnets
 - Successful use of Virtual Distributed Ethernet (VDE)
 - Overlay virtual network -- spans subnets
 - Works well with VirtualBox
 - Investigation and use of Vyatta VM
 - Router/firewall/NAT/DHCP

Problems

- Not enough public IPv4 addresses in subnet
- Too little disk space on cns
- Performance on 64-core hosts
 - Network and disk
- Big Data researchers' needs
 - Consume 32/64GB of RAM per VM
- Adequate power and ventilation
- 2 or 3 VMs per student quickly consumes cores
- Custom code
- No reservation system to keep hardware busy

Future Directions

- Buy more hosting hardware
 - At capacity for Winter 2014 due to multiple VMs per student
- Provide better/easier virtual networking
- Allow variety of VMMs
 - VMware may be too expensive
 - Hyper-V possible
- Improve performance
- Move to Apache VCL
 - Alternatively, an open cloud solution

Links

- [NCSU VCL](https://vcl.ncsu.edu/)
 - <https://vcl.ncsu.edu/>
- [Apache VCL](http://vcl.apache.org/)
 - <http://vcl.apache.org/>
- Institute of Technology VCL
 - [Design](http://css.insttech.washington.edu/~lab/plans/vcl/design.html)
 - <http://css.insttech.washington.edu/~lab/plans/vcl/design.html>
 - [Implementation](http://css.insttech.washington.edu/~lab/plans/vcl/implementation/)
 - <http://css.insttech.washington.edu/~lab/plans/vcl/implementation/>
 - [User's Quickstart Guide](http://css.insttech.washington.edu/~lab/Support/HowtoUse/UsingVCLQS.html)
 - <http://css.insttech.washington.edu/~lab/Support/HowtoUse/UsingVCLQS.html>
 - [Virtual Networks in the VCL](http://css.insttech.washington.edu/~lab/Support/HowtoUse/UsingVCLVirtualNetworks.html)
 - <http://css.insttech.washington.edu/~lab/Support/HowtoUse/UsingVCLVirtualNetworks.html>
- See also: links to technologies used/investigated in slides

VCL Control: Administrator

- Creates VC environment:
 - Creates VM(s) with guest OS installed as template
 - Creates VMs for class, person, or team
 - Based on VM definition and disk template
 - Creates user/team accounts, writes VCL info file
 - Gathers VCL info and distributes to users/teams
 - Usually starts all VMs
 - Can also:
 - Modify VMs/virtual networks (VNs)
 - Control VMs (e.g., poweroff/save state) and VNs
 - Delete VMs

VCL Control: User/Team

- Manages own VMs (VCs) and VNs
 - Windows users use GUI: **manage_vc** app
 - Requires Putty (pscp/plink) & Remote Desktop Client (RDC)
 - Compiled AutoITScript program, downloaded from web
 - Retrieves and parses VCL info associated with user/team
 - Connection to display: RDC
 - Control of VC(s): via predefined GUI interaction or commands
 - Mac OS/X or Linux users use **CLI**
 - Must retrieve VCL info via scp and parse it visually
 - Connection to display
 - Mac: CoRD or Microsoft RDC; Linux: rdesktop
 - Control of VC(s) via restricted ssh commands sent to host

VCL Components

- One switch per VCL (cost and infrastructure issue)
- **Compute nodes (cns):**
 - Host “virtual computers” (Vcs): VMs or bare-metal
 - Host OS: Fedora 19 x86_64
 - for most features/latest fixes
 - Virtual Machine Manager (VMM): [VirtualBox](#) 4.3+
 - Type II VMM runs inside host OS vs. Type I (bare metal)
 - Typical VM:
 - 1 virtual CPU on one real core
 - 1GB virtual RAM
 - 10GB virtual disk
 - 1 Gb NIC
 - Remote display via RDP

VCL Components (cont'd)

- **Storage nodes (sns):**
 - Original rationale
 - Store VMs when no longer in use, load them on demand
 - Original design:
 - Host OS: Fedora x86_64
 - Glusterfs for replication support
 - Nodes paired for redundancy
 - Clients (cns) write to either of pair of sns (round-robin DNS)
 - Sns replicate to each other
 - Read-only areas
 - ISO files, common configurations, shared virtual disks
 - Read-write areas
 - For VM definitions and virtual disks

VCL Components (cont'd)

- **Backup disks** (for r/w section of storage cloud):
 - IOCellNetworks NDAS (originally, Ximeta)
 - “Network Direct Access Storage”
 - Proprietary protocol on top of Ethernet
 - One active and one archived, swapped weekly
- **Management nodes (mns):**
 - Host OS: Fedora x86_64
 - Reservation system (VCL code, mysqld, httpd)
 - Name server for VCL cns/sns
 - Redundant pair of nodes (cold failover)

The InstTech VCLs

- 2009: proposed to Institute of Tech. faculty
 - Convinced WA state to allow us to fund
 - Purchased Super Micro Computer system:
 - 10-blade chassis: SuperBlade SBE-710E
 - Each blade (SBA-7121M-T1): "compute node" or "cn"
 - 2 CPU * 4 cores per CPU, energy-efficient
 - 16GB RAM
 - 150GB 10K RPM SATA drive
 - 2 NICs
 - Remote KVM (IPMI) and virtual media
 - Pass-through network switch
- Added 8-node storage "cloud"

The InstTech VCLs (cont'd)

- 2010: added Qty 20 4-core Dell T3400s
 - No IPMI
 - 8-node storage cloud
 - Decommissioned in 2013
- 2012: added Qty 30 4-core Dell Optiplex 990s
 - Intel AMT for IPMI
 - 12-node storage cloud
- 2013: added Qty 5 64-core Dell PE R815s
 - IDRAC Enterprise for IPMI
 - No storage cloud