Hardware / Virtualization / Architectures

Foundations of the cloud
Implementing Virtualization Technologies

- Cloud Computing relies heavily on the concept of virtualization
  - In fact not possible without it.
  - But they are separate concepts
  - You can have virtualization without doing cloud computing

- Virtualization means to convert physical hosts to logical/virtual ones. This is not really something new. IBM did this from the 60’s on with their "VM" operating system ( /370).
Cont...

- 2 Practical types of virtualization
  - Desktop Level
    - Places virtual machine as a process on top of the existing OS.
    - Great for research or testing of multiple items
    - Base operating system has no idea that there is another operating system.
    - Software abstracts and brokers all the system calls
  - Server Level
VMware Desktop Virtualization on PC
Oracle Virtual Box on Mac OSX
Server Level Virtualization

- Why call it that?
  - We use the entire system

- “To make things more effective than the user space model (mentioned in part 2) we can use a **Hypervisor**. The Hypervisor is a small piece of software which controls the hardware - it assigns memory, CPU cycles, PCI hardware. This hypervisor replaces the kernel of a traditional, "real" hardware system:”
How Hypervisor works

- X86 architecture
  - Most common and cheapest
  - Ring structure in x86 architecture
  - User process runs in Ring 3
  - Kernel or heart of OS Runs in Ring 1
- Hypervisor inserts itself between kernel and hardware
  - Ring -1
  - Separation prevents from Ring 3 causing entire system to crash by making sure system calls pass through proper channels
Hyper-V Architecture

Parent Partition
- WMI Provider
- Applications
- VM Worker Processes

Child Partitions
- Applications
- Applications
- Applications

Provided by:
- ISV / IHV / OEM
- OS
- Microsoft Hyper-V
- Microsoft / Citrix (XenSource)

User Mode
- Ring 3

Kernel Mode
- Ring 0
- Ring - 1

Windows Hypervisor
- "Designed for Windows" Server Hardware

Windows Server 2008
- Windows Kernel
- VSP
- IHV Drivers
- VMBus

Supported Windows OS
- Windows Kernel
- VSC
- VMBus

Non-Hypervisor Aware OS
- Emulation

Xen-Enabled Linux Kernel
- Linux VSC
- VMBus
- Hypercall Adapter
XEN – Open source Linux
KVM – In Linux Kernel
Hypervisors

- Just one part - need some management technology on top of it to utilize fully
- Three types of virtualization available
  - Paravirtualization
    - No hardware emulation, no need for VT-X and AMD-X CPU extensions
    - But also no Windows on Xen
    - Kernel of operating system that runs XEN needs to be modified
  - Full System Virtualization ← This is the one most used currently
    - Uses HVM (Hardware Virtualization Mode)
    - No need to modify underlying Kernel – CPU does all the translating
  - OS Level virtualization
    - FreeBSD Jail and Solaris Zones (give you access to multiple copies of the same OS per operating system)
Non x86 Hypervisor

- IBM has their Z/VM operating system and virtualization platform
  - Allows users to run multiple instances of Linux OS complies for the IBM zOS.
Xen

- Starts as part of the Kernel OS
  - Each virtualized OS has its own domain (called DomU)
  - All succeeding Virtual Machines have to communicate with the hardware through Dom0
  - Dom0 goes through the Hypervisor to the hardware.
  - Dom0 is only thing that has hardware access.
KVM

- Hypervisor uses AMD-V and Intel VT-x technology to change its architecture.
- KVM lives in the kernel as part of the standard Linux kernel as of 2.6.20 (if packages are enabled)
- Uses QEMU to do all the hardware emulation
Datacenter Tech

- Starts with your processors
  - Older processors do not have the on chip instruction sets to effectively do Hardware Emulation
  - All new processors do (enterprise level for sure)
  - Intel Xeon 5600 Nehalem class
    - Intelligent Power technology
      - Drop processor speed and memory speed to lessen power draw when needed – also can over clock when necessary as well.
    - VT-c – direct access to network for Hypervisor (if supported)
    - VT-d – direct access to storage for Hypervisor (if supported)
  - Intel Xeon 7500 Nehalem class
    - Internal self-diagnostics and self healing
    - Supports 16GB DDR3 dimms and 8 core chips
Virtualization pieces

- Virtualization is one piece.
- You need some management component on top of that.
- The idea behind cloud:
  - Utilize commodity hardware
  - Utilize lots of it to abstract your computing resources
- Chip based technologies help
  - You now have a large number of smaller systems instead of a few large systems
  - Power and cooling become a financial cost
  - Also Rip and Replace become the option instead of nurse and repair. (Cost vs. Time)
Datacenter changing with clouds entrance

- Previous data center was wild west of standards
  - Intel working with other groups and companies to produce standard for datacenter
  - [http://www.opendatacenteralliance.org/](http://www.opendatacenteralliance.org/)
- Currently datacenter built around the application, then OS, and then some hardware was bought to run it
  - Example Airline reservation system.
- Virtualization came in and helped reduce need for physical servers but the design principal is still largely the same.
- The problem: this datacenter design is not meant to handle spikes and sags (too expensive)
Cloud based datacenter

- Model of computing is changing
  - Datacenter used to serve internal clients or fixed number of outside clients
  - Clients are now more numerous and mobile
    - Data can surge and swell based on popularity
    - Small example: When singer Michael Jackson died Google was crushed under wave of people checking to see if it was true.
  - Cloud based datacenter allows for users to provision the resources they need—instead of begging an admin for some space or requisitioning a system.
  - Cloud based data center is User driven, User provisioned, and responding to mobile clients.
Networking in the cloud

• Data is abstracted
  • Two new concepts are utilized
  • iSCSI - [http://software.intel.com/file/31966](http://software.intel.com/file/31966)
    • Hard drive commands over TCP to connect to storage on the network
    • Excellent for attaching NAS, SANS
    • Cheaper than Fibre Channel
    • Requires internal private network – not on public network
  • Jumbo Frames
    • Data – larger frames let you pass more data in less processing time
  • Dell, HP, Cisco producing switches that handle Jumbo Frames and have the priority for iSCSI increased – designed for the cloud…
Ideal datacenter

• Similar to what Microsoft, Google, Amazon, and Rackspace are offering.
  • Two types of Cloud computing
    • Offer just a simple set of API’s
      • Amazon has S3 for storage, EBS for permanent storage, and SimpleDB
      • Rackspace has their competing open source version of all Amazon products
      • Google lets you access their Gmail and Picasa API
      • Microsoft gives you program access to their SQL server in their cloud
    • All abstracts the need for separate operating systems.
      • Now you are just offering services not complete operating systems and the user does not need to worry about compatibility.
      • No need to test and build on different hardware – all API based.
  • Really back to the mainframe concept
    • Now IT is focused on Access Control an not so much hardcore IT.
What really makes the cloud go?

- You need some kind of management software
  - to integrate with all your CPU tech and virtualization platforms
- Cloud platforms – open source
  - Eucalyptus
  - Ubuntu Enterprise Cloud (built on top of Eucalyptus)
  - OpenStack (Rackspace – Amazon competitor)
  - Nimbus
- All of these are Amazon service compatible?
  - Why?
Ubuntu Enterprise Cloud Example

[Diagram showing the structure of Ubuntu Enterprise Cloud with clusters, controllers, and storage controllers.]

https://help.ubuntu.com/community/UEC ← Installation link
Example Explained

- We need a server to interact with (user) Cloud Controller
  - No need for fast hard drives of new processors (can reuse older hardware)
- There is a Walrus controller
  - Can be accessed via API for someone to attach temporary storage to another project programmatically
  - No need to interact with our Cloud Controller
  - No need VT-x technology here just a lot of hard drive storage for people to use.
• The cloud controller talks to the Cluster controller
• The Cloud Controller allows us to provision virtual machines we have created previously (or downloaded)
• The actual instances are stored on our Node Controllers
• These use KVM to do the virtualization
  • They need VT-x based CPU’s
  • As much memory and CPU cores as they can get
  • KVM virtualization helps utilize every last drop of resources for multiple virtual machines on a single node controller
  • We can continually add more node controllers and expand our “Cloud”
  • -- hence the name could computing
Cluster Controller can also connect to Storage controller
  • Allows for EBS like permanent storage
  • Cloud advantage is that when finished with instance – resources are recycled
  • But what if you want to save some of the data?
  • Attach some EBS permanent storage

What if I use all of the computers I have available and still need more computing power?
  • All projects currently (UEC, Eucalyptus, OpenStack) are all Amazon compatible and you can add their exhaustive resources to yours or move yours into theirs.
  • Dominos Pizza does this one day of the year every year: which day?
Optimizations

- As we look at CPU and software virtualization where can we improve
  - Direct Access
  - Letting virtualized systems have direct hardware access to Hard Drives and Networking cuts down on overhead and increases throughput in some cases.
Not a Single Box

- Planning your cloud applications and what you will need to create to get them going
  - Images which will run your code
  - Storage mechanism to hold data
- These 2 things are not on the same box. You should not rely on operating system messaging or working with a local filesystem to manage how your software interacts with data.
Creating Images

- What consists of an image?
  - Operating System?
  - Application Servers (Oracle, Windows, Hadoop)?
  - Packages (Map reduce / Apache / Tomcat)?
  - What does your application need?
  - “Golden Images”

Remember, storage should be treated separately.
Amazon EC2 Image Tutorial

- We will log into Amazon Web Services
  - Username: valscarlata@gmail.com, password: itmisfun
  - Create an AMI (amazon machine image)
  - Launch the AMI
  - Stop the AMI
  - Attach EBS (Elastic Block Storage Device) to the AMI
  - Snapshot the EBS volumes
    - This is how you would backup your data / drives
    - You can also mount these drives in different AMI instances
How Can We Access the AMI?

- We need to configure / add packages to the instance after we have launched it.
- SSH into it
- Remote desktop into it (windows)
- Use Amazon developer tools to access it.
Links and sources

- http://www.opendatacenteralliance.org/
- http://www.capgemini.com/ctoblog/2011/01/the_new_data_centre_is_not_you/
(opening slide) // hypervisor pictures
Links and Sources Continued

- http://elite-itsolutions.co.uk/images/diag-hyperv-arch.png // Hyper-V picture
- http://open.eucalyptus.com/wiki/EucalyptusAdvanced_v2.0
- https://help.ubuntu.com/community/UEC/PackageInstallSeparate
- http://software.intel.com/file/31966