***Day 1:***

***Virtualization***

* Virtualization software can turn one machine into multiple machines with ease.
* We have this as a built-in feature in Windows 7 Professional and Windows Server 2008 R2.
* It is not a new concept; software engineers have been trying to achieve this using boot managers or boot loaders.
* Installation of Virtualization software is as easy as installing a software.
* The system on which we install the software would be host OS and OS present in Virtual Software would be guest OS.
* Benefits:
  + Consolidate Servers: Increase the scale of server infrastructure without having new hardware.
  + Conserve Energy: Less hardware, Less energy consumption.
  + Improve ease of management: Need to make changes then need to power off the systems, just change and start the virtual machine.
  + Reduce Backup and recovery time: Safe from hardware failure, less time for backup.
  + Test Software Configurations.
  + Maintain legacy applications: retain old servers which could be useful in future.
  + Support a Cross-platform office. (Restriction on some software to run mac virtually)
* Different Virtualization Software’s
  + VMWare (most popular)
  + Microsoft (Microsoft Virtual Server and Virtual PC, great when working with Windows guests and hosts, not viable with Linux/Mac operating systems)
  + Citrix
  + Parallels (First Commercial software to be able to run on mac)
  + Virtual Box (from Sun Microsystems, popular)

***What is Virtualization***

* Virtualization is a modified solution between centralized and decentralized deployments.
* This allows multiple OS to be run on a single machine, separating hardware from a single operating system.
* The Virtual Machine Monitor (Hypervisor) interacts between guest OS and hardware to control the guests use of CPU, memory and storage.
* This would reduce costs, as we could have a machines in a central hub (Central) and then in each machine (Decentralized) we could build the infrastructure required for the company.

***Why It Matters***

* At this point we know that virtualization separates the guest OS from hardware, which puts light on new tools.
* Using the virtualization, we could assign the resources to the OS so that it uses only the assigned amount of resources rather than using up all the resources.
* As guest is not bound to the hardware, we can move the it from one physical server to other physical server running the guest OS which changes the traditional thinking of server provisioning.
* While Virtualization deployment, as the CPU memory, and storage are handled by hypervisor the application can relocate to receive the required resources at that point.

***Approaches***

* Full Virtualization:
  + This is the basic form of virtualization, where you can provide fully emulated machine in which we could install any guest OS.
  + But in this type we have performance issues (may be trying to emulate a complete set of hardware in software)
* Single Kernel Image (SKI)
  + Host OS spawn’s additional copies of itself which could be found in Swsoft Virtuozzo and Sun Solaris.
  + We think this type as lightweight virtualization which solves performance issues with pure emulation, but at an expense of flexibility.
  + In this type, we cannot have other OS installed rather than having copies of same instance.
  + The point of concern here is security and reliability, if the host kernel is been breached then all the guest instances are compromised.
* ParaVirtualization
  + Found in Xen source, which tries to address the issues from the above two types.
  + In this type, the OS is slightly altered such that you gain access of hardware as the hypervisor does. This is hardware-assisted virtualization. We find improvement in performance.
  + For flexibility, the guest OS will not be tied to host OS. Different versions of OS can run parallel in a hypervisor.
  + This can be thought as a low-overhead full virtualization.

***Xen***

* This is the first virtualization solution which supports Intel’s VT technology which permits guest OS to run at it full processor potential with less overhead incurred by virtualization.
* Guest can be migrated from one machine to other machine in < 100 ms
* With Hypervisor, operators can continue to manage use of CPU memory, block, and I/o devices dynamically.

***Possibilities***

* Virtual LAN’s allows administrators to ignore about physical installations, hardware upgrades can be done seamlessly without the guest OS knowing that its host has been changed. Downtime can be reduced.
* This gives breathing space for administrator in getting in certified before deploying applications into new OS. Testing can be done on similar kind of instance as production.
* Virtualization is an elegant solution for many security issues. Where each systems needs firewall to be protected, multiple systems could reside on same physical machine.
* Using this the developers can have their individual sandbox environment for development so that they are protected from someone modifying the code.
* If virtualization is tied to system monitoring solution or provision & management tool like Red Hat Satellite server, systems could be migrated automatically with ease during the peak use or while maintenance.
* It would be great to see a farm of servers which can be re-tasked in seconds according to workload and time of day.

Xen is undisputed champion in open-source virtualization, many companies and organizations are involved. Red Hat is early adopter of Xen, an active contributor and incorporated in Fedora.