From Grids to Clouds

SPD Course

16/05/2011 (updated on 17/05)

Massimo Coppola

What are Grids

- Ideas, aims and shortcomings
- Computational Grids as the equivalent of power Grids
  - Easy to access (standards)
  - Ubiquous
  - Multiple providers
- "Remote" execution of applications
  - Infrastructure for AAA Access, Author. & Account
  - Need to move app & data
  - Need to find machines properly configured
  - Standards implemented within GRID middleware

What are Grids

- Initial providers & adopters were academic
  - HPC oriented (MPI – like) or client server (Web services)
  - Unclear business model due to adoption overhead
  - Change in paradigm (new API) and data migration
- Grid middlewares made up of
  - implementation layers (fabric, connection, data…), protocols and infrastructures (PKI)
- Full story within the CPA course

What are Clouds

- In a sense, the same idea
  - Ease of use of remote computing power, from multiple users, everywhere, with standard mechanisms
- What changed?
  - Virtualization
    - Push app-specific configuration toward the final user
    - Improve security and resource management
  - Large companies have lots of excess resources which CAN be provided (=sold) this way

What are Clouds

a 'cloud' is
an elastic execution environment of resources,
involving multiple stakeholders
and providing a metered service at multiple granularities
for specified level of quality (of service)

- What resources are provided as services? Multiple levels!
- Infrastructure as a service IaaS
- Platform as a service PaaS
- Software as a service SaaS

What are Clouds

- Quality of Service (QoS) specified formally and agreed between the user and the platform as a Service Level Agreement (SLA)

- Standards and mechanisms are needed to define, agree and enforce SLAs onto virtualized resources.

Types of Cloud

- Public Clouds
- Private Clouds
- Community Clouds
- Hybrid Clouds

- Cloudbursting

- Nowadays, public Cloud providers have complete control over their infrastructure

- Virtualization as a tool used to build the Cloud Abstraction.
- The topic is covered in depth in other courses
- e.g. the CPA course for what concerns system virtualization

RECAP ON VIRTUALIZATION

Virtualization

In short : Virtualization means providing the abstraction of a computing machine in order to execute programs, allowing to decouple the applications from the implementation of the underlying SW/HW

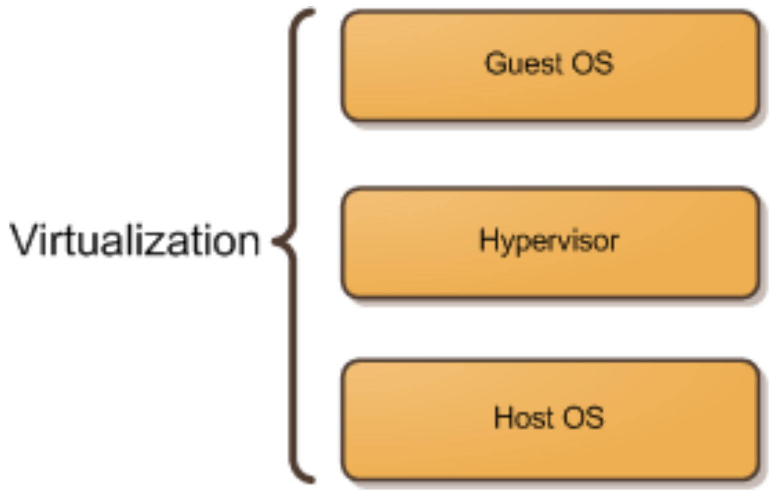
- Two main concepts for any virtual machine (VM)
  - VM interfaces: kind of code/features provided
  - VM implementation: design and techniques employed
- Kind and complexity of the machine abstraction provided vary, as well as its interfaces
- Approaches used to execute code within the virtual machine include code interpretation, on-the-fly compilation, hardware emulation
- The level at which the VM is implemented and its purpose also affect the design of VM support

Virtualization Examples

- System virtualization
  - Emulate a whole HW system (CPU, interfaces and devices, including work and mass memory) using a (possibly) different kind of system
  - You need to install a full software stack to use the VM, possibly starting from the BIOS
  - Techniques range from dynamic interpretation, on-the-fly compilation to unprivileged execution on the CPU with hardware-assisted emulation of devices
- Language-oriented virtual machines
  - Provide a simplified machine abstraction used to ease programming and improve portability of applications
  - Java and .Net frameworks as examples
  - Can provide properties unavailable/unpractical in HW
    - Security, code signing and code verification, code reflection and on-the-fly code modification
  - Again, approaches for execution include full interpretation (traditional JAVA) full compilation (P-code) and dynamic compilation (e.g. Hotspot Java, .Net)

System virtualization

- The Hypervisor (HV) implements the virtual machine emulation to run a Guest OS
- Provides resources and functionalities to the **Guest** OS
- Typical settings: the VM emulation stacks with the **Host** and the **Guest** OS

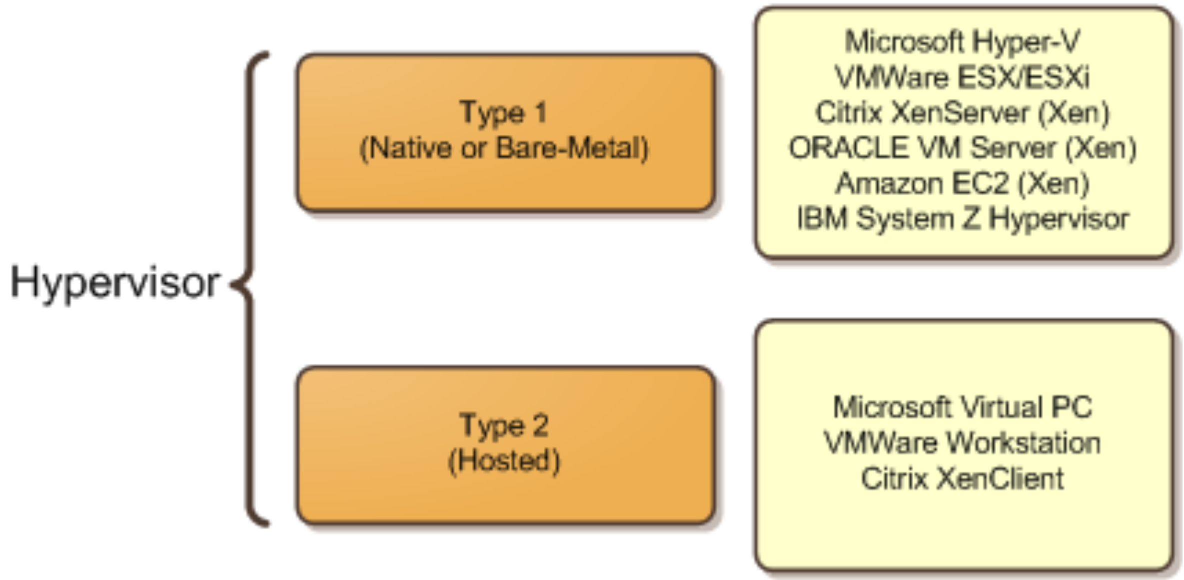


- System-level VM emulation = emulate each HW transition, including ISA, CPU and device user & privileged state
  - Different ISA → full emulation
  - Guest ISA = Host ISA → a subset of the ISA can be **executed**

ISTITUTO DI SCIENZA E TECNOLOGIE DELL'INFORMAZIONE "A. FAEDO"

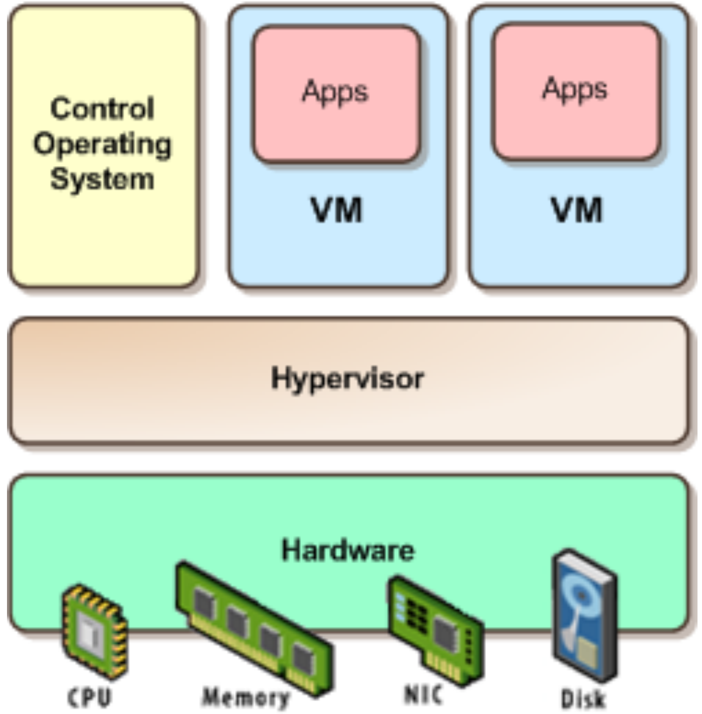
Different Kinds of Hypervisor

- In system virtualization, the Hypervisor is not necessarily run on top of a host O.S.
- Even with no Host O.S. , the HV will still need a **Control** O.S.
  - Create, monitor and manage other VMs



Type 1 Hypervisor : bare-metal

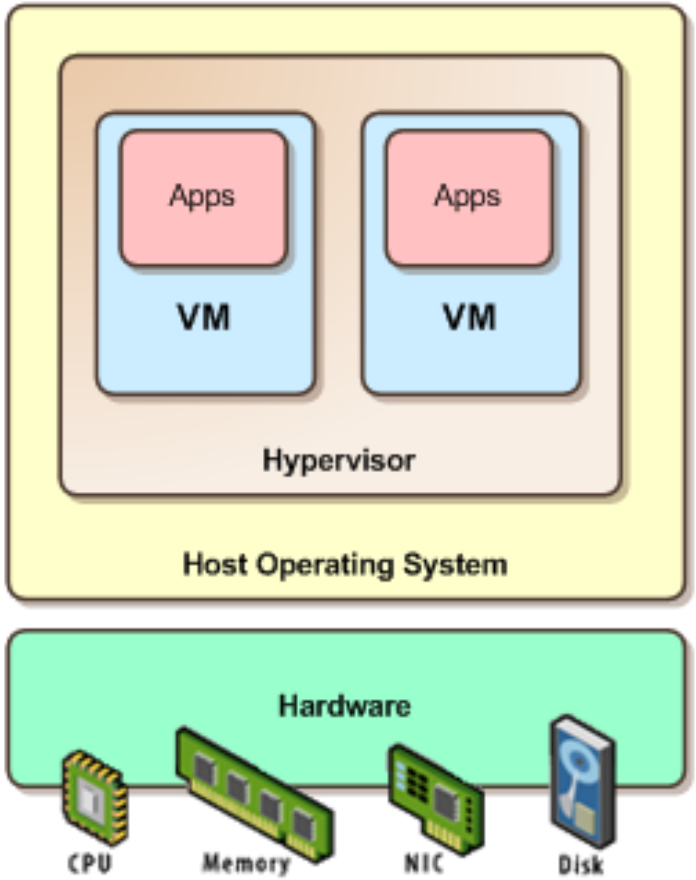
- HV runs directly on the hardware
- Essential management functions: memory, CPU, system bus
- "Control" instance of the OS is special VM
  - Privileged link to the HV to steer other VMs
  - *May* provide device drivers to Guests



- Pros: complete knowledge of HW, can make HW-aware decisions
- Cons: manage full VM emulation, intercept all HW mechanisms

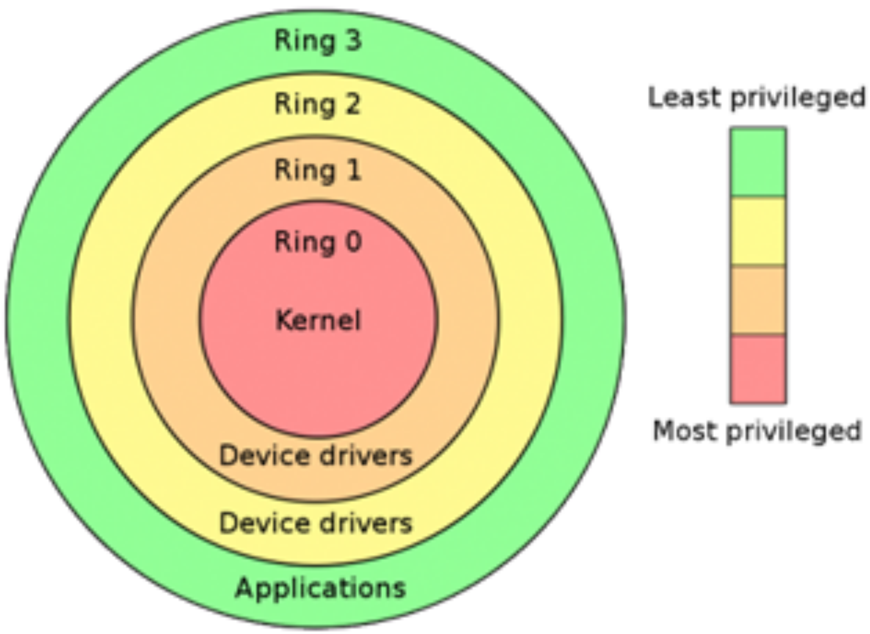
Type 2 hypervisor : Hosted HV

- HV runs within the Host OS
- Resources controlled and managed by the Host
- Need mechanisms to separate host and guests if same ISA



- Pros: less intrusive on Host OS
- Cons: low-level resource management done by host, HV has limited access to HW

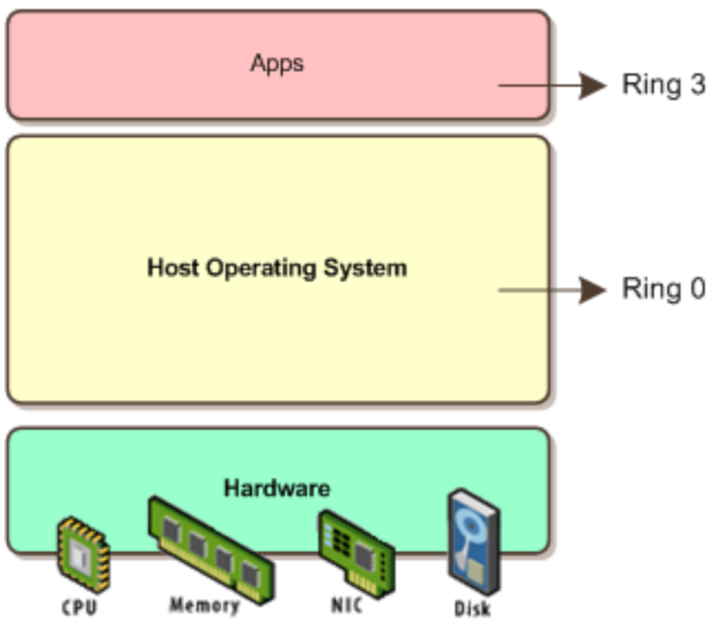
Protection rings

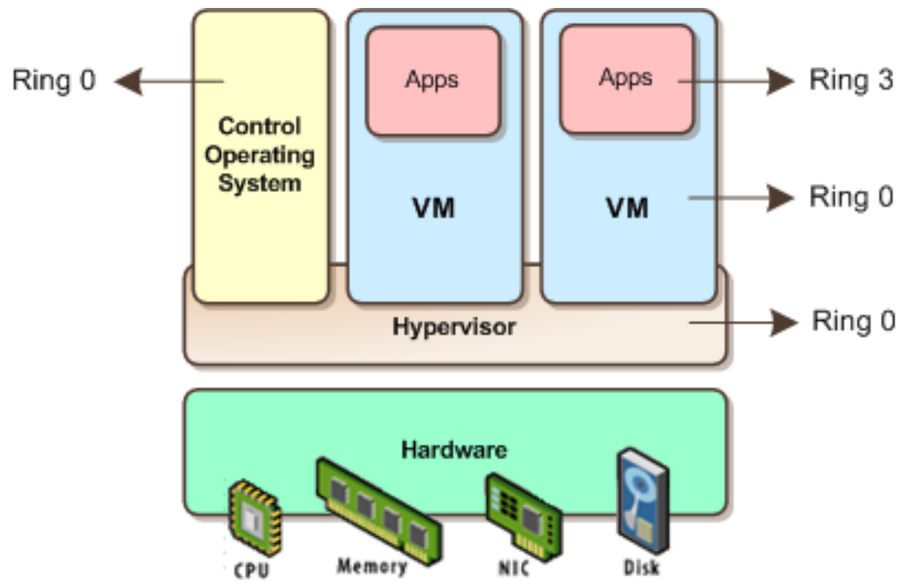


- Current X86 architecture
- A generalization of the basic mechanism of the “supervisor” state
- Different classes of machine instructions (subsets of the ISA) are allowed in different levels
- Can be used as an OS containment mechanism
- In practice, most O.S. kernels use just one ring for OS (then it must be ring 0) and one for apps (ring 3)

Paravirtualization

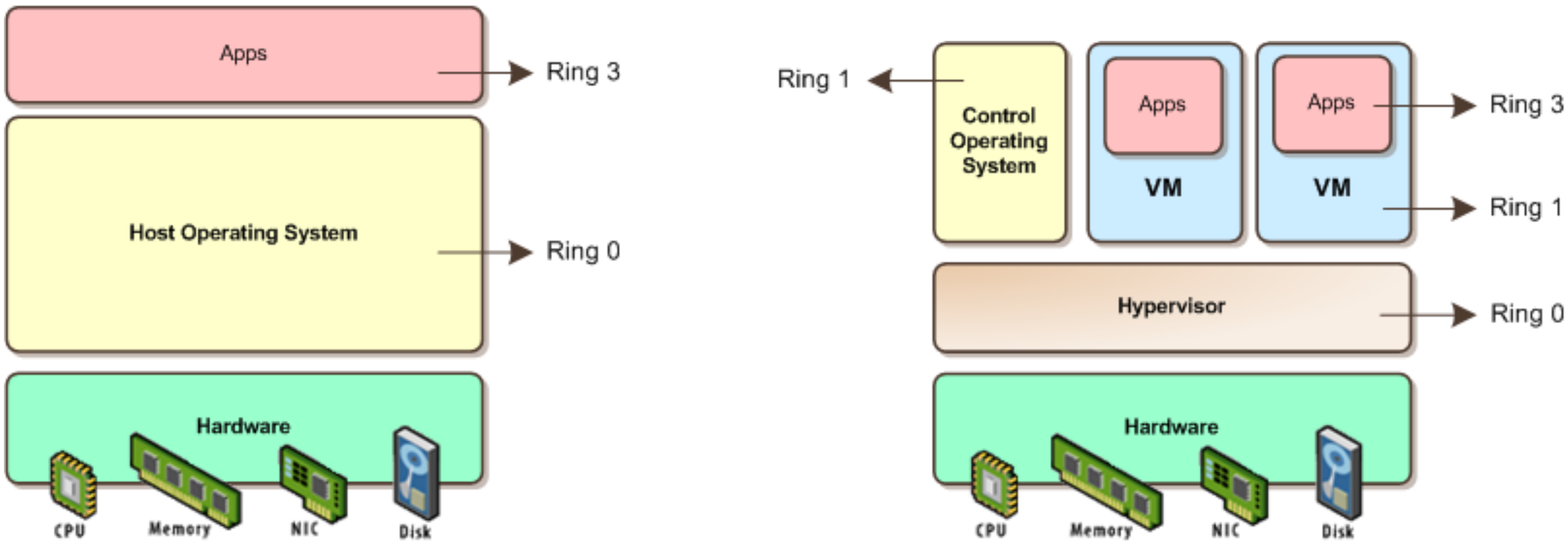
- Hypervisor needs to to run in ring 0
- Host O.S. kernel is modified
  - it runs in ring 0 on top of the hypervisor.
- Guest VM kernel is also modified





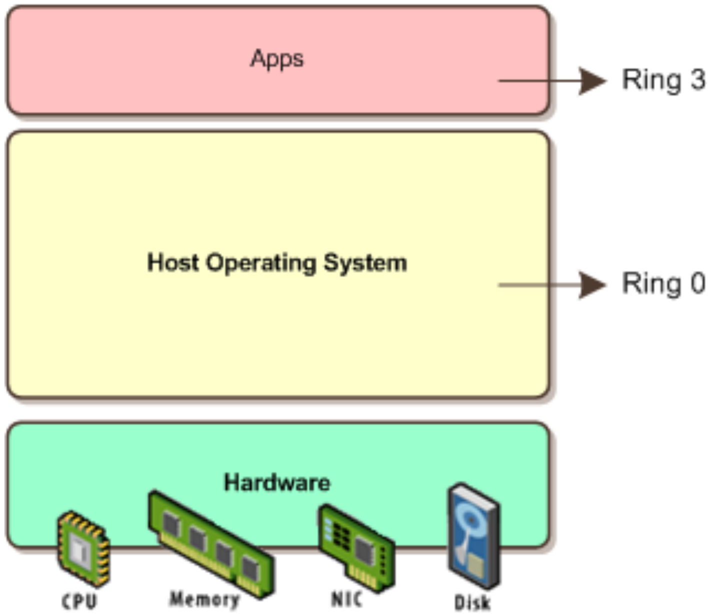
Full virtualization, no HW assist

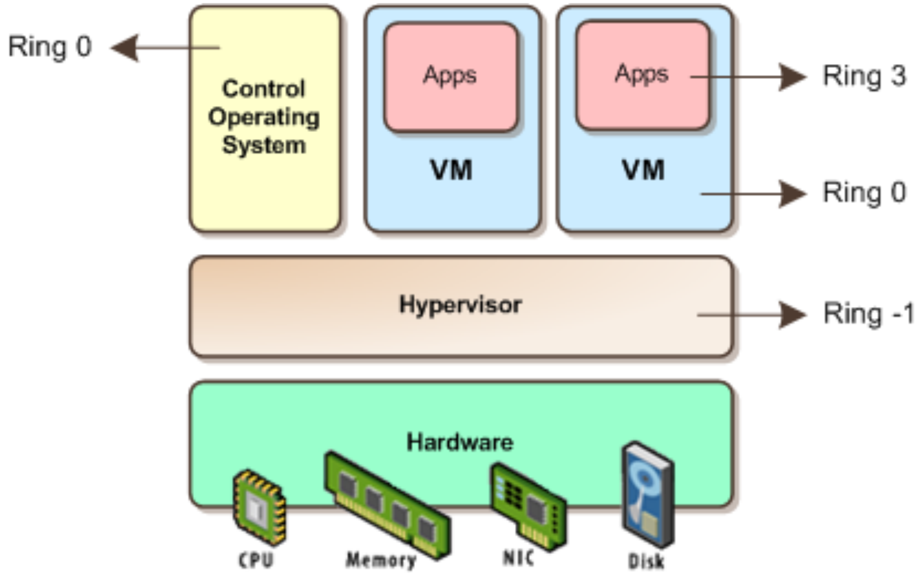
- The HOST O.S. kernel is modified to run in ring 1
  - Some host kernels are easily ported to ring 1
- Guests may be modified, or run on SW emulation
  - A subset of the ISA trapped and emulated by the HV



Full virtualization with HW assist

- Hypervisor is run in a special ring **-1**
- **Intel and AMD** provide HW support in recent CPUs
- Control O.S. kernel is run in ring 0





End of recap

- If you need references concerning Virtualization, please see
  - the CPA course page
  - The CPA/SPD 2009-2010 course page

References

- Ian Foster, Carl Kesselman, Steven Tuecke, *"The Anatomy of the Grid: Enabling Scalable Virtual Organizations" - 2003*

  http://onlinelibrary.wiley.com/doi/10.1002/0470867167.ch6/summary

- Craig Lee, *"A perspective on scientific cloud computing"* – proceedings of HPDC 2010
  - Beware that the definition of Cloud federation is not uniform in the scientific literature

- *"The Future Of Cloud Computing"* – report of EU expert group, see

  http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf

  - Parts A and B summarize a great deal of basic concepts and definitions concerning Clouds