Chapter 4 – Part 1

Big Data Computing Technology Platforms

Contents:

- Computing Platforms
- Design considerations
- Compute Clusters interconnection medium
- Resource sharing in Compute Cluster
- Cloud Architecture

- Part 2 – Cloud security basics

Computing Platforms

Compute Clusters for Big Data Handling

- Collection of interconnected stand-alone computers which can work together collectively and cooperatively as a single integrated computing resource pool.

- *Why clusters?* Clusters attempt to exploit massive parallelism at the job level; Achieves high availability (HA) through stand-alone operations.

Computing Platforms

- The benefits of computer clusters and massively parallel processors (MPPs) include: Scalable performance, high availability (HA), fault tolerance, modular growth, and use of commodity components.

- Beowulf Cluster (BC) - When a single compute job requires frequent communication among the cluster nodes, the cluster must share a dedicated network, and thus the nodes are mostly homogeneous and tightly coupled. This type of clusters is also known as BCs.

Computing Platforms … (Cont’d)

Design space considerations - Comprises 6 exclusive components; These include:

- Scalability
- Packaging
- Control
- Heterogeneity
- Programmability
- Security

Computing Platforms … (Cont’d)

- Scalability – Physical scaling according to the needs of data processing requirements;
- Limited by - Multicore chip technology, cluster topology, packaging method, power consumption, and cooling scheme employed;
- Other limiting factors - Memory capacity, disk I/O bottlenecks (if any), and latency tolerance, etc.

Computing Platforms … (Cont'd)

- Packaging – Two types: (i) Compact (ii) Slack.

In Compact Clusters, the nodes are closely packaged in one or more racks sitting in a room, and the nodes are not attached to peripherals;

In Slack Clusters, the nodes are attached to their usual peripherals (i.e., they are complete SMPs, workstations, and PCs), and they may be located in different rooms, different buildings, or even remote regions.

SMP – Symmetric Multiprocessor

RACK Servers in a DC

Computing Platforms … (Cont'd)

Packaging directly affects communication wire length, and thus the selection of interconnection technology used.

- Compact cluster can utilize a high-bandwidth, low-latency communication network that is often proprietary;
- Slack clusters employ standard LANs or WANs.

Cluster interconnect technologies

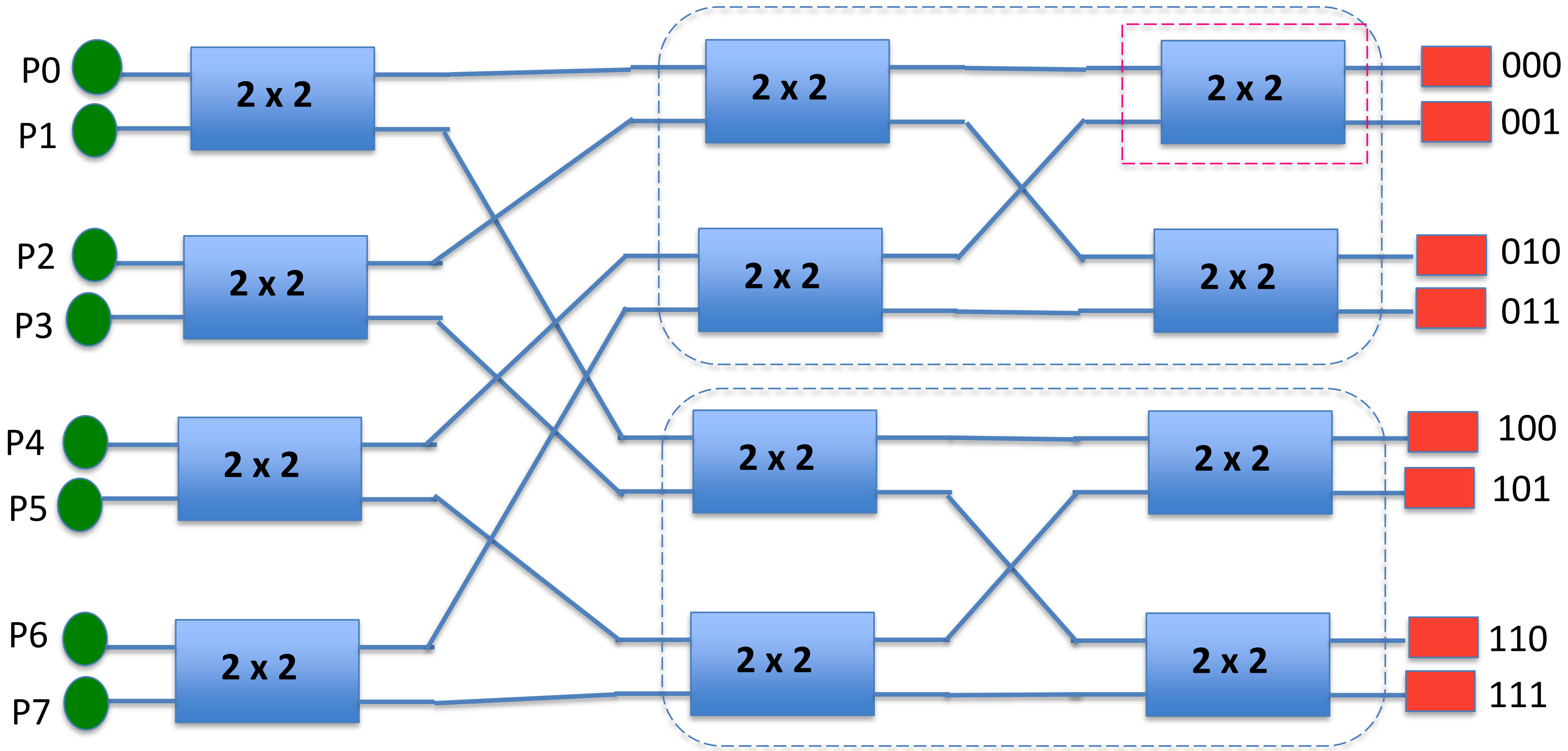
Several types: Myrinet, Infiniband, Ethernet, etc

Following are the influential parameters considered in the design

- Available link speeds
- Message Passing Interface (MPI) latency
- Network processor
- Different topologies
- Different network topologies
- Routing mechanisms
- Flow control

Self-reading:
https://static.googleusercontent.com/media/research.google.com/en//archive/googlecluster-ieee.pdf

Interconnection Medium – A Self-Routing Architecture: *Example of <u>Small-scale Cluster</u> Processor-Memory Interface*



Stage MSB

Stage LSB

Small-scale Cluster system - Self-routing Strategy

- Destination tag routing strategy

  - Each switch either connects to the upper port or to the lower port;
  - Logic: 0 – Connection to the upper port
    1 – Connection to the lower port
  - Use destination tag as a guidance to route the packet across the stages
  - MSB – Stage 0 & LSB – Stage 2

Q: Verify the above strategy for accessing the memory module (a) 110 from P0; (b) 001 from P6

Computing Platforms … (Cont'd)

- Control - (i) Centralized (ii) Decentralized

    -- Compact cluster - Centralized control

    -- Slack cluster - Both (i) and (ii)

In a centralized cluster, all the nodes are owned, controlled, managed, and administered by a central operator;

In a decentralized cluster, the nodes have individual owners;

Q: Can you list the advantages and disadvantages of the above control mechanisms?

Computing Platforms … (Cont’d)

- Homogeneity – (i) Homogeneous (ii) Heterogeneous

Homogeneous cluster - Nodes from the same platform, i.e., the same processor architecture and the same operating system; Often, the nodes are from the same vendors;

Heterogeneous cluster - Nodes of different platforms and different OS.

Q: What is the major disadvantage, if any, in employing these systems?

Computing Platforms … (Cont’d)

- Programmability

Like traditional OSs, cluster operating system (COS) must provide user-friendly interface between: the user, the application and the cluster hardware and additional features like single-system image and system availability;

In addition, COS has to ensure: failure management, load balancing and tools for parallelizing computations;

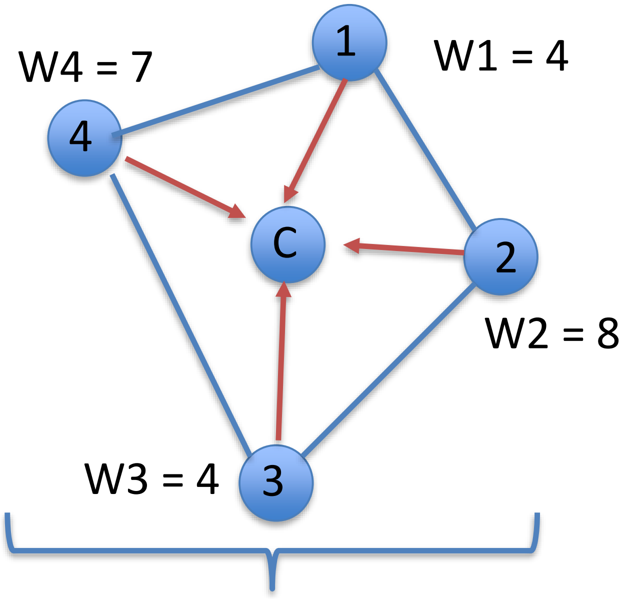
(Parallel compilers & tools are an added advantage in a cluster programming environment)

Computing Platforms … (Cont'd)

Examples of COS

- Solaris MC - Solaris MC is a prototype, distributed operating system for multi-computers (or clusters).
- MOSIX - MOSIX is a software package that extends the Linux1 kernel with cluster computing capabilities. The enhanced Linux kernel allows *any size cluster of Intel based computers to work together like a single system*, very much like a SMP (symmetrical multi processor) system.
- GLUnix (Global Layer Unix for a Network Of Workstations) - GLUnix was originally designed as a global operating system for the Berkeley Network of Workstations (NOW). The NOW project's goal was to construct a platform that would support both parallel and sequential applications on commodity hardware; It has load balancing capability;

A Window-based Load Balancing technique - COS



At t = 0

P1: -1.75; p2: +2.25; p3: -1.75; p4: -1.25

- Step 1: Node C broadcasts REQ msg to all nodes 1,…4, for status collection, say, at t=0;
- Step 2: Upon receiving the REQ msg, each node communicates to C on its current load information;
- Step 3: Node C computes an average load (5.75 in this example); Then it computes the deficit and surplus in each node w.r.t the avg;
- Step 4: Node C then sends MIGRATION msgs to the respective nodes to transfer the excess to deficit nodes; In our case: P2 to P1: 1.75; P2 to P3: 0.5; P4 to P3: 1.25; (Integer requirement to be considered, if needed;)
- Step 5: After an interval t+W, node C repeats the process from Step 1;

Nodes are independent in receiving processing tasks and may update their status using different periods compared to W. Hence, the above algorithm is sensitive to the interval W; A small W implies frequent transfers and hence communication overheads may be high; A large W implies, node C may be working with an outdated information!

Computing Platforms … (Cont'd)

- Security – Intra-cluster communication (ICC) can be either exposed or enclosed.

In an exposed cluster, the communication paths among the nodes are exposed to the outside world. An outside machine can access the communication paths, and thus individual nodes, using standard protocols (e.g., TCP/IP);

Disadvantages - Being exposed, ICC needs additional effort to ensure privacy and security; Outside communications may disrupt ICC in an unpredictable fashion; For instance, heavy Bulletin Board Services (BBS) traffic may disrupt production jobs; Even standard communication protocols tend to have high overhead on such systems;

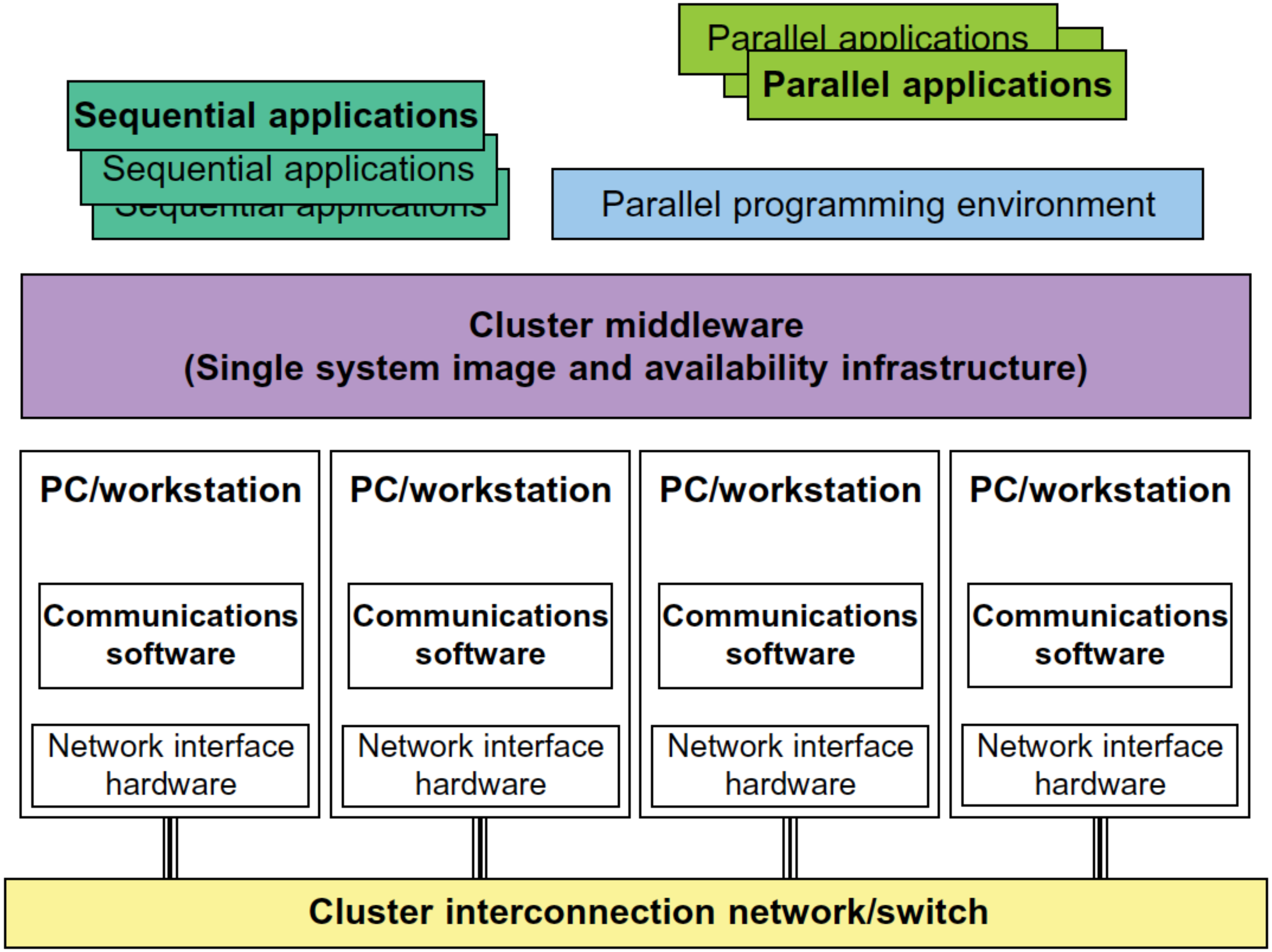
Computing Platforms … (Cont'd)

- In an enclosed cluster, ICC is shielded from the outside world, which alleviates the aforementioned problems with exposed clusters.

Disadvantage - There is currently no standard for

efficient, enclosed ICC. Consequently, most commercial or academic clusters realize fast communications through *one-of-a-kind* protocols.

Architecture of a Compute Cluster



Resource sharing in Compute Clusters

Clustering improves both availability & performance – these two goals are not necessarily in conflict!

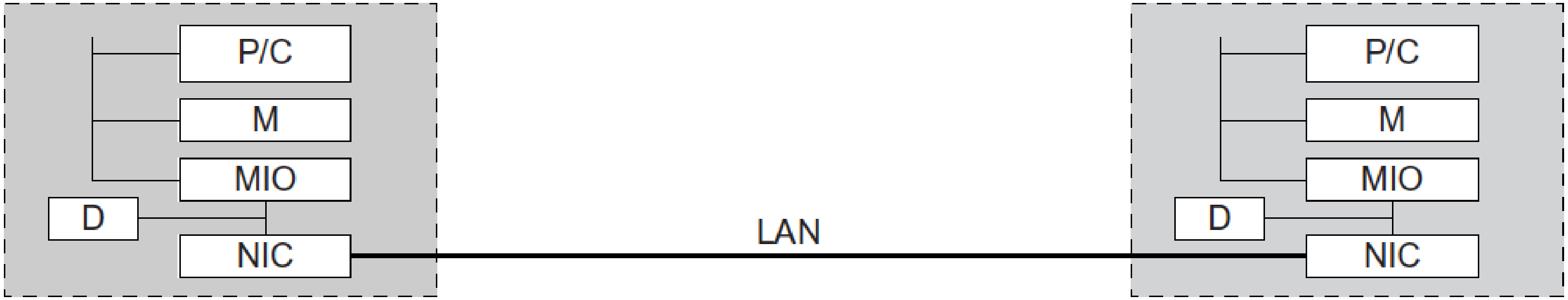
Because some HA clusters use *hardware redundancy for scalable performance*

Three ways to connect or configure a Cluster (See a figure in the next slide)

- Share-nothing architecture
- Shared-Disk architecture
- Shared-Memory architecture

HA – High Availability

Resource sharing in Compute Clusters



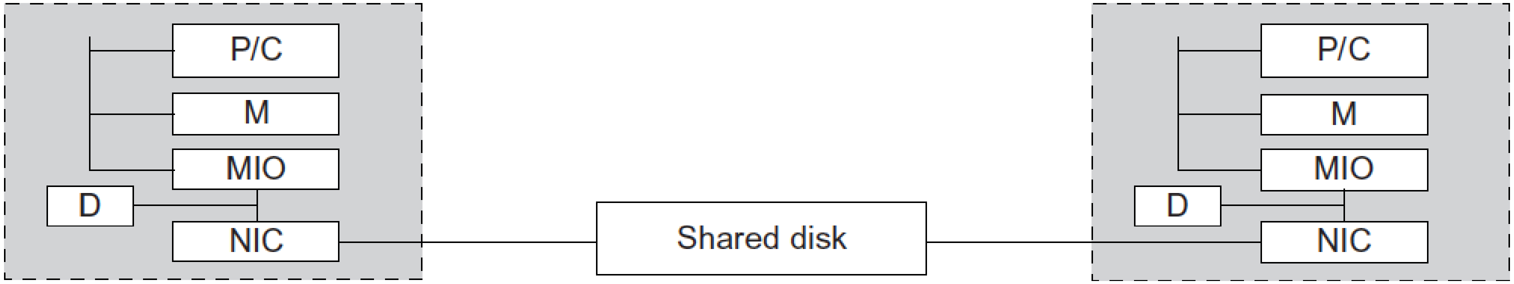
(a) Shared nothing

The *shared-nothing architecture* - simple to configure and used in most clusters;

- Nodes are connected through an I/O bus; It simply connects two or more autonomous computers via a high-speed LAN such as Ethernet;

Reference: K. Hwang, Z. Xu, Support of clustering and availability, in: Scalable Parallel Computing, McGraw-Hill, Chapter 9.

Resource sharing in Compute Clusters



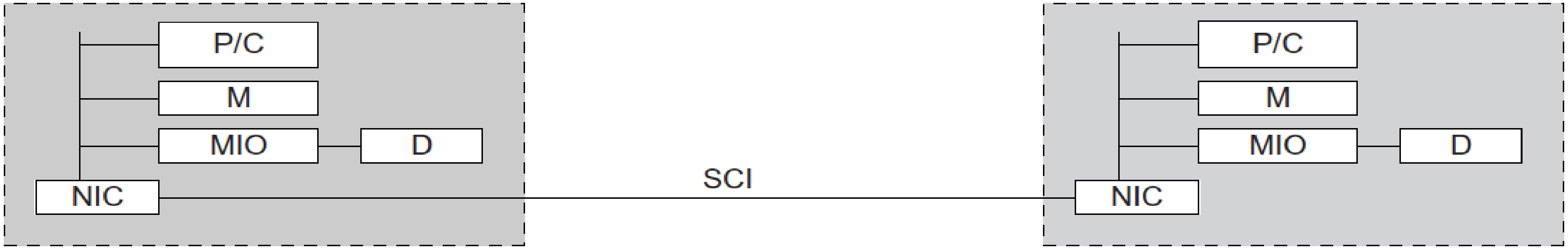
(b) Shared disk

The *shared-disk architecture* is in favor of small-scale availability clusters in business applications - When one node fails, the other node takes over –*fault-tolerance!* *How this is done?* – *Checkpointing!*

- The shared disk holds ***checkpoint files*** or critical system images to enhance cluster availability. Without shared disks, checkpointing, rollback recovery, failover, and failback are not possible in a cluster.

Reference: K. Hwang, Z. Xu, Support of clustering and availability, in: Scalable Parallel Computing, McGraw-Hill, Chapter 9.

Resource sharing in Compute Clusters



(c) Shared memory

- The shared-memory cluster – Nodes could be connected by a *scalable coherence interface (SCI)* ring, which is connected to the memory bus of each node through an NIC module;

All common data/instructions are written in the shared space and an address generated by a processor refers to this global address space.

Remarks: Note that in the other two architectures, the interconnect is attached to the I/O bus. The memory bus operates at a higher frequency than the I/O bus.

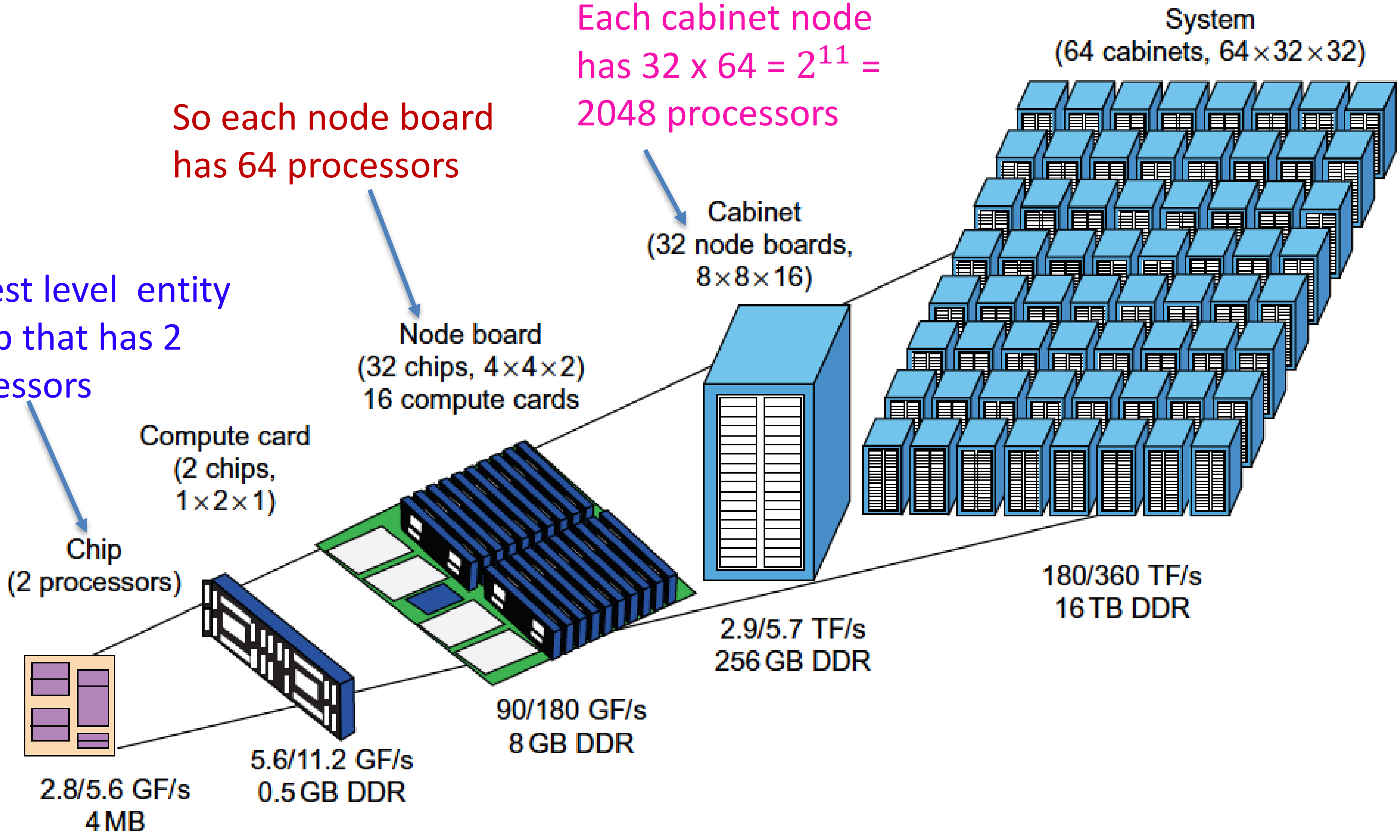
Reference: K. Hwang, Z. Xu, Support of clustering and availability, in: Scalable Parallel Computing, McGraw-Hill, Chapter 9.

Sample Real Compute Cluster – IBM Blue Gene

Lowest level entity - Chip that has 2 processors

So each node board has 64 processors

Each cabinet node has 32 x 64 = 2^{11} = 2048 processors



Cloud Computing Platforms

Evolution of Cloud Computing – From Cluster, Grid, and Utility computing

- Cluster and Grid computing leverage the use of many computers in parallel to solve problems of any size;
- Utility and Software as a Service (SaaS) provide computing resources as a service with the notion of pay per use.
- Utility computing is perhaps is an immediate predecessor of Cloud Computing

Cloud Computing Platforms … (Cont’d)

Cloud computing – A high-throughput computing (HTC) paradigm whereby the infrastructure provides the services through a large data center and/or distributed server farms.

Cloud computing leverages dynamic resources to deliver large numbers of services to end users;

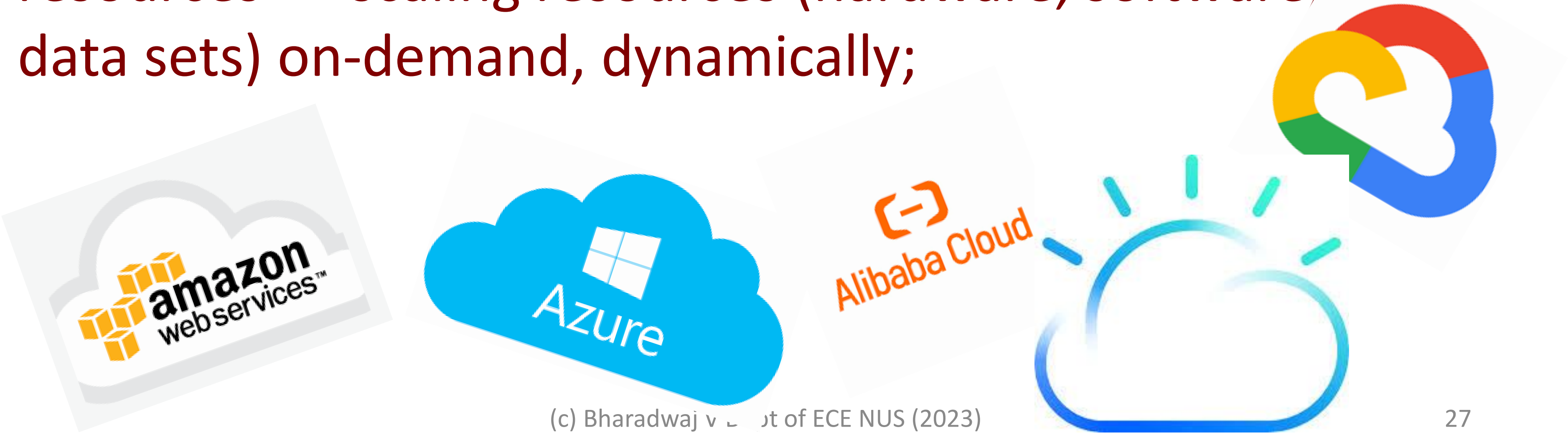
The Cloud Computing model enables users to share access to resources from anywhere, at any time, through their connected devices

Cloud Computing Platforms … (Cont'd)

CC offers significant benefits to IT companies – How?

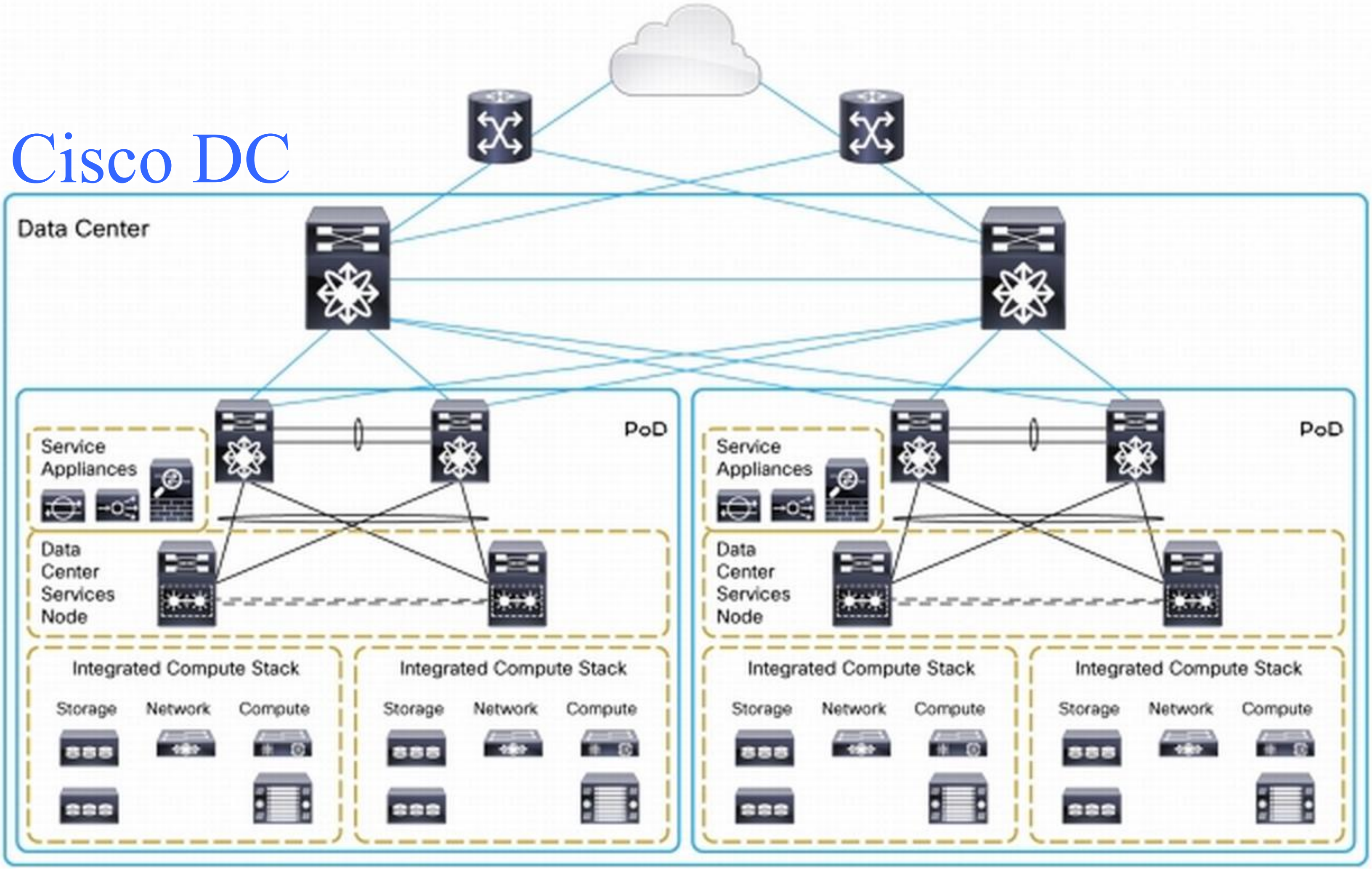
- By freeing them from the low-level tasks of setting up the hardware (servers) and managing the system software at a low cost and easy-to-use manner;

Cloud computing applies a virtual platform with "elastic resources" - scaling resources (hardware, software, data sets) on-demand, dynamically;



Cloud interconnection topology

Cisco DC

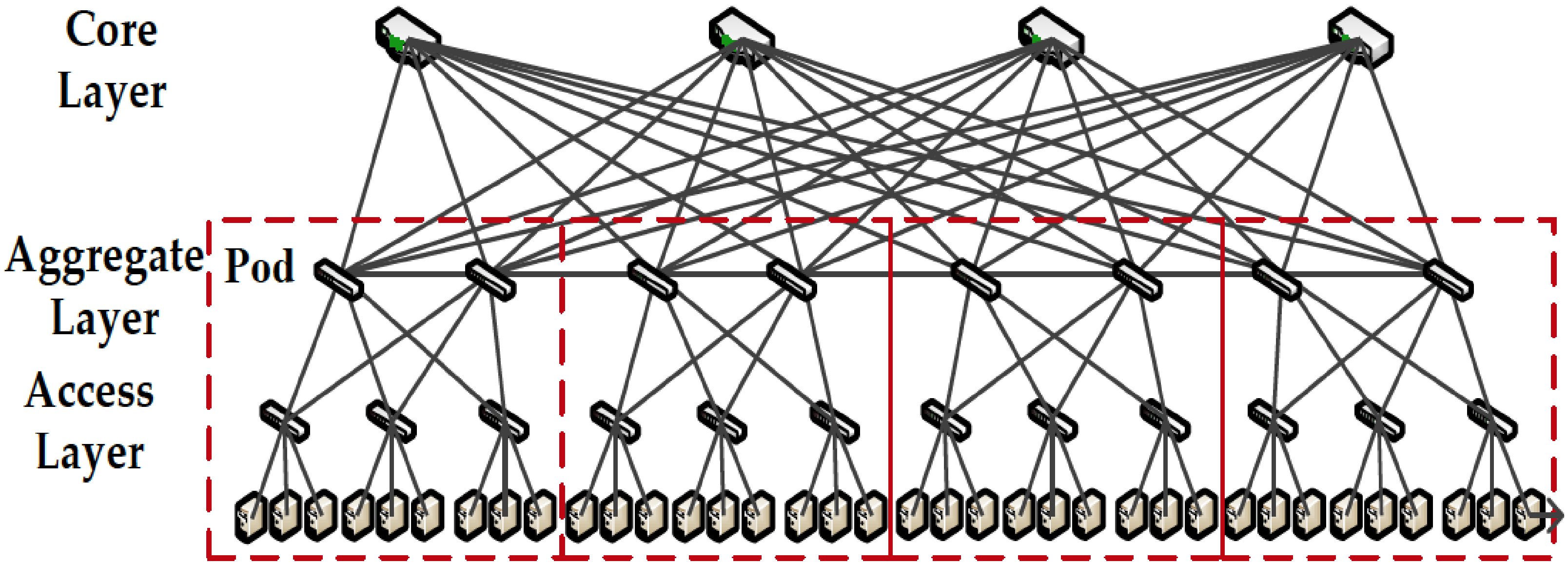


POD: Performance optimized DC

Ref: Cisco DC Design - White_paper_c11-714729

Cloud interconnection… (Cont'd)

Abstract view



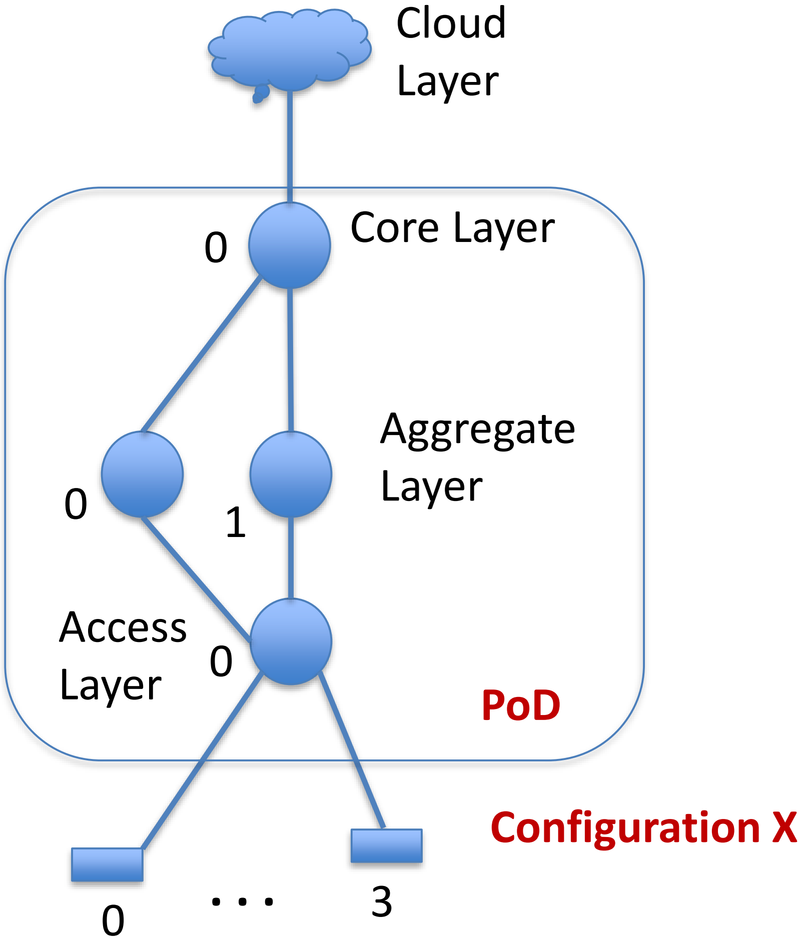
Core Layer ↔ Aggregate Layer ↔ Access Layer

Accessibility – Computing the Probability of a successful access to the Cloud

A DC comprises leaf nodes, an aggregate layer, an access layer, a core layer and a Cloud layer as an hierarchical system.

In this DC, let there be just one PoD that comprises, 1 core node (0), 2 nodes (0,1) in aggregate layer, and 1 node (0) in its access layer. Further there is *no link between nodes of aggregate layer in this DC.*

Accessibility … (Cont'd)



In such a DC system, *consider all the paths between a single node 0 in access layer to the Cloud layer* and answer the following. Let us call this as configuration X.

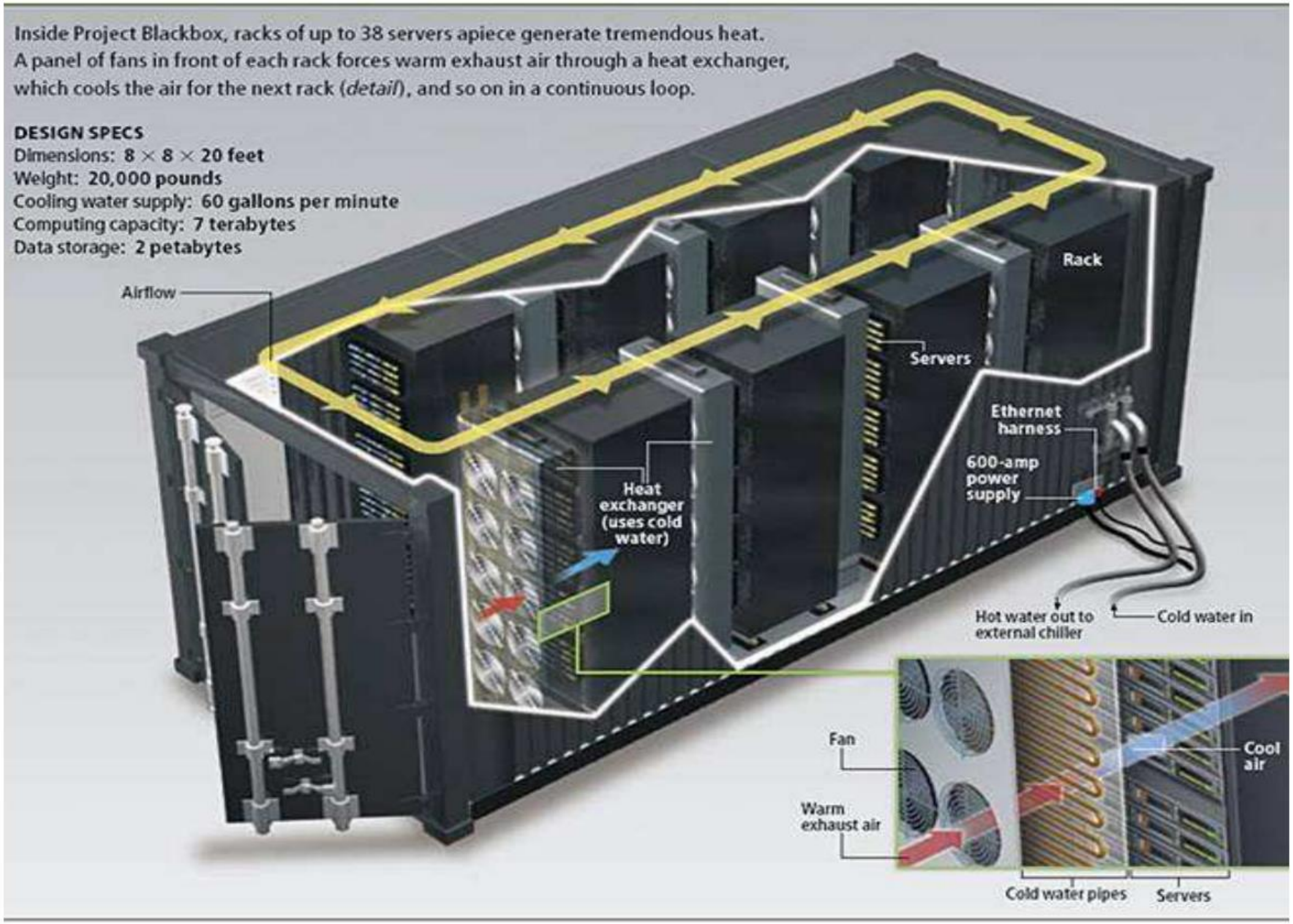
1) How many path exist between an access node 0 to the cloud?

Accessibility … (Cont'd)

2) Assume that the nodes do not fail. Let each link in the paths from the access node 0 to the cloud node fails with probability p. Thus, there is no link in the system that is 100% reliable and available. *Determine the probability that there exists at least one path from the access node 0 to the cloud where every link on that path functions without any failures.* You need to express this probability in terms of p.

Solution?

Data Center Infrastructure



Inside Project Blackbox, racks of up to 38 servers apiece generate tremendous heat. A panel of fans in front of each rack forces warm exhaust air through a heat exchanger, which cools the air for the next rack (*detail*), and so on in a continuous loop.

DESIGN SPECS
Dimensions: 8 × 8 × 20 feet
Weight: 20,000 pounds
Cooling water supply: 60 gallons per minute
Computing capacity: 7 terabytes
Data storage: 2 petabytes

Layout of a DC built inside a Container (Courtesy of HP Project Blackbox).

Ex: HP's DC - 40c - 2008/2009; about 40 foot DC; Max Capacity ~ 27kW/rack; 3500 Compute Nodes and ~12K storage drives;

Cloud vs Data Centers Debate!

- When the data is stored and accessed via online over an internet connection as and when needed, it is referred to as a *Cloud Platform.*

- Whereas, if such data and accessibility is largely governed by any on-site services in which all the software and its components including applications are stored locally in the system, then this is referred to as a *Data Center*.

Often an organization using a DC may also engage a Cloud Service Provider (CSP) for other services.

Cloud vs Data Centers Debate!

FEATURE	DC	Cloud
Scalability	Limited; depends on the capacity of the storages, servers, etc	Easily scalable – pay-as-you-go!
Security	Governed by local norms	One of the QoS factors for CSP!
Cost	High	Pay-as-you-go! Compute/Storage resources made available at cheaper costs!
Availability	Entire control on organization; their norms may dictate policies;	Largely governed by SLAs imposed by CSP; This often may provide better guarantees.

To be cont'd…

A DC, by and large, is viewed as a Private/Hybrid-Cloud!

ANNEX - Virtualization – Quick note!

A virtual machine (VM) is a computer file, *typically called an image*, that behaves like an actual computer. In other words, *creating a computer within a computer!*

VM runs in a window (separate app available to create VMs), much like any other program, giving the end user an identical experience as they would have on the host system itself.

Virtualization – Quick note!

VM is *sandboxed* from the rest of the system - this means that the software inside a VM can’t escape or tamper with the host computer itself!

Advantages - Serves as an ideal environment for:

- Testing other operating systems including beta releases,
- Accessing virus-infected data, creating operating system backups
- Running software or applications on operating systems they weren’t originally intended for, etc.

Virtualization – Quick note!

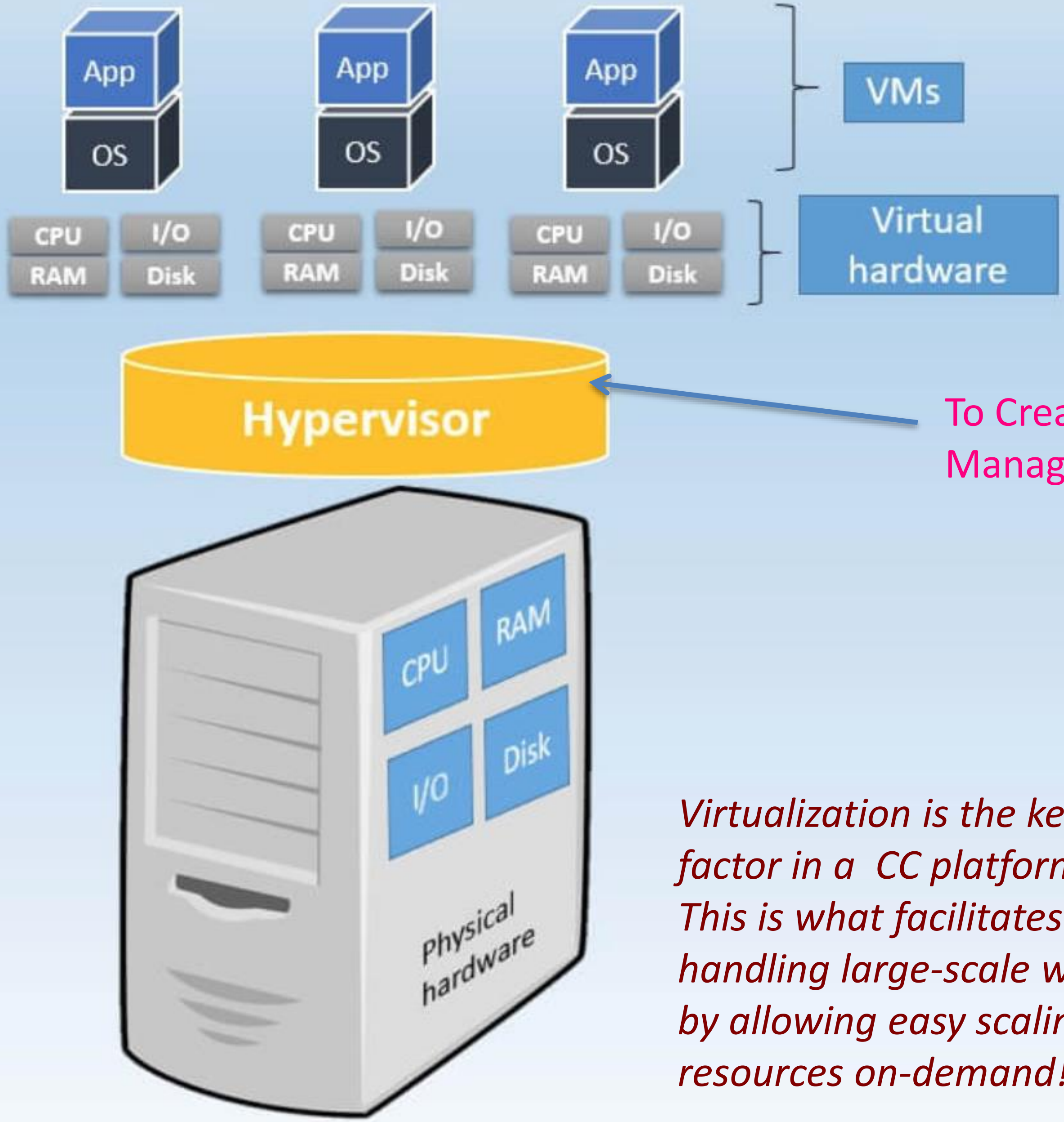
A hypervisor, such as *VMware vSphere or Microsoft Hyper-V*, is installed on top of physical hardware. A hypervisor is then used to create and manage VMs, which have their own virtual computing resources.

Multiple VMs can run simultaneously on the same physical computer.

What about servers?

For servers, the multiple operating systems run side-by-side using a *hypervisor* to manage them!

Virtualization – Quick note!



To Create & Manage VMs

Virtualization is the key factor in a CC platform! This is what facilitates handling large-scale workloads by allowing easy scaling of resources on-demand!