Types of Data Centers

Acknowledements

Images are borrowed from publicly available Internet sources

Classification Parameter

- By certification
  - Models availability, component redundancy and fault-tolerance requirement of a DC
  - Tier-1, 2, 3, 4 (1 is simple/least available/least fault-tolerant/least costly, 4 is complex/most available/most fault-tolerant/most costly)
  - Just to give you an idea
    - Tier 1: 99.671% availability
      - 28.8 hours of downtime per year max
    - Tier 2: 99.741% availability
    - Tier 3: 99.982% availability
    - Tier 4: 99.995% availability
      - 26.3 minutes of downtime per year max
  - Certification given by based on many things
  - We will not do this further in this course

- By ownership and management
  - Enterprise data center
    - Built by an organization for its own use (small/large)
  - Co-location data center
    - DC space, power, cooling etc. built and owned by one organization, rented by other organization to keep their own servers/storage etc. (large)
  - Managed Services Data Center
    - DC and systems all owned and managed by one organization, rented by other organizations for their use (large)
  - Cloud Data Centers
    - DC and systems all owned and managed by one organization, other organizations pay to host their applications and data (rather than renting out the physical infrastructure as in Managed Services DC) (large)

- By use
  - DCs for HPC (High Performance Computing)
    - Used primarily for running many parallel jobs
    - Scientific applications, large simulations,…
  - DCs for cloud computing type use
    - Used primarily to provide computing and storage service to third party users who may rent/subscribe
  - DCs for organization's own use (providing its own services to its own or other users)
  - DCs for physical hosting services
    - Just provides the physical infrastructure, customers rent rack space to put their own servers etc.

- The distinction is sometimes a bit blurred
  - You can rent virtual machines on a cloud from a cloud DC and run HPC applications on it ☺
  - Still, look at what is the primary reason the DC was built? System and network design depends on that

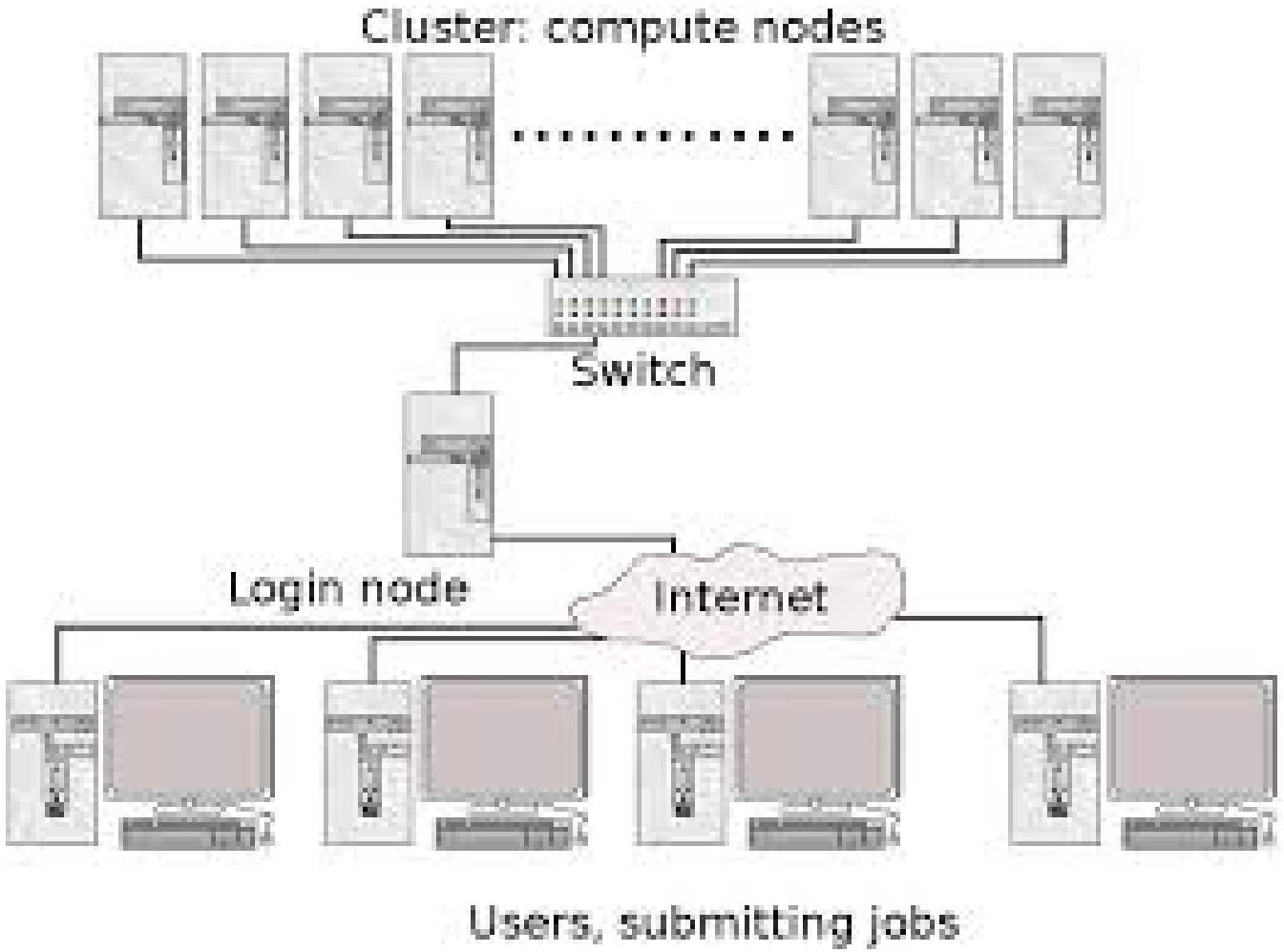
Data Centers for HPC

What is HPC?

- "*High-performance computing (HPC) is the ability to process data and perform complex calculations at high speeds*" - Netapp
- Typically used for computing needed to run large parallel jobs
  - Dependent tasks of the same job
  - Independent jobs
- Basic need for many branches of science and engineering
- Numerous scientific and engineering applications
- Numerous software packages to aid parallelization of applications that can take advantage of the large number of machines/cores

Typical Architecture of a HPC system

- Cluster
  - One (or a few) master node(s) (also called head node or login node sometime)
  - A number of compute nodes
  - Usually also have a central shared storage accessible by all nodes
  - A fast interconnect connecting them
  - The master node is connected to the external network as well as to all compute nodes
  - The compute node need not be connected to the external network (no direct external access) except for management purposes


Cluster: compute nodes
Switch
Login node
Internet
Users, submitting jobs

Basic Operation

- Users log in to the master node
- Jobs are submitted by the users to the master node only
  - No direct access to the compute nodes ideally
- A cluster middleware runs on the master node
  - Main tasks – scheduling jobs to compute nodes, load balance the compute nodes, monitor faults,,…
- The scheduler on the master node allocates the jobs to the compute nodes
  - Maintains a queue (can be more than one queue based on categories, priority etc.)
  - Runs with the privilege of the logged-in user
- The results are returned by the compute node to the master, which can then be picked up by the users
- User views the system as a single machine

Workload Manager

- Very important part of a cluster
- It is a software that runs on the master node
- Queues and schedules the jobs on the compute nodes
  - Typically does more than scheduling
- Well known workload managers
  - PBS Pro
    - Very popular
    - Both open source and commercial version
  - Rocks
  - slurm
  - Spectrum LSF (from IBM, not free)

Some Example Features of PBS Pro

- Takes jobs from users and queues them
- Selects which job to run when on which node based on policy set by administrator
  - Set priority, limit resources, many others…..
- Tracks usage to enforce usage policy specified by administrator, report usage
- Tracks whether a job completes or not, ensuring jobs complete even if some node goes down
- Set of commands given to control PBS operation by administrator
- Very powerful and in wide use, tested on clusters with tens of thousands of nodes

What about Storage?

- Depends on the use
- Two types of physical storage
  - Disks local to nodes
  - Central storage if any
- Two types of logical storage available to users
  - Scratch – storage for temporary use, like temporary files generated during run, temporary storage of some data
    - Can be on local disk or shared file server
  - Permanent storage – storage for more persistent use
    - Usually on shared file servers for large systems
    - Can be on the nodes for smaller systems
- Relevant directories mounted when a user logs in to the master node
- Storage models vary depending on application needs

Managing Shared Storage

- Distributed file systems hide the individual node details to user
- But underneath, traffic goes to the file server from all compute nodes
- Files can be large (terabytes even)
- Can be a bottleneck for large systems
- So what filesystems to use?
  - For smaller systems, NFS is ok
  - For larger system, use a parallel file system (Lustre, GPSF, …)
    - Allows parallel access to files for faster read/write

How do you program on them?

- Can use standard software for specific domains that support parallelization
- Can program on your own also
  - MPI (Message Passing Interface) is the de-facto standard for writing parallel codes
- Can also run sequential code (single job) as is

How to specify the rating of a HPC system?

- Specified usually in Flops (Floating point operations per second), not number of servers etc.
  - 100 TFlops, 1 PFlop etc.
  - Theoretical Peak vs. Actual Peak vs Sustained performance
- Theoretical Peak Flop rating can be computed from the CPUs
  - (No. of cores) x (cycles/sec) x (flops/cycle)
  - First two are in any CPU spec, third is dependent on CPU family

- Example – a server with 2 no. Xeon Gold 6152 CPU
  - 22 cores, 2.1 GHz base frequency
  - Flops per cycle (double precision) for Xeon Gold processor family (Intel skylake/cascadelake architecture) = 32
- So one CPU = 22 x 2.1 x 10^9 x 32 = 1.478 Teraflop (TFlop)
- So one server rating = 3 TFlop (approx.)
- To make a HPC system with theoretical peak performance of 1 Petaflop, you will need around 340 such servers

- But this is theoretical peak rating, won't be achieved
- Ratings in practice are measured by running benchmark suite of programs and measuring
  - Actual peak rating
  - Sustained rating
- LINPACK and its variations – current standard benchmark for testing HPC system's performance for comparison
  - Software to solve a dense system of linear equations
  - Other benchmarks exist

- What kind of scale does such DCs achieve?
  - Take a look at the site top500.org, it is fun and informative!
    - Anyone interested in HPC should follow this site
  - The current (Nov. 2019 list) top one in the world (ORNL Summit, USA) has
    - A theoretical peak rating of 200 PFlops, actual peak 148 PFlop!! (only one above 100 PFlop actual peak though)
    - Around 2.4 million cores with IBM 22 core processors (4000+ nodes
      - How do you connect them?
    - > 10 MW of power consumption
  - More than 1 PFlop theoretical peak is common
    - Sometimes aided also by GPUs in nodes, which have much higher Flop rating per GPU

- Issues
  - How do you connect them?
    - Topology?
    - Protocol?
      - Ethernet (10/25/40/100G)
      - Infiniband (56/100/200G)
      - Some companies have proprietary very high speed interconnects (ex. Aries from Cray Inc.)

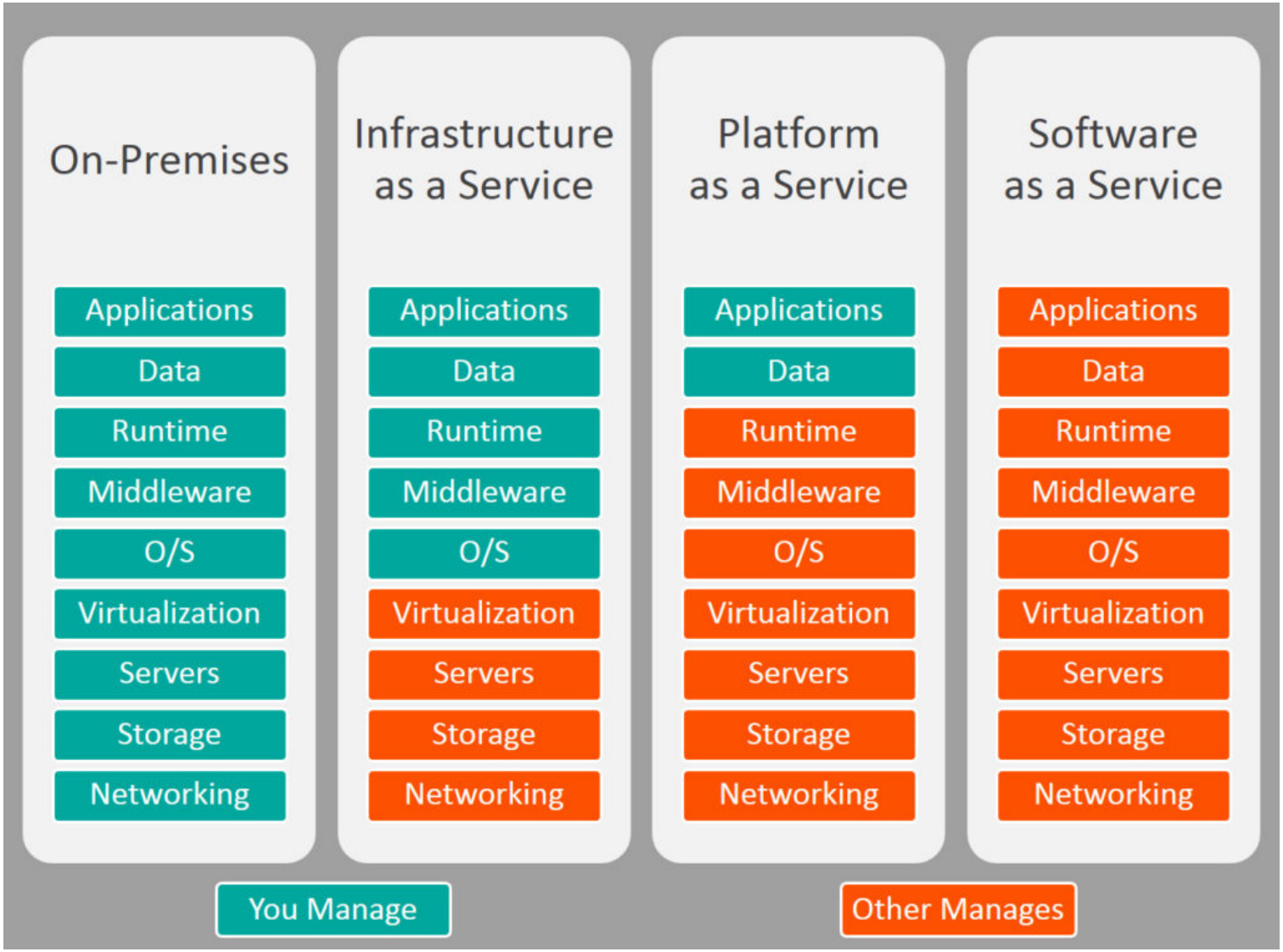
Data Centers for Cloud Computing

What is Cloud Computing?

- "*Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS)*" – Amazon AWS
- "*Simply put, cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet ("the cloud") to offer ….You typically pay only for cloud services you use, helping lower your operating costs, run your infrastructure more efficiently and scale as your business needs change.*" – Microsoft Azure

Cloud Computing Models

- Infrastructure-as-a-Service (IaaS)
  - Cloud provides you servers, storage etc.
- Platform-as-a-service (PaaS)
  - Cloud provides you a runtime on which to develop your applications to give
- Software-as-a-Service (SaaS)
  - Cloud provides you a final service (ex. Dropbox)
- We will consider only IaaS


On-Premises
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Infrastructure as a Service
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Platform as a Service
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
Software as a Service
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You Manage
Other Manages

Infrastructure-as-a-Service

- Users can rent/subscribe servers, storage space, network bandwidth
- Normally a physical server can support multiple applications
- However, the applications need to be hidden from one another
  - I should think I have a physical server to myself
- Data should be protected from each other also
- Achieved through Virtualization

Virtual Machines

- Emulates a server on top of a physical server, with its own CPU, RAM, storage etc.
  - Can run its own OS, and do whatever you can do on a physical server
- A single physical server can run multiple VMs on it, possibly with different OS
  - How to estimate how many? We will see…
  - Underneath, the VMs share the same CPU, RAM etc., but isolated by the virtualization software
  - Storage for a VM can be provisioned on local storage or central storage as per configuration
  - Well-known virtualization software on the market – VMWare, RedHat Virtualization
- Techniques for creating and managing VMs are not in the scope of this course

Number of VMs per server

- vCPU – unit of CPU allocation to a VM
  - Typically, 1 physical core = 2 vCPUs for high end servers
    - Can be 1-to-1 for some servers also
  - Typical minimum allocation per VM = 1 vCPU
- Consider a server with 2 no. Xeon 6152 CPUs and 256 GB RAM
  - Total physical cores = 2x22 = 44
  - Say VM configuration we want is 2 vCPU and 4GB Ram each
  - Total available vCPUs = 44x2 = 88
  - So maximum 44 VMs.
  - Lets check the RAM. 44 VM means 44x4 = 176 GB RAM. So enough RAM
    - Usually leave some RAM for the virtualization software itself, depends on the software
  - Sometimes we will allow more than 44 (overcommit), as all of them may not be used together to the full extent

Sample VM Sizing in Amazon AWS

Instance Name	vCPU	Memory (GB)
t2.nano	1	0.5
t2.micro	1	1
t2.small	1	2
t2.medium	2	4
t2.large	2	8
t2.xlarge	4	16

A Typical Cloud Data Center

- Will have a large number of servers, with multiple VMs running on them
- Shared storage for allocating storage to the VMs
- Interconnect to connect the servers and the storage
- Complex management software
  - Example: If a user requests a new VM, which server do you place it on? If a VM fails, which server (and how) do you migrate it to before failure? Migration for load balancing? How do you ensure everyone gets proper network bandwidth?….
  - Extremely important practical issues as well as research issues
  - Many vendors – IBM, Redhat, VMWare, Rackspace,…..
  - Proprietary software also used by many large cloud service providers

DCs for Organization’s Own Use

- Will still contain servers and storage (local or shared) and some interconnection between them
- Design will depend on exact use
  - Number and type of servers
  - Amount and type of storage
  - Interconnection pattern and technology
  - And of course software/applications that are to be run
- Can be small (Ex. a campus data center for academic/administrative use of a small university) to very large (Ex. Google's data center for providing all google services that you use like search, mail, drive etc., Facebook's data center,…)