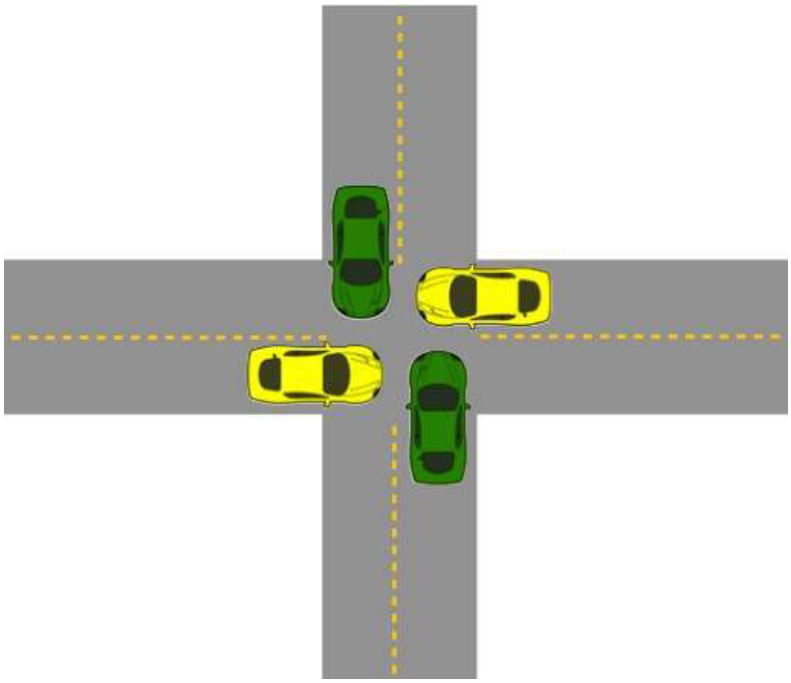
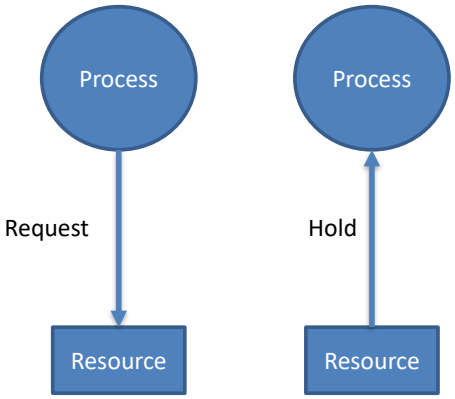
Deadlock

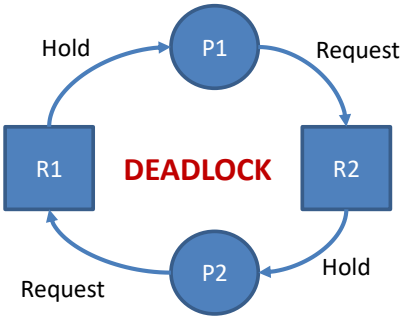
What is Deadlock?



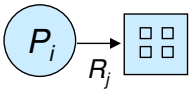
What is Deadlock?

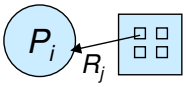
- A set of processes is deadlocked **if each process in the set is waiting for an event** that **only another process in the set can cause**.
- Deadlocks are a **set of blocked processes** each **holding a resource and waiting to acquire a resource held by another process**.



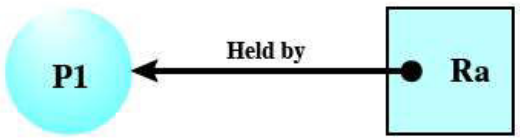


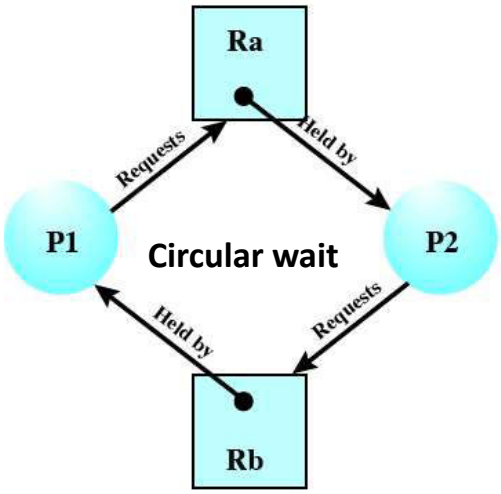
What is Deadlock?

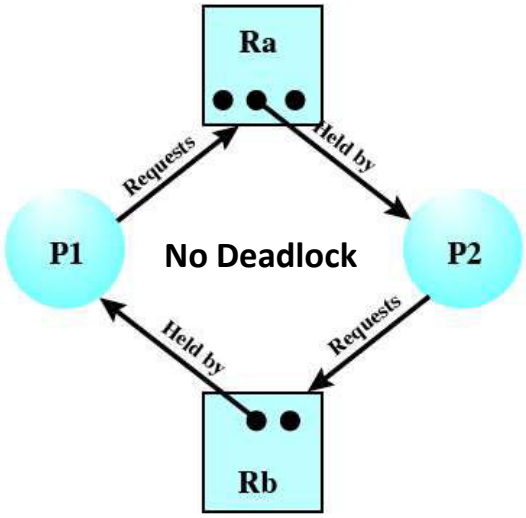
- Process
- Resource Type with 4 instances
- P_i requests instance of R_j 

- P_i is holding an instance of R_j 

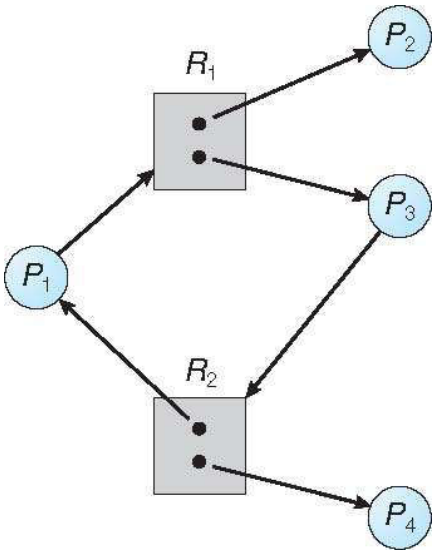
Conventions







Graph With a Cycle But No Deadlock



Preemptable and non-preemptable resource

- **Preemptable**:- Preemptive resources are those which **can be taken away from a process without causing any ill effects** to the process.
  - Example:- Memory.
- **Non-preemptable**:- Non-pre-emptive resources are those which **cannot be taken away from the process without causing any ill effects** to the process.
  - Example:- CD-ROM (CD recorder), Printer.

Deadlock v/s Starvation

Deadlock	Starvation
All processes keep waiting for each other to complete and none get executed.	High priority process keep executing and low priority process are blocked.
Resources are blocked by the process.	Resources are continuously utilized by the higher priority process.
Necessary conditions are mutual exclusion, hold and wait, no preemption, circular wait.	Priorities are assigned to the process.
Also known as circular wait.	Also known as lived lock.
It can be prevented by avoiding the necessary conditions for deadlock.	It can be prevented by Aging.

Conditions that lead to deadlock

1. Mutual exclusion
   - Each resource is either currently **assigned to exactly one process or is available**.
2. Hold and wait
   - Process currently holding resources granted earlier can **request more resources**.
3. No preemption
   - Previously granted resources **cannot be forcibly taken away** from process.
4. Circular wait
   - There must be a **circular chain of 2 or more processes**. Each process is waiting for resource that is held by next member of the chain.

- **First three detect the possibility of the deadlock**
- **All four of these conditions must happen for existence of deadlock.**

Strategies for dealing with deadlock

1. Just **ignore** the problem.
   - ❑ **Detection** and **recovery -** Let deadlocks occur, detect them and act.
2. Dynamic **avoidance** by careful resource allocation.
3. **Prevention**, by structurally negating (killing) one of the four required conditions.

Deadlock ignorance (Ostrich Algorithm)

- When storm approaches, an ostrich puts his head in the sand (ground) and pretend (imagine) that there is no problem at all.
- **Ignore** the **deadlock** and **pretend** that **deadlock never occur**.
- Reasonable if
  - deadlocks **occur very rarely**
  - **difficult to detect**
  - **cost** of prevention is **high**
- UNIX and Windows take this approach

Deadlock detection for multiple resource

E =

Tape Drive	Plotters	Scanners	CD Roms
4	2	3	1

total no of each resource

A =

Tape Drive	Plotters	Scanners	CD Roms
2	1	0	0

no of resources that are available (free)

C =

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	1	0
P2	2	0	0	1
P3	0	1	2	0

no of resources held by each process

R =

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	2	0	0	1
P2	1	0	1	1
P3	2	1	0	0

no of resources still needed by each process to proceed

Deadlock detection for multiple resource

E =

Tape Drive	Plotters	Scanners	CD Roms
4	2	3	1

total no of each resource

A =

Tape Drive	Plotters	Scanners	CD Roms
0	0	0	0

no of resources that are available (free)

C =

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	1	0
P2	2	0	0	1
P3	2	2	2	0

no of resources held by each process

R =

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	2	0	0	1
P2	1	0	1	1
P3	0	0	0	0

no of resources still needed by each process to proceed

Deadlock detection for multiple resource

E =

Tape Drive	Plotters	Scanners	CD Roms
4	2	3	1

total no of each resource

A =

Tape Drive	Plotters	Scanners	CD Roms
2	2	2	0

no of resources that are available (free)

C =

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	1	0
P2	2	0	0	1
~~P3~~	~~0~~	~~0~~	~~0~~	~~0~~

no of resources held by each process

R =

DEADLOCK

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	2	0	0	1
P2	1	0	1	1
~~P3~~	~~0~~	~~0~~	~~0~~	~~0~~

no of resources still needed by each process to proceed

Example

- Five processes P_0 through P_4; three resource types A (7 instances), *B* (2 instances), and *C* (6 instances)

- Snapshot at time T_0:

	Allocation	*Request*	*Available*
	A B C	*A B C*	*A B C*
P_0	0 1 0	0 0 0	0 0 0
P_1	2 0 0	2 0 2	
P_2	3 0 3	0 0 0	
P_3	2 1 1	1 0 0	
P_4	0 0 2	0 0 2	

- Sequence <P_0, P_2, P_3, P_4, P_1> will result in ***Finish[i] = true*** for all ***i***

Example (Conti..)

- P_2 requests an additional instance of type ***C***

	Request
	A B C
P_0	0 0 0
P_1	2 0 2
P_2	0 0 1
P_3	1 0 0
P_4	0 0 2

- State of system?
  - Can reclaim resources held by process P_0, but insufficient resources to fulfill other processes; requests
  - Deadlock exists, consisting of processes P_1, P_2, P_3, and P_4

Deadlocks

- Detection and Resolution
  - The kernel analyzes the resource state to check whether a deadlock exists.
  - If so, it aborts some process(es) and allocate resources held by them to other processes so that deadlock ceases to exist
- Prevention
  - The kernel uses resource allocation policy that ensures that 4 conditions for resource deadlock, mentioned earlier, do not arise simultaneously. It makes deadlock impossible.
- Avoidance
  - Kernel analyzes the allocation state to determine whether granting a resource request can lead to deadlock in future. Requests that do not lead to deadlock are granted; others are not scheduled. This avoids deadlocks

Differences

- **Prevention: Design a system such that deadlocks cannot occur**, at least with respect to serially reusable resources.
- **Avoidance: Impose less stringent conditions** than for prevention, allowing the possibility of deadlock, but sidestepping it as it approaches.
- **Detection:** In a system that allows the **possibility of deadlock**, determine if deadlock has occurred, and which processes and resources are involved.
- **Recovery:** After a deadlock has been detected, clear the problem, allowing the deadlocked processes to complete and the resources to be reused. Usually involves destroying the affected processes and restarting them.

Differences

- **Detection and Recovery –** Kernel analyzes the resource states to check deadlocks; If so, aborts processes or allocates resources held by a process to other processes so that **deadlock ceases to exist**.

- **Prevention –** Kernel uses resource allocation policies to ensure that 4 necessary conditions of deadlock does not arise. **Makes deadlocks impossible**.

- **Avoidance –** Kernel analyzes resource allocation state to determine whether granting a resource to a process leads to deadlock in future. Only grant feasible requests keeping some processes pending. **Deadlocks do not arise**.

Detection Algorithm

- **Available***:* A vector of length ***m*** indicates the number of available resources of each type

- **Allocation***:* An ***n* x *m*** matrix defines the number of resources of each type currently allocated to each process

- **Request***:* An ***n* x *m*** matrix indicates the current request of each process. If ***Request* [*i*][*j*] = *k***, then process P_i is requesting ***k*** more instances of resource type R_j.

Detection Algorithm

1. Let ***Free*** and ***Finish*** be vectors of length ***m*** and ***n***, respectively Initialize:
   (a) ***Free* = *Available***
   (b) For ***i* = 1,2, …, *n***, if $\boldsymbol{Allocation_i \neq 0}$, then
   Finish*[i] = *false; otherwise, ***Finish*[i] = *true***

2. Find an index ***i*** such that both:
   (a) ***Finish*[*i*] == *false***
   (b) $\boldsymbol{Request_i \leq Free}$
   If no such ***i*** exists, go to step 4

3. $\boldsymbol{Free = Free + Allocation_i}$
Finish*[*i*] = *true
go to step 2

4. If ***Finish[i] == false***, for some ***i***, $1 \leq i \leq n$, then the system is in deadlock state

Detection – Algorithm Usage

- When, and how often, to invoke depends on:
  - How often a deadlock is likely to occur?
  - How many processes will need to be rolled back?
    - one for each disjoint cycle

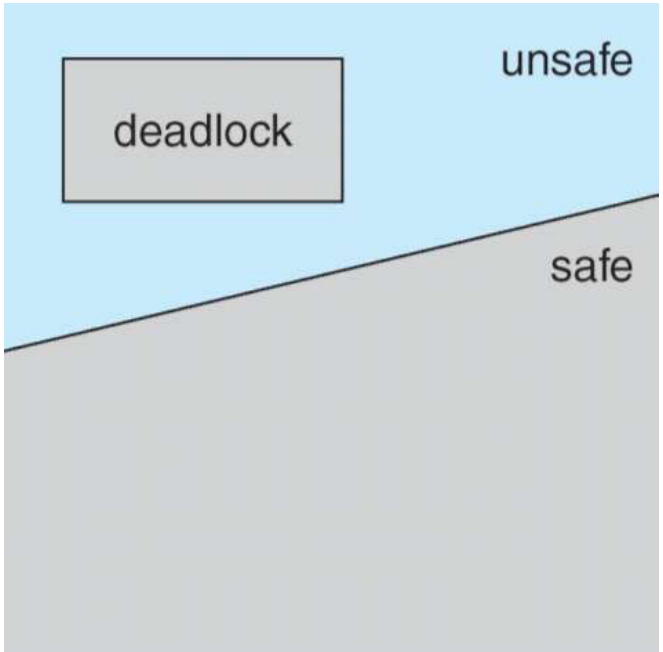
- If detection algorithm is invoked arbitrarily, there may be many cycles in the resource graph and so we would not be able to tell which of the many deadlocked processes "caused" the deadlock.

Safe and unsafe states

- A state is said to be safe if it is not deadlocked and there is some scheduling order in which **every process can run to completion even if all of them suddenly request** their maximum number of resources immediately.
- Total resources are 10
- 7 resources already allocated
- So there are 3 still free
- A need 6 resources more to complete it.
- B need 2 resources more to complete it.
- C need 5 resources more to complete it.

Process	Has	Max
A	3	9
B	2	4
C	2	7
Free : 3		

Safe, Unsafe and Deadlock State



Safe states

Process	Has	Max
A	3	9
B	2	4
C	2	7
Free : 3		

2

Process	Has	Max
A	3	9
B	4	4
C	2	7
Free : 1		

4

Process	Has	Max
A	3	9
B	0	-
C	2	7
Free : 5		

5

Process	Has	Max
A	3	9
B	0	-
C	7	7
Free : 0		

7

Process	Has	Max
A	3	9
B	0	-
C	0	-
Free : 7		

6

Process	Has	Max
A	9	9
B	0	-
C	0	-
Free : 1		

Unsafe states

Process	Has	Max
A	3	9
B	2	4
C	2	7
Free : 3		

1

Process	Has	Max
A	4	9
B	2	4
C	2	7
Free : 2		

2

Process	Has	Max
A	4	9
B	4	4
C	2	7
Free : 0		

4

Process	Has	Max
A	4	9
B	0	-
C	2	7
Free : 4		

Deadlock Avoidance

- Deadlock can be avoided by **allocating resources carefully**.
- Carefully **analyse each resource** request to **see if it can be safely granted**.
- Need an algorithm that can always avoid deadlock by making right choice all the time **(Banker's algorithm).**
- Banker's algorithm for single resource
- Banker's algorithm for multiple resource

A state is said to be safe if it is not deadlocked and there is some scheduling order possible in which every process can run to completion even if all of them suddenly request their maximum number of resources immediately.

Banker's algorithm for single resource

- What the algorithm does is check to see if granting the request leads to an unsafe state. If it does, the request is denied.
- If granting the request leads to a safe state, it is carried out.
- If we have situation as per figure
  - then it is safe state
  - because with 10 free units
  - one by one all customers can be served.

Process	Has	Max
A	0	6
B	0	5
C	0	4
D	0	7
Free : 10		

Banker's algorithm for single resource

Process	Has	Max
A	1	6
B	1	5
C	2	4
D	4	7
Free : 2		

Process	Has	Max
A	1	6
B	1	5
C	4	4
D	4	7
Free : 0		

Process	Has	Max
A	1	6
B	1	5
~~C~~	~~0~~	~~0~~
D	4	7
Free : 4		

Process	Has	Max
A	1	6
B	1	5
~~C~~	~~0~~	-
D	7	7
Free : 1		

Process	Has	Max
A	1	6
B	1	5
~~C~~	~~0~~	-
~~D~~	~~0~~	-
Free : 8		

Process	Has	Max
A	1	6
B	5	5
~~C~~	~~0~~	-
~~D~~	~~0~~	-
Free : 4		

Banker's algorithm for single resource

Process	Has	Max
A	1	6
~~B~~	~~0~~	-
~~C~~	~~0~~	-
~~D~~	~~0~~	-
Free : 9		

Process	Has	Max
A	6	6
~~B~~	~~0~~	-
~~C~~	~~0~~	-
~~D~~	~~0~~	-
Free : 4		

Process	Has	Max
~~A~~	~~0~~	-
~~B~~	~~0~~	-
~~C~~	~~0~~	-
~~D~~	~~0~~	-
Free : 10		

- The order of execution is C, D, B, A. So if we can find proper order of execution then there is no deadlock.

Banker's algorithm for single resource

Process	Has	Max
A	1	6
B	2	5
C	2	4
D	4	7
Free : 1		

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
1	0	2	0

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	3	0	1	1
P2	0	1	0	0
P3	1	1	1	0
P4	1	1	0	1
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	1	1	0	0
P2	0	1	1	2
P3	3	1	0	0
P4	0	0	1	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
1	0	1	0

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	3	0	1	1
P2	0	1	0	0
P3	1	1	1	0
P4	1	1	1	1
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	1	1	0	0
P2	0	1	1	2
P3	3	1	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
2	1	2	1

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	3	0	1	1
P2	0	1	0	0
P3	1	1	1	0
P4	-	-	-	-
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	1	1	0	0
P2	0	1	1	2
P3	3	1	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
1	0	2	1

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	4	1	1	1
P2	0	1	0	0
P3	1	1	1	0
P4	-	-	-	-
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	1	1	2
P3	3	1	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
5	1	3	2

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	-	-	-	-
P2	0	1	0	0
P3	1	1	1	0
P4	-	-	-	-
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	1	1	2
P3	3	1	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
5	0	2	0

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	-	-	-	-
P2	0	2	1	2
P3	1	1	1	0
P4	-	-	-	-
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	0	0	0
P3	3	1	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
5	2	3	2

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	-	-	-	-
P2	-	-	-	-
P3	1	1	1	0
P4	-	-	-	-
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	0	0	0
P3	3	1	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
2	1	3	2

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	-	-	-	-
P2	-	-	-	-
P3	4	2	1	0
P4	-	-	-	-
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	0	0	0
P3	0	0	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	-	-	-	-
P2	-	-	-	-
P3	-	-	-	-
P4	-	-	-	-
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	0	0	0
P3	0	0	0	0
P4	0	0	0	0
P5	2	1	1	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
4	2	3	2

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	-	-	-	-
P2	-	-	-	-
P3	-	-	-	-
P4	-	-	-	-
P5	2	1	1	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	0	0	0
P3	0	0	0	0
P4	0	0	0	0
P5	0	0	0	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	-	-	-	-
P2	-	-	-	-
P3	-	-	-	-
P4	-	-	-	-
P5	-	-	-	-

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	0	0	0	0
P2	0	0	0	0
P3	0	0	0	0
P4	0	0	0	0
P5	0	0	0	0

no of resources still needed by each process to proceed

Banker's algorithm for multiple resource

Tape Drive	Plotters	Scanners	CD Roms
6	3	4	2

total no of each resource

Tape Drive	Plotters	Scanners	CD Roms
5	3	2	2

resources hold

Tape Drive	Plotters	Scanners	CD Roms
1	0	2	0

Available (free) resources

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	3	0	1	1
P2	0	1	0	0
P3	1	1	1	0
P4	1	1	0	1
P5	0	0	0	0

no of resources held by each process

Process	Tape Drive	Plotters	Scanners	CD Roms
P1	1	1	0	0
P2	0	1	1	2
P3	3	1	0	0
P4	0	0	1	1
P5	2	1	1	0

no of resources still needed by each process to proceed

Summarizing Banker's Algorithm

- When the system receives a request for resources, it runs the Banker's algorithm to determine if it is safe to grant the request
- Can the request be granted?
  - If not, the request is impossible and must either be denied or put on a waiting list
- Assume that the request is granted
- Is the new state safe?
  - If so, grant the request
  - If not, either deny the request or put it on a waiting list

Trade-offs in Banker's Algorithm

- The algorithm needs to know how much of each resource a process could possibly request
  - In most systems, this information is unavailable, making the Banker's algorithm useless
  - it is unrealistic to assume that the number of processes is static
  - Algorithm works with assumption that a process will eventually release all its resources upon termination. This is practically impossible for any process to wait indefinitely to acquire resources with this mechanism

Deadlock Prevention

- Deadlock can be **prevented by attacking the one of the four conditions** that leads to deadlock.
  1. Attacking the Mutual Exclusion Condition
  2. Attacking the Hold and Wait Condition
  3. Attacking the No Preemption Condition
  4. Attacking the Circular Wait Condition

Attacking the Mutual Exclusion Condition

- Removing the mutual exclusion (**non-blocking synchronization technique**)
- Removing the mutual exclusion condition means that no process may have exclusive access to a resource
  - **each resource can be assigned to more than one process**.
- We **can not assign some resources to more than one process** at a time such as **CD-Recorder**, **Printer** etc…
- Files for multiple read is okay but multiple writes is not possible
- So, this solution is not feasible.

Attacking the Hold and Wait Condition

- Removing hold and wait needs algorithms, such as serializing tokens, also known as the all-or-none algorithms
- Require processes to **request all their resources before starting execution**.
- A process is **allowed to run if all resources it needs is available**. Otherwise, no resource will be allocated, and it will just wait.
- Problem with this strategy is that a **process may not know required resources at start of run**.
- This advance knowledge is frequently difficult to satisfy and, in any case, is an inefficient use of resources.

Attacking the No Preemption Condition

- Removing this lock-out condition is done as:
- When a process **P0 request some resource R which is held by another process P1** then **resource R is forcibly taken away from the process P1 and allocated to P0**.
- A process has to be able to have a resource for a certain amount of time, or the processing outcome may be inconsistent, or thrashing may occur.
- Breach of process priority
- Generally, not a possible option unless the resources are like CPU where status can be saved for continuing execution.

Attacking the circular wait condition

- Provide a **global numbering of all the resources**.
- Now the rule is that: **processes can request resources whenever they want to, but all requests must be made in sequential order**.
- A process need not acquire them all at once.
- Assign a precedence to each resource and force processes to request resources in order of **increasing precedence**.
  - if a process holds some resources, and the highest precedence of these resources is m, then this process cannot request any resource with precedence smaller than m.
    - Orderly request ONLY
- Allow holding only one resource per process; if a process requests another resource, it **must first free the one it's currently holding**
- Resource graph can never have cycle.

Basic Concepts of Cloud Datacenter

Why Study Data Centers

- From an end-to-end cloud infrastructure perspective, the large proportion of cost (both capex and opex) lies within the data center facility.
- The data center facility is the first layer of cloud computing. Sometimes, it is also called Data Center as a Service
- It can be considered the foundation of the cloud cost model.
- Formulated as 3+2 model:
  - Three key functional elements
    - Space
    - Power
    - Cooling
  - Two auxiliary components
    - Fire
    - Physical Security

Why Study Data Centers

- Therefore, it is critical to understand
  - What a data center facility is
  - How is it associated with cloud computing infrastructure?
  - How will the cost impact the cloud business solution?

Components of Cloud Cost Analysis

- Data Center Facilities
- Data Center Power
- Power Distribution Unit and Cabling
- Data Center Cooling
- Effective Air Distribution in Data Centers
- Cooling Strategies
- Fire Suppression and Physical Security

Ways to Measure Data Center Capacity

- The total number of racks installed in the data center
- The total physical size of the building in square meters or square feet
- The total power capacity in megawatts

Data Centers

- Data center (DC) is a physical facility that enterprises use to house computing and storage infrastructure in a variety of networked formats.

- Main function is to deliver utilities needed by the equipment and personnel:
  - Power
  - Cooling
  - Shelter
  - Security

Data Center - An Overview

Engineering Perspective	Communication Perspective	Storage Perspective	Computer Perspective
Mechanical rooms	Wiring closet	Network storage room	Server room
Electrical rooms	LAN rooms	Storage space	Computer room
	Network closet or room	Backup room	
	Telecommunication room	Disaster recovery site	
	Console room or Network operation center		

Rooms in Data Center

- Entrance Room
- Main Distribution Area (Computer Room) - MDA
- Telecom Room
- Mechanical Room
- Electrical Room
- Network Operation and Support Room (or NOC)
- Staging Area, Storage Room, Loading Dock
- Common Areas
- General Office

Yotta - India's Biggest Hyperscale Tier 4 Data Center

Yotta – G1 Data Center

At Yotta, we are dedicated to propelling innovation across Gujarat, through our world-class products in AI, Cloud Computing, IT Services, and data centers. Welcome to Yotta, where our vision is to help power the evolution of Gujarat's technological landscape.

Built-up Area	Racks	IT Power	Design PUE
21,000 sq. ft	350	2MW	<1.6

Yotta D1, NCR-Delhi

Yotta's Greater Noida campus is spread across a 20-acre space. This will be the first data center park in the region, which will consist of 6 interconnected data center buildings offering 30,000 racks capacity powered by more than 250 MW of power.

Racks	Area (sq.ft)	IT Power	Design PUE
5000	300,000	30MW	1.4

Yotta NM1, Navi Mumbai

Located in the 600-acre Hiranandani Fortune City, Yotta Data Center Park at Panvel has Yotta NM1, its first data center, Yotta NM2 (Ready in core and shell) and space to develop additional three data center facilities on the same campus.

Racks	IT Power	Fiber Paths	Design PUE
7200	50MW	4	1.4

Yotta - in Action

Yotta - in Action

Hosted At India, Asia's Largest Data Center : Yotta NM1

Shakti Cloud at Yotta's 700 MW datacentre campus offers high-density space, power, cooling, and robust cybersecurity, with rack densities exceeding 48 kW and efficient heat management for optimized power usage.

Know More

Data Centers

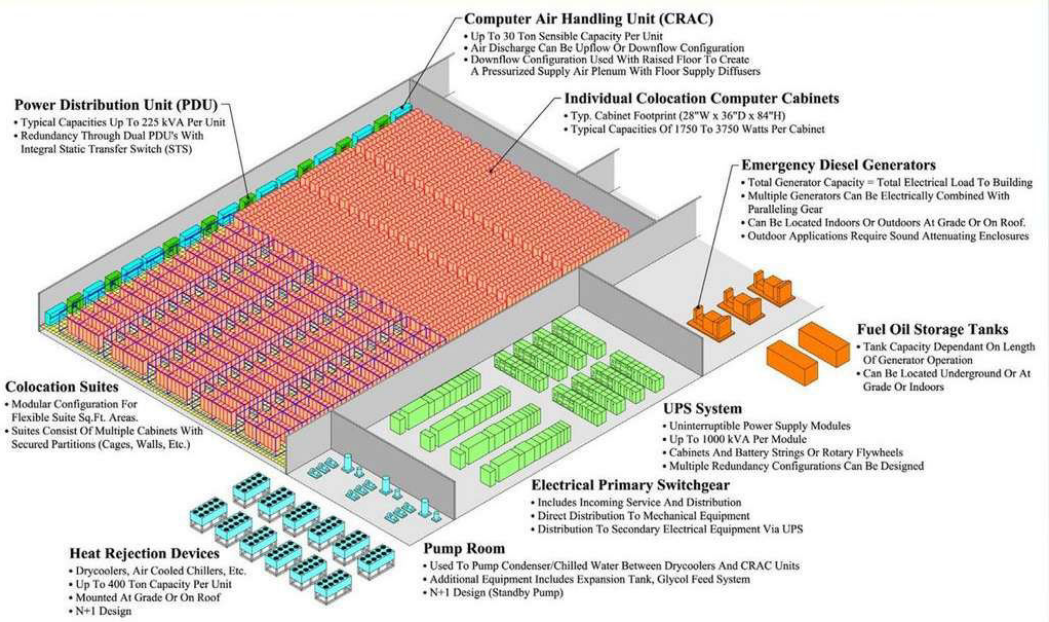
	Traditional Data Center	Cloud Data Center
Servers	Co-located Dependent Failure	Integrated Fault-Tolerant
Resources	Partitioned Performance Interrelated	Unified Performance Isolated
Management	Separated Manual	Centralized Full Control With Automation
Scheduling	Plan Ahead Overprovisioning	Flexible Scalable
Renting	Per Physical Machines	Per Logical Usage
Application / Services	Fixes on Designated Servers	Runs and Moves across All VMs

Data Centers

	Traditional Data Center	Cloud Data Center
Location	On-premises, Physical proximity to the organization	Virtually accessible due to cloud deployment
Scalability	Require multiple servers and more data space for the development	Offers subscription-based plans
Repairing & Upgrades	Excess downtime and connection issues due to single location	Centralized Full Control With Automation
Cost	High-cost infrastructure due to installation	Enable extra resources with additional payment

Datacenter Architecture

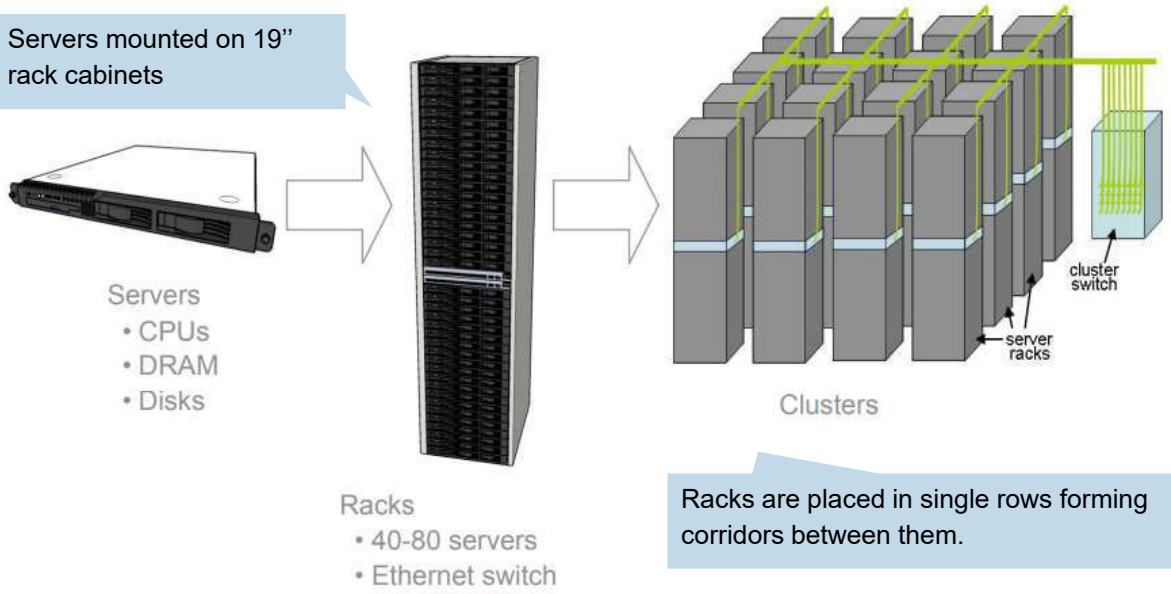
Main components of a datacenter



src: The Datacenter as a Computer – Barroso, Clidaras, Holzle

Traditional Data Center Architecture

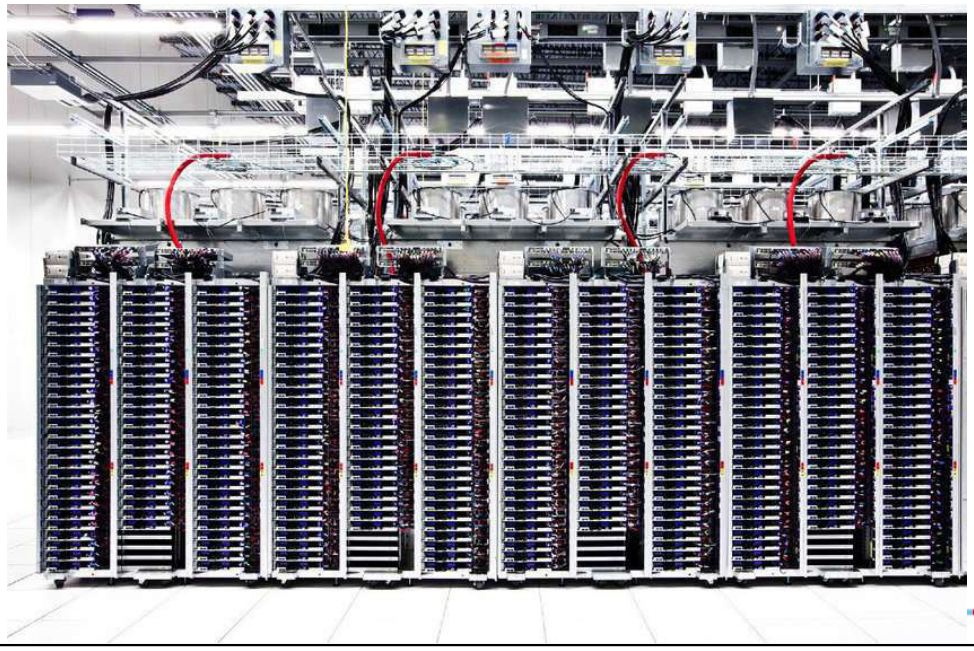
Servers mounted on 19'' rack cabinets



Racks are placed in single rows forming corridors between them.

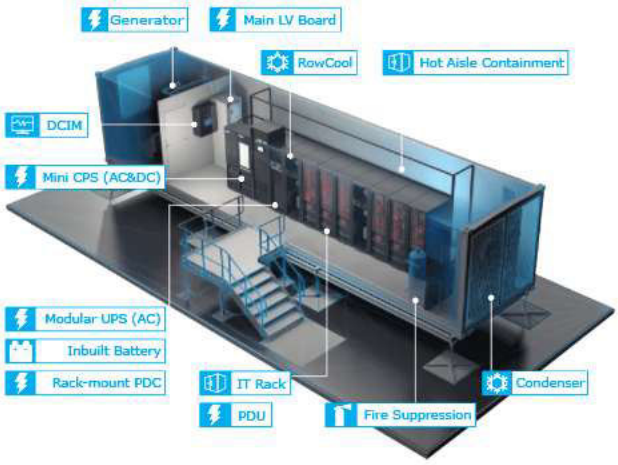
Src: the datacenter as a computer – an introduction to the design of warehouse-scale machines

A Row of Servers in a Google Data Center



Src: the datacenter as a computer – an introduction to the design of warehouse-scale machines

Inside a modern data center

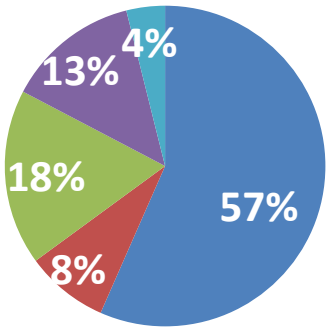


- Today's DC use shipping containers packed with 1000s servers each.
- For repairs, whole containers are replaced.

Costs for operating a data center

- DCs consume 3% of global electricity supply (416.2 TWh > UK's 300 TWh)
- DCs produce 2% of total greenhouse gas emissions
- DCs produce as much CO2 as The Netherlands or Argentina

Monthly cost = $3'530'920



31% power

- Servers
- Networking Equipment
- Power Distribution & Cooling
- Power
- Other Infrastructure

45,978 servers, 3yr server & 10 yr infrastructure amortization

Major Contributors for Cost

- These cost components usually consist of more than 90% of the total data center facility costs:
  - Space
  - Power
  - Cooling
- Additional Mandatory cost components
  - On-site security
  - Cable and cabling
  - Fire suppression system
  - Racks or cabinets

Capacity Planning

Scenarios to cater for Cost of Data Center Capacity Planning

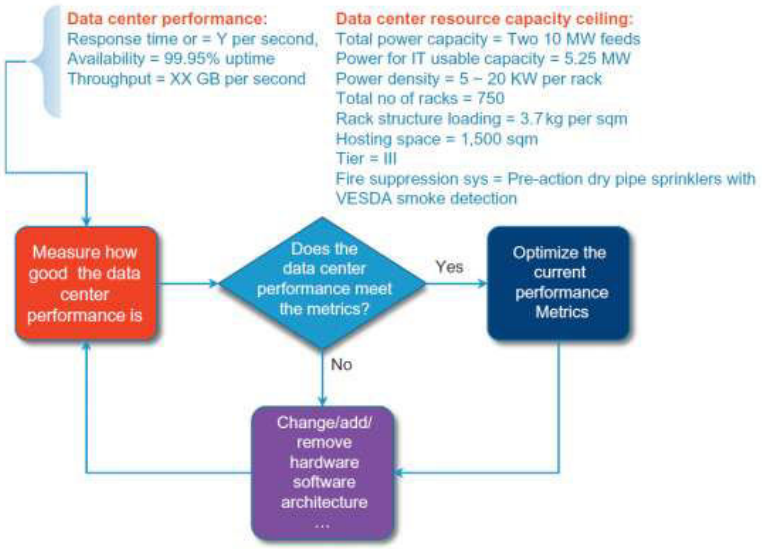
- Building a brand new data center
- Purchasing or selecting colocation space
- Expanding the existing data center capacity
- Consolidating the existing data centers
- Moving or migrating current business applications into the cloud
- Building a business continuity facility
- Data center relocation

Capacity Planning - Additional

Additional Scenarios to cater that doesn't impact much on cost

- Performance of data center facility
  - Service Level Agreement (SLA)
  - Customer experiences
  - User or stakeholder expectations
- Resource capacity of data center facility
  - Resource ceilings
  - System capability
  - Reliability and availability

Data Center Capacity Planning



Scale-up vs. scale-out

- **Scale-up**: high cost powerful CPUs, more cores, more memory
- **Scale-out**: adding more low cost, commodity servers

- Supercomputer vs. data center
- **Scale**
  - Blue waters = 40K 8-core “servers”
  - Microsoft Chicago Data centers = 50 containers = 100K 8-core servers
- **Network architecture**
  - Supercomputers: InfiniBand, low-latency, high bandwidth protocols
  - Data Centers: (mostly) Ethernet based networks
- **Storage**
  - Supercomputers: separate data farm
  - Data Centers: use disk on node + memory cache

Power Usage Effectiveness (PUE)

- **PUE is the ratio of**
  - The total amount of energy used by a DC facility
  - To the energy delivered to the computing equipment

- **PUE is the inverse of data center infrastructure efficiency**

- **Total facility power** = covers **IT systems** (servers, network, storage) +
  other equipment (cooling, UPS, switch gear, generators, lights, fans, etc.)

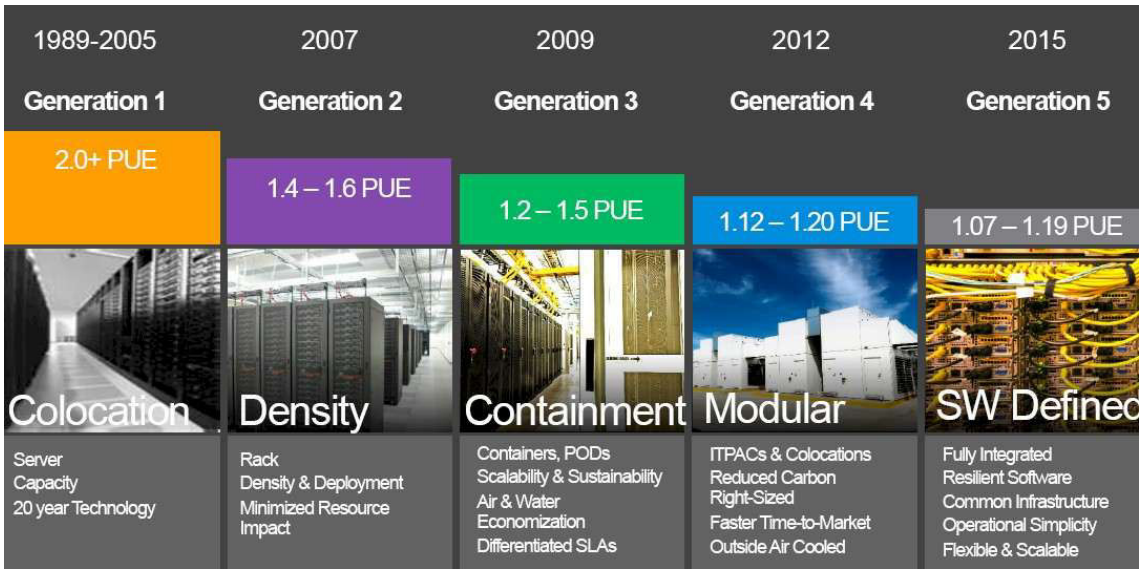
Achieving PUE

- **Location of the DC** – cooling and power load factor

Price per Kilo Watt Hour	Where?	Possible Reason Why
3.6 cents	Idaho	Hydroelectric Power; Not Sent Long Distance
10.0 cents	California	Electricity Transmitted Long Distance over the Grid; Limited Transmission Lines in the Bay Area; No Coal Fired Electricity Allowed in California.
18.0 cents	Hawaii	Must Ship Fuel to Generate Electricity

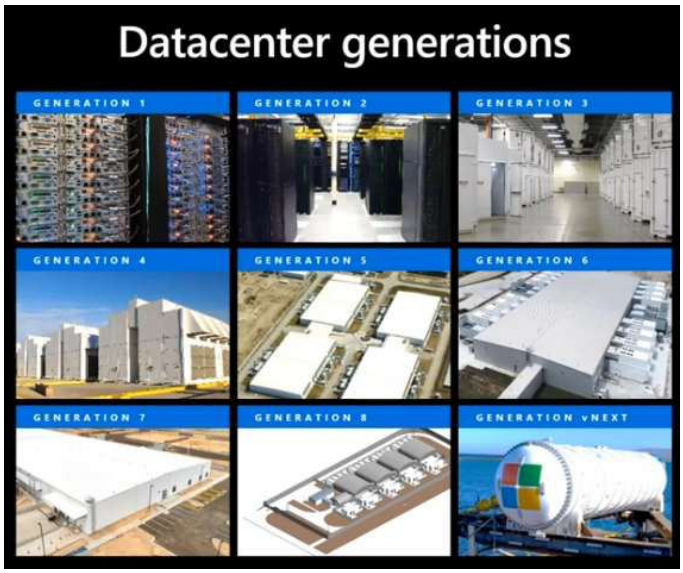
- **Raise temperature of aisles**
  - Usually 18-20 C; Google at 27 C
  - Possibly up to 35 C (trade off failures vs. cooling costs)
- **Reduce conversion of energy**
  - E.g., Google motherboards work at 12V rather than 3.3/5V
- **Go to extreme environments**
  - Arctic circle (Facebook)
  - Floating boats (Google)
  - Underwater DC (Microsoft)
- **Reuse dissipated heat**

Evolution of data center design



https://www.nextplatform.com/2016/09/26/rare-tour-microsofts-hyperscale-datacenters/

Evolution of datacenter design

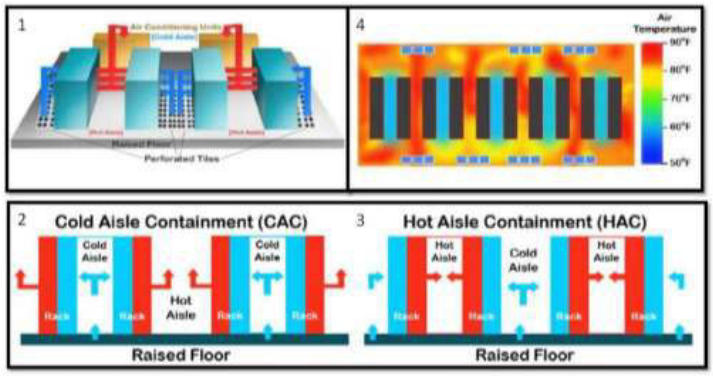


- **Gen 6: scalable form factor (2017)**
  - Reduced infrastructure, scale to demand
  - 1.17-1.19 PUE
- **Gen 7: Ballard (2018)**
  - Design execution efficiency
  - Flex capacity enabled
  - 1.15-1.18 PUE
- **Gen 8: Rapid deploy datacenter (2020)**
  - Modular construction and delivery
  - Equipment skidding and preassembly
  - Faster speed to market
- **Project Natick (future) – 1.07 PUE or less**

Src: Inside Azure Datacenter Architecture with Mark Russinovich

Datacenter Challenges

Challenge 1: Cooling data centers



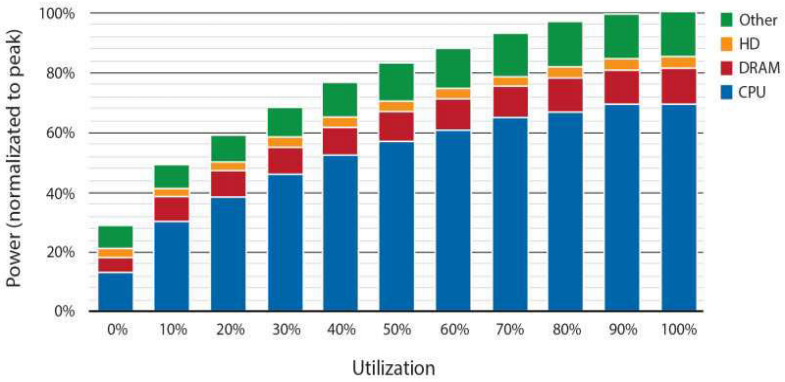
1- Conventional cooling
2- Cold Aisle Containment (CAC)
3- Hot Aisle Containment (HAC)
4- Thermal modelling

Cooling plant at a Google DC in Oregon

Challenge 2: Energy Proportional Computing

- **Average real world DC and servers are too inefficient.**
  - waste 2/3+ of their energy
- **Energy consumption is not proportional to the load**
  - CPUs are not so bad but the other components are
  - CPU is the dominant energy consumer in servers – using 2/3 of energy when active/idle.
- **Try to optimize workloads**
- **Virtualization and consolidation**.

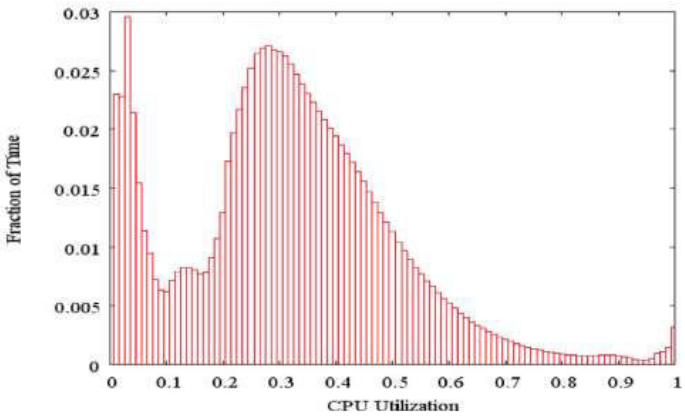
Sub-system power usage in an x86 server as the compute load varies from idle to full (reported in 2012).



src: "The Datacenter as a Warehouse Computer"

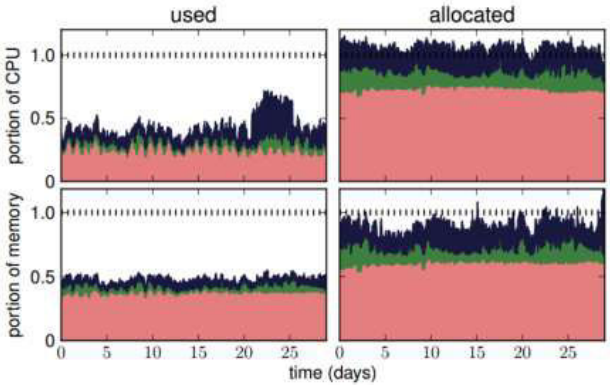
Challenge 3: Servers are idle most of the time

- For **non-virtualized servers 6-15% utilization**
- **Server virtualization** can **boost** to an average **30% utilization**
- **Need** for **resource pooling** and application and server **consolidation**
- Need for **resource virtualization**



src: Luiz Barroso, Urs Hölzle "The Datacenter as a Computer"

Challenge 4: Efficient monitoring



- Even with virtualization and software defined DC, **resource utilization can be poor**.
- Need for **efficient monitoring** (measurement) and **cluster management**.
- Goal to **meet SLOs** and **SLIs.**
- **Job's tail latency matters!**

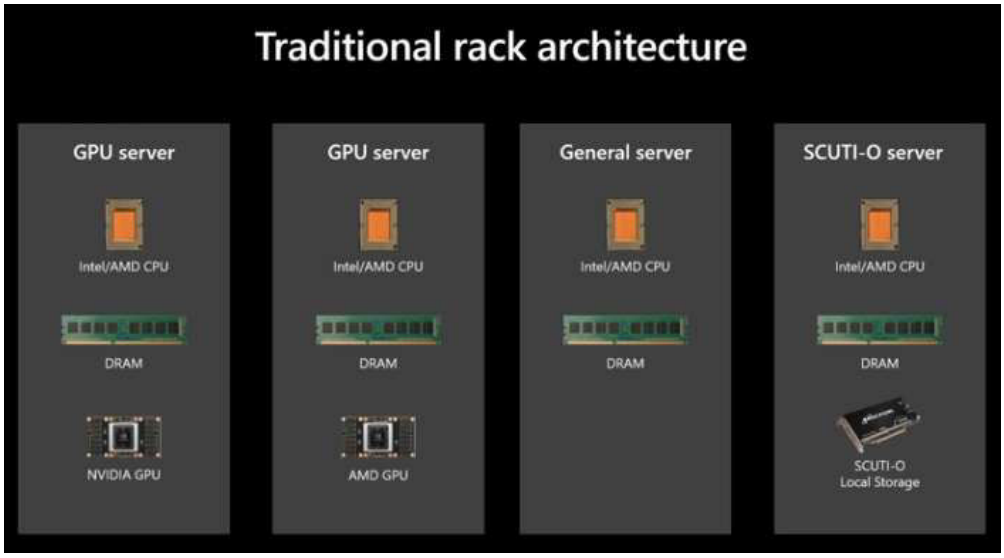
src: "Heterogeneity and dynamicity of clouds at scale: Google trace analysis" SoCC'12

Improving resource utilization

- **Hyper-scale system management software**
  - Treat the datacenter as a warehouse scale computer
  - Software defined datacenters
  - System software that allows DC operations to manage the entire DC infrastructure
  - Compose a system using pooled resources of compute, network, and storage based on workload requirement

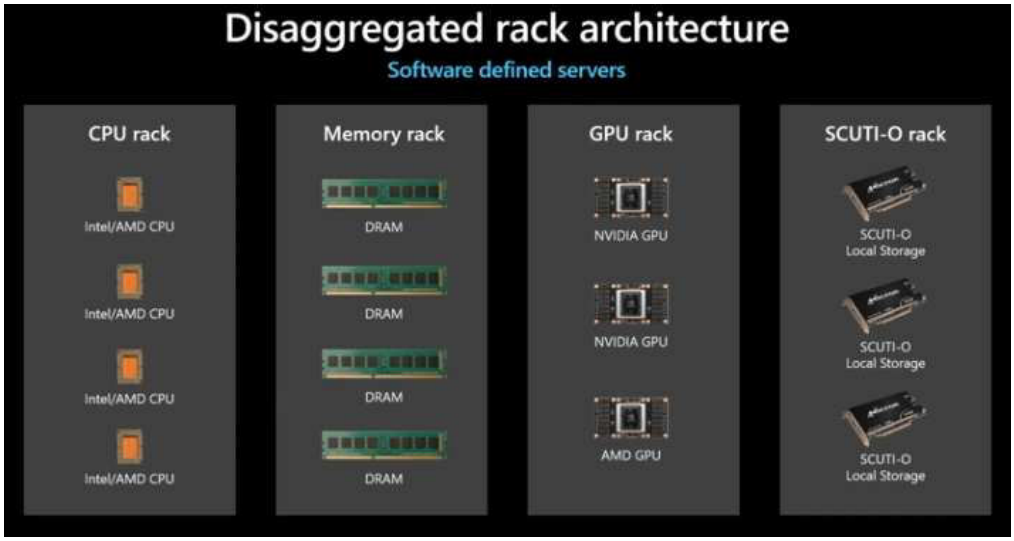
- **Dynamic resource allocation**
  - Virtualization is not enough to improve efficiency
  - Need the ability to dynamically allocate CPU resources across servers and racks, allowing admins to quickly migrate resources to address the shifting demand
  - Drive 100-300% better utilization for virtualized WLs, and 200-600% for bare-metal WLs.

Software defined DCs



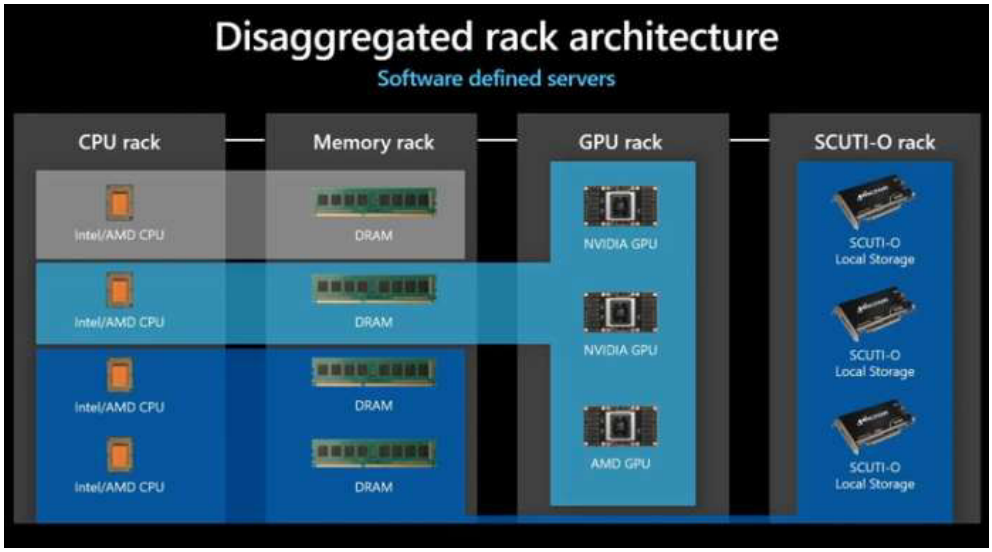
Src: Inside Azure Datacenter Architecture with Mark Russinovich

Disaggregation across racks



Src: Inside Azure Datacenter Architecture with Mark Russinovich

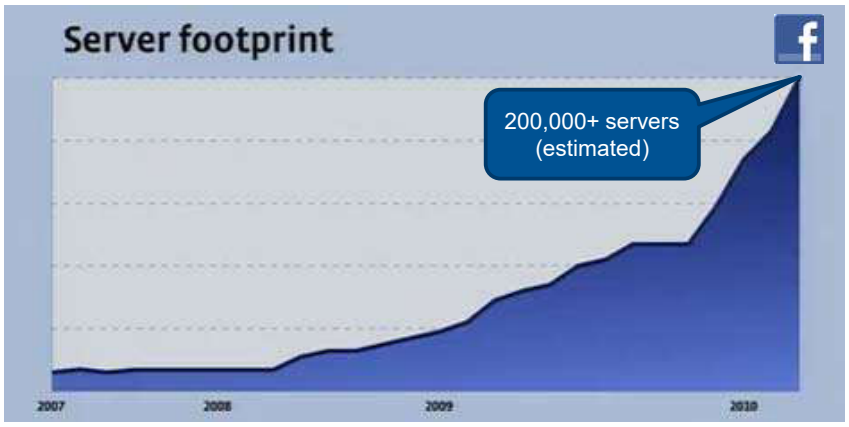
Software defined Servers



Src: Inside Azure Datacenter Architecture with Mark Russinovich

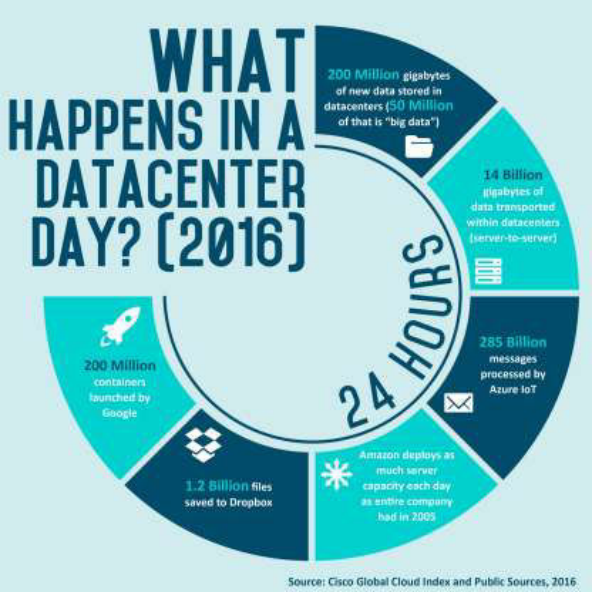
Challenge 5: Managing scale and growth

- In 2016, Gartner estimated that Google has 2.5 million servers
- In 2017, Microsoft Azure was reported to have more than 3 million servers.



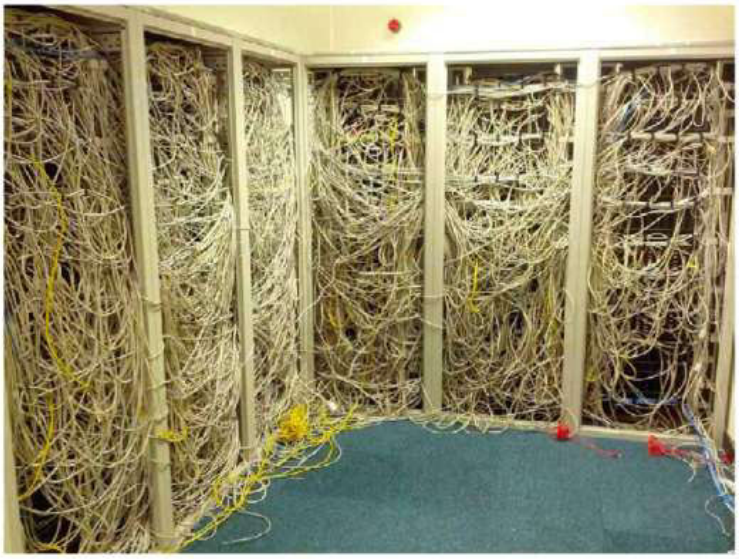
Size and growth of DC (2016-2020)

- The scale and complexity of DC operations grows constantly.
- By 2030, Cisco estimated that we would have 1200 million GB of new data saved each day (600 million GB big data)
- Data center market is expected to grow to USD 438 billion by 2030 (approx.)



Challenge 6: networking at scale

[David Samuel Robbins, gettyimages.ch]



[@AlexCWheeler, Twitter]

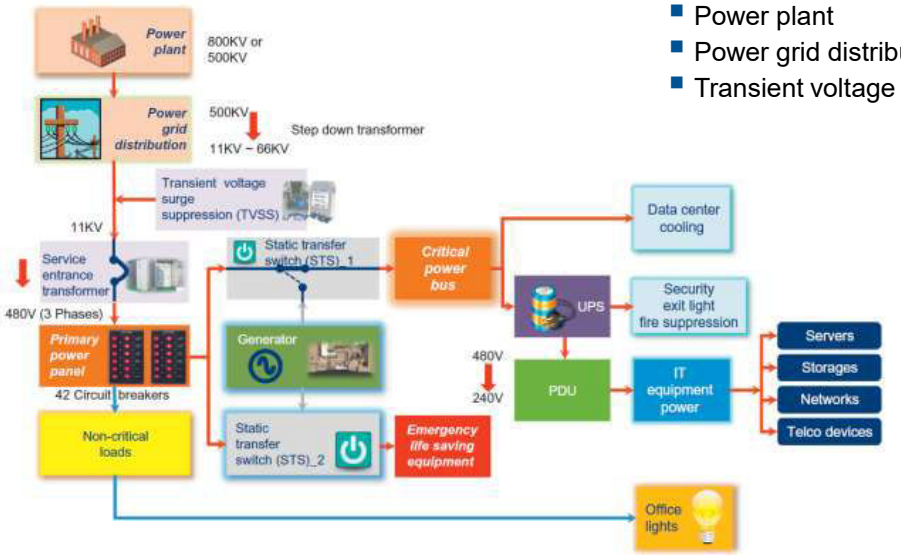
DC Power - Introduction

"DC runs out of power before running out of space"

- One of the biggest cost components for data center operators
- Represent between 10%-25% of data center cost
- Biggest hurdle to grow the cloud business
- Hence, to save on data center costs, the energy bill is one of the major items to pay close attention

Data Center Power Distribution System

- Power plant
- Power grid distribution
- Transient voltage surge suppression

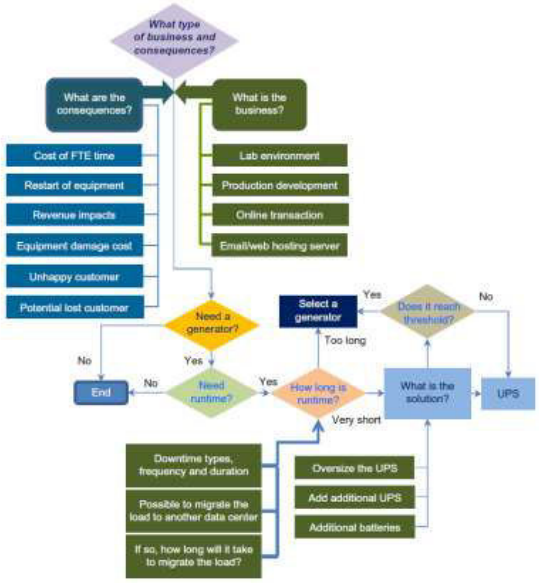


Suggested Learning

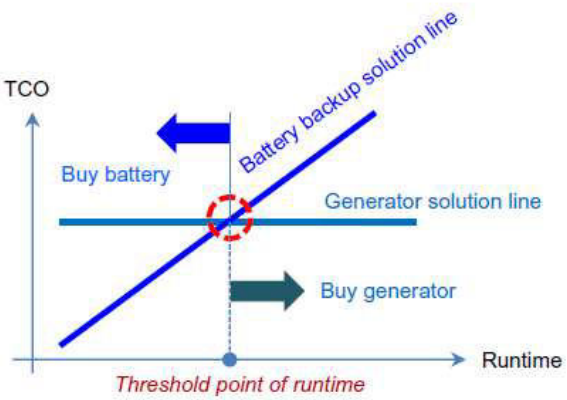
- Power, voltage, current
- Power factor for AC power

Voltage (V)	500					
Amps (A)	100					
Apparent Power (VA)	50,000					
Power Factor	1.0	0.9	0.8	0.7	0.6	0.5
Real Power (KW)	50	45	40	35	30	25

Process to Decide Runtime & Generator

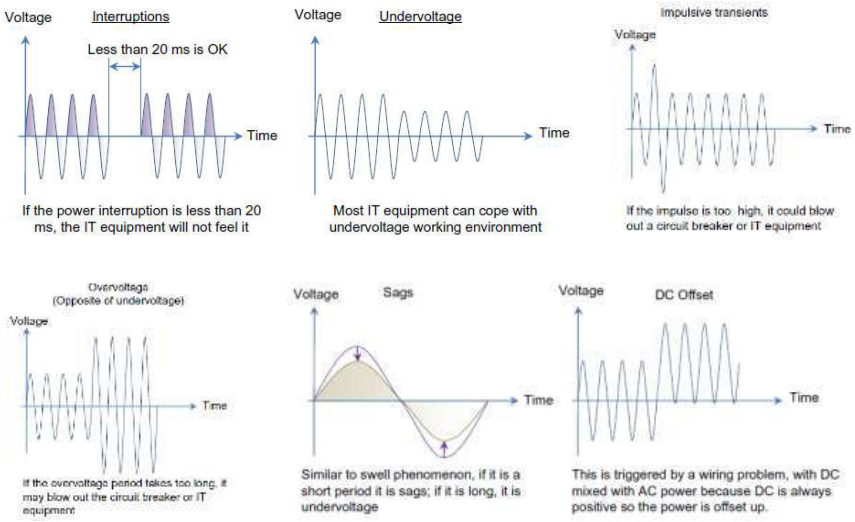


Choice Between Generator & Batteries



Common Power Problems

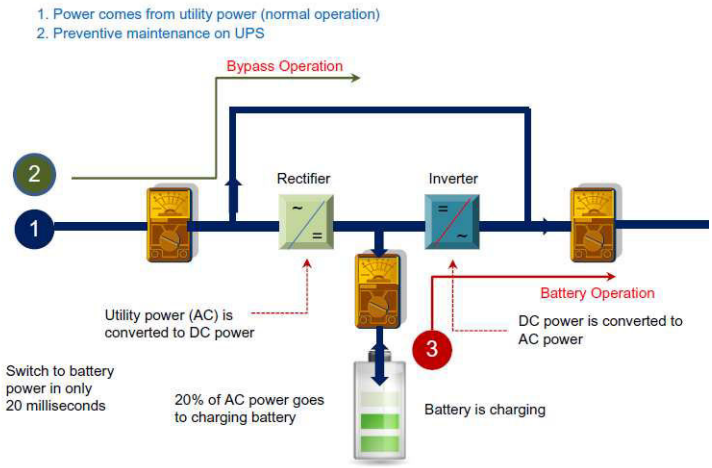
- Interruptions
- Impulsive transients
- Sags and undervoltages
- Swells and overvoltages
- DC offset
- Harmonic distortion
- Voltage fluctuation
- Frequency variations



Solution?

Uninterruptible Power Supply (UPS)

- The purpose of UPS is to provide a reliable and quality power source to a data center.
- It has three alternative paths for three different functions or modes



UPS Topologies

- In order to improve power source quality and protect IT loads, we have to use different types of UPS solutions to solve different types of power source problems.
- There are five basic UPS topologies:
  - Standby or offline single UPS topology
  - Line interactive UPS topology
  - Online double conversion topology
  - Delta conversion UPS topology
  - Rotary UPS topology

Power Distribution Unit

- Power Distribution Unit (PDU)
- Data Center Power Calculation
- Strategies for reducing data center power or increasing power consumption

Power Feed to 42RU Rack

- It is important to map out which power cable to feed to which rack or how many power cables to feed one rack.
- Normally, the data center operation people do not know what IT professionals exactly want.

Considering following assumptions:

- Assumed power factor = 0.8
- 1RU server draw = 350 W
- 2RU server draw = 550 W
- Blade Chassis draw = 4,000 - 4,500 W

- Based on the above assumptions, if we would like to fully populate one 42RU rack with dual-corded 1RU servers, how many power feeds should we have?

Power Feed to 42RU Rack

Simple Calculation for the Power Feed

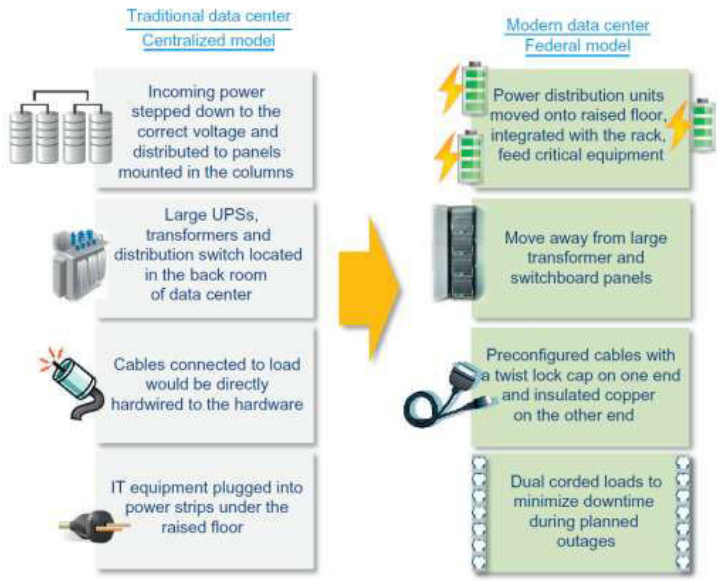
Power Feed	Details of Power Calculation	Available Power	No. of Blade Chassis (4–4.5 kW)	No. of 2RU Server (550 W)	No. of 1RU Server (350 W)
240 V 5 A	$240 \times 5 \times 0.8 = 0.96$ kW	0.96 kW	0	1	2
240 V 10 A	$240 \times 10 \times 0.8 = 1.92$ kW	1.92 kW	0	3	5
240 V 20 A	$240 \times 20 \times 0.8 = 3.84$ kW	3.84 kW	0	7	10
240 V 30 A	$240 \times 30 \times 0.8 = 5.76$ kW	5.76 kW	1	10	16
240 V 30 A 3 phase	$240 \times 30 \times 1.73 \times 0.8 = 9.97$ kW	9.97 kW	2	18	28
240 V 50 A 3 phase	$240 \times 50 \times 1.73 \times 0.8 = 16.61$ kW	16.61 kW	4	30	47

[1] *Nameplate value = assume highest power draw CPU running 100% with full memory load running very low line condition.*

Power Feed to 42RU Rack

Power Cable Feed for One 42 RU Rack with Different Server Configurations					
Type of 42RU Rack Configuration	**No. of Power Feeds (240 V 10 A)**	**No. of Power Feeds (240 V 20 A)**	**No. of Power Feeds (240 V 30 A)**	**No. of Power Feeds (240 V 30 A 3 Phase)**	**No. of Power Feeds (240 V 50 A 3 Phase)**
42 × 1RU servers (350 W)	10	4	3	2	1
21 × 2RU servers (550 W)	6	4	3	2	1
3 × 10RU blade chassis (4–4.5 kW)	7	4	3	2	1
10 × 1RU + 5 × 2RU + 2 × 10RU blades	8	4	3	2	1

Traditional DC vs Modern DC in terms of PDU



How to Calculate Required Power?

- We should make these key assumptions accurate:
  - Required workload
  - Future workload (forecasting growth)
  - Type of power redundancy
- If we overestimate the power load, we will have higher capex and opex.
  - The unit cost for the data center facility or cloud infrastructure will be higher
  - TCO will become higher
  - ROI will be lower.
- However, if we underestimate the workload, we will end up having power disruptions.

Process of Calculating DC Power Requirements

- Understand the business needs.
  - Based on the business requirements or assumptions, we should be able to determine data center space and number of racks.
- Assessment of IT equipment load:
  - Servers
  - Storage devices (SAN, NAS, iSCSI, and FC)
  - Routers and switches (core, distribution, and edge)
  - Load balancing
  - Firewall
- Assessment of Non-IT equipment load:
  - Network monitoring system
  - On-site security
  - Fire suppression system

Process of Calculating DC Power Requirements (Contd..)

- Verify and list power factors of all the devices
  - Type of power supply source (single or three phase)
  - Power rating
  - Voltage

An Example of Power Calculation

- Based on our understanding of the business requirements, we assume that a data center needs 1000 square meters of space, 120 racks, peak power of 10%, and the capability for 25% future growth.

Required IT Workload per Rack				
Devices	**Unit Power Rating (W)**	**Qty. of Devices**	**No. of RU**	**Power Draw (kW)**
Server 1 (1RU)	350	3	3 × 1RU	1.05
Server 2 (2RU)	450	2	2 × 2RU	0.9
Server 3 Blade Chassis	2500	2	2 × 10RU	5
Storage Array 1	300	2	2 × 2RU	.6
Storage Array 2	520	3	3 × 2RU	1.56
Network Devices	100	2	2 × 1RU	0.2
Total Rack Power	4,220	14	39RU	9.31

- Because the data center has 120 racks, the total power draw for the data center would be:
  - 120 racks x 9:31 kW = 1; 117:2 kW

An Example of Power Calculation (Contd...)

- Workload adjustments with peak power capacity

$$1,117.2\ \text{kW} \times 0.25 = 279.3\ \text{kW (Future Workload Estimation)}$$
$$1,117.2\ \text{kW} + 279.3\ \text{kW} = 1,396.5\ \text{kW (Total IT Load Including Future Growth)}$$
$$1,396.5\ \text{kW} \times 1.1 = 1,536.15\ \text{kW (Total Peak Power)}$$
$$1,536.15\ \text{kW} - 1,396.5\ \text{kW} = 139.65\ \text{kW (Net Peak Power)}$$

An Example of Power Calculation (Contd...)

- Calculate the UPS load, considering efficiency = 90% and under the normal operation mode, the power needed for battery charging is very little and can be neglected. However, when the battery is partially and completely discharged, it will draw 20% of the UPS power load

$$1,117.2\ \text{kW} \times 0.25 = 279.3\ \text{kW (Future Workload Estimation)}$$
$$1,117.2\ \text{kW} + 279.3\ \text{kW} = 1,396.5\ \text{kW (Total IT Load Including Future Growth)}$$
$$1,396.5\ \text{kW} \times 1.1 = 1,536.15\ \text{kW (Total Peak Power)}$$

Calculating UPS load		
Total Workload	**Inefficiency of UPS & Battery**	**Subtotal**
1536.15 kW	10% (UPS)	153.62 kW
	20% (Battery Charging)	307.23 kW
Total	30% (Subtotal)	460.85 kW

An Example of Power Calculation (Contd...)

- Estimate lighting and other power load. We assume 2.5 W per square meter

1000 sqm x 2.5 W = 25 kW

An Example of Power Calculation (Contd...)

- Estimate power required by all cooling loads

Summary of All Cooling Loads in the Data Center

Cooling Calculation Items	Cooling Efficiency Factor Calculation	Cooling Load
Chilled Water Power[1]	2,022 × 0.49% = 990.78 kW and 990.78 × 0.7 = 693.55 kW	693.55 kW
Direct Expansion Power[2]	990.78 × 1.0 = 990.78 kW	990.78 kW
Total Cooling Power		1,684.33 kW

[1]*Note: Here, we evenly split the power load into two different cooling methods: chilled water cooling and direct expansion. Direct expansion is not as efficient as a chilled water system.*

[2]*A direct expansion air conditioning (DX) system is another way of cooling. It is different than chilled water cooling. It adopts a refrigerant vapor expansion/compression (RVEC) cycle to directly cool the air in an occupied space.*

An Example of Power Calculation (Contd...)

Summary of all Power Load Estimates in the Data Center	
Power Draw Estimates	**Power Load**
Total Required IT load	1,117.2 kW
Future Load	279.3 kW
Peak Power Draw	139.65 kW
UPS and Battery	460.85 kW
Lighting & Others	25 kW
Total Power Draw	2,022 kW

$$\text{Total Power} = 2,022\ \text{kW} + 1,684.3\ \text{kW} = 3,706.3\ \text{kW} \approx 3.71\ \text{MW}$$

An Example of Power Calculation (Contd...)

- **To tell power utility company how much utility power the data center needs.**
- This is the demanded power load for the data center, but when the data center operation people are asking for utility power voltage, the total power draw should be no more than a certain percentage of a threshold, such as **80% or 85%**. This is dependent on national utility regulations.
- Therefore, the total power should be multiplied by this utility threshold factor.
- In this example, we use 80%. Therefore, the data center operator should ask125% more than the actual required power from the utility company. Here is the estimate data:

Total Utility Power = 3:71 MW x 125% ~ 4.63 MW

An Example of Power Calculation (Contd...)

Standby Generator Capacity Calculation			
Load Components	**Power Load**	**Generator Factor[1]**	**Subtotal**
Total Power Draw	2022 kW	1.25	2.53 MW
Cooling Load	1684.33 kW	1.55	2.61 MW
Total Required Generator Capacity			5.14 MW

[1]*Note: The generator factor should be decided differently for different power draws. For the cooling system, many mechanical components, such as pumps are involved, so the factor would be greater. In this case, we use 1.55.*

Power Usage Effectiveness

- To tell power utility company how much utility power the data center needs.
- This is the demanded power load for the data center, but when the data center operation people are asking for utility power voltage, the total power draw should be no more than a certain percentage of a threshold, such as 80% or 85%. This is dependent on national utility regulations.
- Therefore, the total power should be multiplied by this utility threshold factor.
- In this example, we use 80%. Therefore, the data center operator should ask125% more than the actual required power from the utility company. Here is the estimate data:
- **The total utility power requirement is more than four times the critical IT load.**

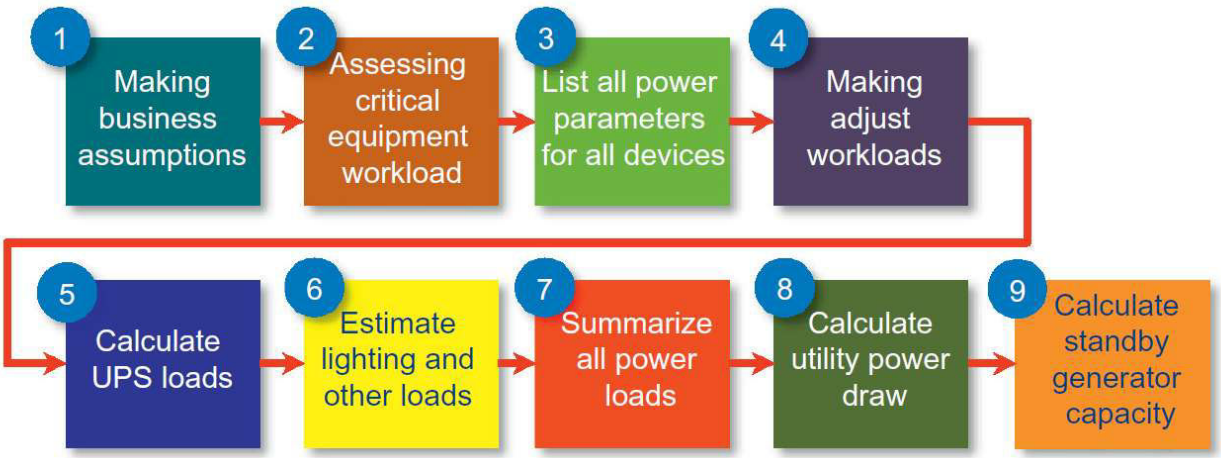
Total Utility Power = 3:71 MW x 125% ~ 4.63 MW

Power Usage Effectiveness (PUE)

- Ideally, all power draw should be consumed by IT equipment but due to power transmission and conversion, power loss is inevitable. For a green data center, we use Power Usage Effectiveness (PUE) to measure the energy efficiency of a data center:

$$\text{PUE} = \frac{\text{Power to Data Center}}{\text{Power to IT Load}}$$

Nine Step - Power Calculation



Strategies for Power Saving

- Improve efficiency of ups or remove redundant power equipment
- Improve power configuration
- Reducing data center capacity

Improve Efficiency of UPS

- When we select the UPS model for a data center power system, we should always check the TCO rather than just the purchase cost.

In summary, UPS B is \$37,449 (or \$51,246 − \$13,797) better than UPS A per annum. If the UPS life cycle is 7 years, UPS B can save \$262,143 (\$37,449 × 7) over seven years without taking into consideration NPV. Today, real cost savings should rely on both capex and opex, which is TCO, rather than just capex or purchase cost.

UPS Dimensions	UPS A	UPS B
UPS efficiency	85%	95%
UPS power	150 kW	150 kW
Required power inputs	176 kW	157 kW
Power loss after UPS	26 kW	7 kW
Cooling cost to support extra heat (\$0.15 kW/hour)	\$17,082 per annum	\$4599 per annum
Cost of inefficiency	\$34,164 per annum	\$9,198 per annum
Total cost of inefficiency	\$51,246 per annum	\$13,797 per annum

Cooling in Datacenter

Cooling in Datacenter - Introduction

- Cooling is one of the three keys or critical elements for any enterprise class data center.
- Without a cooling system, the data center would not work properly.
- It could quickly go bust.
- It is very important, but also costly. For some high-density data centers, the cooling capex could be as high as 50% of the total project cost. Therefore, if we can save costs in cooling, we can save large proportion of data center cost (both capex and opex).

Cooling in Datacenter - Definition

- Cooling is the process of removing heat and maintaining the right temperature conditions either for humans or objects.
- The right cooling conditions consist of three critical metrics:
  - Temperature
  - Humidity
  - Airflow

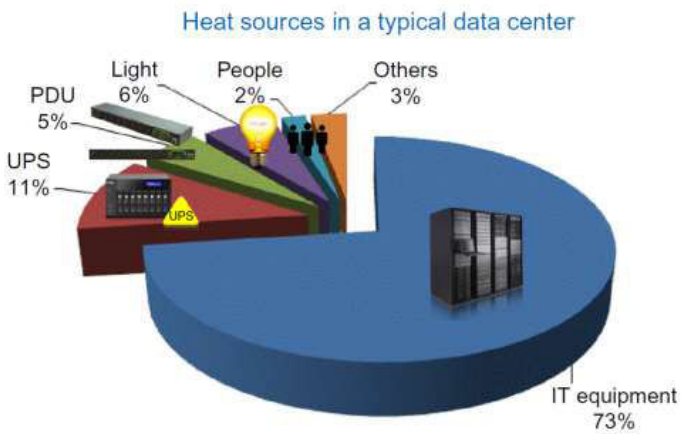
Types of Cooling

- Comfort Cooling
  - HVAC systems designed primarily for human comfort, not for the precise temperature and humidity control required for critical IT equipment

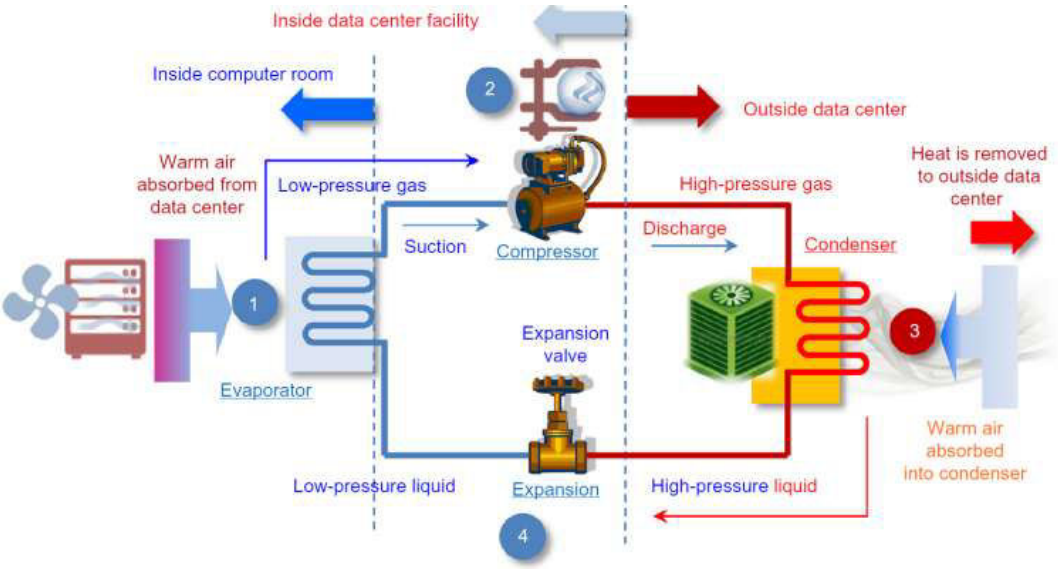
- Precision Cooling
  - purpose of precision cooling is to cool IT equipment in a data center that requires precise and stable environment control.
  - A precision cooling system has to work around the clock 24 x 7 and every day. It has to maintain a constant temperature and precise humidity and airflow because the IT equipment constantly produces a concentrated heat load and is very sensitive to both temperature and humidity changes.

Sources of Cooling

- The IT equipment may generate as much as 80% of the total data center heat. We can categorize these heat sources into four different origins or levels:
  - CPU socket or core
  - Rack unit level or individual equipment
  - Rack
  - Room



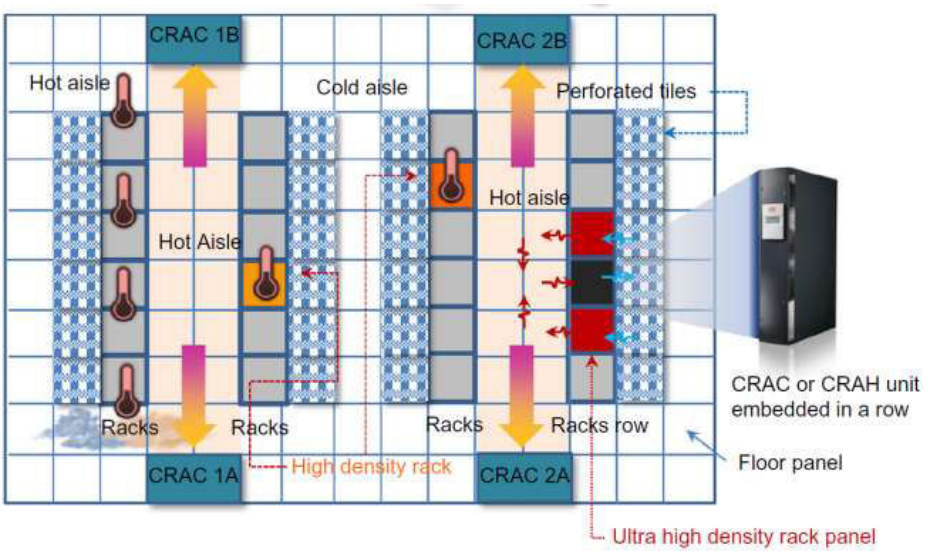
Refrigeration Cycle Process



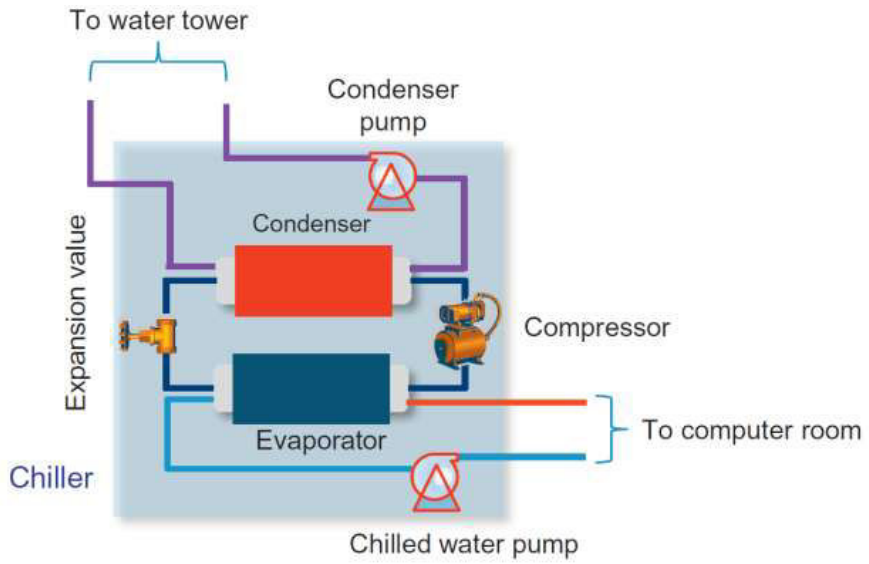
Data Center Cooling Components

- Inside of a data center, we must have four controllable devices to implement the refrigeration cycle.
- Without these critical components, it would be impossible to maintain the right temperature and
- humidity in a data center. These four components are:
  - Computer room air conditioning (CRAC) units
  - Computer room air handling (CRAH) units
  - Chiller
  - Humidifier

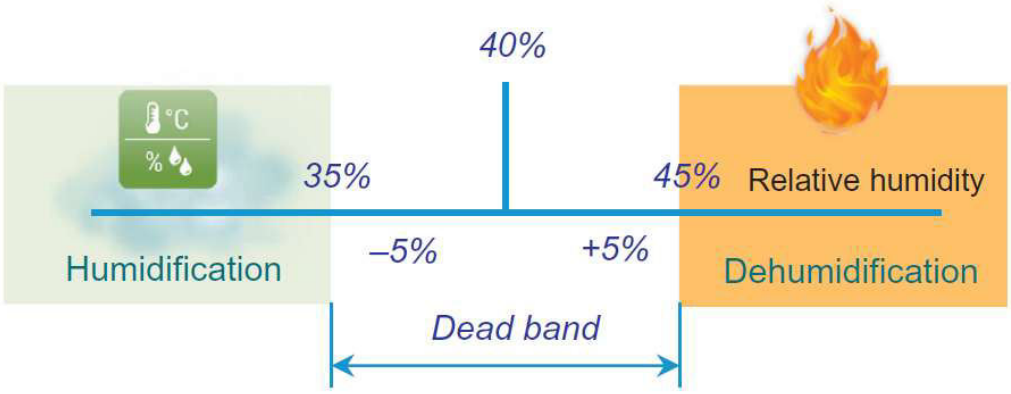
Composition of Cooling Components



Chiller and CRAH



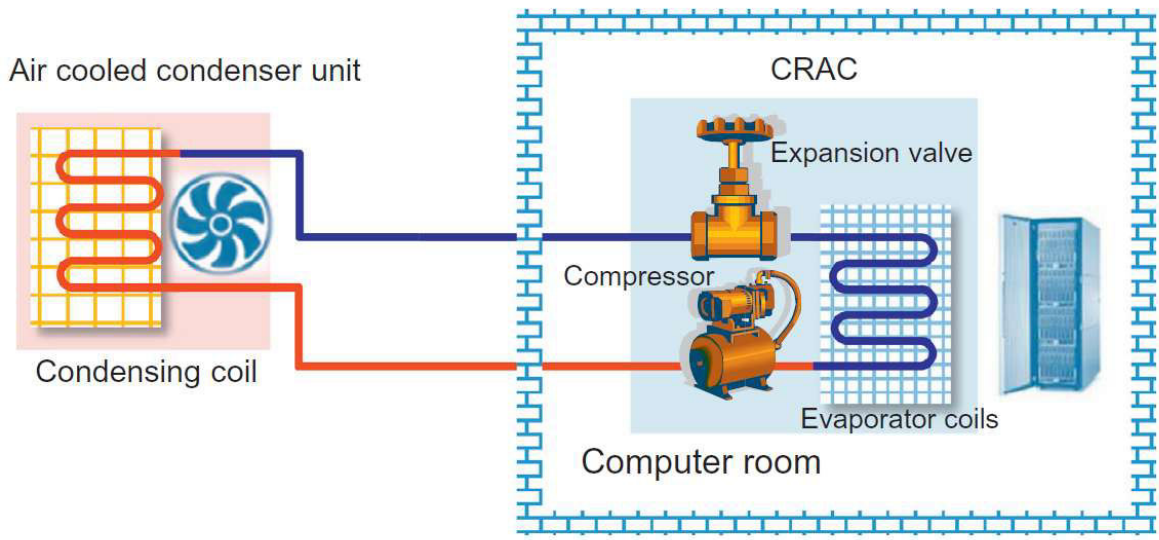
Humidification vs Dehumidication



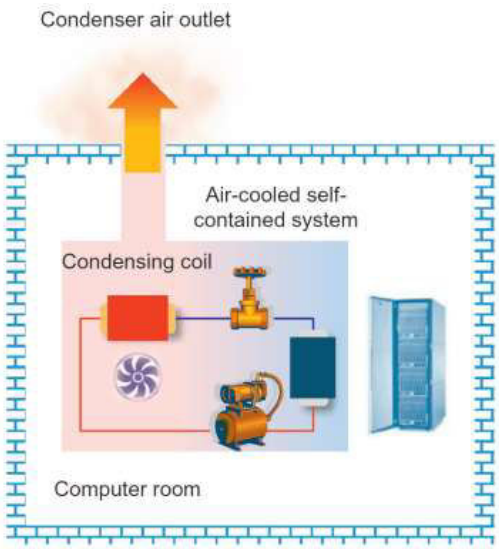
Technologies for Removal of Data Center Heat

- Air cooled DX system (Two pieces)
- Air cooled self-contained system (one piece)
- Ceiling mounted system
- Glycol cooled system
- Water cooled system
- Chilled water system

Air Cooled DX System (Two Pieces)

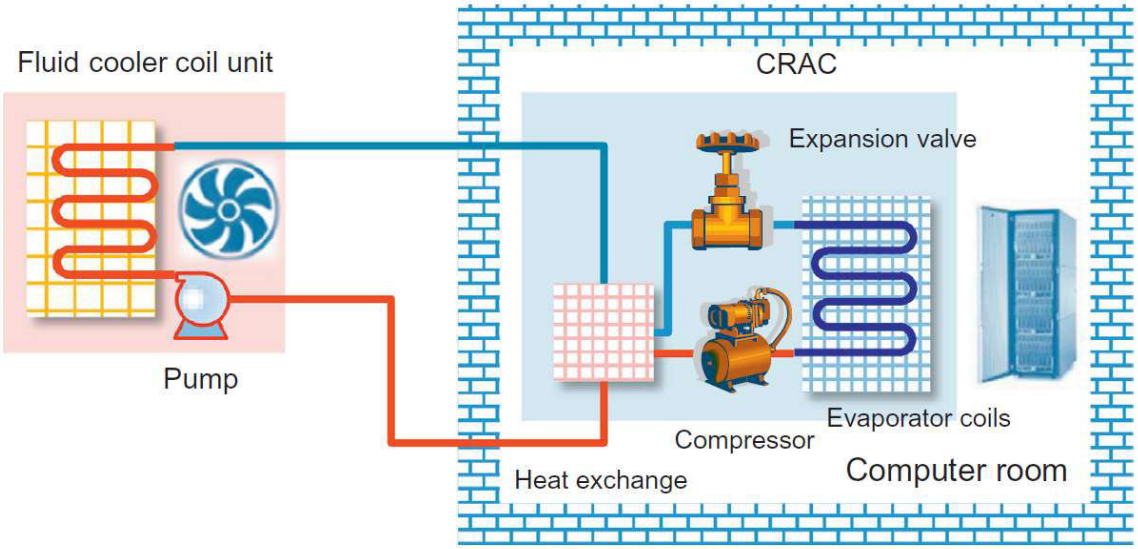


Air Cooled Self Contained System

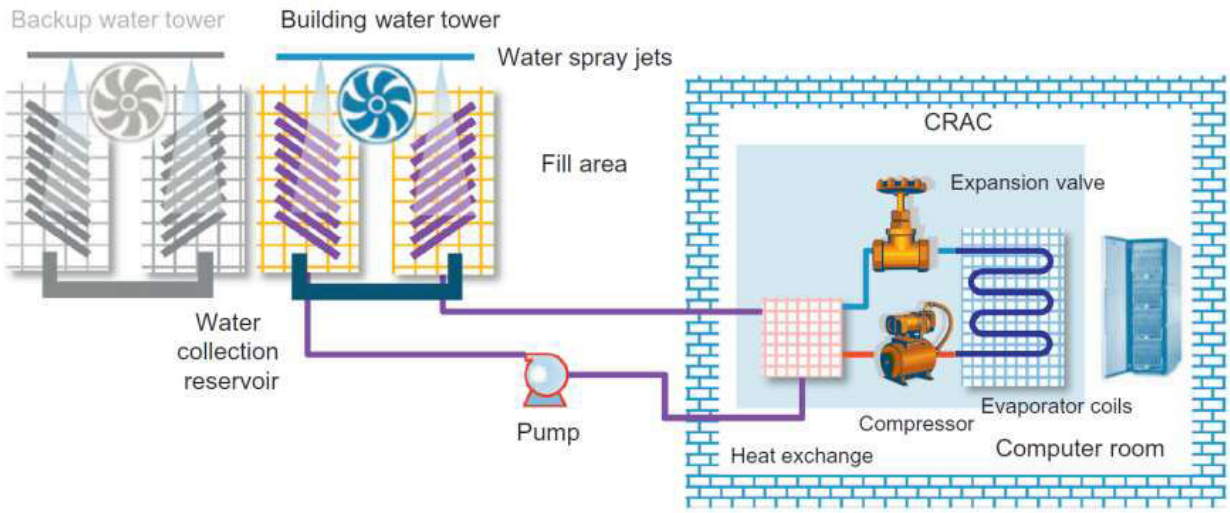


Ceiling Mounted System

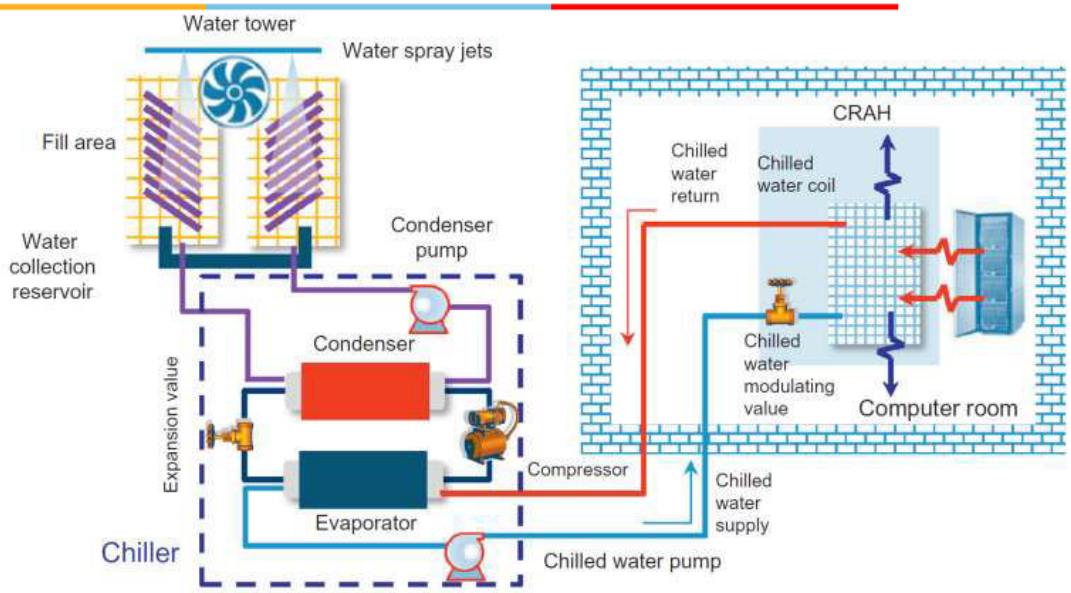
Glycol Cooled System



Water Cooled System



Chilled Water System

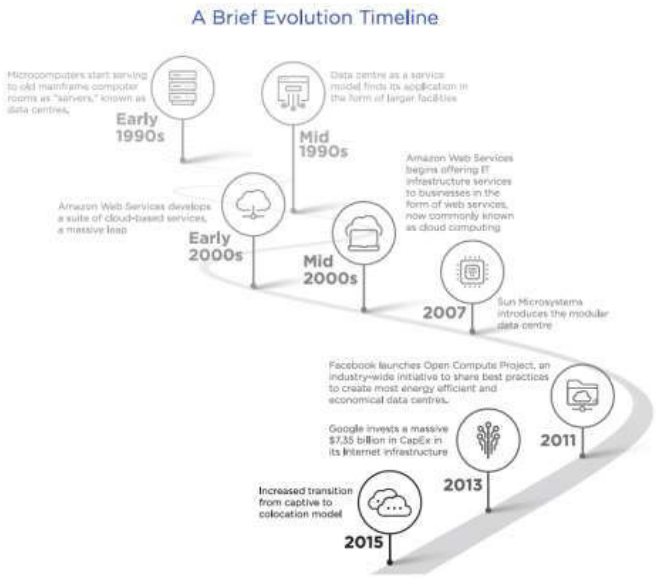


Cooling Technologies Comparison

Cooling Technologies	Advantages	Disadvantages	Applications
Air-Cooled DX	1. Lowest TCO 2. Easiest to maintain	1. Refrigerant piping must be installed 2. The pipe cannot be run very long, less than 60 m 3. Multiple CRAC units impossible for one system 4. It is a one-to-one relationship	For wiring closets, small & medium data centers with moderate availability requirements
Air-Cooled Self-Contained	1. Lowest installation cost 2. Very reliable 3. Portable 4. Easy and quick to be adopt	1. Smaller heat removal capacity 2. Requires ductwork 3. Less energy efficient 4. Not for precision cooling application	For wiring closets, laboratory environment with moderate availability requirements Use to fix hot spots in data center
Celling Mounted	Do not need IT floor space	Complex installation and maintenance	Cooling 3-17 kW IT equipment
Glycol Cooled	1. High reliability 2. Not much space required 3. Glycol pipes can be longer than DX 4. It is capable of supporting several CRAC units with one outdoor unit 5. When temperature drops, it can directly flow to specially installed economizer coil, which provides "free cooling." 6. Reducing opex	1. Requires additional components (pump package, valves) 2. Increased capex in comparison with DX 3. Additional glycol maintenance required	For small to medium data center with moderate availability requirements
Water Cooled	1. The system has the highest reliability 2. Condenser water can easily be run long distance 3. One outdoor cooling tower can support multiple CRAC units 4. Less expensive than chilled water system	1. High initial capex for cooling tower, pump, and pumping installation 2. Very high maintenance cost due to frequently cleaning and water treatment requirements 3. If no dedicated cooling tower, it may reduce system reliability	In conjunction with building cooling system for small, medium, and large data centers with moderate to high availability requirements
Chilled Water	1. CRAH is often less expensive and capable of removing a large amount of heat with the same footprint of air conditioners 2. Chilled water pipe loops can run very long distances 3. One cooling tower and one chiller unit can support several CRAH units 4. It can be extremely reliable 5. For large installations, the unit cost (per kW) is the lowest	1. It has the highest capex cost for installation below 100 kW of workload 2. CRAH unit removes more moisture than CRAC unit so it needs extra humidifier	In conjunction with other systems in medium and large data center with moderate to high availability requirements or high availability dedicated solution in large enterprise data centers

Rack, aisle, & Pod

Data Center Evolution



Rack

Rack - Today's Environment

- Data centers are 24/7/365 mission-critical systems
- Growing business demands needs even more processing power and data storage
- New generations of high-density servers and networking equipment have increased rack densities and overall facility power requirements.
- The need now exists for taller, wider and deeper racks to accommodate the changes in IT equipment and densities
- High density configurations also increase cabling density which can make heat removal more difficult.
- This also reduces access to equipment. Racks must position and route cabling correctly and provide ready access to equipment.
- Failures caused by high temperature or humidity in the rack are clearly unacceptable.
- Optimized airflow, organized cabling and a monitored sensor network in the rack provide a safe environment for heat emitting equipment.

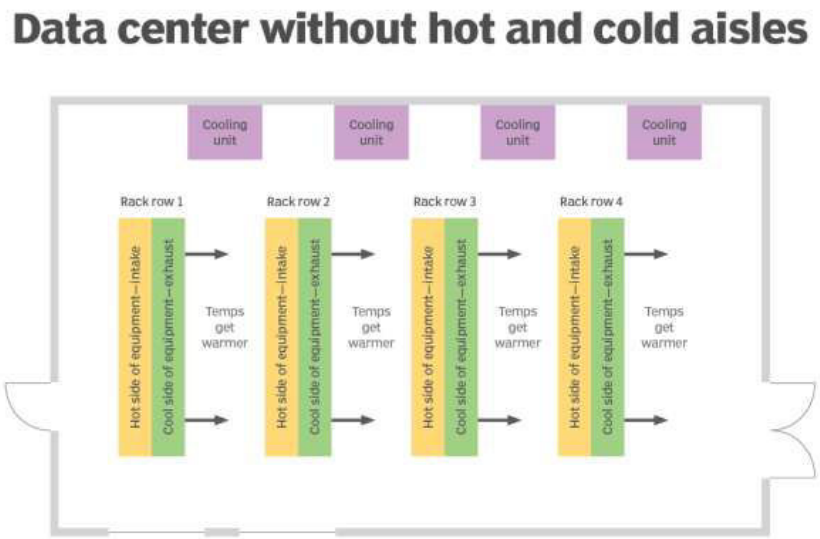
Evolution in Rack Technologies

- The need to improve IT system performance and ensure reliability has driven the evolution in rack technology.
- Today's racks offer features that improve equipment installation speed, while increased height, width and depth options better contain the larger equipment that is being introduced into the data center.
- No longer strictly technicians or maintenance personnel to install components.
- Tool-less installation helps to save time during equipment deployment.
- The ability to add new equipment or change configurations at potentially a moment's notice makes component mounting a fast "slide in, slide out" procedure, a necessity in today's data center.

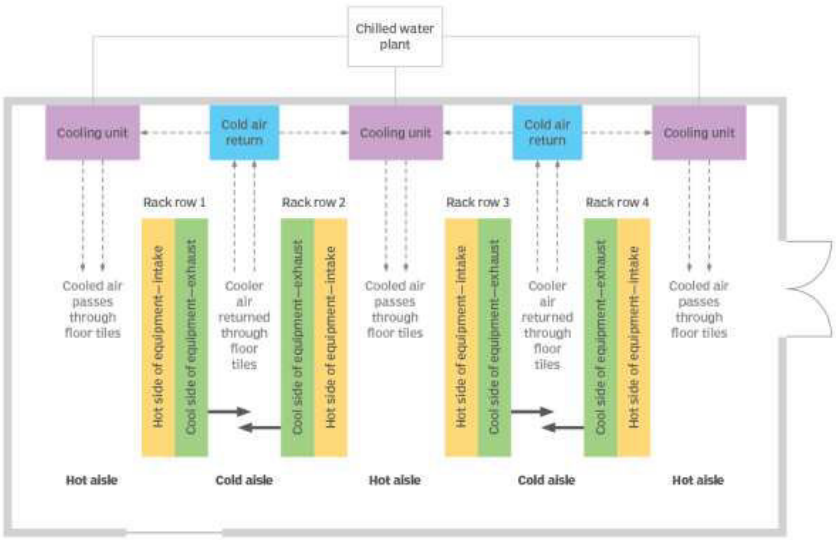
Evolution in Rack Dimensions

- Taller racks, beyond the common 47U (2200 mm) are also becoming more popular as data centers with room to expand vertically take advantage of headroom.
- The depth of rack-mounted equipment is also increasing. For example, the greatest server depth requirements in the past resulted in a rack depth of 1200 mm.
- The introduction of side-breathing equipment makes rack width a factor as well.
- Racks widths up to 1000 mm are becoming common to meet equipment manufacturers' specifications that call for clearances from 150 - 280 mm.
- These clearances are required for proper airflow to the equipment and to provide ample cable management space. Often, airflow management accessories are required to make them compatible with hot aisle/cold aisle arrangements.

Data Center without Hot and Cold Aisle



Data Center with Hot and Cold Aisle



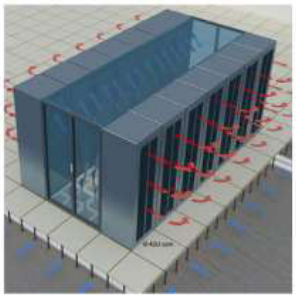
Hot and Cold Aisle

- The hot and cold aisles in the data center are part of an energy-efficient layout for server racks and other computing equipment.
- The goal of a hot/cold aisle configuration is to manage airflow in a way that conserves energy and lowers cooling costs.
- Hot aisle/cold aisle data center design involves lining up server racks in alternating rows with cold air intakes facing one way and hot air exhausts facing the other.
- The rows composed of rack fronts are called cold aisles. Typically, cold aisles face air conditioner output ducts.
- The rows the heated exhausts pour into are called hot aisles. Typically, hot aisles face air conditioner return ducts.

Hot/Cold Aisle & Containment System

- Data center containment is the separation of cold supply airflow from hot air coming out of equipment exhaust.
- A containment system isolates hot aisles and cold aisles from each other and prevents hot and cold air from mixing.
- By separating the air, you can save revenue by reducing your energy consumption, and improve overall equipment performance and efficiency.
- The amount of energy used to maintain acceptable temperatures in the data center can be greatly reduced by containment.

Containment System

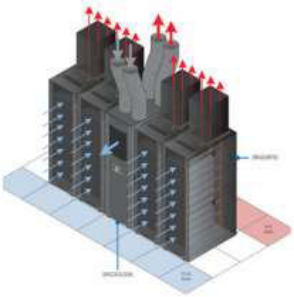


Cold Aisle Containment

Enclose a cold aisle with ceiling panels above the aisle, between adjoining racks, and with doors at the end of the aisle.

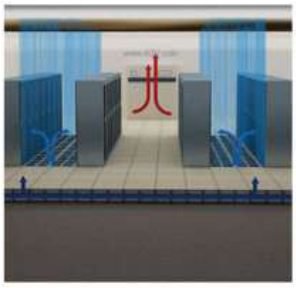
Hot Aisle Containment

Keep hot exhaust air, emitted from server racks, separated from the cold air.



Chimney Systems

Push hot air through the chimney up to the ceiling return air plenum.



Curtain Systems

An economical energy saving solution often utilizing curtain materials that have been around for a long time in clean rooms.

Containment System

Hard Panel Systems

This unique hard panel roof fully segregates hot and cold air, improving the cooling capacity of any data center.

Modular Systems

Get the benefits of traditional containment with greater flexibility and lower cost.

Aisle End Doors

Great for hot aisle/cold aisle containment applications, adapt to fit to existing racks, and can be configured to fit any site needs.

Inrow Cooling

Precisely cool and condition air in close proximity and is targeted cooling at the rows of server cabinets that fill the data center.

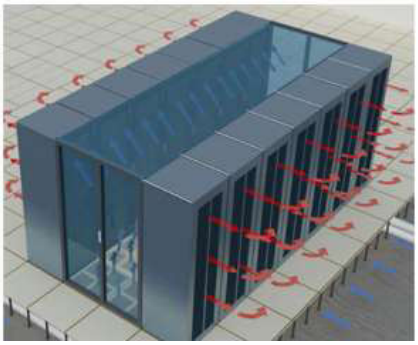
Cold Aisle Containment System

Cold Aisle Containment

The Cold Aisle Containment (CAC) approach encloses the cold aisle with ceiling panels above the aisle between adjoining racks and with doors at the end of the aisle. This allows the cold air from the perforated floor tiles in front of the cabinets to be contained and delivered to the server equipment air inlets. The CAC prevents cold air from mixing with warm air or being obstructed by surrounding equipment/sources before it reaches the servers. Because CAC possesses this ability, it is called focused cooling.

Benefits

- ✓ Provides focused cooling at the rack
- ✓ Can cool higher heat loads typically up to 10-15kW (raised floor)
- ✓ Minimizes cold and hot air mixing so the return temperature to the cooling unit is higher
- ✓ Cold aisle containment combined with Inrow cooling can cool more than 30kW heat load per rack



Hot Aisle Containment System

Hot Aisle Containment

In its simplest form, Hot Aisle Containment (HAC) consists of a barrier that guides hot aisle exhaust airflow upward back to the CRAC (computer room air conditioner) return. This method keeps hot exhaust air emitted from server racks separated. Keep in mind though, while HAC improves energy efficiency compared to no HAC; it can add cooling (and power) load because of larger fans to overcome additional pressure drops.

Benefits

- ✓ One redundant cooling unit can serve several aisle containments.
- ✓ HAC with Inrow cooling improves energy efficiency compared to traditional raised floor cooling with HAC.
- ✓ Typically a rack hat style containment with end row doors.
- ✓ Minimizes cold and hot air mixing so the return temperature to the CRAC is higher.
- ✓ HAC with Inrow cooling can cool more than 30kW heat load per rack.

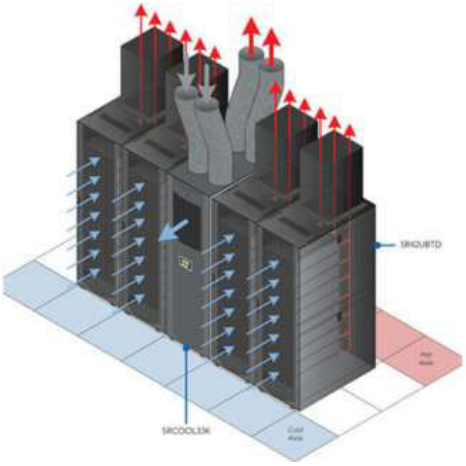
Chimney Containment System

Chimney Systems

A chimney system consists of a solid metal chimney that extends from the top rear of the IT racks. The hot air is pushed through the chimney up to the ceiling return air plenum. This type of system is highly compatible with most data centers due to the fact that it can be installed on an already existing enclosure. They can be installed not only on an individual cabinet but also on a system which can span several cabinets.

Benefits

- ✓ Hot air is completely eliminated from the space and the room remains cool, in turn lowering operational costs.
- ✓ Increased flexibility with installation and configuration.



Curtain Containment System

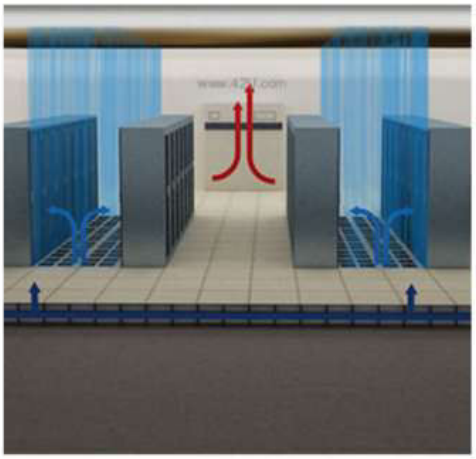
Curtain Systems

Data center curtains are an economical energy saving solution often utilizing curtain materials that have been around for a long time in clean rooms. Combining the curtains with an aluminum track system allows you to attach the data center curtains to the drop ceiling and to the tops of the racks.More intelligently designed aisle containment curtains utilize UL listed fuse links. These melt at a temperature that is below the temperature at which your fire suppression system would be activated – 135 degrees is the melt point for them to drop away. This allows the curtains to drop to the floor before the temperature would be sufficient to trigger a data center fire suppression system.

Cold aisle/hot aisle containment curtains are available in flexible strip and soft wall panel formats that can be customized to meet the specific requirements of any facility.

Benefits

- ✓ Increased energy efficiency and performance capability of climate control.
- ✓ Superior output density is facilitated, thanks to guaranteed cold air supply.
- ✓ Easily installed and retrofitted.
- ✓ An inexpensive way to boost the performance of your existing data center, lengthening the investment cycle until a replacement needs to be purchased.



Hard Panel Containment System

Hard Panel Systems

Hard panel systems work with both cold or hot aisle containment applications and are pre-assembled for easy installation. This unique hard panel roof fully segregates hot and cold air, improving the cooling capacity of any data center. It attaches to the top of the rack, fully supported by the racks, and lies flat to avoid interference with other overhead obstructions and extrusions. Should a fire occur, the roof panels drop away to allow an overhead suppression system to work within the aisle.

Benefits

- ✓ Perimeter compression gaskets efficiently seal and minimize air leakage.
- ✓ Sturdy aluminum framing.
- ✓ Multiwall panel option adheres to new NFPA codes*.
- ✓ Ergonomically designed with angled handles to reduce pinch points.

**Always check with local codes before any containment installation.*

Modular Containment System

- Modular Containment is an off-the-shelf, out-of-the-box containment solution that can be self-installed in minutes.
- Provides data center managers an alternative to the complex traditional containment systems.
- Helps data center managers achieve similar results to traditional containment (i.e. improved inlet temperatures), without all the complexities of measuring, designing, and installing.
- Modular aisle containment system usually attaches to the racks magnetically, creating a truly tool-less design that can be moved, reconfigured, and reused as many times as needed.
- The two main components of the Modular Containment are the Rack Top Baffle and the Bi-Directional Doors, with an optional row gap filler. The Rack Top Baffle can be purchased in either a 30° angle from horizontal position (best for cold aisle containment), or in a vertical 90° angle from horizontal position (best for hot aisle containment). Both components attach directly to the racks.
- The open-sealing design ensures fire safety with an elegant appearance that brings professional aesthetics to your server room.

Aisle End Doors in Containment System

innovate achieve lead

Aisle End Doors

Closing the end of the aisles with a functional door system can not only optimize the design and operation of your data center, but it can also improve the efficiency. Aisle end doors are great for hot aisle/cold aisle containment applications, adapt to fit to existing racks, and can be configured to fit any site needs.

Benefits

- ✓ Improved efficiency
- ✓ Easy to configure to existing setup
- ✓ Eliminates hot/cold air mixing
- ✓ Increased cooling capacity

Inrow Cooling in Containment System

innovate achieve lead

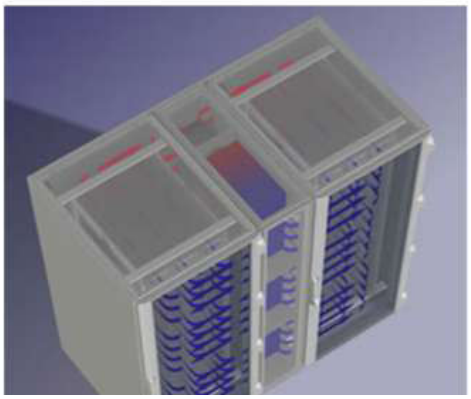
Inrow Cooling

Inrow cooling precisely cools and conditions air in close proximity and is targeted cooling at the rows of server cabinets that fill the data center. They can be installed on the floor or suspended from overhead making them closer to the actual rack. This setup offers both capacity and efficiency gains.**Cooling Medium**

Whether floor mounted or overhead, inrow cooling solutions consist of fans and a cooling coil. Depending on the product, the cooling coil will utilize chilled water or refrigerant as the cooling medium. In the case of refrigerant based units, they require a connection to a remote condenser system; while chilled water based units traditionally connect to chiller systems.

Benefits

- ✓ Warm and cool air does not have far to travel.
- ✓ Allows the units to dissipate high heat loads quicker.
- ✓ Scalable approach to cooling your data center.



Benefits of using Hot/Cold Aisle

- Manage heating, ventilation and air conditioning (HVAC) systems in the most effective way
- Conserve energy.
- Saves energy cost considerably

Best Practices to use Hot/Cold Aisle

- Raise the floor 1.5 feet so air conditioning equipment can push air through that space.
- Deploy high cubic-feet-per-minute rack grills that have airflow output in the range of 600 CFM.
- Place devices with side or top exhausts in their own part of the data center.
- Install automatic doors in the data center.

Migrating to Hot/Cold Aisle

- Hiring data center architects and specialized environment and power engineers;
- Costs to move equipment racks and reconfigure aisles;
- Costs to reconfigure routing of electrical cables and power distribution units;
- HVAC system modifications or replacement;
- System downtime during relocation of equipment and racks;
- Labor costs to perform all of the above.

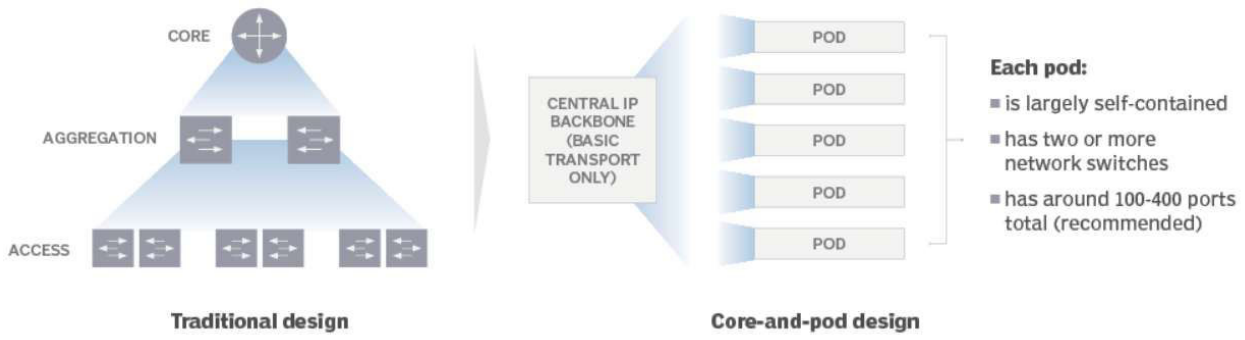
Pod

Data Center facility, comprised of
IT Rooms, comprised of
IT Pods, comprised of
IT racks comprised of
IT devices

Pod

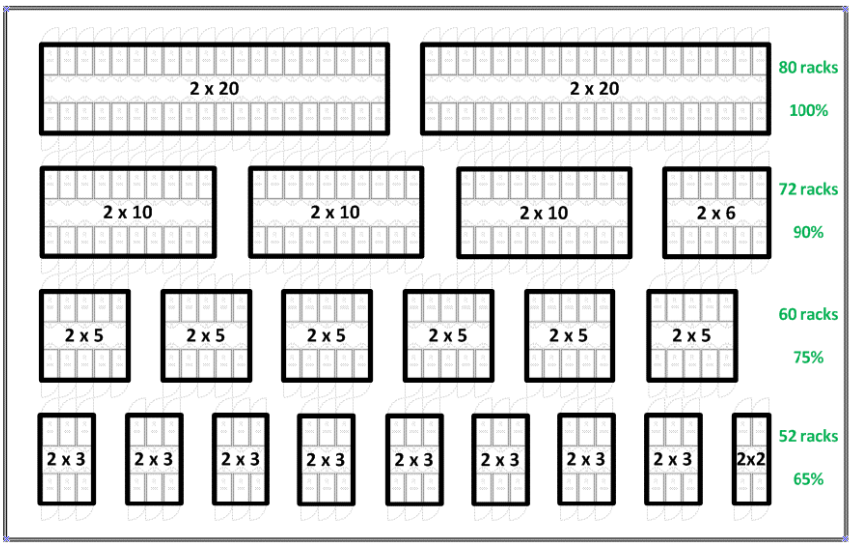
- A pod, is a largely self-contained unit with as few as two network switches, ideally with just 100 to 400 ports total.
- A central backbone connects the data center pods, performing only basic transport functions.
- This type of design enjoys widespread usage among web-scale cloud providers, which have found it "the only way to sustain high growth rates and high availability,“
- The modular design makes the data center network more responsive to innovation, allowing managers to add capacity on demand -- simply deploying additional pods to shoulder new workloads as necessary.
- Data center pods also enable staggered upgrades -- limiting hardware and software changes to one confined area at a time
- Importantly, the pod architecture also supports the core DevOps concept of continuous processes.
- With self-contained zones, network pros can more easily embrace iterative development, continuous testing and blue/green deployment in the network.
  - Rather than waiting several years for a major refresh, data center pods constantly improve, individually and as a group.

Pod



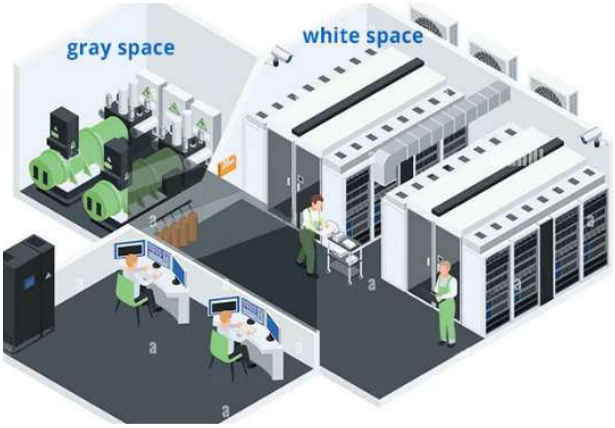
Pod

The larger the pod size, the more racks can be placed into a given space (percentages represent the ratio of racks to the base case of 80 racks)

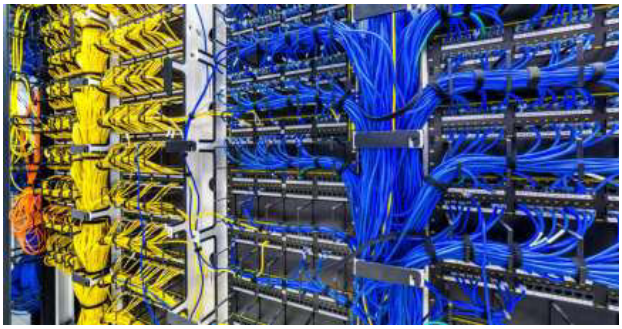


White Space vs Grey Space

- White space in data center refers to the area where IT equipment are placed.
- Grey space in the data centers is the area where back-end infrastructure is located.



Data Center Network Architecture



Data Center Network

innovate achieve lead

- **Data Center Networks** are large clusters of servers interconnected by network switches.
- It interconnects all of the data center resources together.
- Need to be scalable and efficient to connect hundreds of thousands of servers to handle the growing demands of cloud computing.

These servers are used to **host applications** which provide different **concurrent services**. Example)

- Web services like DNS, Web server, Mail server, gaming server, chat server.
- Compute services like suggestion systems, indexing and scientific computing.

DCN Usage Scenarios:

- Compute Intensive: Heavily loaded servers, but low inter-server comm. Ex) HPC
- Data Intensive: Huge intra-DCN data transfer, but low load at servers. Ex) Video and File Streaming
- Balanced: Communication links and computing servers are proportionally loaded. Ex) Geographic Information System

Data Center Networks

- 20-40 servers per rack
- Each server connected to 2 access switches with 1 Gbps (10 Gbps becoming common)
- Access switches connect to 2 aggregation switches
- Aggregation switches connect to 2 core routers
- Core routers connect to edge routers
- Aggregation layer is the transition point between L2-switched access layer and L3-routed core layer
- Low Latency: In high-frequency trading market, a few microseconds make a big difference.
  - Cut-through switching and low-latency specifications.

Data Center Networks (contd.)

- Core routers manage traffic between aggregation routers and in/out of data center
- All switches below each pair of aggregation switches form a single layer-2 domain
- Each Layer 2 domain typically limited to a few hundred servers to limit broadcast
- Most traffic is internal to the data center.
- Network is the bottleneck.
  Uplinks utilization of 80% is common.
- Most of the flows are small.

Data Center Networking Requirements

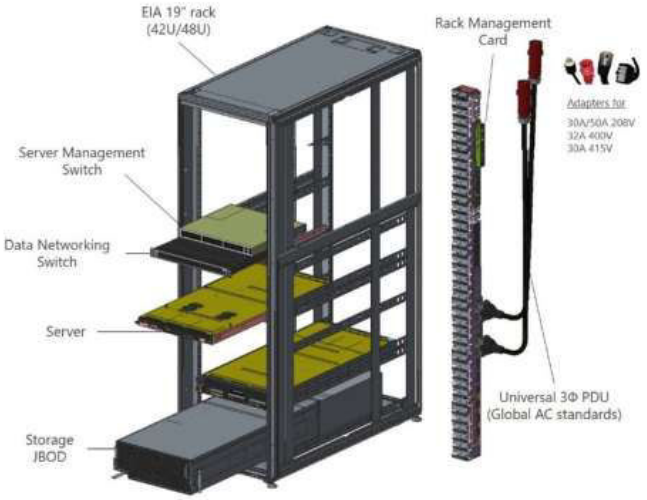
- Needs to be Scalable, Secure, Shared, Standardized, and Simplified (5 S's)
- Converged Infrastructure: Servers, storage, and network have to work together
- Workload Mobility: Large L2 domains required for VM mobility
- East-West Traffic: Significant server-to-server traffic as compared to server to user. One Facebook request required 88 cache looks, 35 database lookups, 392 backend RPC calls.
- Storage traffic on Ethernet: Congestion management on Ethernet

Server Rack Design & Infrastructure

- **Data Center Networks** equipment or the server rack equipment play an important role in data centers in terms of managing the space inside data center building, keeping track of heat generation from each device and for easy troubleshooting.

Server Rack Design & Infrastructure (contd.)

- Having properly arranged and managed server rack equipment in a data center will reduce the complexity of managing servers and also will be helpful in performing equipment audits which is one of the important steps in migration planning process.



Other Related Equipment in Server Rack

Fig 16 : CISCO – SLM2048 – 48 Port 10/100/1000 Gigabit Smart Switch

Fig 17: CISCO 1800 Series Router

- Switch - Allows data to enter from one source, and then directs it to a specific device (LAN)
- Router - Connects two or more different networks together (WAN)

Supporting Equipment in Server Rack

Fig 18 : Patch Panels

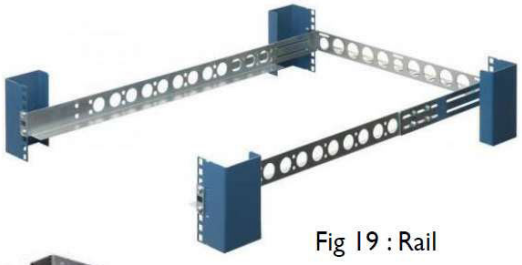


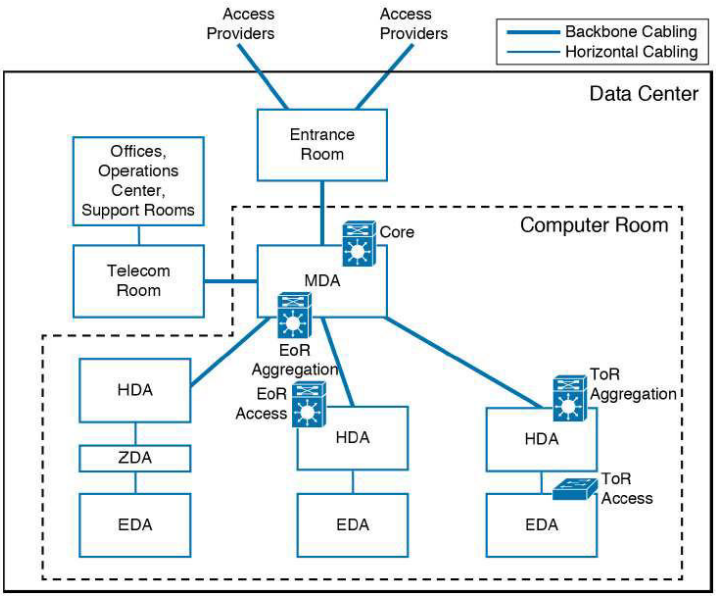
Fig 19 : Rail

Fig 20 : Blanking Panels

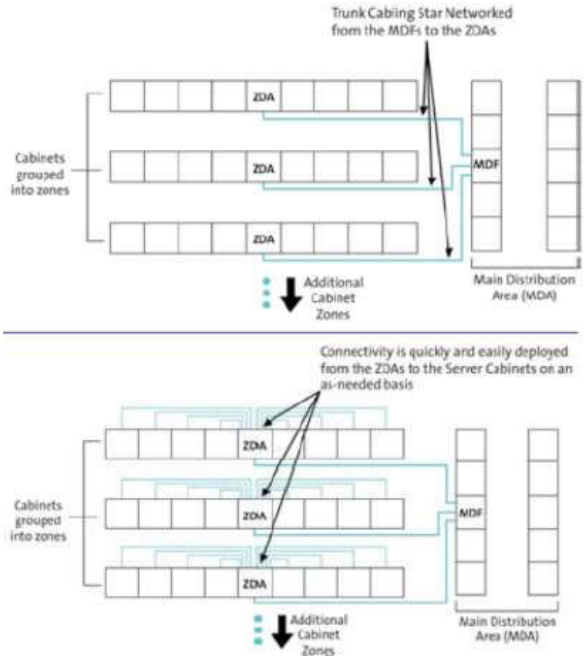
Fig 21 : Cable management bars

ANSI/TIA-942-2005 Standard

- Main Distribution Area (MDA)
- Horizontal Distribution Area (HDA)
- Equipment Distribution Area (EDA)
- Zone Distribution Area (ZDA)



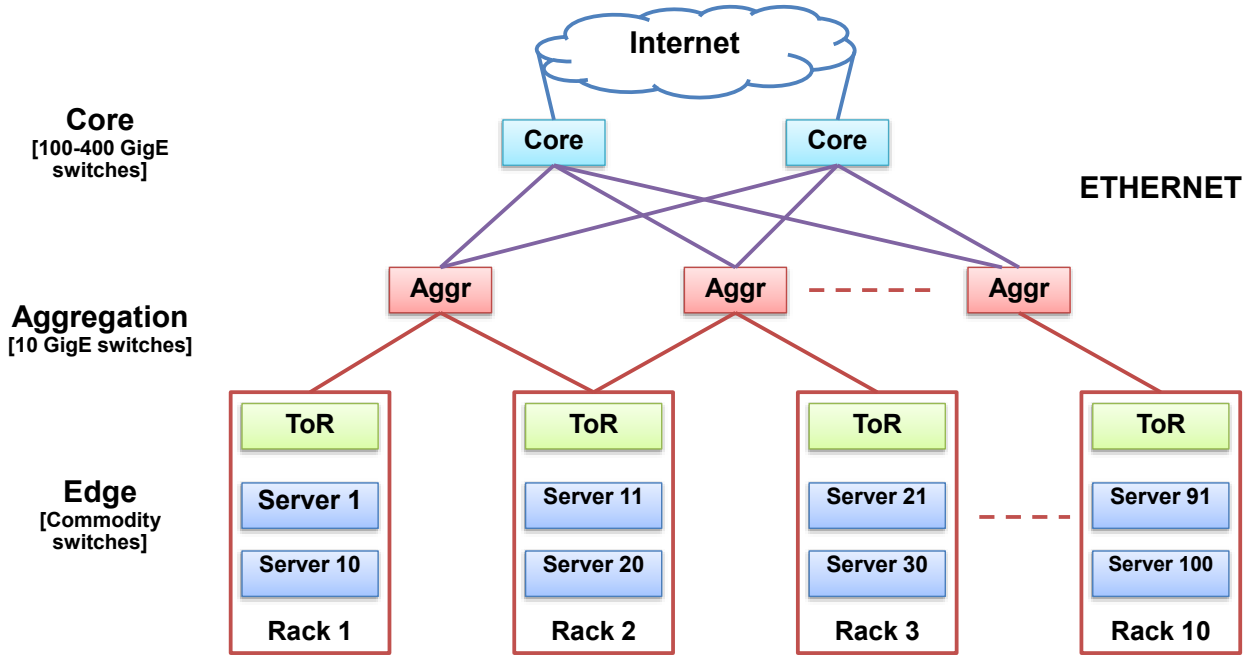
ANSI/TIA-942-2005 Standard



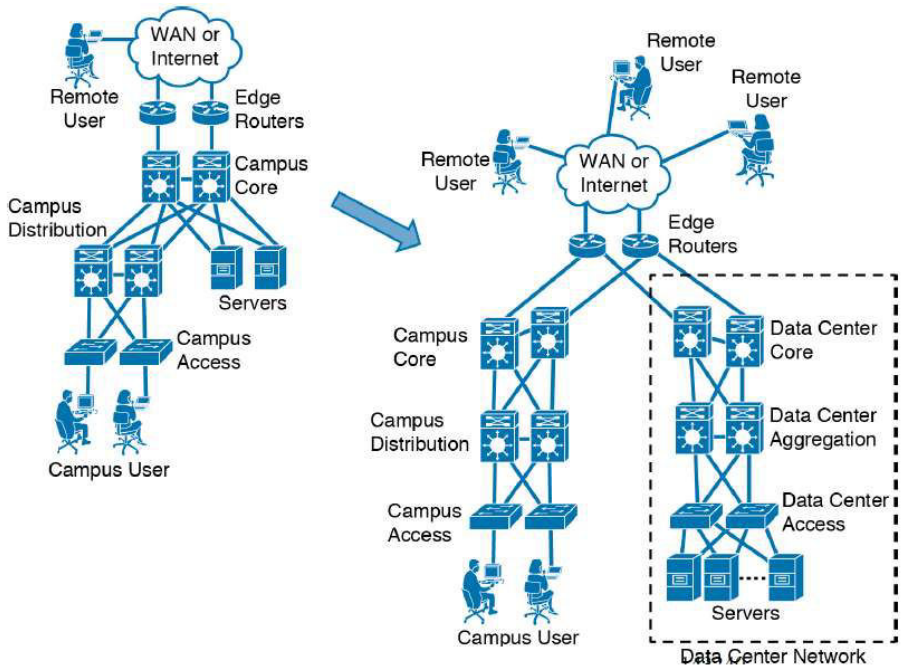
ANSI/TIA-942-2005 Standard

- **Computer Room:** Main servers
- **Entrance Room:** Data center to external cabling
- **Cross-Connect:** Enables termination of cables
- **Main Distribution Area (MDA):** Main across connect. Central point of structured cabling. Core network devices
- **Horizontal Distribution Area (HDA):** Connections to active equipment
- **Equipment Distribution Area (EDA) :** Activer servers+switches. Alternate hot and cold aisle.
- **Zone Distribution Area (ZDA) :** Optionally between HDA and EDA
- **Backbone Cabling:** Connections between MDA, HDA, and Entrance room

Conventional Data Center Architecture

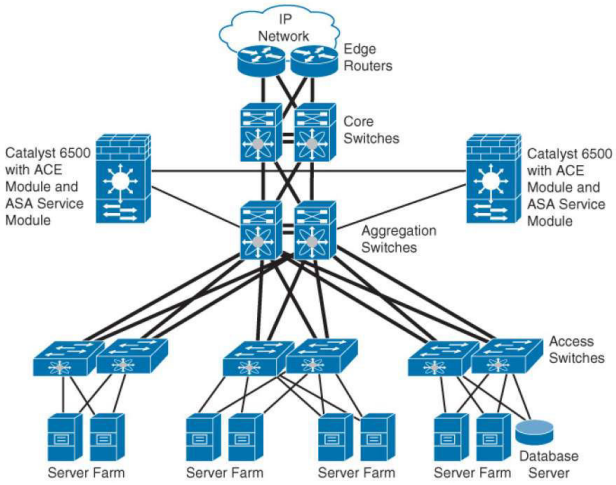


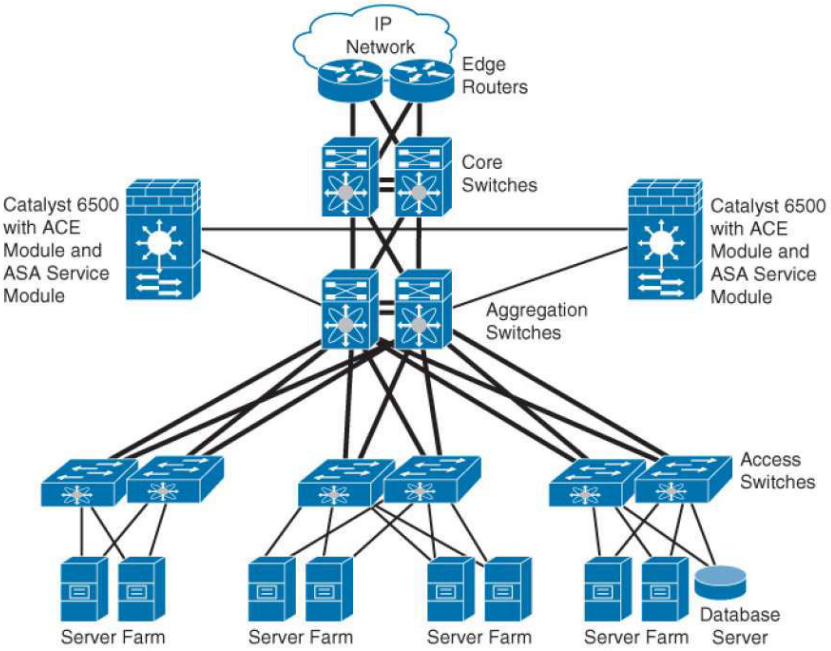
Conventional DC Architecture (contd.)



Hierarchical Network Design

- All servers require application delivery services for
  - Security (VPN, Intrusion detection, firewall)
  - Performance (load balancer)
  - Networking (DNS, DHCP, NTP, FTP etc.)
  - Database services (SQL)
- ADCs are located between the aggregation and core routers and are shared by all servers
- Stateful devices (firewalls) on aggregation layer





Core Layer

- Provides the high-speed packet switching backplane for all flows going in and out of the data center.
- Provides connectivity to multiple aggregation modules and provides a resilient layer 3 routed fabric with no single point of failure.
- Runs an interior routing protocol, such as OSPF or EIGRP, and load balances traffic between the campus core and aggregation layers using Cisco Express Forwarding-based hashing algorithms.

Aggregation Layer

- Provide functions, such as service module integration, Layer 2 domain definitions, spanning tree processing, and default gateway redundancy.
- Server-to-server multi-tier traffic flow occurs through the aggregation layer and can use services, such as firewall and server load balancing, to optimize and secure applications.

Access Layer

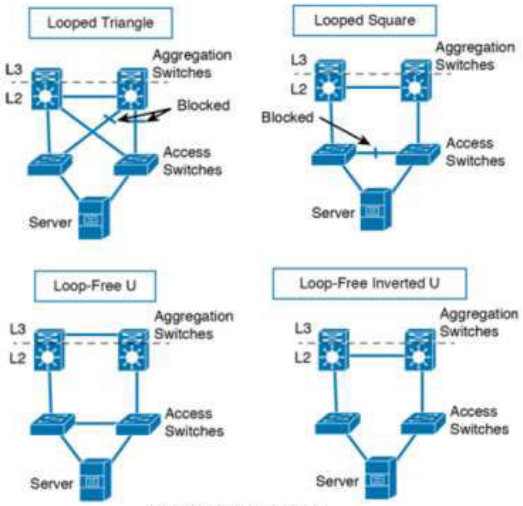
- Servers physically attach to the network within this layer
- The server components consist of RU servers, blade servers with integral switches, blade servers with pass-through cabling, clustered servers, and mainframes with OSA adapters.
- The access layer network infrastructure consists of modular switches, fixed configuration | or 2RU switches, and integral blade server switches.
- Switches provide both Layer 2 and Layer 3 topologies, fulfilling the various server broadcast domain or administrative requirements.

Rack vs Blade Server

Feature	Rack Server	Blade Server
Form Factor	Self-contained server	Thin, modular server
Design	Individual case, power supply, cooling	Slides into chassis, shared resources
Scalability	Linear	High density, hot-swappable
Management	Individual configuration	Centralized management
Cost (Upfront)	Lower	Higher (including chassis)
Cost (Management)	Higher (multiple servers)	Lower (centralized)
Use Cases	General purpose computing, remote locations	Data centers, virtualization, web hosting

Access Aggregation Connection

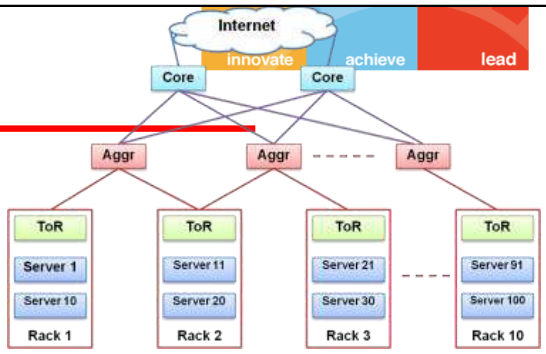
1. **Looped Triangle**:
   Most common. Spanning Tree Protocol (STP) blocks links. Paid but not used.
2. **Looped Square**:
   Oversubscription doubles if failure.
3. **Loop-Free U**:
   No L2 communication between aggregation switches if any switch links fail
4. **Loop-Free Inverted U**:
   Black-holes on some failures



Source: Santana 2014

DCN Design Goals

- Availability and Fault tolerance: Multiple paths and replicated servers.



Challenges:

- **Reduced Utilization**

DCN Design Goals

- Availability and Fault tolerance: Multiple paths and replicated servers. Graceful Degradation.
- Scalability: Incrementally increase DCN size as and when needed.
- Low Cost: Lower power and cooling costs.

Amortized Cost	Component	Sub-Components
45%	Servers	CPU, memory, storage
25%	Infrastructure	Power distribution, cooling
15%	Power draw	Electrical utility costs
15%	Network	Links, transit, equipment

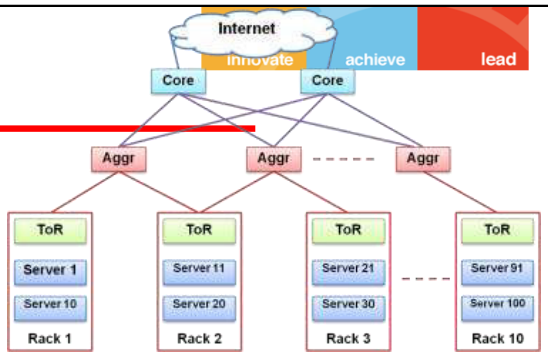
	Hierarchical design			Fat-tree		
Year	10 GigE	Hosts	Cost/ GigE	GigE	Hosts	Cost/ GigE
2002	28-port	4,480	$25.3K	28-port	5,488	$4.5K
2004	32-port	7,680	$4.4K	48-port	27,648	$1.6K
2006	64-port	10,240	$2.1K	48-port	27,648	$1.2K
2008	128-port	20,480	$1.8K	48-port	27,648	$0.3K

Challenges:

- **Reduced Utilization**
- **Scale-out vs Scale-up:** per-port cost, cabling and packaging complexity, scalable cooling.
- **Placement, Air-Flow and rack-density**

DCN Design Goals

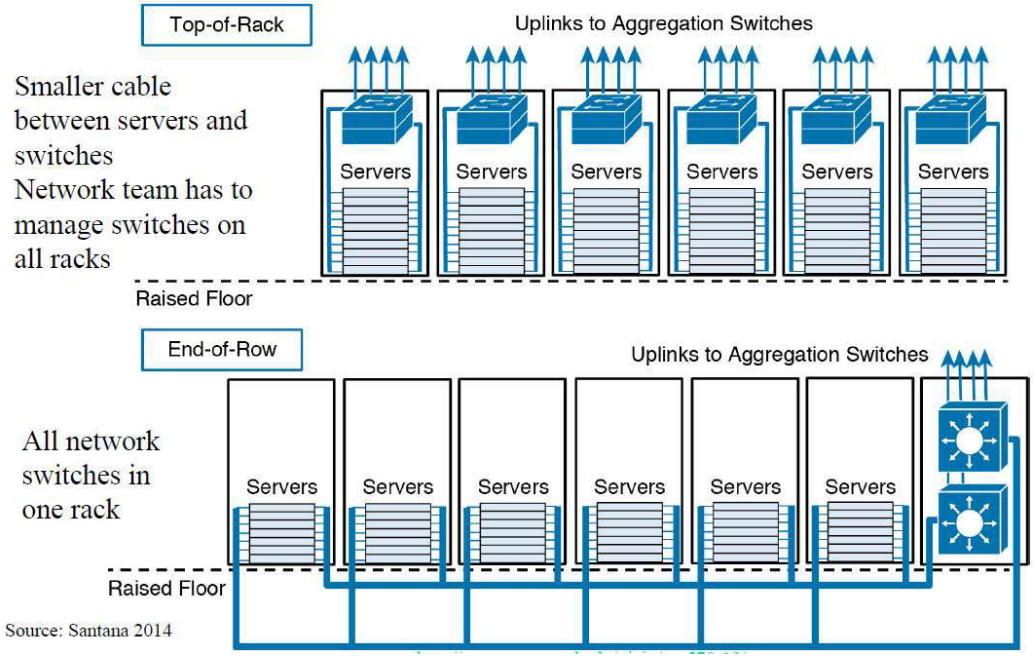
- Availability and Fault tolerance: Multiple paths and replicated servers. Graceful Degradation.
- Scalability: Incrementally increase DCN size as and when needed.
- Low Cost: Lower power and cooling costs.
- Throughput: The number of requests completed by the data center per unit of time. (Compute + Transmission+ Aggregation Time)
- Economies of scale: Utilize the benefits of its huge size.
- Scalable interconnect bandwidth: Host to host communication at full bandwidth.
- Load balancing: Avoid hot-spots, to fully utilize the multiple paths.



Challenges:

- **Reduced Utilization**
- **Scale-out vs Scale-up:** per-port cost, cabling and packaging complexity, scalable cooling.
- **Placement, Air-Flow and rack-density**
- **TCP Incast, Large Buffer switches**
- **Resource fragmentation:** VLANs
- **Manual Configuration**
- **Oversubscription:** 1:1 vs 1:240
- **Flooding and Routing n/w overhead**

Switch Locations



Source: Santana 2014

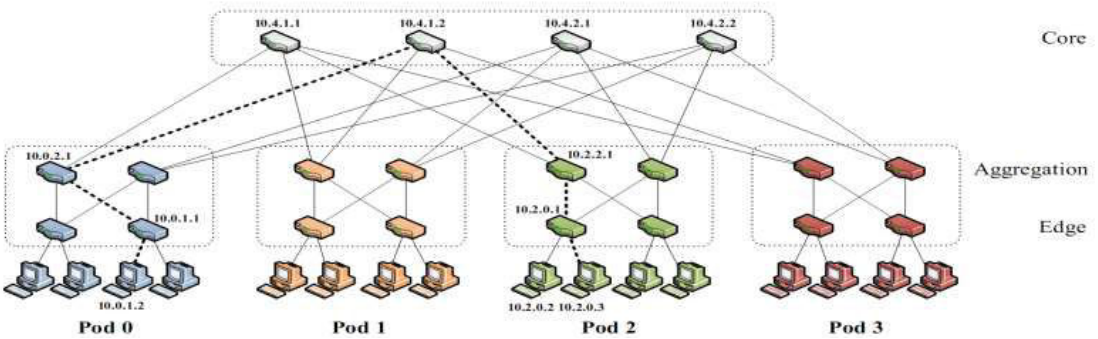
ToR vs EoR

- ToR:
  - Easier cabling
  - If rack is not fully populated □ unused ToR ports
  - If rack traffic demand is high, difficult to add more ports
  - Upgrading (1G to 10G) requires complete Rack upgrade
- EoR:
  - Longer cables
  - Severs can be place in any rack
  - Ports can easily added, upgraded

Fat-Tree Based DC Architecture

K-ary fat tree: three-layer topology (edge, aggregation and core)

- each pod consists of $(k/2)^2$ servers & 2 layers of k/2 k-port switches
- each edge switch connects to k/2 servers & k/2 aggr. switches
- each aggr. switch connects to k/2 edge & k/2 core switches
- $(k/2)^2$ core switches: each connects to k pods
- i,e, **$(k/2)^2$ core switches for k^2 pod switches and $(k/2)^2$ servers.**



1:1 Oversubscription ratio. Commodity Fat-tree with K=4

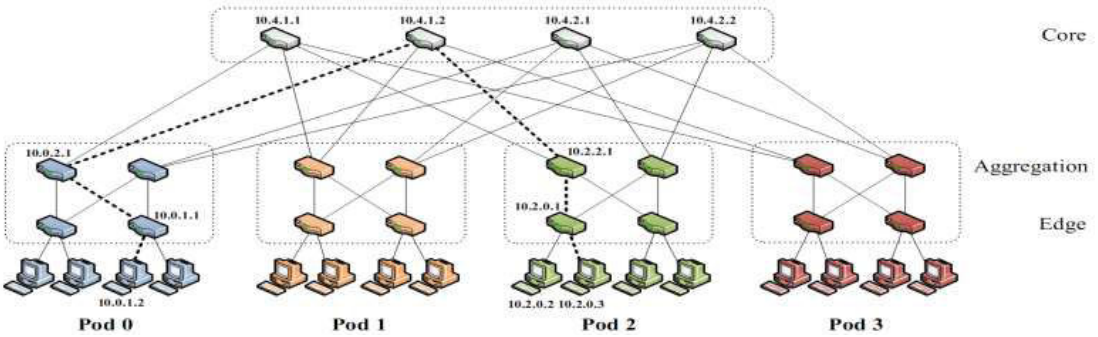
Fat-Tree Based DC Architecture

Advantages:

- **Full Bisection BW:** 1:1 Oversubscription ratio
- **Low Cost:** Commodity switches

Disadvantage:

- **Scalability:** Size of n/w dependent upon ports per switch.48 ports => maximum 27,648 hosts.
- **Agility and Performance Isolation**: Not supported

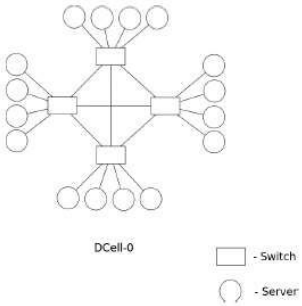


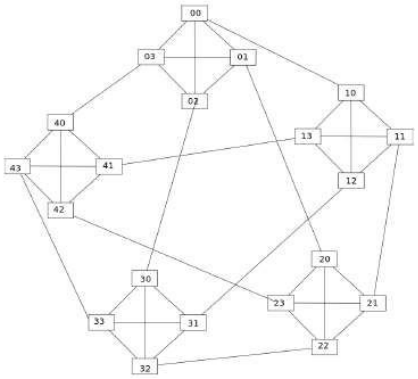
1:1 Oversubscription ratio. Commodity Fat-tree with K=4

Recursive DCN Architecture

- A Level-0 subnet is the basic building block. It contains inter-connected servers.
- Each level-k subnet has multiple level-(k-1) subnets.
- Ex) DCell, BCube, 4-4 1-4, etc

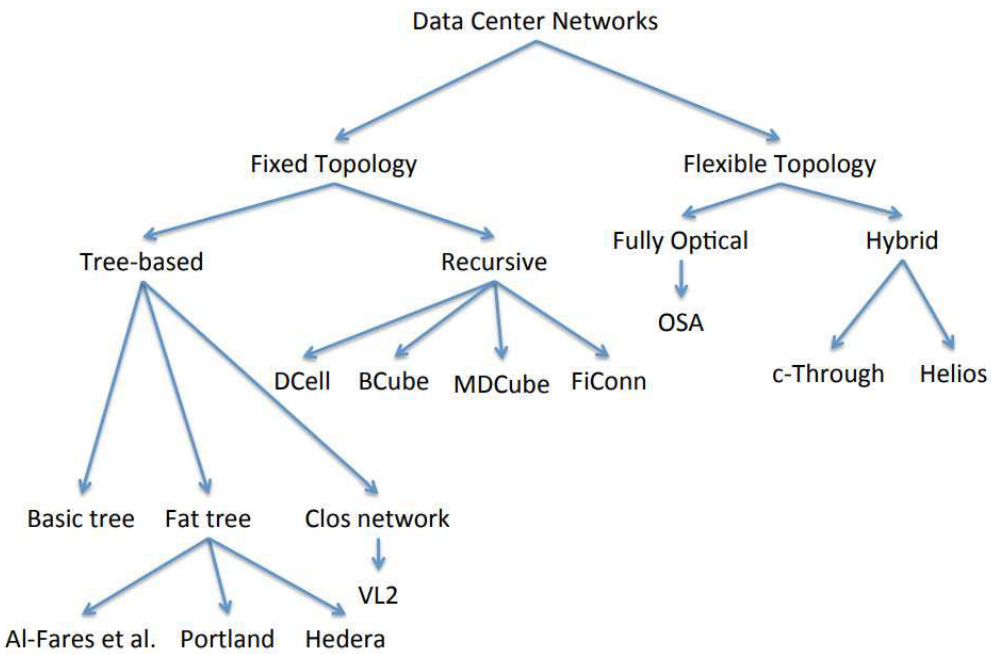
- **Advantages**:
  - Highly Scalable commodity n/w
  - Low CapEx and OpEx.
- **Disadvantage**:
  - Cabling and packaging





DCell-1 formed by 5 DCell-0

Data Center Networks



East-West, North-South traffic

Network Traffic

- Network traffic is the amount of data that moves across a network during any given time. Network traffic may also be referred to as data traffic.
- In search engine optimization, traffic to a network can be characterized as being direct, organic or paid.
  - Direct traffic occurs when someone enters a website's URL in a browser.
  - Organic traffic is the direct result of someone using a search engine to locate content.
  - Paid traffic means someone has clicked on a sponsored advertisement.
- In data center administration, network traffic can be characterized as being either north-south or east-west.

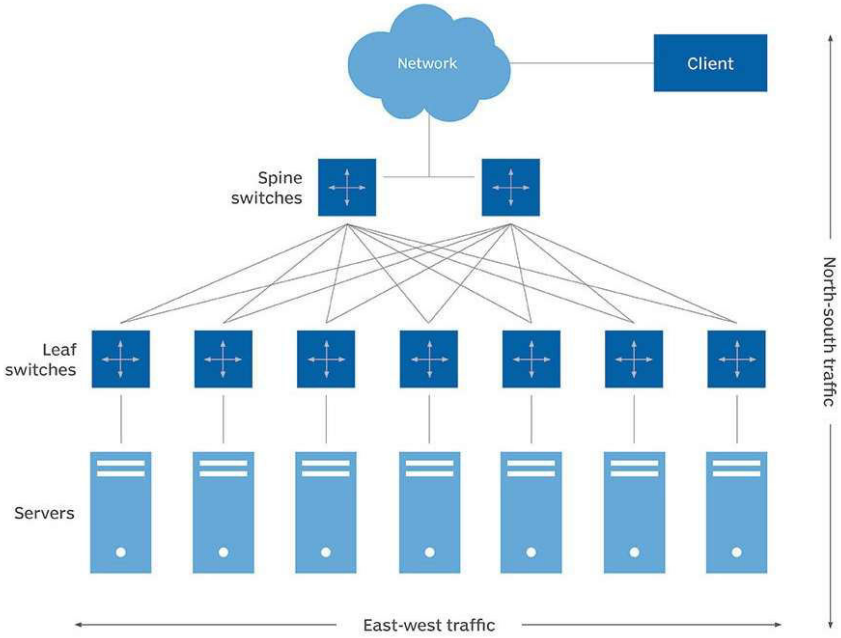
East-West Traffic

- East-west network traffic characterizes data packets that move from server to server within a data center.
- The term east-west was inspired by network diagrams that depict local area network traffic horizontally.
- Hardware virtualization and the internet of things are two concepts that have driven the growth of east-west network traffic.

East-West Traffic

- The volume of east-west traffic has grown as a result of virtualization and data center trends, such as converged infrastructure.
- Network controllers, virtual machines (VMs) and other devices perform various functions and services that previously ran on physical hardware.
- As these components relay data to each other, they increase traffic on the network, which can cause latency issues that negatively affect network performance. For example, if hosts on one access switch need to quickly communicate with systems on another access switch, uplinks among the access layer and aggregation layer become congested.

East-West Traffic



North South Traffic

- North-south traffic describes client-to-server traffic that moves between the data center and a location outside the network.
- North-south traffic is typically depicted vertically to illustrate traffic that flows in and out of the data center.
- In the early days of the internet, most network traffic was north-south.
- This type of traffic often involves the exchange of sensitive information, such as user credentials, and is critical to the functioning of an organization's IT systems.
- Examples of north-south traffic include web browsing, email, and cloud-based applications.

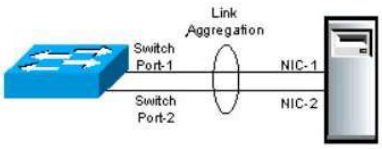
North South Traffic

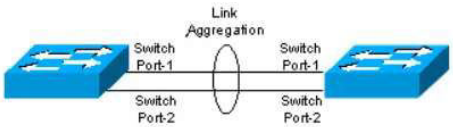
- In a modern, cloud-based infrastructure, north-south traffic is becoming increasingly important as more and more applications are deployed across multiple data centers and cloud platforms such as Amazon Web Services and Google Cloud.
- Proper management of north-south traffic is crucial for ensuring the performance and security of these applications.
- To achieve this, organizations may use firewalls, intrusion detection systems, and VPNs to secure and control the flow of north-south traffic.

Link Aggregation

Link Aggregation

- Link aggregation, also known as port trunking, channel bonding, or NIC teaming, is an Ethernet bundling technology to combine multiple physical links between two networking devices into a single logically virtual link.
- The connection of two devices is generally found between switches, switch-to-server, switch-to-NAS, switch-to-AP, etc.
- The goal of Ethernet port aggregation is to increase bandwidth, improve fault tolerance and provide link redundancy.





Link Aggregation

- Link aggregation is commonly used in data centers, where high availability and performance are critical, and it provides a scalable and cost-effective solution for managing network traffic.

Link Aggregation - Types

- **Static Link Aggregation:** This is a manual configuration process where the network administrator manually configures the switch ports to form a Link Aggregation Group (LAG). In this case, the switch doesn't communicate with other switches to form a LAG.

- **Dynamic Link Aggregation:** This is an automated process where the switch ports automatically form a LAG by communicating with other switches. Dynamic Link Aggregation requires the use of a protocol, such as Link Aggregation Control Protocol (LACP).

Link Aggregation Configuration Protocol

- The Link Aggregation Control Protocol (LACP) is an IEEE Ethernet standard defined in the IEEE 802.3 ad specification.
- Switches supporting LACP are also called LACP switches or 802.3 ad switches.
- LACP allows for the bundling of multiple physical ports into a single logical channel known as a Link Aggregation Group (LAG).
- By creating LAGs
  - the available bandwidth is increased
  - port flexibility is improved
  - link redundancy is established between two devices.

LACP (contd.)

- LACP is a standardized protocol that ensures interoperability between different vendors' network equipment and provides a way for devices to negotiate the parameters of the link aggregation group, such as the number of links, the algorithm used for load balancing, and the failure detection mechanism.

Link Aggregation vs LACP

- Static Link Aggregation is a manual process that requires the network administrator to configure the switch ports to form a LAG, whereas LACP is an automated process that allows switches to detect and configure LAGs automatically.
- LACP is a protocol used for the automatic setup and configuration of link aggregation groups (LAGs), while link aggregation refers to the process of combining multiple physical network links into a single logical link.

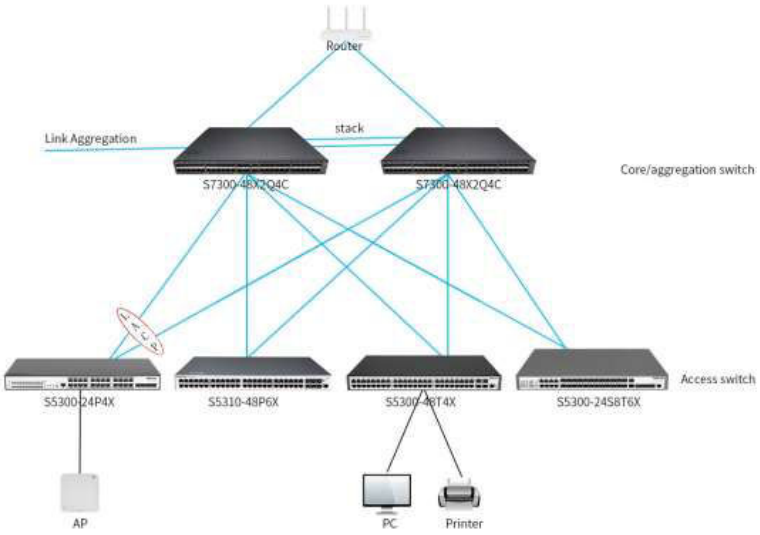
Link Aggregation vs LACP

Feature	LAG	LACP
Definition	Link Aggregation Group (LAG)	Link Aggregation Control Protocol (LACP)
Nature	Logical link	Protocol
Function	Bundling multiple links	Dynamic link aggregation and de-aggregation
Configuration	Can be manually or automatically configured	Automatically negotiated
Dependency	Can be used with or without LACP	LACP is required for dynamic link aggregation
Link failure	Basic link failure detection	Automatic link failure detection
Load balancing	Basic load balancing	Advanced load balancing based on traffic demand
Network management	Can be more complex	Can be more simplified
Cost-effectiveness	Less cost-effective	More cost-effective

How to Setup LACP

- Verify that the switch and the network devices support LACP.
- Configure the physical interfaces you want to aggregate into a LAG on the switch. Assign each interface to a specific VLAN, if needed.
- Enable LACP bonding on the switch. (By issuing the command "LACP enable" on the switch's command line interface (CLI)).
- Configure the LAG on the switch by assigning it an IP address, subnet mask, and any other necessary settings.
- Configure the network devices that will connect to the LAG to use LACP. This typically involves setting the network device to use LACP and configuring the device with the same LAG settings as the switch.
- Connect the network devices to the LAG ports on the switch.

Link Aggregation vs LACP



Example - Link Aggregation Switches

Model	A	B	C
Port	24x 1G RJ45, 4x 10G SFP+	48x 10G SFP+, 2x 40G QSFP+, 4x 100G QSFP28	24x 25G SFP28, 4x 100G QSFP28
Switching Capacity	128 Gbps	1920 Gbps	2 Tbps
Forwarding Rate	96 Mpps	1440 Mpps	1488 Mpps
Feature	stackable	stackable	MLAG, VXLAN, RDMA
Layer	Layer 2 access/aggregation	Layer 3 aggregation/core layers	Layer 3+ aggregation/core layers

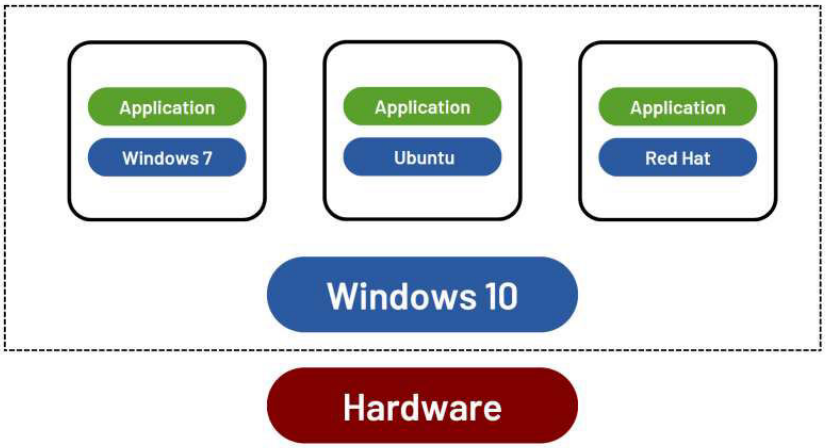
Link Aggregation - Benefits

- **Increased bandwidth**: By combining multiple links into a single logical link, link aggregation bonding can increase the available bandwidth and throughput for network traffic.
- **Improved fault tolerance**: If one physical link fails, traffic can be automatically rerouted to other links in the LAG, improving network availability and reducing downtime.
- **Load balancing**: Network traffic can be distributed across multiple links in the LAG, helping to balance the load and avoid congestion on any single link.
- **Simplified network management**: Link aggregation can simplify network management by reducing the number of physical links that need to be managed and configured, which can save time and reduce errors.
- **Cost-effective**: Link aggregation can be a cost-effective way to increase network performance and availability, as it can make use of existing network infrastructure and equipment.

link aggregation can provide a number of benefits for network performance, availability, and management, making it a useful tool for many different types of networks and applications.

VIRTUALIZATION

Operating System



ABSTRACTION

Keywords

Level of Abstraction

Degree of Freedom

Virtualization

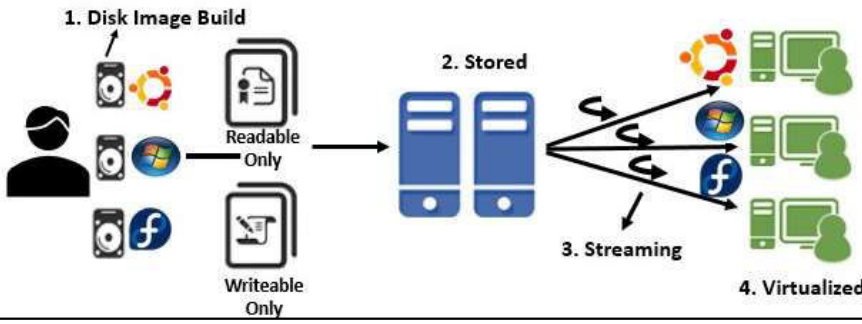
- It is technique which allows to share a single physical instance of an application or resource among multiple organizations or tenants.

Virtualization - Definition

- Virtualization is the **creation of a virtual** (rather than actual) **version** of computing resource, such as **computer hardware, an operating system, a server, a storage device or network resources**.
- This is usually done by introducing a layer of abstraction at appropriate place in computing stack.

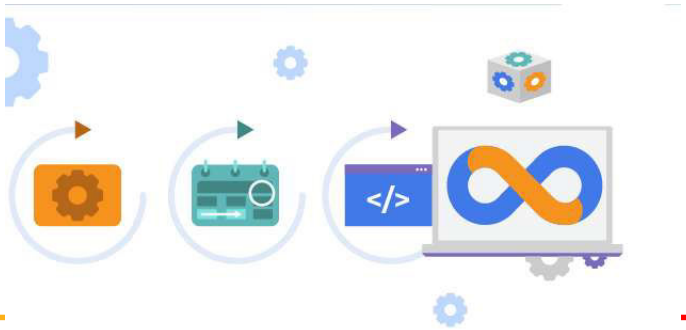
Virtualization

- Virtualization is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources*
- It is the process by which one computer hosts the appearance of many computers.



Virtualization

- Virtualization is used to improve IT throughput and costs by using physical resources as a pool from which virtual resources can be allocated.



Why called as Virtualization?

- Because physically (or actually) the device or software is not what it is pretending to be. Actual configuration is different and hence called virtual.
- However, there is always some physical entity down the line.

Driving Forces for Virtualization

- Increased performance and computing capacity
- Underutilized hardware and software resources
- Lack of space
- Green initiative
- Rise of administrative cost

Virtualization in General

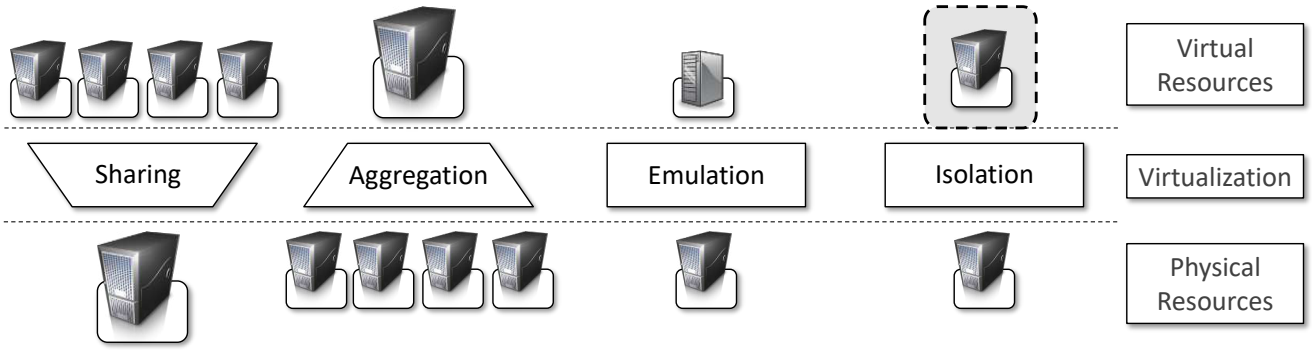
Advantages of virtual machines:

- Run operating systems where the relevant physical hardware is unavailable,
- Easier to create new machines, backup machines, etc.,
- Software testing using "clean" installs of operating systems and software,
- Emulate more machines than are physically available,
- Debug problems (suspend and resume the problem machine),
- Easy migration of virtual machines (shutdown needed or not).
- Run legacy systems!

Characteristics of Virtualization

- Increased security
- Managed execution
  - Sharing
  - Aggregation
  - Emulation
  - Isolation
- Portability

Functions Enabled by Managed Execution



Virtualization Levels

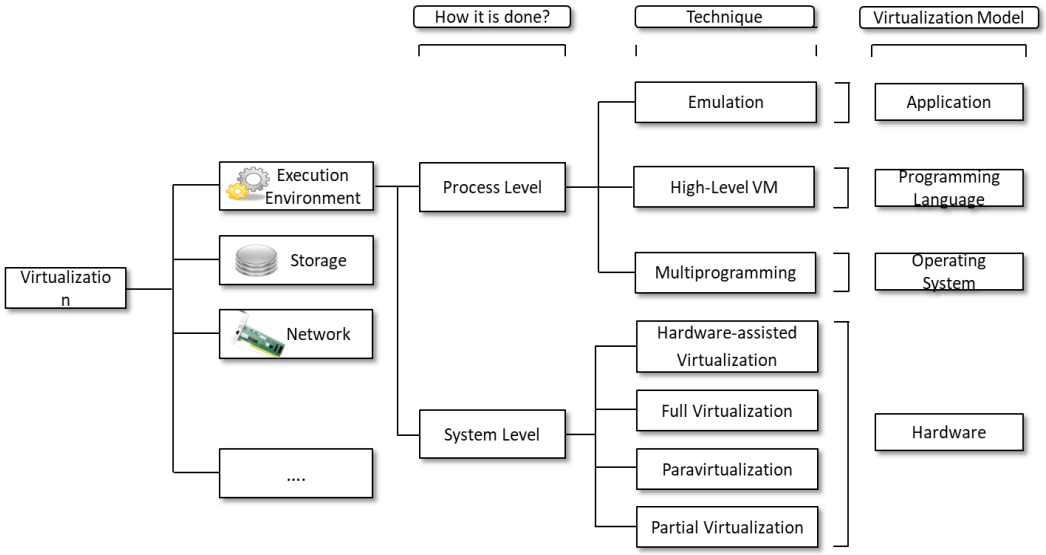
Application level
- JVM / .NET CLR / Panot

Library (user-level API) level
- WINE/ WABI/ LxRun / Visual MainWin / vCUDA

Operating system level
- Jail / Virtual Environment / Ensim's VPS / FVM

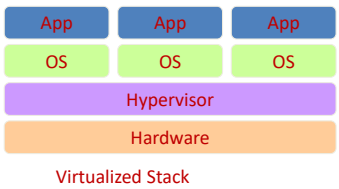
Hardware abstraction layer (HAL) level
- VMware / Virtual PC / Denali / Xen / L4 / Plex 86 / User mode Linux / Cooperative Linux

Instruction set architecture (ISA) level
- Bochs / Crusoe / QEMU / BIRD / Dynamo



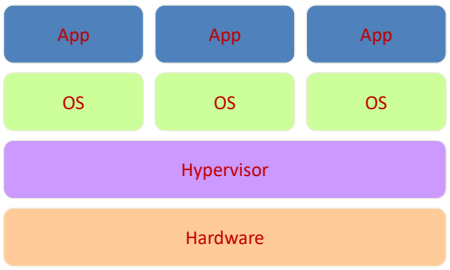
Implementation: Levels of Virtualization

- Virtualization technology benefits the computer and IT industries by enabling users to share expensive hardware resources by multiplexing VMs on the same set of hardware hosts. Virtual workspaces:
  - *An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols, resource quota (e.g. CPU, memory share), software configuration (e.g. O/S, provided services).*



Implementation: Levels of Virtualization

- Implement on Virtual Machines (VMs):
  - Abstraction of a physical host machine,
  - Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs, VMWare, Xen, etc.
- Provide infrastructure API:
  - Plug-ins to hardware/support structures



Virtualized Stack

Taxonomy of Virtualization Techniques

- Execution Virtualization
  - Hardware level virtualization
    - Hardware assisted virtualization
    - Full virtualization
    - Para-virtualization
    - Partial virtualization
  - Operating system level virtualization
  - Programming language level virtualization
  - Application-level virtualization
    - Interpretation
    - Binary translation

Other Type of Virtualization

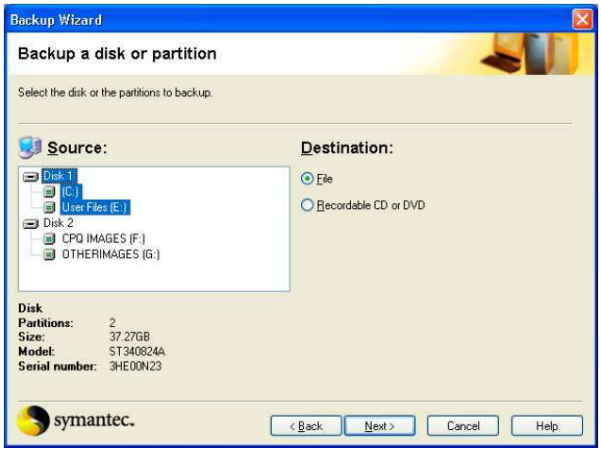
- Storage virtualization
- Network virtualization
- Desktop virtualization
- Application server virtualization

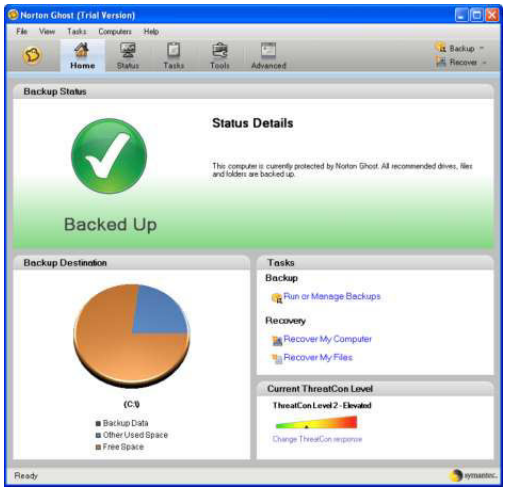
Need for Virtual Machine

How easy or hard?

Clone complete installation to new computer

Virtual Machines - Analogy





Norton Ghost

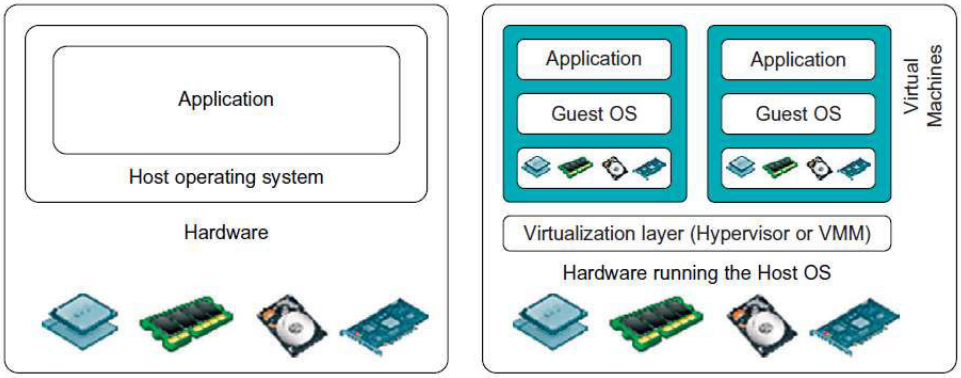
Virtual Machines - Analogy

Image

Virtual Machine

Virtualization Levels

The virtualization software creates the abstraction of VMs by interposing a **virtualization layer (Hypervisor or VMM)** at various levels of a computer system.



(a) Traditional computer (b) After virtualization

Virtualization Layer

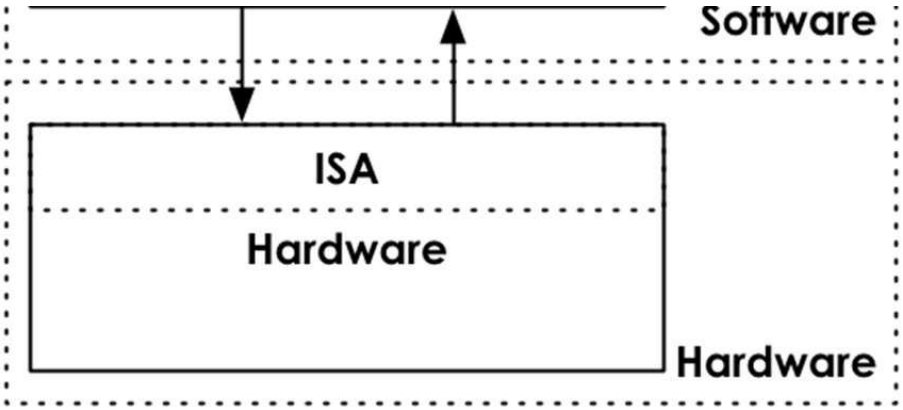
- The main function of the software layer for virtualization is to virtualize the physical hardware of a host machine into virtual resources to be used by the VMs, exclusively.

Virtualization Levels

Common virtualization layers include

1. Instruction Set Architecture (ISA) level
2. Hardware level
3. Operating system level
4. Library support level
5. Application level

Instruction Set Architecture Levels



Instruction Set Architecture (ISA) level

- At the ISA level, virtualization is performed by emulating a given ISA by the ISA of the host machine. Instruction set emulation leads to virtual ISAs created on any hardware machine. e.g, MIPS binary code can run on an x-86-based host machine with the help of ISA emulation.
- With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.

Instruction Set Architecture (ISA) level

- Code interpretation – Dynamic Binary Translation - **Virtual Instruction Set Architecture (V-ISA)**
- The basic emulation method is through code interpretation.
  - An interpreter program interprets the source instructions to target instructions one by one.
  - One source instruction may require tens or hundreds of native target instructions to perform its function.
  - Obviously, this process is relatively slow.

Instruction Set Architecture (ISA) level

- **Advantage**:
  - It can run a large amount of legacy binary codes written for various processors on any given new hardware host machines
  - Best application flexibility
- **Shortcoming & limitation**:
  - One source instruction may require tens or hundreds of native target instructions to perform its function, which is relatively slow.
  - V-ISA requires adding a processor-specific software translation layer in the complier.

Hardware Abstraction level

- Hardware-level virtualization is performed right on top of the bare hardware.
- On the one hand, this approach generates a virtual hardware environment for a VM.
- On the other hand, the process manages the underlying hardware through virtualization.

Hardware Abstraction level

- The idea is to virtualize a computer's resources, such as its processors, memory, and I/O devices. The intention is to upgrade the hardware utilization rate by multiple users concurrently.

Advantage:

- Has higher performance and good application isolation

Operating System (OS) Level

- OS-level virtualization creates isolated containers on a single physical server and the OS instances to utilize the hardware and software in data centers. The containers behave like real servers.
- OS-level virtualization is commonly used in creating virtual hosting environments to **allocate hardware resources among a large number of mutually distrusting users**.

Advantage:

- Has minimal startup/shutdown cost, low resource requirement, and high scalability;

Operating System (OS) Level

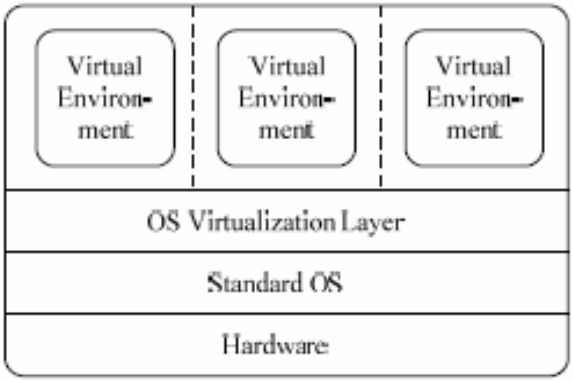


Figure 6.3 The virtualization layer is inserted inside an OS to partition the hardware resources for multiple VMs to run their applications in virtual environments

Library Support level

- Since most systems provide well-documented APIs, such an interface becomes another candidate for virtualization.
- Virtualization with library interfaces is possible by controlling the communication link between applications and the rest of a system through API hooks.
- The software tool WINE has implemented this approach to support Windows applications on top of UNIX hosts.

Advantage:

- It has very low implementation effort

User-Application Level

- Virtualization at the application level virtualizes an application as a VM. On a traditional OS, an application often runs as a process.
- Therefore, application-level virtualization is also known as process-level virtualization.
- The most popular approach is to deploy high level language (HLL) VMs. In this scenario, the virtualization layer sits as an application program on top of the operating system, and the layer exports an abstraction of a VM that can run programs written and compiled to a particular abstract machine definition.

User-Application Level

- Other forms of application-level virtualization are known as
  - application isolation,
  - application sandboxing, or application streaming.

Advantage:

- has the best application isolation

Shortcoming & limitation:

- low performance, low application flexibility and high implementation complexity.

Hypervisor

- A hypervisor is a hardware virtualization technique allowing multiple operating systems, called guests to run on a host machine. This is also called the Virtual Machine Monitor (VMM).

Type 1: Bare metal hypervisor

- sits on the bare metal computer hardware like the CPU, memory, etc.
- All guest operating systems are a layer above the hypervisor.
- The original CP/CMS hypervisor developed by IBM was of this kind.

Hypervisor

Type 2: Hosted hypervisor

- Run over a host operating system.
- Hypervisor is the second layer over the hardware.
- Guest operating systems run a layer over the hypervisor.
- The OS is usually unaware of the virtualization

The XEN Architecture

- Xen is an open source hypervisor program developed by Cambridge University. Xen is a micro-kernel hypervisor, which separates the policy from the mechanism.
- Xen does not include any device drivers natively . I t just provides a mechanism by which a guest OS can have direct access to the physical devices.
- As a result, the size of the Xen hypervisor is kept rather small. Xen provides a virtual environment located between the hardware and the OS.

Binary Translation with Full Virtualization

- Full virtualization does not need to modify the host OS. It relies on binary translation to trap and to virtualizes the execution of certain sensitive, non virtualizable instructions.
- The guest OSes and their applications consist of non-critical and critical instructions.

Guest OS Requests Using a VMM

- This approach was implemented by VMware and many other software companies.
- VMware puts the VMM at Ring 0 and the guest OS at Ring 1. The VMM scans the instruction stream and identified the privileged, control- and behavior sensitive instructions.
- When these instructions are identified, they are trapped into the VMM, which emulates the behavior of these instructions.
- The method used in this emulation is called binary translation. Therefore, full virtualization combines binary translation and direct execution.

Full Virtualization vs. Para-Virtualization

Full virtualization

- Does not need to modify guest OS, and critical instructions are emulated by software through the use of binary translation.
- VMware Workstation applies full virtualization, which uses binary translation to automatically modify x86 software on-the-fly to replace critical instructions.

Advantage: no need to modify OS.

Disadvantage: binary translation slows down the performance.

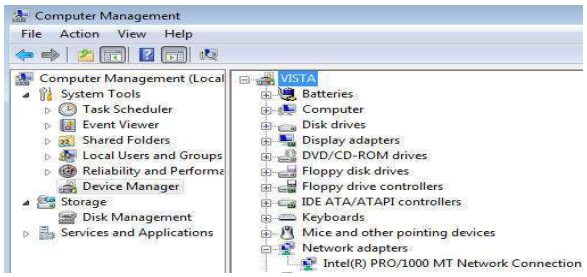
Full Virtualization vs. Para-Virtualization

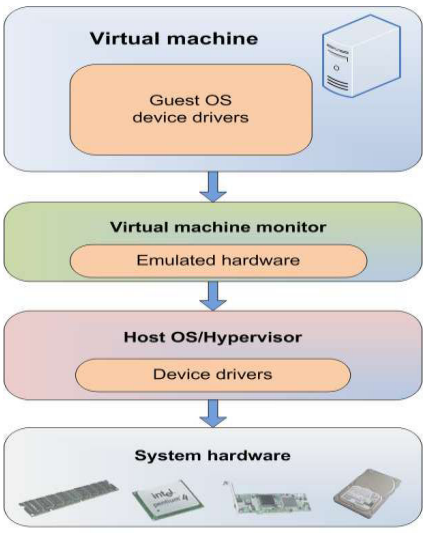
Para virtualization

- Reduces the overhead, but cost of maintaining a para-virtualized OS is high.
- The improvement depends on the workload.
- Para virtualization must modify guest OS, non-virtualizable instructions are replaced by hyper calls that communicate directly with the hypervisor or VMM.
- Para virtualization is supported by Xen, Denali and VMware ESX.

Full Virtualization

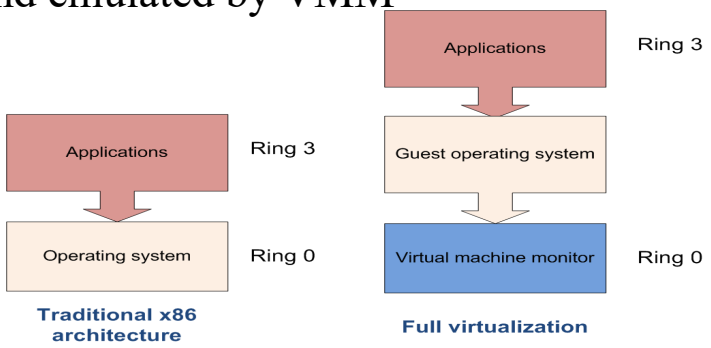
- Everything is virtualized
- Full hardware emulation
- Emulation = latency





Privileged Instructions

- Privileged instructions: OS kernel and device driver access to system hardware
- Trapped and emulated by VMM



Pros and Cons – Full Virtualization

- **Pros**
  - Disaster recovery, failover
  - Virtual appliance deployment
  - Legacy code on non-legacy hardware
- **Cons** – LATENCY of core four resources
  - RAM performance reduced 25% to 75%
  - Disk I/O degraded from 5% to 20%
  - Network performance decreased up to 10%
  - CPU privileged instruction dings nearing 1% to 7%

Para-virtualization

- OS or system devices are virtualization aware

- Requirements:
- OS level – recompiled kernel
- Device level – para-virtualized or "enlightened" device drivers

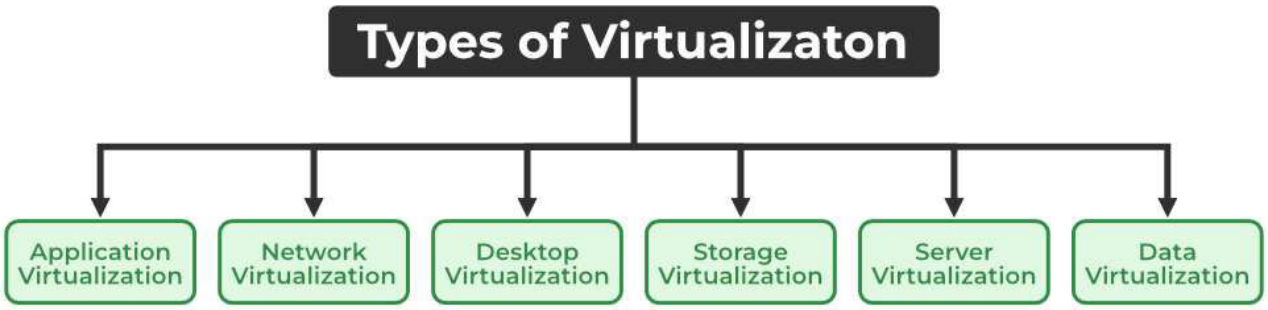
Para-virtualization

- **Pro:** Fast

- **Con:** requires a specially modified guest OS, thus precludes the ability to run off-the-shelf and legacy OS in paravirtual environments

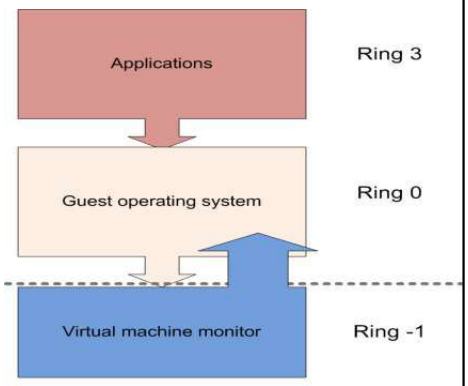
Virtualization - Other Levels

Virtualization Types



Hardware-assisted Virtualization

- Server hardware is virtualization aware
- Hypervisor and VMM load at privilege Ring -1 (firmware)
- Removes CPU emulation bottleneck
- Memory virtualization coming in quad core AMD and Intel CPUs



Hardware-assisted virtualization

CPU Virtualization

- A VM is a duplicate of an existing computer system in which a majority of the VM instructions are executed on the host processor in native mode. Thus, unprivileged instructions of VMs run directly on the host machine for higher efficiency. Other critical instructions should be handled carefully for correctness and stability.
- The critical instructions are divided into three categories: privileged instructions, control-sensitive instructions, and behavior-sensitive instructions.
- Privileged instructions execute in a privileged mode and will be trapped if executed outside this mode.
- Control-sensitive instructions attempt to change the configuration of resources used. Behavior-sensitive instructions have different behaviors depending on the configuration of resources, including the load and store operations over the virtual memory.

CPU Virtualization

- A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode while the VMM runs in supervisor mode.
- When the privileged instructions including control- and behavior-sensitive instructions of a VM are executed, they are trapped in the VMM. I n this case, the VMM acts as a unified mediator for hardware access from different VMs to guarantee the correctness and stability of the whole system. However, not all CPU architectures are virtualizable.
- RISC CPU architectures can be naturally virtualized because all control and behavior-sensitive instructions are privileged instructions.
- On the contrary, x86 CPU architectures are not primarily designed to support virtualization.

Memory Virtualization

- Virtual memory virtualization is similar to the virtual memory support provided by modern operating systems. I n a traditional execution environment, the operating system maintains mappings of virtual memory to machine memory using page tables, which is a one-stage mapping from virtual memory to machine memory.
- However, in a virtual execution environment, virtual memory virtualization involves sharing the physical system memory in RAM and dynamically allocating it to the physical memory of the VMs.
- That means a two-stage mapping process should be maintained by the guest OS and the VMM, respectively: virtual memory to physical memory and physical memory to machine memory.

I/O Virtualization

- There are three ways to implement I /O virtualization: full device emulation, para-virtualization, and direct I /O.
- I /O virtualization. Generally, this approach emulates well-known, real-world devices. All the functions of a device or bus infrastructure, such as device enumeration, identification, interrupts, and DMA, are replicated in software. This software is located in the VMM and acts as a virtual device.
- The para-virtualization method of I /O virtualization is typically used in Xen. I t is also known as the split driver model consisting of a frontend driver and a backend driver. It achieves beer device performance than full device emulation, it comes with a higher CPU overhead
- Direct I /O virtualization lets the VM access devices directly. It can achieve close-to native performance without high CPU costs.

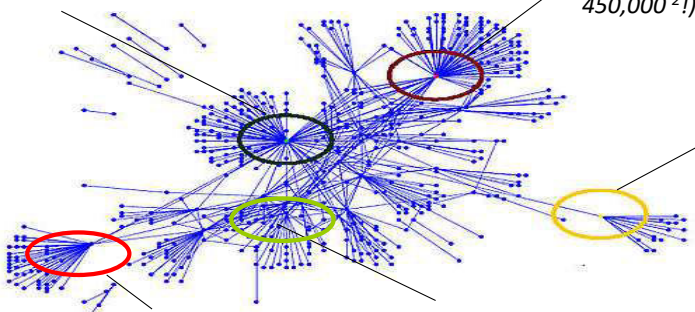
Storage Virtualization

- ***Storage Virtualization*** *is the next frontier in Storage Advances that* ***aims to provide a layer of*** abstraction to reduce complexity.
- Storage Networking Industry Association (SNIA) defines Storage Virtualization as:
  - The act of abstracting, hiding, or isolating the internal functions of a storage (sub) system or service from applications, host computers, or general network resources, for the purpose of enabling application and network-independent management of storage or data.
  - The application of virtualization to storage services or devices for the purpose of aggregating functions or devices, hiding complexity, or adding new capabilities to lower level storage resources.

Why Storage Virtualization?

Provided continuous availability despite exponential growth (e.g. *FaceBook- Over 55 billion page views a month, 41 million active users*[1])

Effectively group and manage heterogeneous storage devices & servers (*e.g. Estimated number of Google Servers 450,000*[2]*!)*



Allocate and manage storage in accordance to the Quality of Service (QoS) associated with the data *(e. g. Gartner estimates average data center doubling its storage every 18 to 24 months)*

Mergers and Acquisitions *(e.g. Microsoft & Yahoo!)*

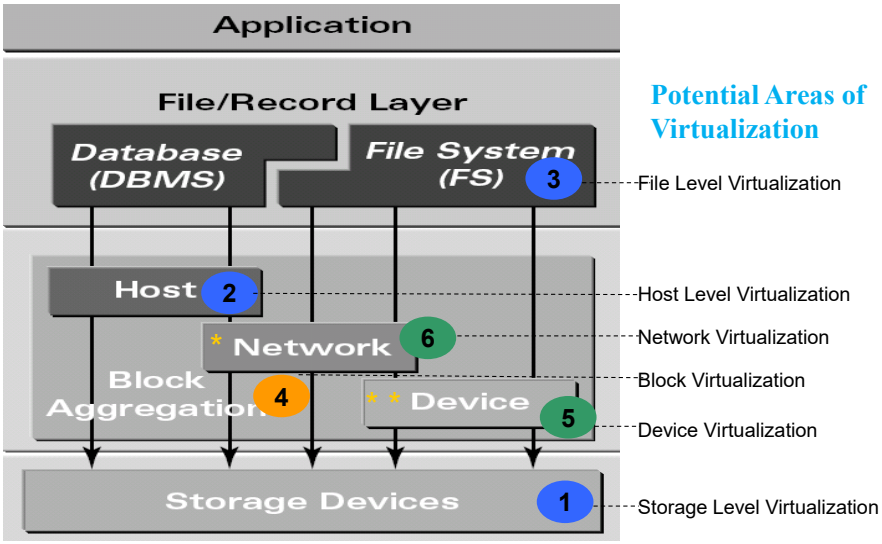
Multiple Storage Software Platforms (*e.g. IBM, EMC, HP,..)*

Storage Virtualization aims to provide a layer of abstraction to manage storage and reduce complexity !!!

NAS vs SAN

	NAS	SAN
Access Methods	**File access**	**Disk block access**
Access Medium	Ethernet	Fiber Channel
Architecture	Decentralized	Centralized
Transport Protocol	Layer over TCP/IP	SCSI/FC and SCSI/IP
Efficiency	Less	More
Sharing and Access Control	Good	Poor
Typical Applications	Web	Database
Typical Clients	Workstations	Database servers

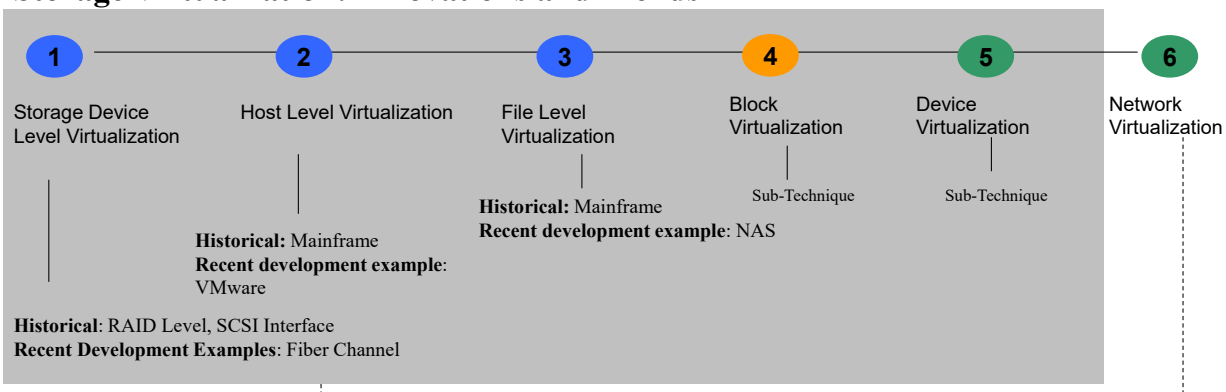
Where can Storage be virtualized?



Source: The Storage Networking Tutorials, SNIAVIRT- Page 20
http://www.snia.org/education/tutorials/

Where can Storage be virtualized?

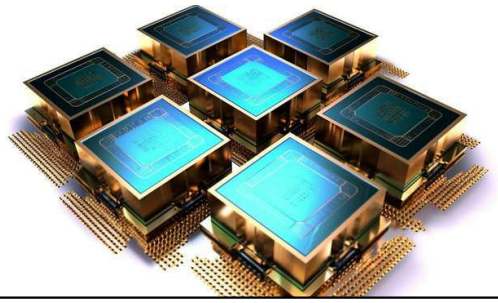
Storage Virtualization: Innovations and Trends



Major innovations continue to emerge even in historical areas of storage virtualization

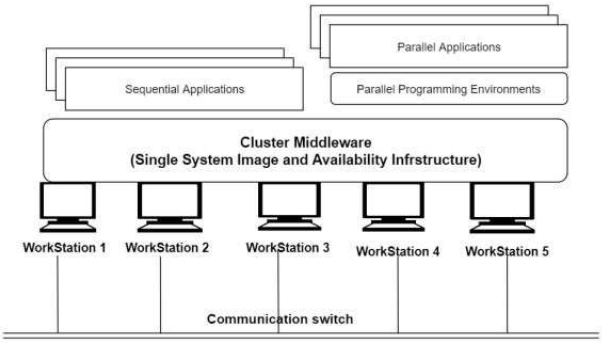
Symmetrical (aka in-band) and Asymmetrical (aka Out-of-Band) are emerging as key areas of abstraction and virtualization.

Cluster Computing | Virtual Cluster | Virtual Data Center | Protection/Isolation



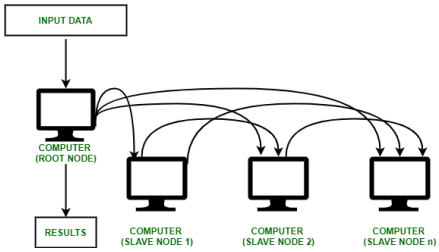
Definition

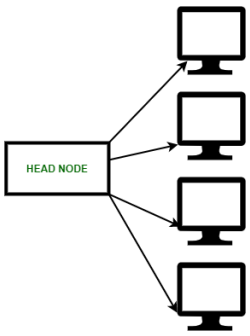
- A computer cluster is a collection of interconnected stand-alone computers which can work together collectively and cooperatively as a single integrated computing resource pool.



Definition

- Cluster computing is a collection of tightly or loosely connected computers that work together so that they act as a single entity. The connected computers execute operations all together thus creating the idea of a single system. The clusters are generally connected through fast local area networks (LANs)





Concepts

- Clustering explores massive parallelism at the job level and achieves high availability
- Benefits with scalable performance, fault tolerance, modular growth, and use of commodity components
- These features can sustain the generation changes in hardware, software, and network components.

Concepts

- Clusters can range in size from just a few computers to thousands of nodes and can be built using off-the-shelf hardware or customized components.
- Cluster Computing requires specialized knowledge and skills to design, deploy, and manage effectively.
- More suitable for use by expert users and organizations with dedicated HPC teams.

Attributes of Cluster Computing

- Scalability
  - Over passes constraints such as the multicore chip technology, cluster topology, packaging method, power consumption, and cooling scheme applied, memory wall, disk I/O bottlenecks, and latency tolerance
- Packing
  - Packaged in a compact or a slack fashion
    - **Compact -** the nodes are closely packaged in one or more racks in a room, not attached to peripherals individually, utilize a high-bandwidth, low-latency communication network
    - **Slack -** are complete SMPs, workstations, and PCs attached with peripherals, located in different rooms, different buildings, or even remote regions.

Attributes of Cluster Computing contd..

- Control
  - Centralized – all the nodes are owned, controlled, managed, and administered by a central operator
  - Decentralized – individual owner can reconfigure, upgrade, or even shut down the workstation at any time
    - lack of a single point of control makes system administration of such a cluster very difficult.
    - It also calls for special techniques for process scheduling, workload migration, checkpointing, accounting, and other similar tasks

Attributes of Cluster Computing contd..

- Homogeneity
  - **Homogeneous cluster** uses nodes from the same platform, that is, the same processor architecture and the same operating system; often, the nodes are from the same vendors
    - Any binary process image can migrate to another node and continue execution
  - **Heterogeneous cluster** uses nodes of different platforms
    - Interoperability is an important issue

Attributes of Cluster Computing contd..

- Security
  - Defined with intra-cluster communication
  - Exposed cluster - **Insecure**
    - The communication paths among the nodes are exposed to the outside world and is possible to access to nodes using standard protocols
  - Enclosed
    - Intra-cluster communication is shielded from the outside world
    - Disadvantage is that there are no standards for efficient communication.

Importance of Cluster Computing

- Cluster computing gives a relatively inexpensive, unconventional to the large server or mainframe computer solutions.
- It resolves the demand for content criticality and process services in a faster way.
- Many organizations and IT companies are implementing cluster computing to augment their scalability, availability, processing speed and resource management at economic prices.
- It ensures that computational power is always available.
- It provides a single general strategy for the implementation and application of parallel high-performance systems independent of certain hardware vendors and their product decisions.

Types of Clusters (Location)

- **Dedicated Clusters**
  - Typically installed in a deskside rack in a room
  - Homogeneously configured with the same type of computer nodes and managed by a single administrator group like a frontend host.
  - Used as substitutes for traditional mainframes or supercomputers
  - Offers much enhanced throughput, as well as reduced response time.
- Enterprise Clusters

Types of Clusters (Location)

- Dedicated Clusters
- **Enterprise Clusters**
  - Heterogenous nodes are usually full-fledged SMPs, workstations, or PCs, with all the necessary peripherals attached
  - Typically, geographically distributed
  - Nodes are individually owned by multiple autonomous owners

Types of Clusters (Functionality)

- High Performance (HP) Clusters
- Load Balancing Clusters
- High Availability (HA) Clusters

High Performance (HP) Clusters

- HP clusters use computer clusters and supercomputers to solve advance computational problems.
- They are designed to take benefit of the parallel processing power of several nodes.
- They are used to performing functions that need nodes to communicate as they perform their jobs.

Load Balancing Clusters

- Incoming requests are distributed for resources among several nodes running similar programs or having similar content.
- This prevents any single node from receiving a disproportionate amount of task.
- This type of distribution is generally used in a web-hosting environment.

High Availability (HA) Clusters

- HA clusters are designed to maintain redundant nodes that can act as backup systems in case any failure occurs.
- Consistent computing services like business activities, complicated databases, customer services like e-websites and network file distribution are provided.
- Designed to give uninterrupted data availability to the customers.

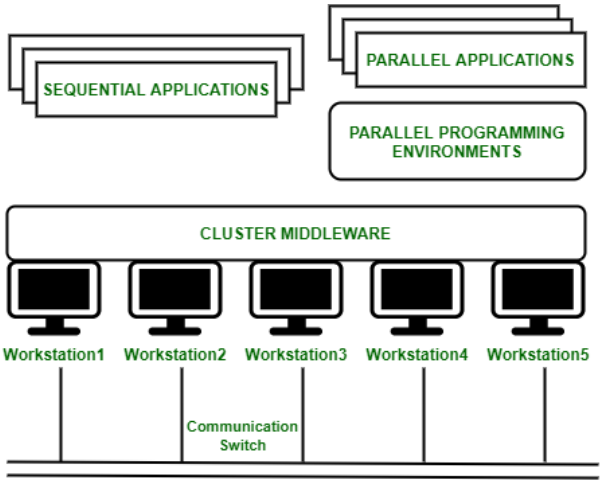
Types of Clusters (Visibility)

- Open Cluster
  - IPs are needed by every node and those are accessed only through the internet or web. This type of cluster causes enhanced security concerns.
- Close Cluster
  - The nodes are hidden behind the gateway node, and they provide increased protection. They need fewer IP addresses and are good for computational tasks.

Summarizing Attributes and Classification

Attributes	Attribute value	
Packaging	Compact	Slack
Control	Centralized	Decentralized
Homogeneity	Homogeneous	Heterogeneous
Security	Enclosed	Exposed
Example	Dedicated cluster	Enterprise cluster

Cluster Architecture



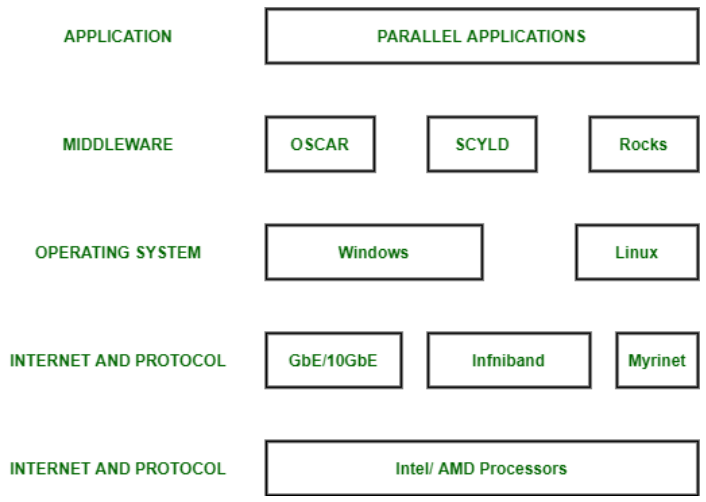
Basic Cluster Architecture

- Array of interconnected individual computers and the computer systems operating collectively as a single standalone system.
- It is a group of workstations or computers working together as a single, integrated computing resource connected via high-speed interconnects.
- A node – Either a single or a multiprocessor network having memory, input and output functions and an operating system.
- Two or more nodes are connected on a single line, or every node might be connected individually through a LAN connection.

Basic Cluster Architecture

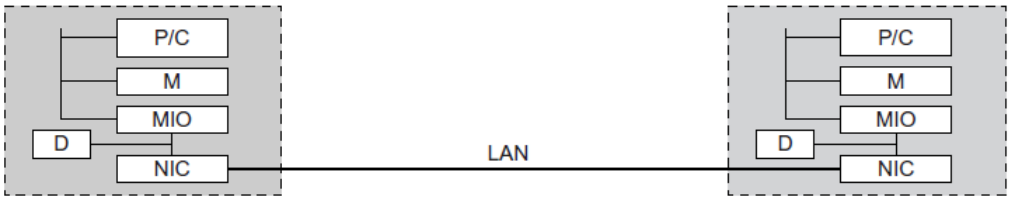
- The processing nodes are commodity workstations, PCs, or servers
  - Easy to replace or upgrade with new generations of hardware
- The node operating systems should be designed for multiuser, multitasking, and multithreaded applications
- Nodes are interconnected by one or more fast commodity networks
- Network interface card is connected to the node's standard I/O bus

Components of Cluster Computer



Resource Sharing in Clusters

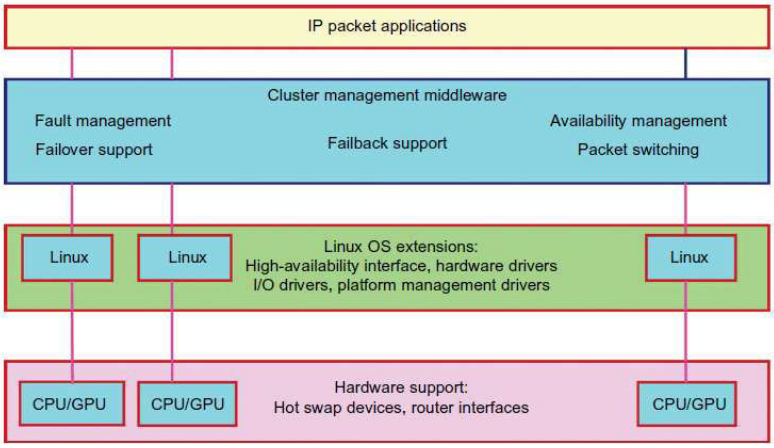
- Shared - Nothing Architecture
  - Mostly used where the nodes are connected through the I/O bus.
  - Connects two or more autonomous computers via a LAN such as Ethernet



(a) Shared nothing

GPU Clusters

- Middleware, Linux extensions, and hardware support for achieving massive parallelism and HA in a Linux cluster system built with CPUs and GPUs



Tools & Frameworks of Cluster

Cluster Management Software

hadoop

Apache MESOS™

kubernetes

slurm workload manager

High Performance Computing (HPC) Tools

Portable Batch System

HTCondor
High Throughput Computing

Parallel Computing Libraries & Frameworks

MPI

APACHE Spark

OpenMP

NVIDIA
CUDA

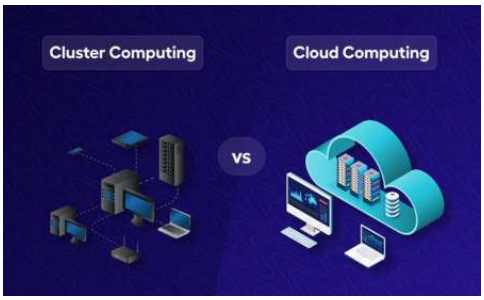
Distributed File System

Applications of Cluster

- Various complex computational problems can be solved
- It can be used in the applications of aerodynamics, astrophysics, and in data mining
- Weather forecasting
- Image Rendering
- Various e-commerce applications
- Earthquake Simulation
- Petroleum reservoir simulation

Cluster Computing
vs
Cloud Computing

Cluster Computing refers to a group of interconnected computers that work together to perform complex tasks in parallel. Cloud Computing refers to a network of remote servers that provide computing resources over the internet.



Cluster vs Cloud

- Differences lies in
  - architecture
  - deployment
  - management
- Cluster Computing is typically used for high-performance computing tasks that require a lot of processing power and memory, such as scientific simulations and data analysis.
- Cloud Computing is more flexible and can be used for various tasks, including web hosting, data storage, and software development.

The Difference

Factors	Cluster Computing	Cloud Computing
Security	Each node in a Cluster Computing system has its security measures.	In Cloud Computing, security is managed by Cloud Service Providers (CSPs).
Cost	Cluster Computing requires dedicated hardware, resulting in high upfront costs.	Cloud Computing uses shared hardware resources, resulting in low, upfront costs with a pay-as-you-go model.
Resource Sharing	Resources in Cluster Computing are shared within the cluster, making it efficient.	Resources in Cloud Computing are shared across multiple users, making it more flexible.
Virtualization	Cluster Computing uses virtualization to partition workloads, which increases efficiency.	Cloud Computing uses virtualization to create multiple virtual machines, which is scalable.
Hardware	Cluster Computing requires dedicated hardware.	Cloud Computing uses shared hardware resources.
Maintenance	Cluster Computing requires skilled IT staff for maintenance.	In Cloud Computing, CSPs manage maintenance and updates, making it hassle-free.
Scalability	Cluster Computing has limited scalability, and adding nodes can be costly.	Cloud Computing is highly scalable and can add resources on demand.

Advantages of Cluster

- High Performance
- Easy to Manage
- Scalable
- Expandability
- Availability
- Flexibility

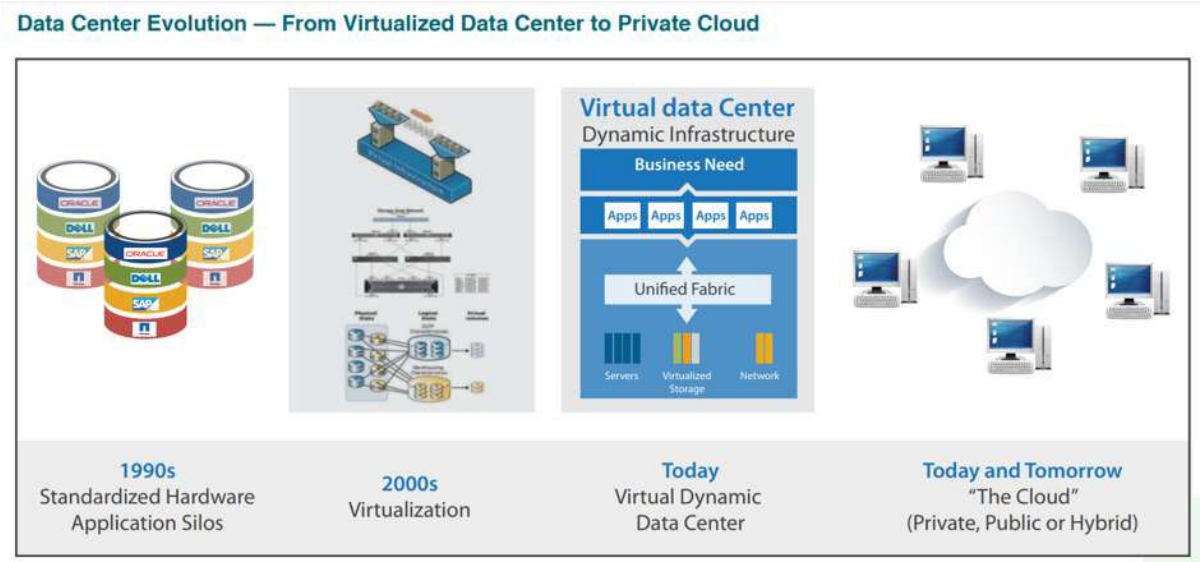
Disadvantages of Cluster

- High upfront cost
- Problem in finding fault
- Need of more space

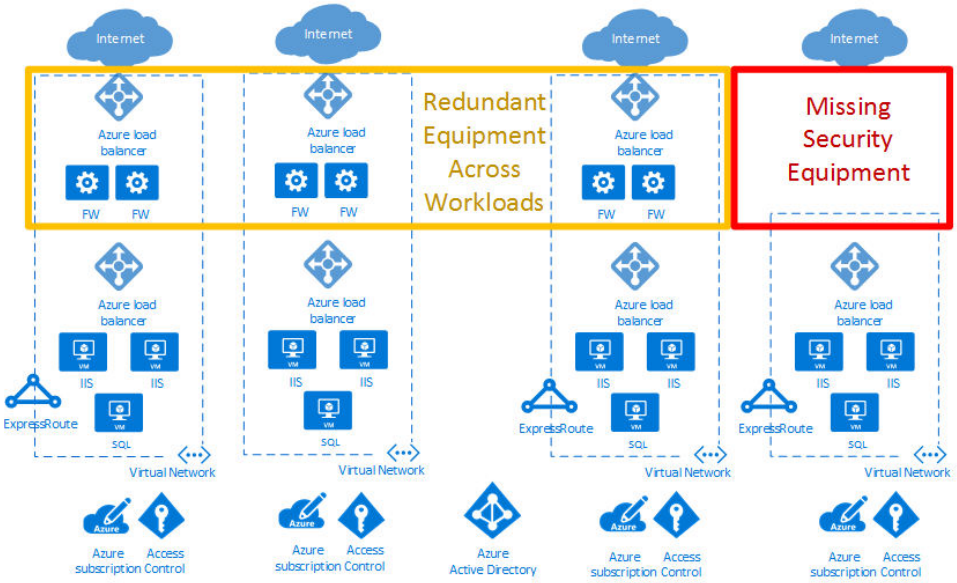
Virtual Data Center

(The cloud is great for individual workloads, but now that we have many isolated workloads, can we get more efficient?

Virtual Data Centers



Common Issues with Data Centers



Server Consolidation in Data Centers

- Server consolidation is an approach to improve the low utility ratio of hardware resources by reducing the number of physical servers.
- Among several server consolidation techniques such as centralized and physical consolidation, virtualization-based server consolidation is the most powerful.
- Data centers need to optimize their resource management and hence the concept of Virtual Data Center.

Benefits of Server Consolidation

- Consolidation enhances hardware utilization.
- This approach enables more agile provisioning and deployment of resources.
- The total cost of ownership is reduced.
- This approach improves availability and business continuity.

Benefits of Server Consolidation

- Consolidation enhances hardware utilization.
- This approach enables more agile provisioning and deployment of resources.
- The total cost of ownership is reduced.
- This approach improves availability and business continuity.

Virtual Data Center

- A virtual data center (VDC) is a set of cloud resources that support a business with computing capabilities.
- A VDC eliminates the need for a company to set up and run an on-premise data center.
- Another common name for a VDC is a software-defined data center.

Virtual Data Center contd...

- A VDC offers the same capabilities as its physical counterpart. It allows a business to set up:
  - Servers.
  - Processing power (CPU).
  - Storage clusters (RAM and disk space).
  - Networking components and bandwidth.
- The main upside of virtual data centers is the ability to add or remove capacity without having to set up or take down hardware. Every abstracted component runs on the provider's virtual machine (VM), and the client pays the usage on a pay-as-you-use basis.

Virtual Data Center contd...

- Similar to regular data center, a VDC provides computing capabilities that enable workloads of business apps and activities, such as:
  - File sharing.
  - Email operations.
  - Productivity apps.
  - CRM and ERP platforms.
  - Database operations.
  - Big data.
  - AIOps and machine learning.
  - Communication and collaboration apps.

Why companies opt to deploy a VDC

- Flexible and scalable infrastructure.
- Shorter time-to-market and idea-to-cash cycles.
- High availability.
- Higher levels of IT setup customization.
- Cost reductions (no rental, power, cooling, maintenance, or hardware costs).

Traditional DC vs VDC

Traditional data center	Point of comparison	Virtual data center
A facility that houses computer hardware and provides computing capabilities.	**Definition**	A pool of cloud resources that uses virtualization to provide computing capabilities.
A high upfront cost as companies need to buy hardware and rent space. High electricity, cooling, and maintenance costs.	**Costs**	A cost-effective pay-as-you-use payment model. No initial investment necessary.
Capital expenditure (CapEx) as companies need to acquire and maintain physical assets.	**Main investment type**	Operating expenses (OpEx) as the company pays ongoing costs only.
Designing and building a data center can take months. Each new piece of hardware requires purchasing, configuring, and racking.	**Setup speed**	Building a new VDC is typically a matter of days. Adding new VMs and capabilities requires minutes.

Traditional DC vs VDC

Traditional data center	Point of comparison	Virtual data center
The data center owner makes full use of CPU, memory, storage, and network resources.	**Hardware dedication**	A single machine can host multiple VMs, so clients can see performance issues if the vendor's device has too many tenants.
The team deploys physical servers with fixed CPUs, memory, and storage. Limited upgrade options and time-consuming server management.	**Servers**	The team deploys resizable virtual servers that keep up with current workload demands.
The team must plan for and set up switch ports, routers, and cabling.	**Networking gear**	Relies on software-defined networks (SDN) and virtual routers to scale network capacity up or down.
Data center security starts with entry restrictions and verifiable access to server racks. The in-house team is in charge of IT-level security, too.	**Security considerations**	The team focuses on IT-level security, while the provider takes care of physical protection. Most vendors offer services for IT-security as well.

Traditional DC vs VDC

Traditional data center	Point of comparison	Virtual data center
Hard to implement and manage centralized security.	**Security centralization**	Has centralized security and management.
Companies need trained personnel to rack and stack equipment. Most companies have separate compute, storage, and network teams.	**Staff requirements**	As little as two or three people can manage a VDC, but staff members require strong expertise.
Migration is a slow and expensive project.	**Data center migration**	Migrating a VDC is quick, simple, and cheap.
Difficult to move the workload from one hardware to another.	**Workload migration**	Easy workload migration between hardware platforms.
Relatively static and predictable, and typically goes one way (adding more equipment).	**Scalability**	Dynamic provisioning enables teams to scale the number of VMs up and down with speed and ease.

Traditional DC vs VDC

Traditional data center	Point of comparison	Virtual data center
A traditional data center is a large consumer of power.	**Power consumption**	Users do not cover power expenses.
Many repetitive tasks and coordination work, but not a lot of necessary expertise.	**Maintenance complexity**	Less repetitive tasks, but the team requires deep expertise.
Requires backup agents that the team must deploy, patch, and manage.	**Backups**	The hypervisor provides LAN-free and agentless backup services.
Each server needs a separate anti-virus program.	**Server anti-virus management**	Anti-virus operates at the hypervisor level.
Firewalls are centrally located and typically not part of the server.	**Firewalls**	A built-in property of the VM.

Traditional DC vs VDC

Traditional data center	Point of comparison	Virtual data center
DR occurs on a per-application basis, and every app has a different solution.	**Disaster recovery**	DR is a service and enables center-wide strategies.
Requires accurate estimation of future needs to avoid unnecessary overhead.	**Future planning**	The company pays only for the needed capacity and can scale up and down to meet the current requirements. No overhead.

Who should Implement Virtual Data Center

- Depending on the size, even single applications can benefit from using the patterns and components used to build a VDC implementation
- Implementing a VDC can help
  - enforce policy points
  - separate responsibilities
  - ensure the consistency of underlying common components.
- Organizations with a DevOps approach can also use VDC concepts to provide authorized pockets of cloud resources

Who should Implement Virtual Data Center

- Organizations requiring network and security compliance
  - At VDC, compliance is defined by a centralized policy in the hub network and centrally managed resource group.

Considerations to Implement Virtual Data Center

- Identity and Directory Service
- Security Infrastructure
- Connectivity to the Cloud
- Connectivity within the Cloud

Benefits of a Virtual Data Center

- Flexibility & Scalability
- On-demand resource provisioning
- Cost-saving opportunities
- Simplified, centralized management
- Lack of downtime
- More business agility
- Simpler Security

Flexibility & Scalability

- Cloud resources make a VDC highly flexible. Adding and removing virtual components is simple, cheap, and quick, traits an on-prem data center cannot provide. These features speed up:
  - Release cycles.
  - Project turnaround.
  - Time-to-market for new products and services.
- Developers get to enjoy more agility with a VDC. The team can:
  - Deploy from public and private catalogs of VM templates.
  - Build and take VMs down quickly and easily.
  - Create and use virtual applications (vApps).
- The near-zero deployment time and flexible VMs increase overall operational efficiency.

Fast Resource Provisioning

- With an on-prem data center, provisioning a new piece of hardware can take weeks. The company must purchase, wait for delivery, configure, and install each new component.

Cost-Saving Opportunities

- An on-site data center requires vast upfront and continuing investments, including:
  - Building, equipping, and maintaining the facility.
  - Hiring and training teams to manage the data center.
  - Acquiring the necessary hardware.
  - Setting up standby components in case the demand or traffic goes up.

Simplified Management

- If you run a virtual data center, the cloud provider is responsible for maintaining the infrastructure stack. Your in-house team only administrates the VMs, a process that occurs remotely, quickly, and in real-time.

- Clients typically manage a virtual data center from a single pane of glass. Centralized tools and interfaces grant complete visibility into the computing resources and allow admins to optimize usage.

Lack of Downtime

- A VDC helps avoid downtime due to:
  - Top-tier cloud providers ensure high availability, typically in the 99.999% range (an average of about 6 minutes of downtime per year).
  - A data recovery occurs in a matter of minutes.

Business Agility

- A virtual data center grants companies the ability to adapt to different settings and requirements quickly. You can drive business operations forward through:
  - On-demand resource provisioning.
  - The ability to test new business models quickly and without risk.
  - Improving the performance of legacy apps.
- Virtual data centers are also compatible with existing physical networks and setups. Businesses have several options when working with a VDC, including:
  - Shifting the entire IT infrastructure to the cloud.
  - Creating a hybrid solution that mixes the use of physical and virtual components.
  - Making virtual resources available only for emergencies and spikes in usage.

Simpler Security

- Managing security in a VDC is simpler than protecting a traditional data center. You do not need to worry about physical security.

VDC & Cloud Computing

- A virtual data center falls under the Infrastructure as a Service (IaaS) delivery model of cloud computing. IaaS enables a company to request physical components and form a data center with:

- Server racks.
- Storage hardware.
- Networking equipment.
- Software and hardware backups.

Concluding comment on VDC

- A virtual data center is a popular choice for companies regardless of where they are on their cloud adoption journey.
- Whether there is an urgent need to decommission hardware or a plan to make a strategic move to new IT cost models, a VDC is an excellent way to start benefiting from cloud-based resources.

Protection/Isolation among resources

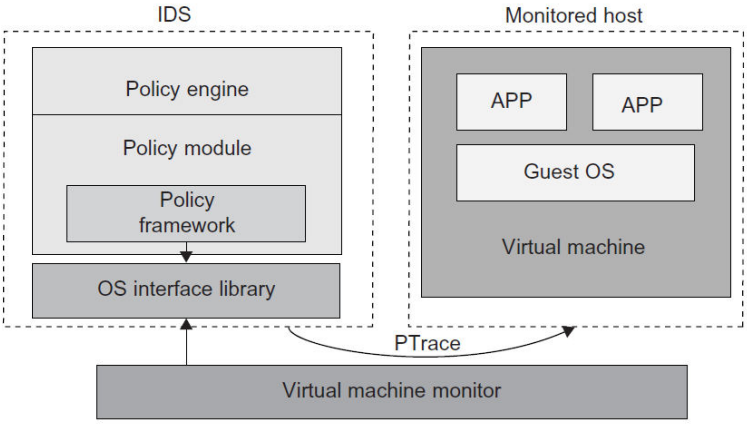
Trust management in Virtual Data Center

Trust management in virtual data center

- A VM entirely encapsulates the state of the guest operating system running inside it.
- Encapsulated machine state can be copied and shared over the network and removed like a normal file, which proposes a challenge to VM security.
- In general, a VMM can provide secure isolation and a VM accesses hardware resources through the control of the VMM, so the VMM is the base of the security of a virtual system. Normally, one VM is taken as a management VM to have some privileges such as creating, suspending, resuming, or deleting a VM.

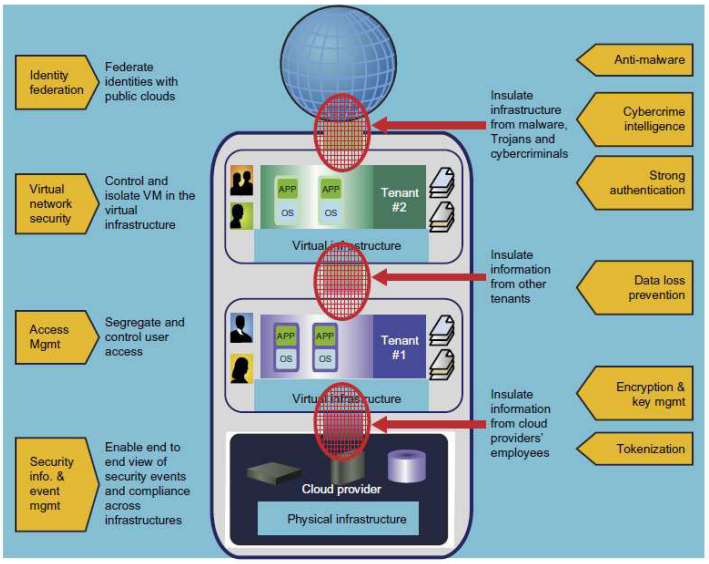
- Once a hacker successfully enters the VMM or management VM, the whole system is in danger.

IDS using a dedicated VM



The architecture of livewire for intrusion detection using a dedicated VM.

Establishing trusted zones for virtual cluster



Techniques for establishing trusted zones for virtual cluster insulation and VM isolation.

Cloud Resource Management and Scheduling

Contents

- Resource management and scheduling.
- Policies and mechanisms.
- Applications of control theory to cloud resource allocation.
- Stability of a two-level resource allocation architecture.
- Proportional thresholding.
- Coordinating power and performance management.
- A utility-based model for cloud-based Web services.
- Resource bundling and combinatorial auctions.
- Scheduling algorithms.
- Cloud scheduling subject to deadlines.

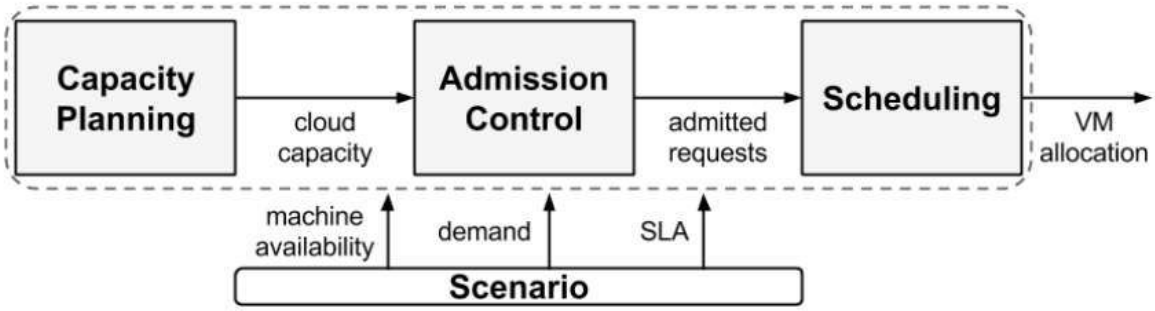
Resource management and scheduling

- Critical function of any man-made system
- It affects the three basic criteria for the evaluation of a system:
  - Functionality
  - Performance
  - Cost
- Scheduling in a computing system → deciding how to allocate resources of a system, such as CPU cycles, memory, secondary storage space, I/O and network bandwidth, between users and tasks.
- Policies and mechanisms for resource allocation.
  - Policy → principles guiding decisions.
  - Mechanisms → the means to implement policies.

Policy vs Mechanism

- A policy typically refers to the principle guiding decisions, whereas mechanisms represent the means to implement policies.
- Policy - What to do?
  - Users should not be able to read another user's file
- Mechanism - How to accomplish the requirement or enforce the policy
  - File protection properties are checked on system call

Cloud resource management (CRM) policies



Cloud resource management process.

Cloud resource management (CRM) policies

Cloud resource management policies can be grouped into five classes:

1. Admission control → prevent the system from accepting workload in violation of high-level system policies.

2. Capacity allocation → allocate resources for individual activations of a service.

3. Load balancing → distribute the workload evenly among the servers.

4. Energy optimization → minimization of energy consumption.

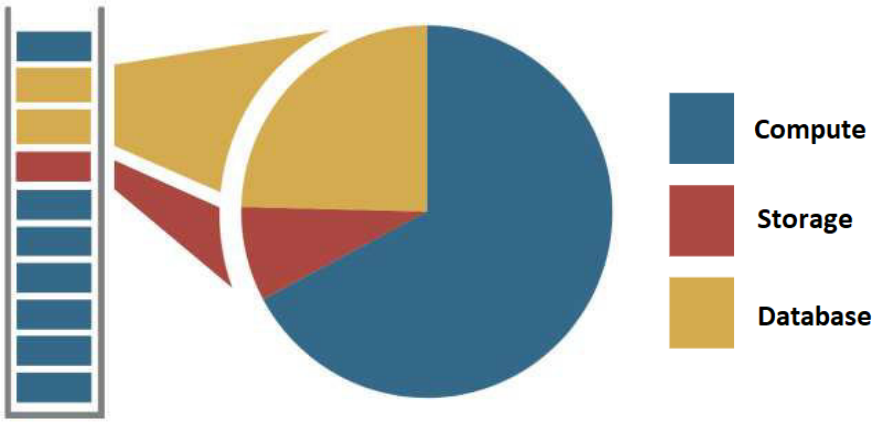
5. Quality of service (QoS) guarantees → ability to satisfy timing or other conditions specified by a Service Level Agreement.

Admission Control

1. Prevent the system from accepting workloads in violation of high-level system policies
   1. a system may not accept an additional workload that would prevent it from completing work already in progress or contracted
2. Limiting the workload requires some knowledge of the global state of the system.
3. In a dynamic system such knowledge, when available, is at best obsolete.

Capacity Allocation

1. Capacity allocation means to allocate resources for individual instances; an instance is an activation of a service.
2. Locating resources subject to multiple global optimization constraints requires a search of a very large search space when the state of individual systems changes rapidly.



Load Balancing & Energy Optimization

1. Load balancing and energy optimization are correlated and affect the cost of providing the services.
2. It was predicted that by 2026 up to 40% of the budget for IT enterprise infrastructure would be spent on energy
3. Gartner estimates global IT spending in the power and utilities market to see a five-year CAGR of about 8.2%, to reach an estimated $242 billion by 2026.

Load Balancing & Energy Optimization

In cloud computing a critical goal is minimizing the cost of providing the service and, in particular, minimizing the energy consumption. This leads to a different meaning of the term load balancing.

A (70%)	B (60%)	C (30%)	D (20%)	Situation
A (45%)	B (45%)	C (45%)	D (45%)	Idle case
A (90%)	B (90%)	C (Standby)	D (Standby)	Cloud Strategy

Quality of Service

1. Quality of service is that aspect of resource management that is probably the most difficult to address and, at the same time, possibly the most critical to the future of cloud computing.
2. Correlated with SLA

Summary - Resource Management Strategy

1. Often resource management strategies jointly target performance and power consumption.
2. Dynamic voltage and frequency scaling (DVFS)
   a. DVFS is a power management technique to increase or decrease the operating voltage or frequency of a processor to increase the instruction execution rate and, respectively, reduce the amount of heat generated and to conserve power
   b. Intel's SpeedStep andAMD's PowerNowlower the voltage and the frequency to decrease power consumption.
   c. Motivated initially by the need to save power for mobile devices, these techniques have migrated to
   d. virtually all processors, including the ones used for high-performance servers.

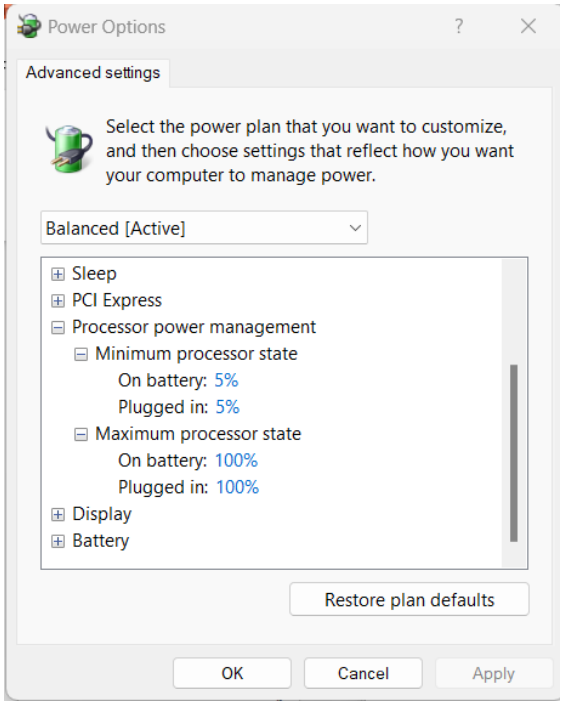
Summary - Resource Mgmt Strategy (contd.)

1. As a result of lower voltages and frequencies, the performance of processors decreases, but at a substantially slower rate than the energy consumption.

The normalized performance and energy consumption function of the processor speed. The performance decreases at a lower rate than does the energy when the clock rate decreases.

CPU Speed (GHz)	Normalized Energy (%)	Normalized Performance (%)
0.6	0.44	0.61
0.8	0.48	0.70
1.0	0.52	0.79
1.2	0.58	0.81
1.4	0.62	0.88
1.6	0.70	0.90
1.8	0.82	0.95
2.0	0.90	0.99
2.2	1.00	1.00

Summary - Resource Mgmt Strategy (contd.)



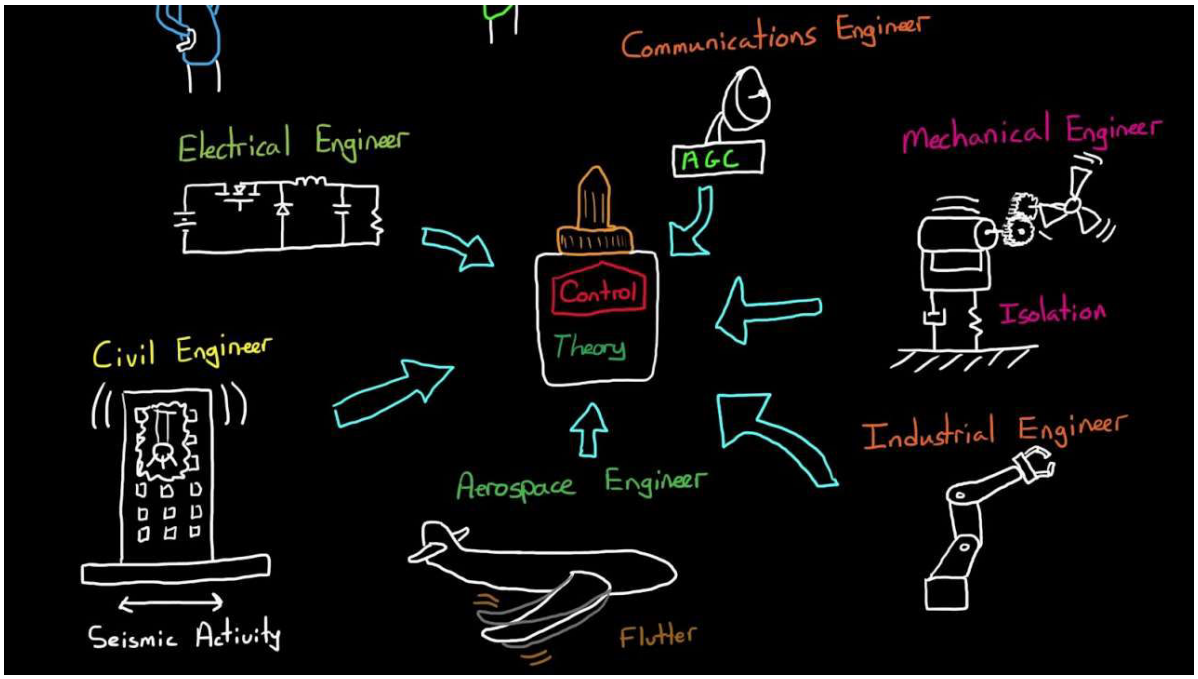
Mechanisms for the implementation of resource management policies

- Control theory → uses the feedback to guarantee system stability and predict transient behavior.
- Machine learning → does not need a performance model of the system.
- Utility-based → require a performance model and a mechanism to correlate user-level performance with cost.
- Market-oriented/economic → do not require a model of the system, e.g., combinatorial auctions for bundles of resources.

Control Theory and Resource Management

- Control theory has been used to design adaptive resource management for:
  - power management
  - task scheduling
  - QoS adaptation in Web servers
  - load balancing
- The classical feedback control methods are used in all these cases to regulate the key operating parameters of the system based on measurement of the system output;

Control Theory and Resource Management



Control theory application to cloud resource management (CRM)

- The main components of a control system:
  - The inputs → the offered workload and the policies for admission control, the capacity allocation, the load balancing, the energy optimization, and the QoS guarantees in the cloud.
  - The control system components → *sensors* used to estimate relevant measures of performance and *controllers* which implement various policies.
  - The outputs → the resource allocations to the individual applications.

Feedback and Stability

- Control granularity → the level of detail of the information used to control the system.
  - Fine control → very detailed information about the parameters controlling the system state is used.
  - Coarse control → the accuracy of these parameters is traded for the efficiency of implementation.
- The controllers use the feedback provided by sensors to stabilize the system. Stability is related to the change of the output.

Sources of Instability in Control System

- The delay in getting the system reaction after a control action.
- The granularity of the control, the fact that a small change enacted by the controllers leads to very large changes of the output.
- Oscillations, when the changes of the input are too large and the control is too weak, such that the changes of the input propagate directly to the output.

The model requires a large number of parameters

Name	Description
I	the set of servers
K	the set of classes
Λ_k	the aggregate rate for class $k \in K$, $\Lambda_k = \lambda_k + \sum_{k' \in K} \Lambda_{k'} \pi_{k,k'}$
a_i	the availability of server $i \in I$
A_k	minimum level of availability for request class $k \in K$ specified by the SLA
m_k	the slope of the utility function for a class $k \in K$ application
N_k	number of applications in class $k \in K$
H_i	the range of frequencies of server $i \in I$
$C_{i,h}$	capacity of server $i \in I$ running at frequency $h \in H_i$
$c_{i,h}$	cost for server $i \in I$ running at frequency $h \in H_i$
$\bar{c}_i$	average cost of running server i
$\mu_{k,j}$	maximum service rate for a unit capacity server for tier j of a class k request
cm	the cost of moving a virtual machine from one server to another
cs_i	the cost for switching server i from the stand-by mode to an active state
$RAM_{k,j}$	the amount of main memory for tier j of class k request
$\overline{RAM}_i$	the amount of memory available on server i

Resource bundling

- Resources in a cloud are allocated in bundles.
- Users get maximum benefit from a specific combination of resources: CPU cycles, main memory, disk space, network bandwidth, and so on.
- Resource bundling complicates traditional resource allocation models and has generated an interest in economic models and, in particular, in auction algorithms.
- The bidding process aims to optimize an objective function *f(x,p)*.
- In the context of cloud computing, an auction is the allocation of resources to the highest bidder.

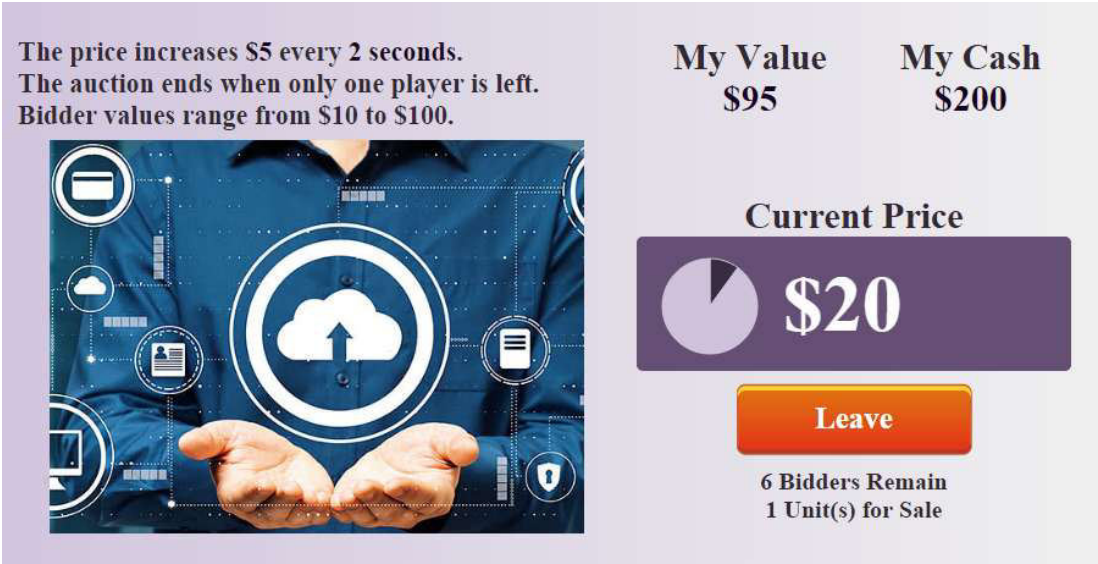
Combinatorial auctions for cloud resources

- Auctions in which participants can bid on combinations of items, or packages, are called combinatorial auctions.
- Such auctions provide a relatively simple, scalable, and tractable solution to cloud resource allocation.
- Two recent combinatorial auction algorithms are:
  - simultaneous clock auction
  - clock proxy auction
  - The algorithm discussed in current session is called the AScending Clock Auction (ASCA).
- In all these algorithms the current price for each resource is represented by a “clock” seen by all participants at the auction.

Combinatorial auctions for cloud resources

- Users provide bids for desirable bundles and the price they are willing to pay.
- Prices and allocation are set as a result of an auction.
- Ascending Clock Auction, (ASCA) → the current price for each resource is represented by a "clock" seen by all participants at the auction.
- The algorithm involves user bidding in multiple rounds; to address this problem the <u>user proxies</u> automatically adjust their demands on behalf of the actual bidders.

Combinatorial auctions for cloud resources



Combinatorial auctions for cloud resources

- In the auction, users provide bids for desirable bundles and the price they are willing to pay. We assume a population of U users,

$$U \text{ users}, u = \{1, 2, \ldots, U\}, \text{ and } R \text{ resources}, r = \{1, 2, \ldots, R\}$$

- The bid of user u is

$$\mathcal{B}_u = \{\mathcal{Q}_u, \pi_u\}$$

- with

$\mathcal{Q}_i = (q_u^1, q_u^2, q_u^3, \ldots)$ an R-component vector; each element of this vector, q_u^i, represents a bundle of resources user u would accept and, in return, pay the total price π_u.

Combinatorial auctions for cloud resources

Each vector component q_u^i is a positive quantity and encodes the quantity of a resource desired or, if negative, the quantity of the re- source offered.

A user expresses her desires as an *indifference set* $\mathcal{I} = (q_u^1 \text{ XOR } q_u^2 \text{ XOR } q_u^3 \text{ XOR} \ldots)$.

vector $p = (p^1, p^2, \ldots, p^R)$	final auction prices for individual resources
$x_u = (x_u^1, x_u^2, \ldots, x_u^R)$	amounts of resources allocated to user u
$[(x_u)^T p]$	total price paid by u for the bundle of resources if the bid is successful at time T
$[\min_{q \in \mathcal{Q}_u} (q^T p)]$	final price established through the bidding process

The bidding process aims to optimize an *objective function* $f(x, p)$. This function could be tailored to measure the net value of all resources traded, or it can measure the *total surplus* – the difference between the maximum amount users are willing to pay minus the amount they pay. Other optimization functions could be considered for a specific system, e.g., the minimization of energy consumption or of security risks.

Function for combinatorial auctions algorithm

The function to be maximized is

$$\max_{x,p} f(x, p)$$

Constraint for combinatorial auctions algorithm

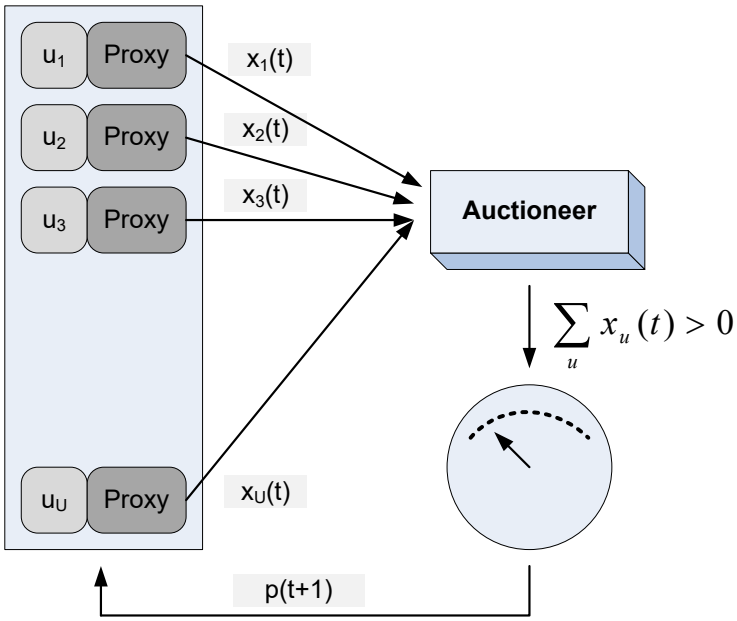
$x_u \in \{0 \cup \mathcal{Q}_u\}, \forall u$	A user gets all resources or nothing.
$\sum_u x_u \leqslant 0$	Final allocation leads to a net surplus of resources.
$\pi_u \geqslant (x_u)^T p, \forall u \in \mathcal{W}$	Auction winners are willing to pay the final price.
$(x_u)^T p = \min_{q \in \mathcal{Q}_u} (q^T p), \forall u \in \mathcal{W}$	Winners get the cheapest bundle in $\mathcal{I}$.
$\pi_u < \min_{q \in \mathcal{Q}_u} (q^T p), \forall u \in \mathcal{L}$	The bids of the losers are below the final price.
$p \geqslant 0$	Prices must be nonnegative.

Pricing and allocation algorithms

- In ASCA algorithm the participants at the auction specify the resource and the quantities of that resource offered or desired at the price listed for that time slot. Then the following excess vector is computed:

$$z(t) = \sum_u x_u(t)$$

- If all its components are negative, the auction stops; negative components mean that the demand does not exceed the offer.
- If the demand is larger than the offer, the auctioneer increases the price for items with a positive excess demand and solicits bids at the new price.



The schematics of the ASCA algorithm; to allow for a single round auction users are represented by proxies which place the bids $x_u(t)$. The auctioneer determines if there is an excess demand and, in that case, it raises the price of resources for which the demand exceeds the supply and requests new bids.

Notation for Proxy

These proxies can be modeled as functions that compute the "best bundle" from each Q_u set given the current price

$$\mathcal{Q}_u = \begin{cases} \hat{q}_u & \text{if } \hat{q}_u^T p \leqslant \pi_u \quad \text{with} \quad \hat{q}_u \in \arg\min\,(q_u^T p) \\ 0 & \text{otherwise} \end{cases}$$

Pseudo code for ASCA algorithm

The input to the ASCA algorithm: U users, R resources, $\bar{p}$ the starting price

```
set t = 0, p(0) = p̄
loop
   collect bids x_u(t) = G_u(p(t)), ∀u
   calculate excess demand z(t) = Σ_u x_u(t)
   if z(t) <0 then
      break
      else
         update prices p(t + 1) = p(t) + g(x(t), p(t))
         t ← t + 1
   end if
end loop
```

Cloud scheduling algorithms

- Scheduling → responsible for resource sharing at several levels:
  - A server can be shared among several virtual machines.
  - A virtual machine could support several applications.
  - An application may consist of multiple threads.
- A scheduling algorithm should be efficient, fair, and starvation-free.
- The objectives of a scheduler:
  - Batch system → maximize throughput and minimize turnaround time.
  - Real-time system → meet the deadlines and be predictable.
- Best-effort: batch applications and analytics.
- Common algorithms for best effort applications:
  - Round-robin.
  - First-Come-First-Serve (FCFS).
  - Shortest-Job-First (SJF).
  - Priority algorithms.

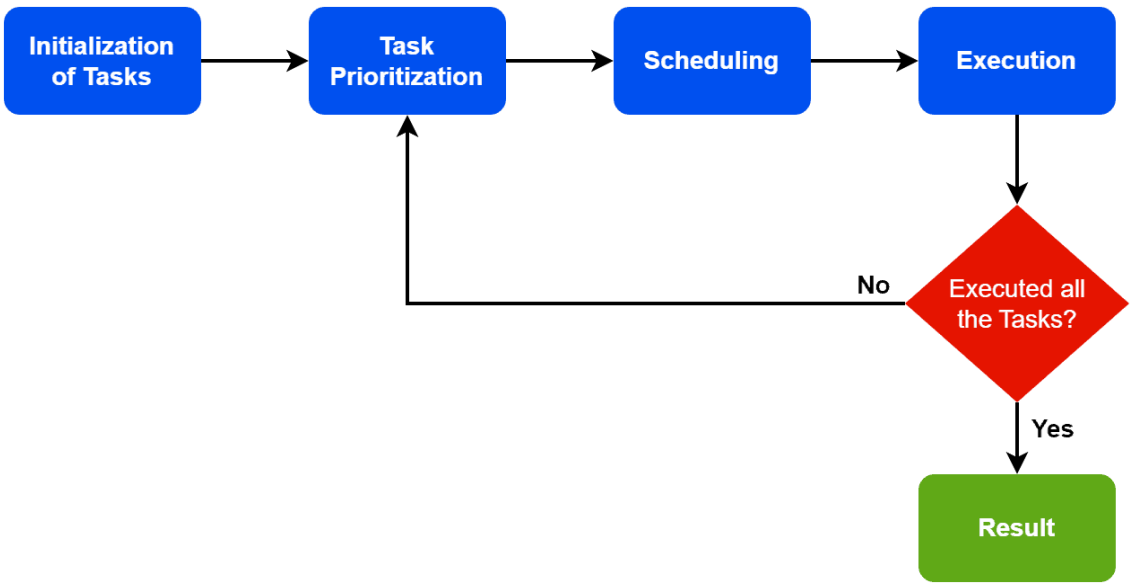
Cloud scheduling algorithms (cont'd)

- Multimedia applications (e.g., audio and video streaming)
  - Have soft real-time constraints.
  - Require statistically guaranteed maximum delay and throughput.
- Real-time applications have hard real-time constraints.
- Scheduling algorithms for real-time applications:
  - Earliest Deadline First (EDF).
  - Rate Monotonic Algorithms (RMA).
- Algorithms for integrated scheduling of several classes of applications:
  - Resource Allocation/Dispatching (RAD) .
  - Rate-Based Earliest Deadline (RBED).

Earliest Deadline First (EDF)

- Dynamic scheduling algorithm
- Can be used in real-time operating systems for scheduling processes with specific deadlines.
- Priority-based algorithm where we assign the highest priority to the tasks with the earliest deadline.
- The idea is to minimize the probability of missing a deadline.
- If multiple tasks have the same deadline, the CPU executes the task with the highest priority.
- EDF is an optimal algorithm for scheduling a set of periodic tasks with known deadlines and execution times as long as the total utilization of the tasks doesn't exceed unity. Hence, the total execution time of all tasks should be less than the total available time.

Earliest Deadline First (EDF)



Rate Monotonic Scheduling (RMA)

- RMA is a priority algorithm that belongs to the static priority scheduling category of Real Time Operating Systems.
- It is preemptive in nature.
- The priority is decided according to the cycle time of the processes that are involved.
- If the process has a small job duration, then it has the highest priority.
- Thus, if a process with highest priority starts execution, it will preempt the other running processes.

Rate Monotonic Scheduling (RMA) contd.

- Necessary condition to check is the processor utilization

$$U = \sum_{i=1}^{n} \frac{C_i}{T_i} \leq n(2^{1/n} - 1)$$

- Where n is the number of processes in the process set, C_i is the computation time of the process, T_i is the Time period for the process to run and U is the processor utilization.
- The above expression should hold true to proceed further

Example: Rate Monotonic Scheduling (RMA)

Processes	Execution Time (C)	Time period (T)
P1	3	20
P2	2	5
P3	2	10

- Priority: Lower the time period in which the process should execute, highest will be the priority of such process
- Execution time & Time period: In a time duration of 5 units P2 will get the execution for total 2 units

Example: Rate Monotonic Scheduling (RMA)

Processes	Execution Time (C)	Time period (T)
P1	3	20
P2	2	5
P3	2	10

Utilization calculation:

- n(2^1/n - 1) = 3 (2^1/3 - 1) = 0.7977
- U = 3/20 + 2/5 + 2/10 = 0.75
- It is less than 1 or 100% utilization. The combined utilization of three processes is less than the threshold of these processes which means the above set of processes is schedulable and thus satisfies the above equation of the algorithm.

Example: Rate Monotonic Scheduling (RMA)

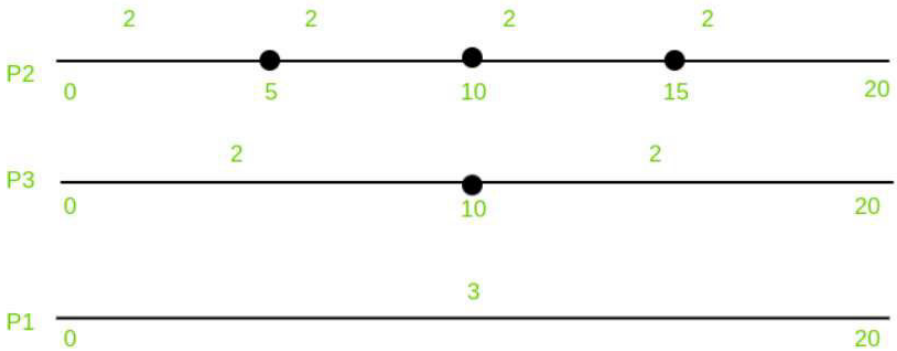
Processes	Execution Time (C)	Time period (T)
P1	3	20
P2	2	5
P3	2	10

Scheduling time – For calculating the Scheduling time of the algorithm take the LCM of the Time period of all the processes. LCM (20, 5, 10) of the above example is 20. Thus we can schedule it by 20-time units.

Priority – As discussed above, the priority will be the highest for the process which has the least running time period. Thus P2 will have the highest priority, and after that P3 and lastly P1.

P2 > P3 > P1 (5 < 10 < 20)

Example: Rate Monotonic Scheduling (RMA)



The above figure says that Process P2 will execute two times for every 5 time units, Process P3 will execute two times for every 10 time units and Process P1 will execute three times in 20 time units. This has to be kept in mind for understanding the entire execution of the algorithm below.

P2	P2	P3	P3	P1	P2	P2	P1	P1		P2	P2	P3	P3		P2	P2			

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Condition to met in RMA

- The analysis of Rate monotonic scheduling assumes few properties that every process should possess. They are :

- Processes involved should not share the resources with other processes.
- Deadlines must be similar to the time periods. Deadlines are deterministic.
- Process running with highest priority that needs to run, will preempt all the other processes.
- Priorities must be assigned to all the processes according to the protocol of Rate monotonic scheduling.

Advantages of RMA

- It is easy to implement.
- If any static priority assignment algorithm can meet the deadlines then rate monotonic scheduling can also do the same. It is optimal.
- It consists of a calculated copy of the time periods, unlike other time-sharing algorithms as Round robin which neglects the scheduling needs of the processes.

Disadvantages of RMA

- It is very difficult to support aperiodic and sporadic tasks under RMA.
- RMA is not optimal when the task period and deadline differ.

RMA - Points to ponder

- Focus on Periods: RMA prioritizes processes based on their periods, not burst times. The process with the shortest period gets the highest priority.
- Deterministic Execution: RMA assumes processes have a fixed burst time that executes entirely within each period. There's no concept of a total burst time that needs to be accumulated.
- Meeting Deadlines: The goal of RMA is to ensure all processes complete their execution (burst time) within their period (deadline).

Performance Analysis & Benchmarking

Performance Analysis

Requirement for quality metrics

- Correlation
  - there should be a linear correlation between the observed attribute and the metric output
- Tracking
  - If the quality changes over time, the metric output should change as well and do so quickly.
- Monotonicity:
  - If the building height, this means that if a building under construction currently has a height value X, then the new height value Y should be greater than X
- Discriminative power:
  - The metric should be able to clearly differentiate between low and high quality levels. Tall, medium, small doesn't have the discriminative power
- Reliability
  - We should be able to rely on the accuracy of the metric. Example, the building height value should be correct most of the time if not always.

Requirement for quality metrics contd.

- Reproducibility
  - measuring the exact same situation more than once should yield approximately the same measurement result every time – independent of measurement errors.
- Fine-grained

Cloud analysis parameters

- Performance
  - Throughput
  - Latency
- Availability & Fault-Tolerance
  - Uptime
  - Time between failures (MTBF)
  - Time to repair (MTTR)
- Security
- Elastic scalability
  - Elasticity
  - Scalability
- Data consistency

Cloud industry challenges - Innovation

- *New technologies emerging faster and faster each day
- *This drives demand for new cloud services, product offerings mainly based customer demand
- *Do you wait for new technology to be “stable” or do you hop on early adopter wave?
- *How can you be sure you are making the right choice?

Cloud industry challenges - Changing Requirements

- *Many enterprises and small businesses focus on being agile to stay competitive, this oftentimes means that what worked last week won't work tomorrow
- *If competitors are leveraging new tech they might have an advantage over those that are "behind the times"
- *How do these changes impact performance or cost?
- *But what about containers??

Cloud industry challenges - Pricing

- Cloud Pricing / Cost is not always an easy thing to figure out
- Often times the expected spend and actual spend don't match up
- Trying to compare the pricing between two or more clouds can be head spinning
- Cloud Spectator is tackling this challenge with our Cloud Transparency Platform (shameless plug)

The need for cloud transparency

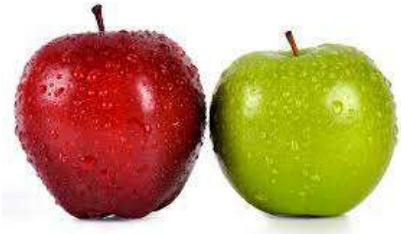
- Clearly there are many challenges for enterprises using or moving to the cloud
- Measure three times, cut once – don't just move because everyone else is…make sure you understand what you are moving to!
- Having access to transparent data around cloud performance and pricing can save you lots of headaches and dollars in the future!
- Benchmarking price and performance can go a long way here

Transparency - All clouds are equal right?

- Cloud Spectator specializes in validating performance claims and bringing transparent data to the enterprise
- On paper most cloud infrastructure appears similar
- Performance and Pricing all over the place for similar spec'd VMs
- VM performance is only one aspect, PaaS and SaaS also show large difference in performance
- Head hurt yet?

Comparing Apple to Apple

- Comparing similar Cloud services from various providers can seem like comparing apples to oranges.
- Decide what matters to you most
  - Control?
  - Performance?
  - Price?
  - Support?
  - Ease of use?



Evaluating Cloud Provider - Other things

- Regions and countries that providers operate in matter on the business level
- Data center locations – hardware differences, temperature, latency
- Size and business stability – perf doesn't matter if your boss won't go with a small cloud
- Security and risk
- Reliability and trustworthiness
- Audit and compliance opacity
- Pricing models and contract terms
- Service-Level Agreements (SLA)
- Ease of use, self-service, robust API

amazon web services

Azure

IBM SOFTLAYER

DigitalOcean

Evaluating Cloud Provider - AWS

General Purpose

Compute Optimized

Memory Optimized

Accelerated Computing

Storage Optimized

HPC Optimized

Instance Features

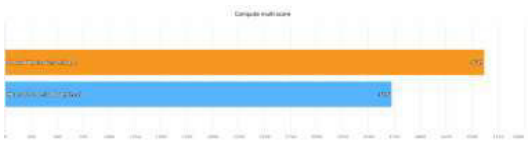
Measuring Instance Performance

Instance Type	Description
T-Series (T2, T3)	Burstable performance instances suitable for workloads with variable CPU usage, such as development and testing environments.
M-Series (M5, M4)	General-purpose instances with a balance of compute, memory, and networking resources. Ideal for small and medium-sized databases, backend servers, and enterprise applications.
C-Series (C6g, C5, C4)	Compute-optimized instances designed for applications that require high performance from the CPU, such as batch processing, high-performance web servers, and scientific modeling.
R-Series (R5, R6g)	Memory-optimized instances designed for applications that require large memory capacity, such as in-memory databases and real-time big data analytics.
X-Series (X2gd, X1e)	High-memory instances optimized for large-scale, in-memory databases like SAP HANA. Suitable for high-performance database workloads.
P-Series (P4, P3)	GPU-based compute instances for machine learning, AI training, high-performance computing, and rendering.
G-Series (G5, G4dn)	GPU instances optimized for machine learning inference, gaming, and graphics-intensive applications.
D-Series (D3, D2)	Storage-optimized instances designed for applications requiring high sequential read and write access to large datasets, such as data warehouses and Hadoop clusters.
I-Series (I3, I4i)	Storage-optimized instances designed for applications requiring high random IOPS and low-latency performance, such as NoSQL databases.
F-Series (F1)	FPGA-based instances for hardware acceleration in custom workloads like genomics research and financial analytics.
H-Series (H1)	Storage-optimized instances for data-intensive workloads such as MapReduce, distributed file systems, and log processing.
Z-Series (z1d)	High-compute, high-memory instances with a high clock speed processor for electronic design automation (EDA) and other compute-intensive tasks.
Inf-Series (Inf1)	Machine learning inference-optimized instances designed for high-performance and cost-effective AI inference workloads.
Mac-Series (Mac1)	Instances for macOS applications, ideal for Apple-based CI/CD pipelines and app development.

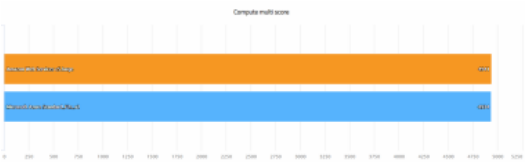
Evaluating Cloud Provider – CPU performance AWS

- AWS offers slightly less variation between VM series compared to Azure
- T3 burstable VMs add another layer of complexity, measure burst or not?

General Purpose – 40% perf diff between AWS and Azure!



Compute Optimized – Very similar performance



Evaluating Cloud Provider - Azure

Instance Type	Description
B-Series	Burstable VMs for workloads with low to moderate baseline CPU utilization, such as development, testing, and small-scale applications.
D-Series	General-purpose VMs offering a balance of CPU, memory, and disk. Ideal for web servers, databases, and medium-scale enterprise applications.
E-Series	Memory-optimized VMs for applications requiring high memory-to-CPU ratios, such as SAP HANA, in-memory analytics, and big data workloads.
F-Series	Compute-optimized VMs for CPU-intensive tasks such as analytics, gaming, and batch processing.
M-Series	High-memory VMs designed for large-scale SAP HANA workloads and other in-memory database applications.
N-Series	GPU-optimized VMs for compute and graphics-intensive workloads like machine learning, AI training, and visualization. Subtypes include: - NC-series: For compute-intensive workloads. - ND-series: For deep learning and AI training. - NV-series: For visualization and rendering tasks.
L-Series	Storage-optimized VMs for low-latency and high-throughput applications, such as NoSQL databases and data analytics.
H-Series	High-performance compute VMs for scientific and engineering workloads, such as molecular modeling and finite element analysis.
A-Series	Entry-level VMs for development, testing, and small applications, offering low-cost compute resources.
Av2-Series	General-purpose, cost-effective VMs for low-demand workloads like development and small-scale web servers.
Ddv4/Dsv4-Series	General-purpose VMs with local SSD storage for enhanced performance in applications requiring temporary storage.
DC-Series	Confidential compute VMs with support for secure enclaves, ideal for sensitive data processing.
G-Series	Memory and storage-optimized VMs for large database workloads, such as SQL and NoSQL databases.
HBv3-Series	High-performance VMs optimized for MPI workloads and distributed applications like weather modeling and fluid dynamics simulations.
VMSS (Scale Sets)	Automatically scalable VMs for large-scale, high-availability applications. Ideal for web apps and microservices.

Evaluating Cloud Provider - GCP

Instance Type	Description
E2	Cost-effective general-purpose instances with predictable performance. Ideal for web applications, small databases, and development environments.
N1	General-purpose instances suitable for a wide variety of workloads, offering a balance of CPU and memory.
N2	Next-generation general-purpose instances with better price-performance compared to N1. Used for general workloads like application servers and databases.
N2D	Cost-optimized general-purpose instances powered by AMD EPYC processors, ideal for memory-bound and compute-bound workloads.
C2	Compute-optimized instances designed for CPU-intensive tasks like scientific modeling, gaming, and media transcoding.
C2D	Compute-optimized instances powered by AMD EPYC processors, providing cost-effective performance for CPU-heavy applications.
M1	Memory-optimized instances for applications requiring very high memory, such as large in-memory databases and real-time data analytics.
M2	Next-generation memory-optimized instances offering even more memory per vCPU than M1, ideal for SAP HANA and similar workloads.
A2	GPU-accelerated instances for AI training, machine learning, deep learning, and computationally intensive workloads.
E2 Micro/Small	Micro and small general-purpose instances for lightweight tasks, such as low-traffic web servers and development environments.
N1 High-CPU	General-purpose instances with higher vCPU-to-memory ratios, suitable for compute-intensive tasks.
N1 High-Memory	General-purpose instances with higher memory-to-vCPU ratios for memory-intensive tasks like data processing.
T2D	Compute-optimized instances powered by AMD EPYC processors, offering high performance at a lower cost.
GPU Instances	Instances with NVIDIA GPUs for high-performance AI and ML training, scientific computing, and rendering.
Spot Instances	Preemptible instances for temporary workloads at a lower cost, suitable for batch jobs and fault-tolerant applications.
Custom VMs	Fully customizable instances allowing users to define the number of vCPUs and memory to match specific workload needs.
Shared-core Instances	Cost-effective instances with shared vCPU resources, ideal for lightweight applications and development/testing.
VM Scale Sets	Automatically scalable VMs for highly available and elastic applications. Suitable for large-scale applications and microservices.

Key Definitions

- **Benchmarking**: The continual and collaborative discipline of measuring and comparing the results of key work processes.

- A performance or key performance indicator (KPI) is a measure of performance.
- Performance measurement is the process used by an organization to establish the parameters to measure if programs are reaching the desired results.

Why benchmarking?

- If you can measure it you can improve it!

- VM CPU and DISK performance is not static, is ever changing

- Not all clouds use the same hypervisor or configuration settings

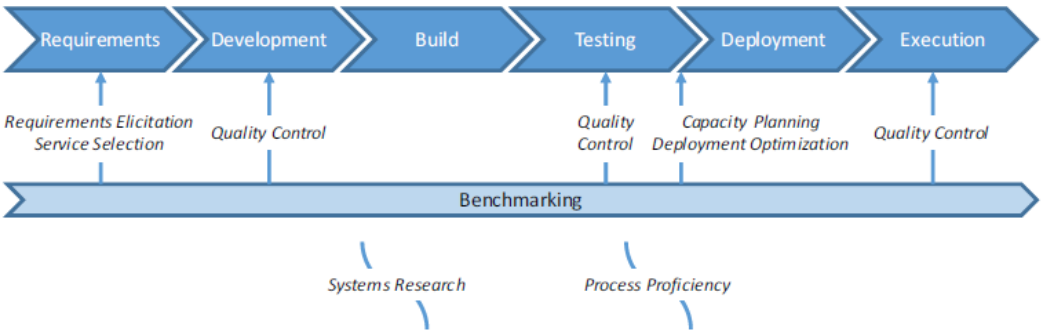
- Getting work done in the cloud costs money, therefore getting work done as quickly or efficiently as possible is key

Why cloud benchmarking?

- Choosing the best cloud platform for application deployment.
- Validating migration decisions (on-premise to cloud or between cloud providers).
- Optimizing cost and performance for existing cloud deployments.
- Evaluating multi-cloud or hybrid cloud strategies.

Motivation for running benchmarks

- Comparing different software and hardware systems
- Comparing different machines in a compatible family
- Comparing different releases of a software product



Who uses data?

- Decision makers
  - State
  - Funding agency
  - Providers
- Consumers
  - Clients
  - Employers
  - Employees

Challenge in Benchmarking

- **Dynamic Pricing and Services**: Frequent changes in cloud services and pricing.
- **Regional Variability**: Performance and costs vary by geographic location.
- **Vendor Lock-in**: Features specific to one provider may not generalize well.
- **Hidden Costs**: Egress charges, support costs, and compliance-related expenses.

Challenge

The challenge is to bring it all together to create a cohesive picture of a cloud.

Data without a context

- We are deluged with data without meaning
- Data without a context is just a bunch of numbers

Steps in Comparative Benchmarking

- Define the objective
- Select metrics
- Identify benchmarked services
- Use benchmarking tools
- Design benchmarking scenarios
- Analyze results
- Summarize findings

Define the Objectives

- Define the objective
- Performance comparison (e.g., CPU, memory, storage, or network throughput)
- Cost analysis for specific workloads
- Evaluating reliability and availability
- Scalability and elasticity assessment
- Specialized requirements like AI/ML support or IoT capabilities

Select the Metrics

- Compute: CPU/memory performance, virtual machine startup time, scalability.
- Storage: Read/write IOPS, latency, data transfer speeds, backup times.
- Networking: Latency, throughput, jitter, packet loss.
- Pricing: Hourly rates, data transfer costs, and pricing for additional services.
- Support and SLA: Technical support quality, SLA guarantees, and downtime.

Identify Benchmarked Services

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP)
- IBM Cloud
- Oracle Cloud
- Alibaba Cloud

Use Benchmarking Tools

- **CloudHarmony**: Compares performance and reliability metrics across cloud providers.
- **Phoronix Test Suite**: Evaluates performance of compute and storage resources.
- **SPEC Cloud Benchmark**: Measures scalability and elasticity of cloud platforms.
- **Iperf or Netperf**: Assesses network performance.
- **YCSB (Yahoo! Cloud Serving Benchmark)**: Evaluates cloud storage services.
- **Custom Tools**: Build workloads specific to the application and use system-level profilers.

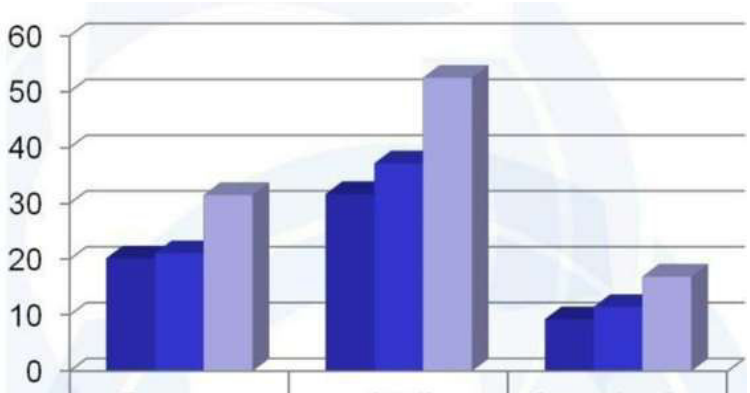
Design Benchmarking Scenarios

- **General-purpose workloads:** Compare VMs or containers.
- **Data-intensive workloads:** Test data analytics, large file storage, and retrieval.
- **Compute-intensive workloads:** Benchmark high-performance computing scenarios.
- **AI/ML workloads:** Run AI frameworks like TensorFlow or PyTorch.
- **Multi-cloud or hybrid scenarios:** Test interoperability and data movement across platforms.

Analyze Results

- Raw performance metrics.
- Price-performance ratio (value delivered per cost unit).
- Additional features such as managed services, ease of use, and integration.

Example: Comparative Benchmarking



	Process	Staff	Organization
Group 1 Avg	20.2	31.7	9.3
Group 2 Avg	21.1	37.2	11.5
Max Score	31.5	52.5	16.9

How to improve?

- Benchmarks do not give any indication as how to improve

How -> Process benchmarking

- Based on a simple premise: The methods of top performing cloud providers differ from those of other
- Process benchmarking systematically compares tactics of top cloud providers with those of others
- Methods that distinguish between top performers and others may be regarded as potential best practices

Benchmarking Tools

Benchmarking storage with FIO

- Excellent tool for stressing out storage
- Random Read - seems to be most common
- Random Write - my favorite metric
- Databases and apps need to write to disk, sometimes causes a lock = I/O wait = Load
- 95th percentile response times
- https://github.com/axboe/fio

Benchmarking CPU/vCPU - GeekBench

- Commercial tool but worth the price if you do a lot of benchmarking
- Easy to run and compare results regardless of specifications
- vCPUs are arbitrary - GeekBench MultiCore result is not
- Single Socket generally provides highest performance
- Multi Core doesn't seem to scale as well
- http://geekbench.com/

Benchmarking up the stack - Web

- Start with what matters, your businesses or your customer's site / app!
- Monitor website performance regularly.
- Load Test / DoS your servers / infrastructure before going live (let your provider know first!)
- Commercial tools like LoadStorm make it easy to impress your clients
- Can also use free software like Siege and ApacheBench
- Resource / APM help figure out what magic formula is
- Don't just look at numbers and be done, act on / improve this, explain to your customers

Benchmarking database with Sysbench OLTP

- Helps provide a somewhat standard baseline
- Don’t guess if a setting helped, test and measure!
- Has OLTP Read-Only, Read/Write, Transactional, many other modes
- Also used as a second opinion for Disk I/O benchmarks
- https://github.com/akopytov/sysbench

Benchmarking everything else - Phoronix

- 100+ benchmarks to run
- Apache, Nginx, PHP are a few relevant ones
- Great for a quick, basic benchmark
- Not the best tool for extensive / in depth benchmarking (imho)
- https://www.phoronix-test-suite.com/

Determining cloud value - Universal Metric

- Solving a business or customer problem is always good, but how do you overdeliver?
- Focus on value and efficiency!
- Cloud price-performance is an easy value to calculate.
- Value (bang for your buck or performance obtained per dollar spent) sheds much more light on pure price or performance metrics
- Once you have all three metrics you can decide where you want to optimize.

Performance / Price = Value

Cloud service benchmarking

How to decide on a cloud service

- How good is the performance of a cloud service under some stress situation?
- How well does the service scale?
- How does the service behave in the presence of failures?
- Is the data that I manage in the cloud consistent?
- Do all clients accessing the data see the same data?

Cloud service benchmarking

- Cloud service benchmarking is about IT benchmarking of software services. The focus is on
  - technical IT qualities of services - performance.
  - inter-relationships and trade-offs between conflicting qualities, such as performance and security.
- Provider-side access as well as a benchmarking approach from a service consumer viewpoint as necessary.

Cloud service benchmarking

- Cloud service benchmarking is a way to systematically study the quality of cloud services based on experiments.
- A benchmark tool creates an artificial load on the cloud service under test – or System under Test (SUT) – while carefully
- tracking detailed quality metrics. A key design goal of cloud service benchmarking is to mimic an application as close as possible to get meaningful results.
- Benchmark runs also aim to extensively stress the service, e.g., through system load or even injected failures.

Monitoring vs benchmarking

- Benchmarking and monitoring relate to the process of measuring quality and collecting information on system states so that both activities are closely related and inherently interested in the same kind of data. There are also some special cases which cannot be clearly categorized as either benchmarking or monitoring.

Monitoring vs benchmarking contd..

- Monitoring is about passive observation of a system or service to detect undesired states.
- Monitoring systems are designed to be as nonintrusive

- Benchmarking, is inherently driven by the desire to answer a specific question, e.g., how a system will react to specific changes or a specific kind of stress.
- Benchmarking is, therefore, about comparison: either directly by comparing different systems and configurations or indirectly by comparing measurement values recorded at different points in time under (presumably) identical measurement conditions.

Monitoring vs benchmarking contd..

- Monitoring is used to drive a dashboard representing the current system state

- The visibility delay for information must be as low as possible – Monitoring often requires that data is discarded and only summaries are recorded and displayed

- Benchmarking, results are often determined based on offline analysis.
- Utmost importance to collect detailed fine-grained data to be able to find the root causes of observed phenomena.

Monitoring vs benchmarking contd..

- Monitoring system is an entirely passive observer which measures values based on an externally caused workload.
- A benchmarking system always contains a load generator component to stress the system under test in particular ways with the goal of answering very specific questions.
- In benchmarking practice, monitoring tools are often used along with benchmarking tools to collect additional information on the SUT but also on the benchmarking system itself to avoid certain measurement flaws

Monitoring vs benchmarking contd..

- Monitoring typically makes use of low-level performance metrics provided by the operating system (e.g., CPU utilization) or hardware switches (e.g., packet throughput) as well as very high-level (but discrete) events like node failures or custom business events.
- While benchmarking has access to arbitrarily complex metrics, monitoring is more limited in this regard.
- Again, it may be hard to differentiate whether a given metric will only be available in monitoring, benchmarking, or both in practice.

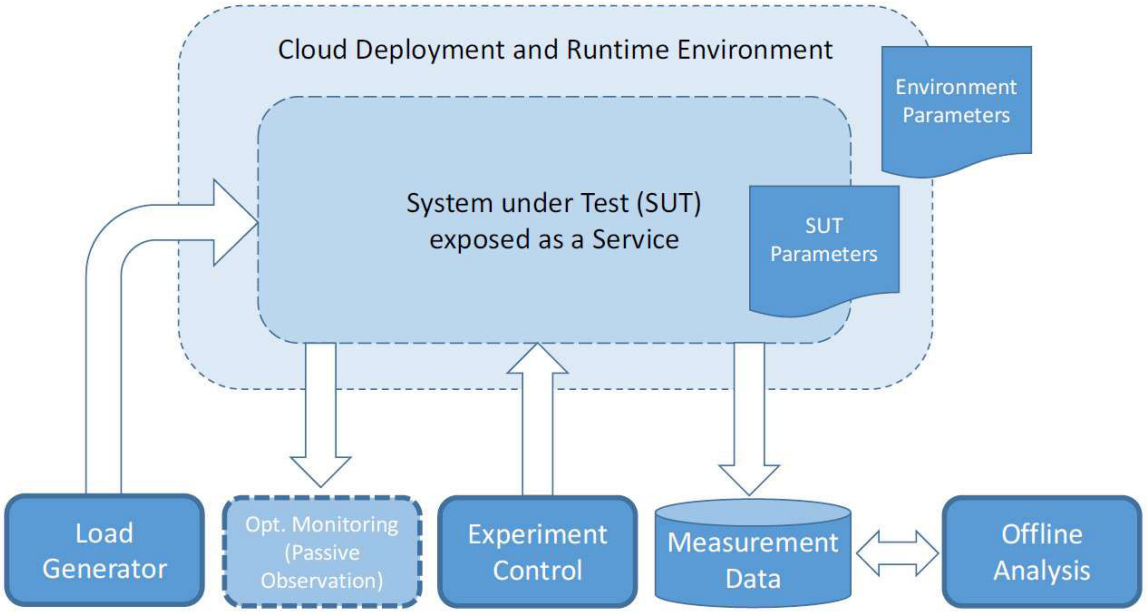
Monitoring vs Benchmarking

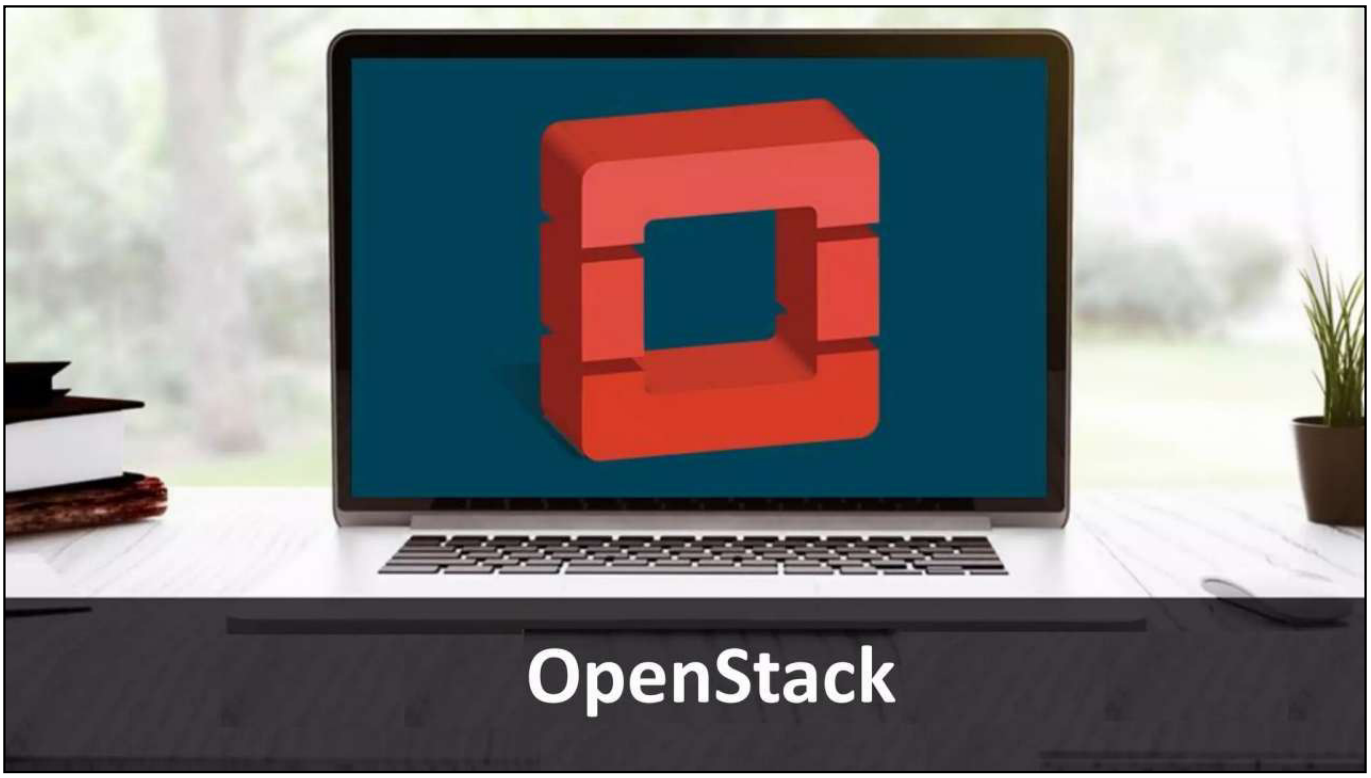
Category	Benchmarking	Monitoring
Motivation	Answer a specific question	Detect undesirable system states
System under test (SUT)	Test deployment	Production system
Point in time	Repeated short test runs	Permanently running process
Level of control	Complete control	Passive observation, no control
Stress on system	Artificially created	Actual production load
Observable metrics	Arbitrary	OS level and custom, discrete events
Impact on SUT	Strong	Negligible
Visibility delay	Often offline analysis	Near real-time

Example: Monitoring vs benchmarking

- Netflix uses their Simian Army to inject arbitrary failures into production systems with the goal of detecting fault-tolerance issues – it is hard to decide whether this falls more into the category of monitoring (production system, continuous) or benchmarking (intrusive, follows a specific goal).

Standard components of benchmarking

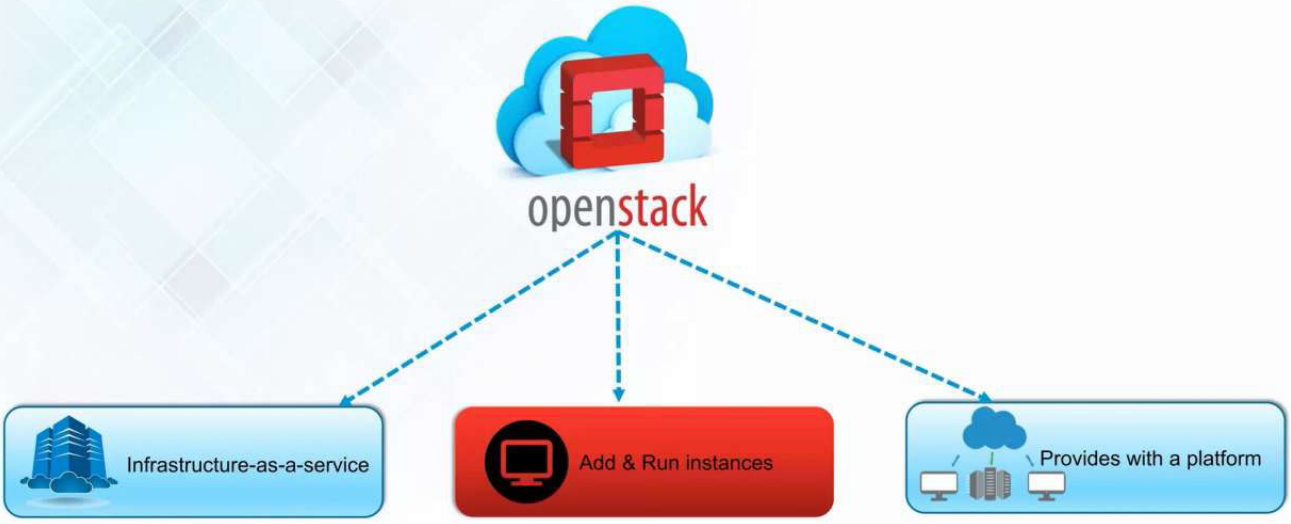




What is OpenStack?

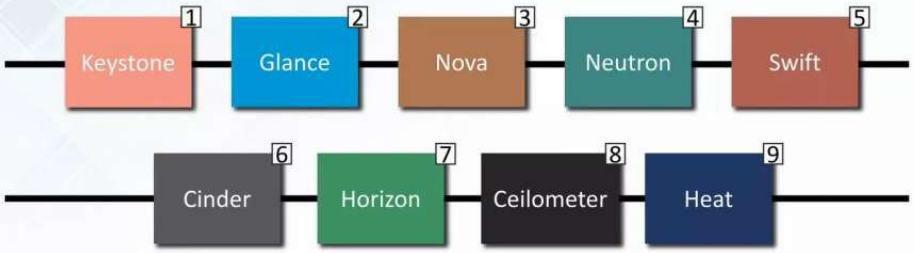
- OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds.
- OpenStack is managed by the OpenStack Foundation, a non-profit that oversees both development and community-building around the project.
- OpenStack is the future of cloud computing backed by some of the biggest companies.

Figure: *Companies contributing to OpenStack*



What is Openstack?

"OpenStack is a collection of open source technologies delivering a massively scalable cloud operating system" - openstack.org



What is OpenStack ?

In July 2010, Rackspace Hosting and NASA jointly launched

OpenStack was launched to help organizations with cloud-computing services

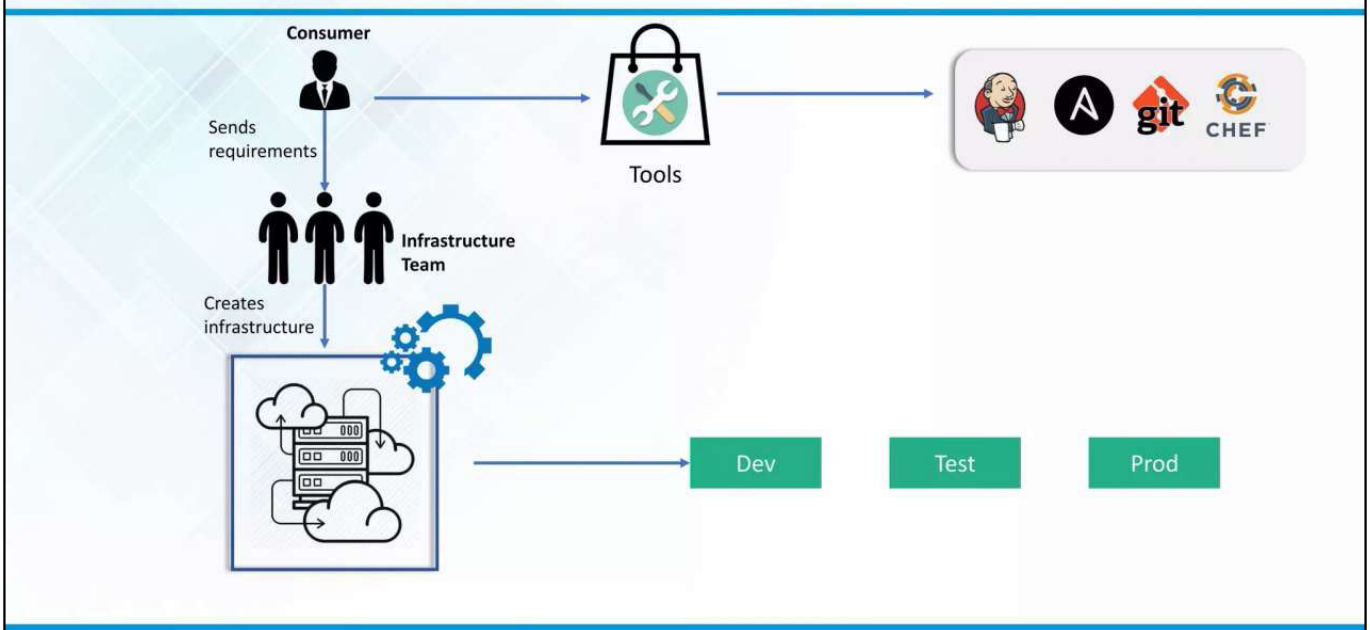
Non-profit and Corporate funded entity

Open-source and built by community of developers

More than 500 companies have joined this project

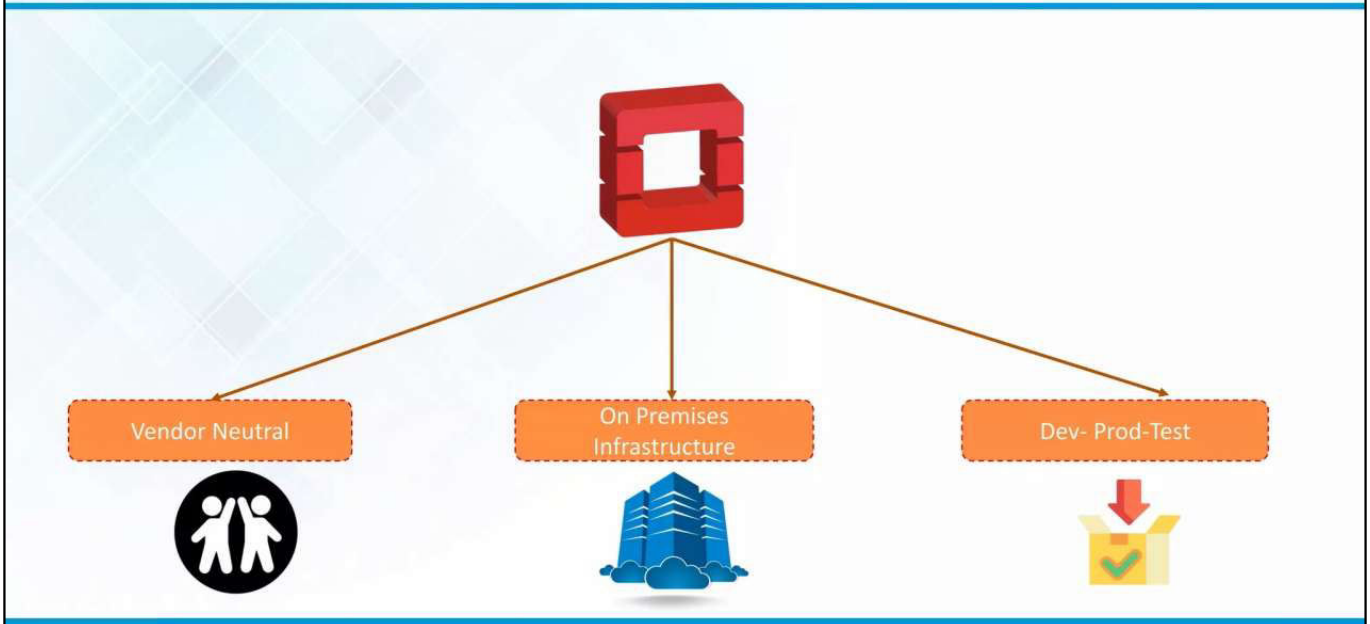
One of the fastest growing Open source communities

Why OpenStack?

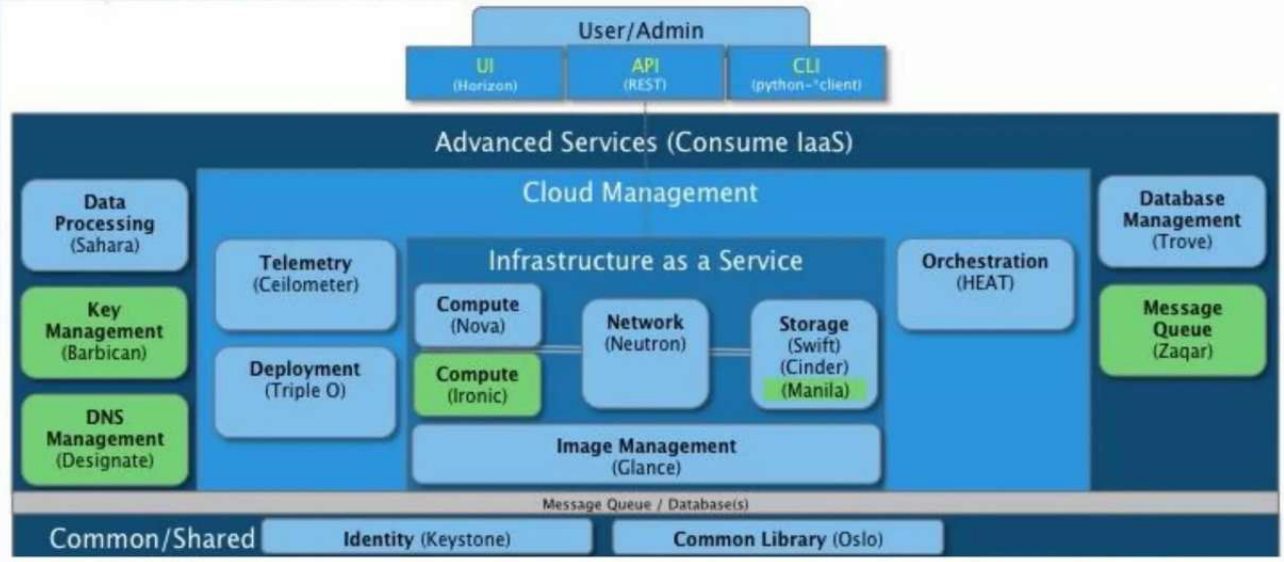


Openstack vs Public cloud

Benefits with OpenStack

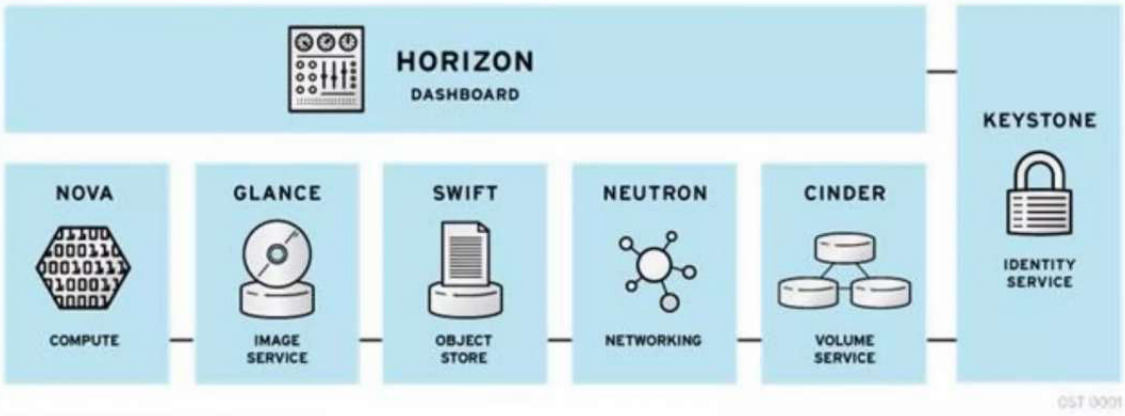


Components of OpenStack



Source : EMC Corporation

Components of OpenStack



Who's Behind OpenStack ?

Platinum Members

at&t, Hewlett Packard Enterprise, Hewlett Packard Enterprise, intel, IBM, rackspace (the #1 managed cloud company), redhat, SUSE

Golden Members

EMC², YAHOO!, CISCO, DELL, aptira, FUJITSU, DreamHost, ERICSSON, EasyStack, HUAWEI, inwinSTACK, NEC, NetApp, MIRANTIS, HITACHI (Inspire the Next), JUNIPER NETWORKS, Virtuozzo, Symantec, UnitedStack

Deployment Models

1. On-premises distribution
2. OpenStack-based Public Cloud
3. Hosted OpenStack Private Cloud
4. OpenStack-as-a-Service
5. Appliance based OpenStack

openstack.

On-premises distribution

- In this model, a customer downloads and installs an OpenStack distribution within their internal network

OpenStack-based Public Cloud

- A vendor provides a public cloud computing system based on the OpenStack project

Hosted OpenStack Private Cloud

- A vendor hosts an OpenStack-based private cloud including the underlying hardware and the OpenStack software

OpenStack-as-a-Service

- A vendor hosts OpenStack management software as a service and customers use the software service

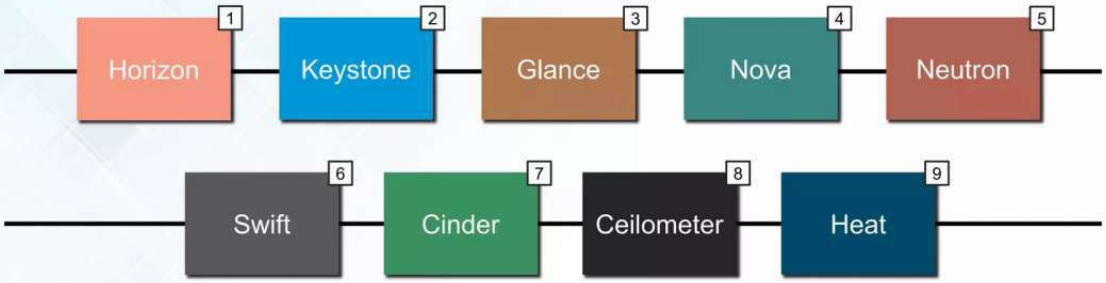
Appliance based OpenStack

- Nebula was a vendor that sold appliances that could be plugged into a network which spawned an OpenStack deployment.

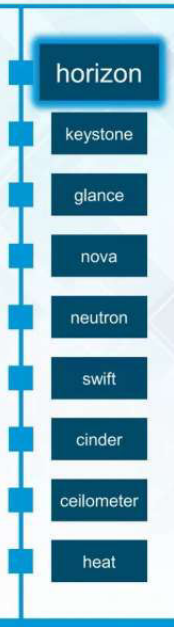
OpenStack Components

OpenStack Components

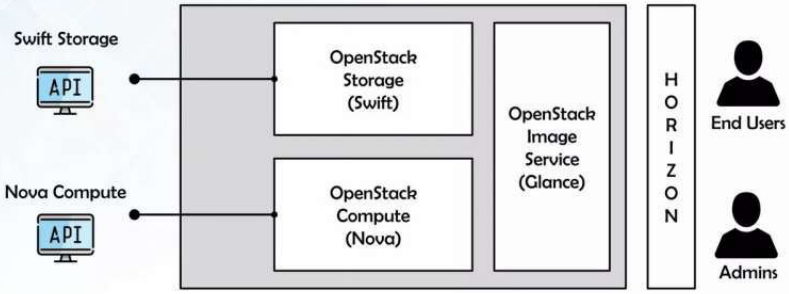
The following gives a flow of all the components in OpenStack that we will discuss in the coming slides:



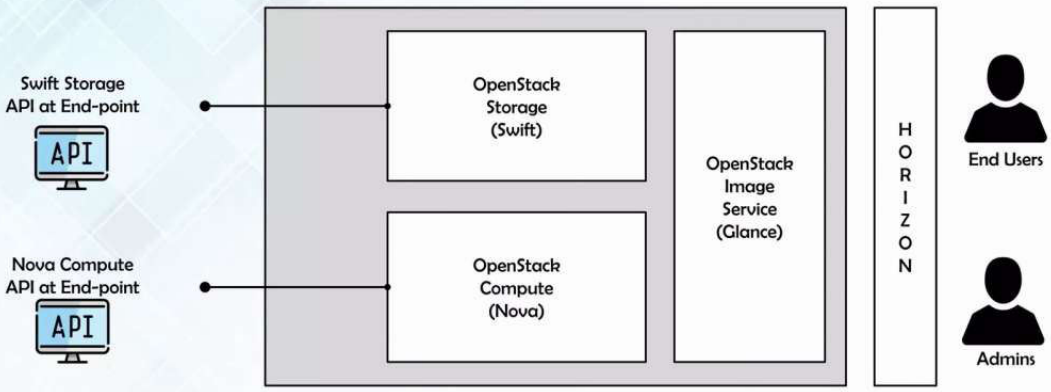
Horizon



- Horizon is the OpenStack's Dashboard which provides a web based user interface to OpenStack services including Nova, Swift, Keystone, etc.
- It allows users and administrators of the environment to interact with and manage the various functional components without having to install any local client tools other than a web browser.



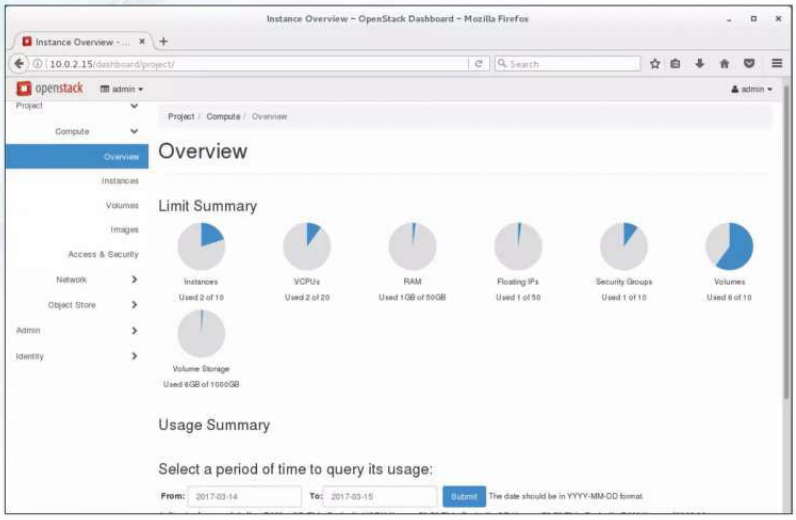
Horizon Architecture



Horizon

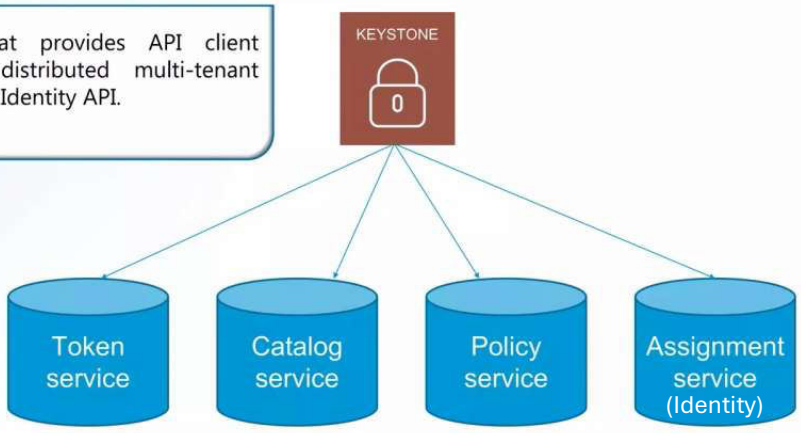
- horizon
- keystone
- glance
- nova
- neutron
- swift
- cinder
- ceilometer
- heat

Horizon provides the Dashboard for OpenStack



Keystone

- Keystone is an OpenStack service that provides API client authentication, service discovery and distributed multi-tenant authorization by implementing OpenStack's Identity API.

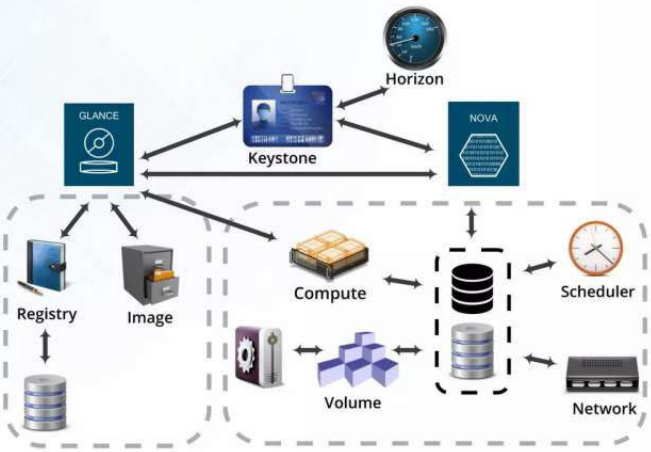


Keystone

- horizon
- keystone
- glance
- nova
- neutron
- swift
- cinder
- ceilometer
- heat

❑ Keystone is an OpenStack service that provides API client authentication, service discovery and distributed multi-tenant authorization by implementing OpenStack's Identity API.

KEYSTONE



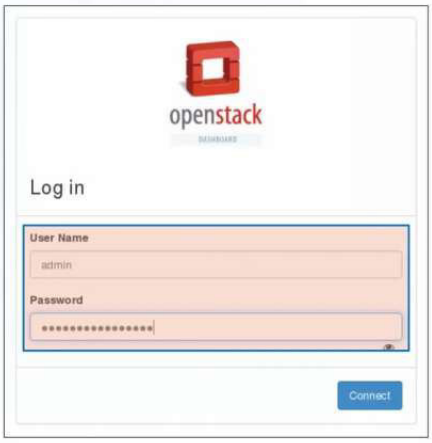
Keystone

- horizon
- keystone
- glance
- nova
- neutron
- swift
- cinder
- ceilometer
- heat

Keystone is used for authentication into OpenStack Horizon

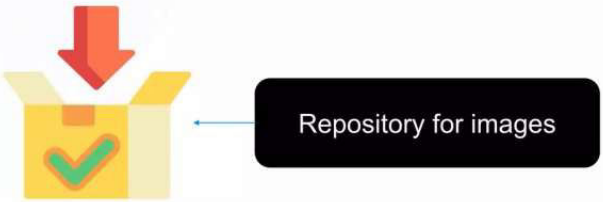
Keystone currently supports token-based and user-service authorization

KEYSTONE



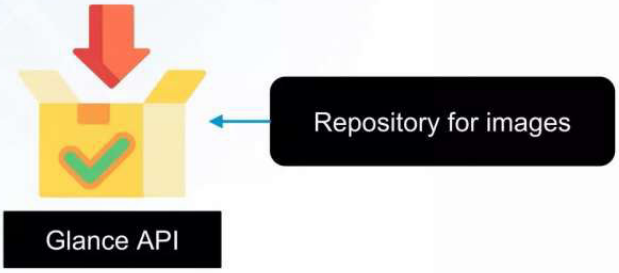
Glance

- The Glance project provides a service where users can upload and discover data assets that are meant to be used with other services.
- This currently includes images and metadata definitions.
- Glance image services include discovering, registering and retrieving virtual machine images.



Glance

- The Glance project provides a service where users can upload and discover data assets that are meant to be used with other services.
- This currently includes images and metadata definitions.

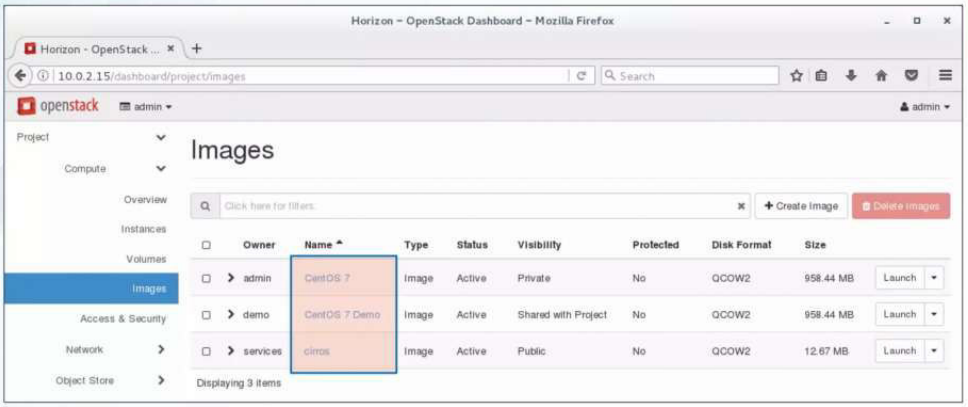


Glance

- horizon
- keystone
- **glance**
- nova
- neutron
- swift
- cinder
- ceilometer
- heat

Images in Horizon are managed by the Glance project

GLANCE

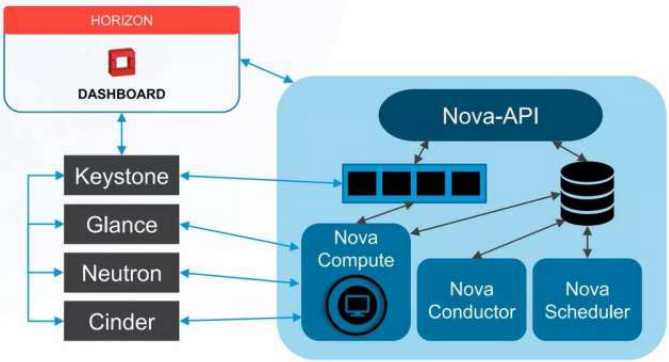


Nova

- horizon
- keystone
- glance
- **nova**
- neutron
- swift
- cinder
- ceilometer
- heat

- Nova is an OpenStack project designed to provide power massively scalable, on demand, self service access to compute resources.
- It is fault tolerant, recoverable and provides API-compatibility with systems like Amazon EC2.

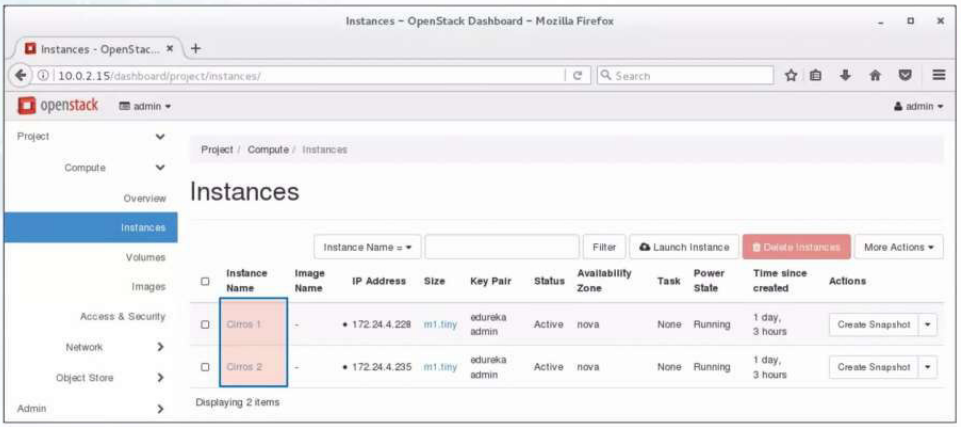
NOVA



Nova

horizon
keystone
glance
nova
neutron
swift
cinder
ceilometer
heat

Instances in Horizon are handled by the Nova project



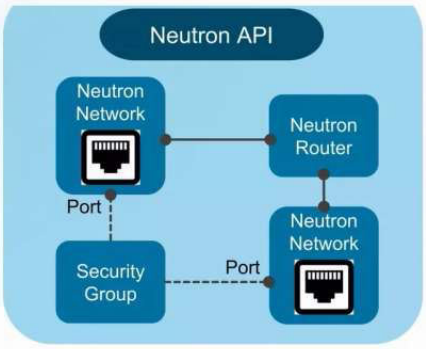
NOVA

Neutron

horizon
keystone
glance
nova
neutron
swift
cinder
ceilometer
heat

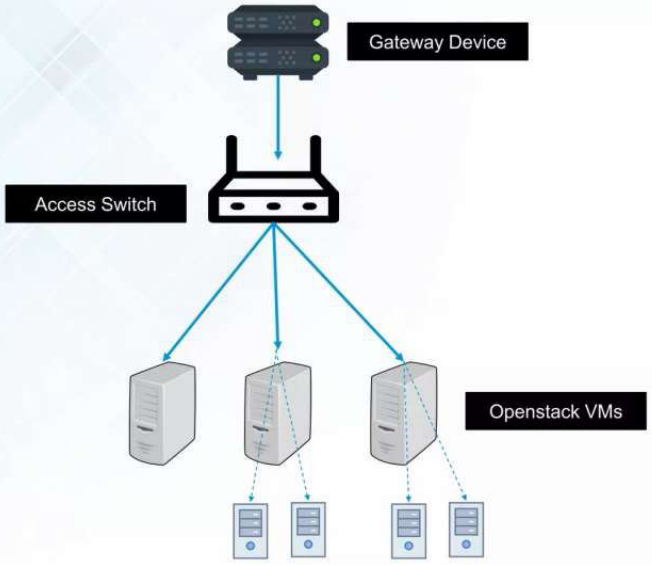
- Neutron is a networking project focused on delivering Networking-as-a-Service (NaaS) in virtual compute environments.
- Neutron has replaced the original networking application program interface (API) called Quantum in OpenStack.

  Neutron is designed to address deficiencies in "baked-in" networking technology found in cloud environments.



NEUTRON

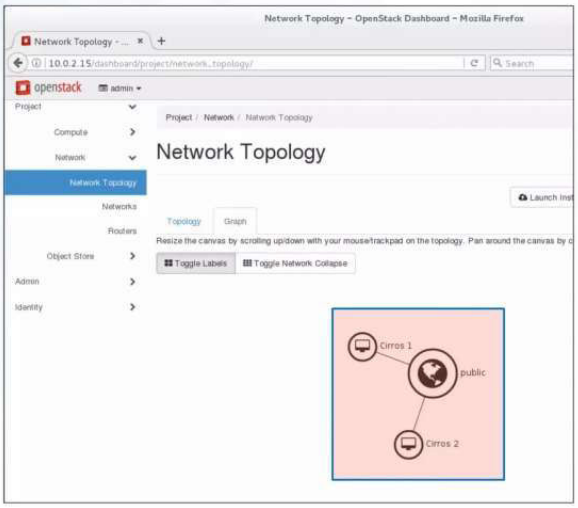
Neutron architecture



Neutron

- horizon
- keystone
- glance
- nova
- **neutron**
- swift
- cinder
- ceilometer
- heat

Network service in Horizon is managed by the Neutron project

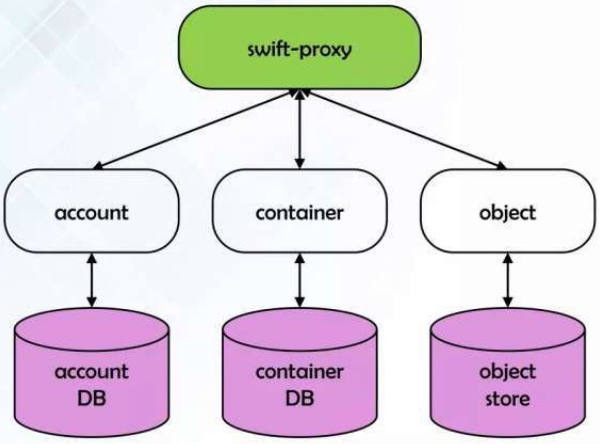


NEUTRON

Swift

- Swift is a highly available, distributed and consistent object store.
- Organizations can use Swift to store lots of data efficiently, safely and cheaply.
- Swift powers storage clouds at Comcast, Time Warner, Globo and Wikipedia as well as public clouds like Rackspace, NTT, OVH and IBM SoftLayer.

Swift Architecture

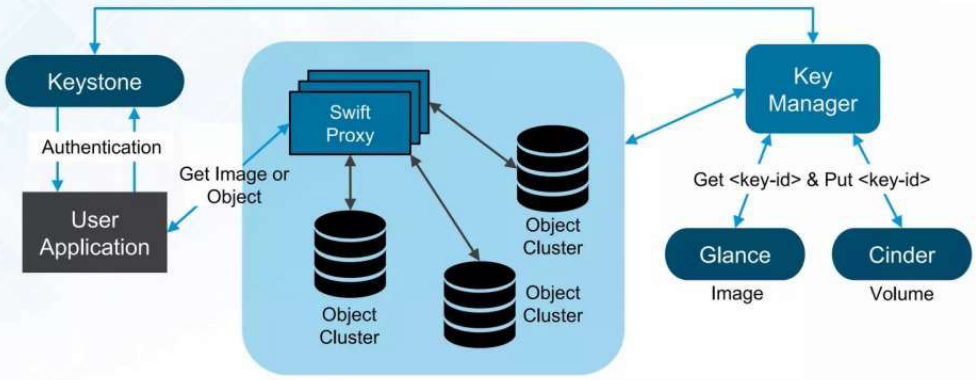


Swift

horizon
keystone
glance
nova
neutron
swift
cinder
ceilometer
heat

- Swift is a highly available, distributed and consistent object store.
- Swift powers storage clouds at Comcast, Time Warner, Globo and Wikipedia as well as public clouds like Rackspace, NTT, OVH and IBM SoftLayer.

SWIFT



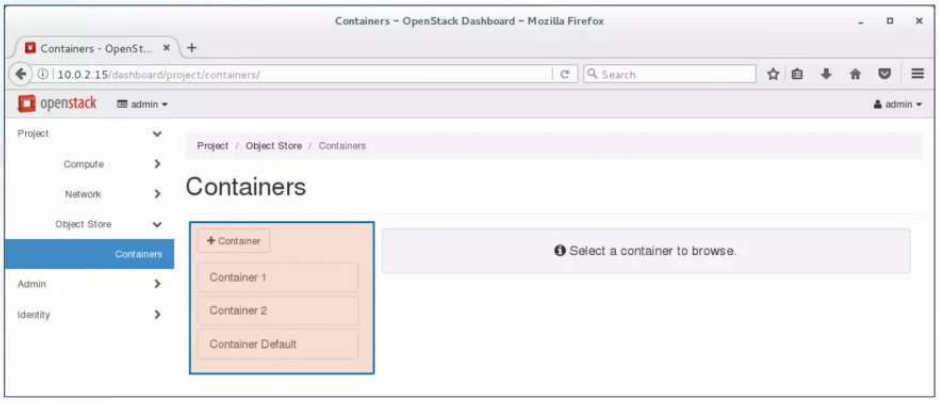
Swift

horizon
keystone
glance
nova
neutron
swift
cinder
ceilometer
heat

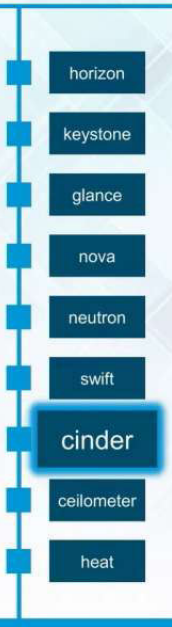
Object Store and Containers are managed by the Swift project

Organizations can use Swift to store lots of data efficiently, safely and cheaply.

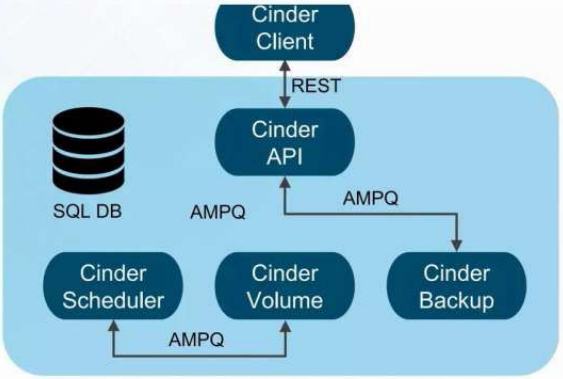
SWIFT



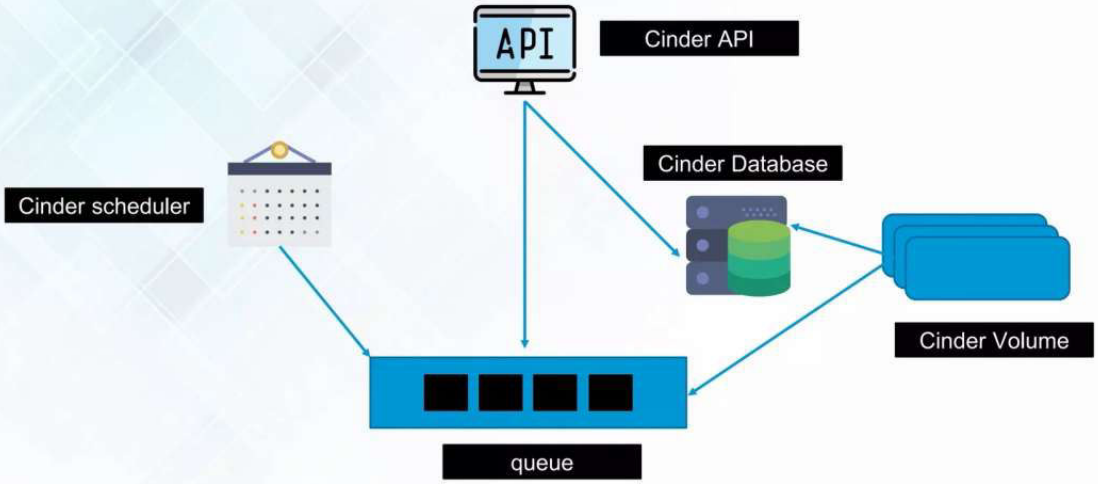
Cinder



- Cinder is OpenStack's Block Storage Service and provides persistent block storage resources that OpenStack Compute instances can consume.
- This includes secondary attached storage similar to the Amazon Elastic Block Storage (EBS) offering.
- In addition, you can write images to a Block Storage device to use as a bootable persistent instance.



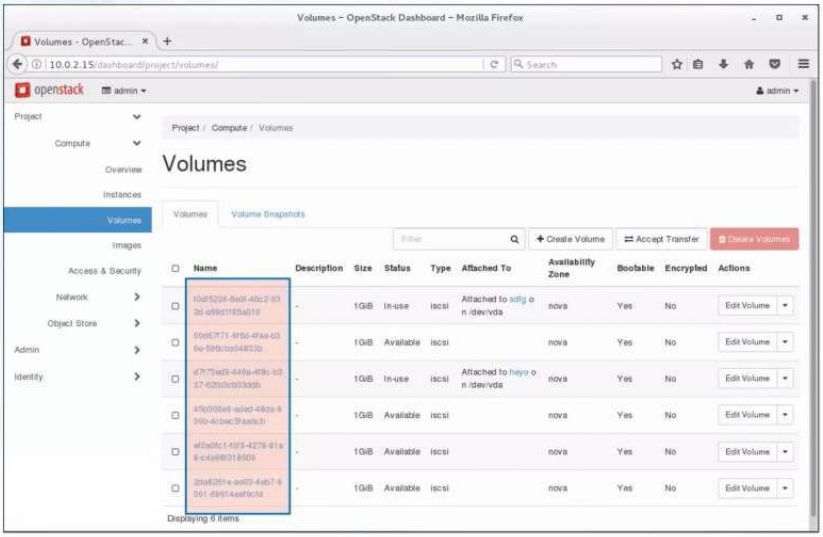
Cinder Architecture



Cinder

- horizon
- keystone
- glance
- nova
- neutron
- swift
- **cinder**
- ceilometer
- heat

Volumes in Horizon are managed by the Cinder project



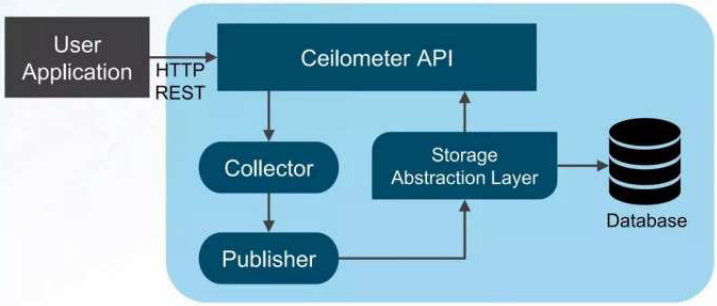
CINDER

Ceilometer

- horizon
- keystone
- glance
- nova
- neutron
- swift
- cinder
- **ceilometer**
- heat

- Ceilometer is a data collection service that provides the ability to normalise and transform data across all current OpenStack core components with work underway to support future OpenStack components.
- Its data can be used to provide customer billing, resource tracking and alarming capabilities across all OpenStack core components.

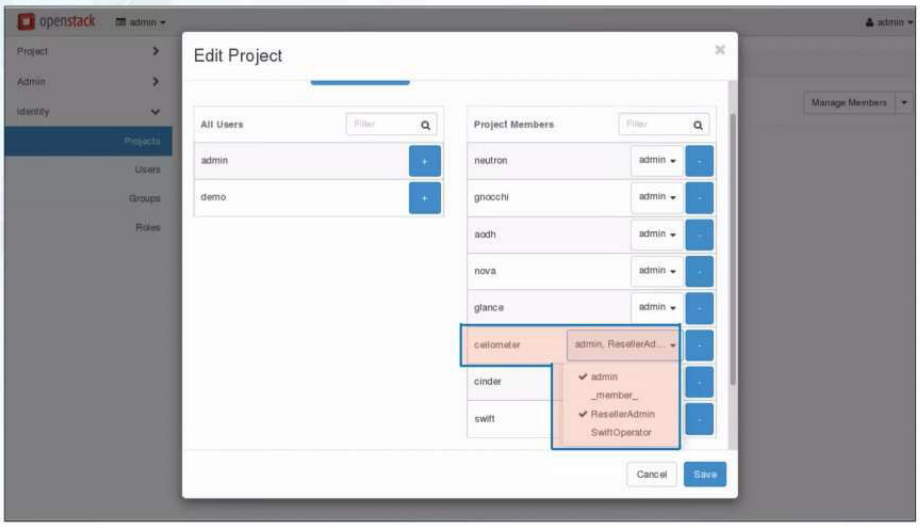
CEILOMETER



Ceilometer

horizon
keystone
glance
nova
neutron
swift
cinder
ceilometer
heat

Ceilometer in Horizon handles normalization and transformation



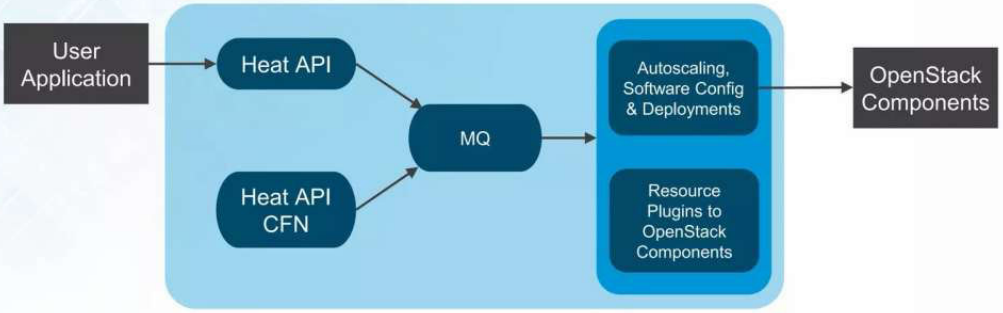
CEILOMETER

Heat

horizon
keystone
glance
nova
neutron
swift
cinder
ceilometer
heat

- Heat is the main project in the OpenStack Orchestration program.
- It implements an orchestration engine to launch multiple composite cloud applications based on templates in the form of text files that can be treated like code.

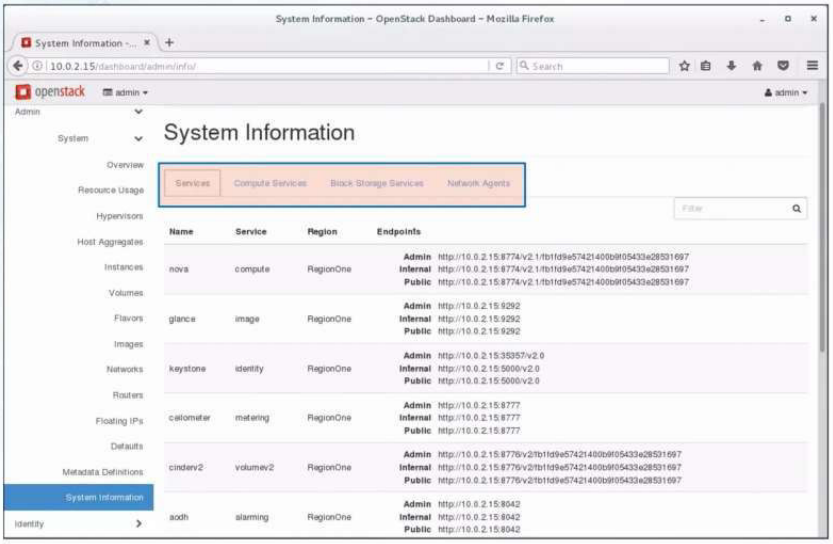
HEAT



Heat

horizon
keystone
glance
nova
neutron
swift
cinder
ceilometer
heat

Heat in Horizon implements orchestration engine



Openstack Architecture

OpenStack Architecture

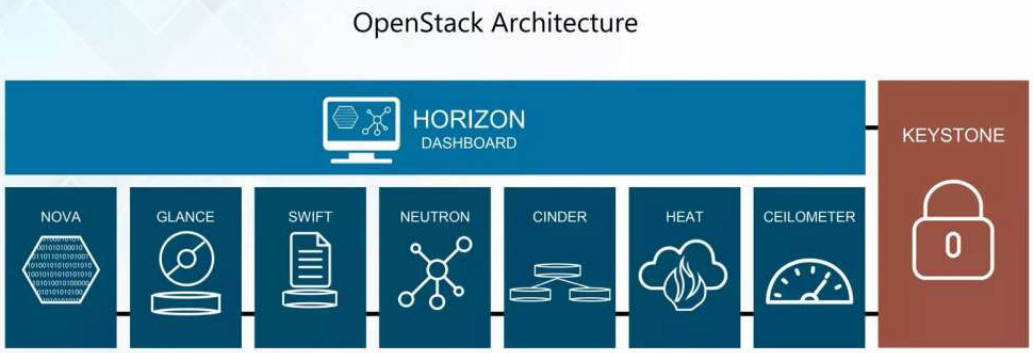
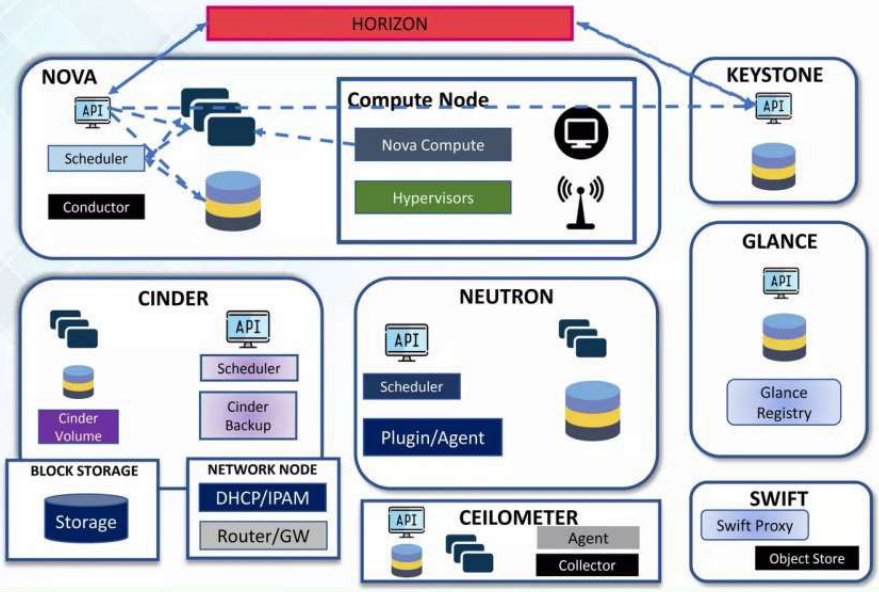
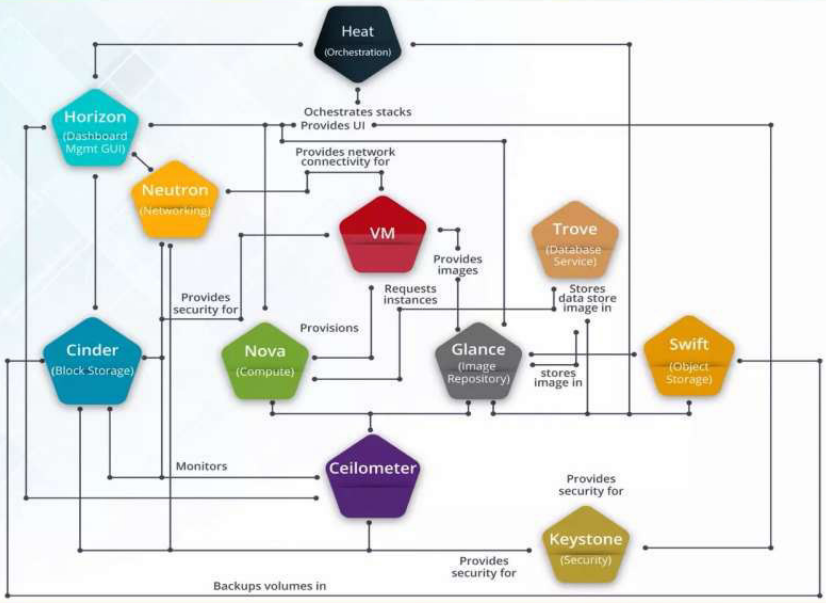


Figure: *OpenStack Architecture*

OpenStack Architecture:



Openstack Cloud Architecture