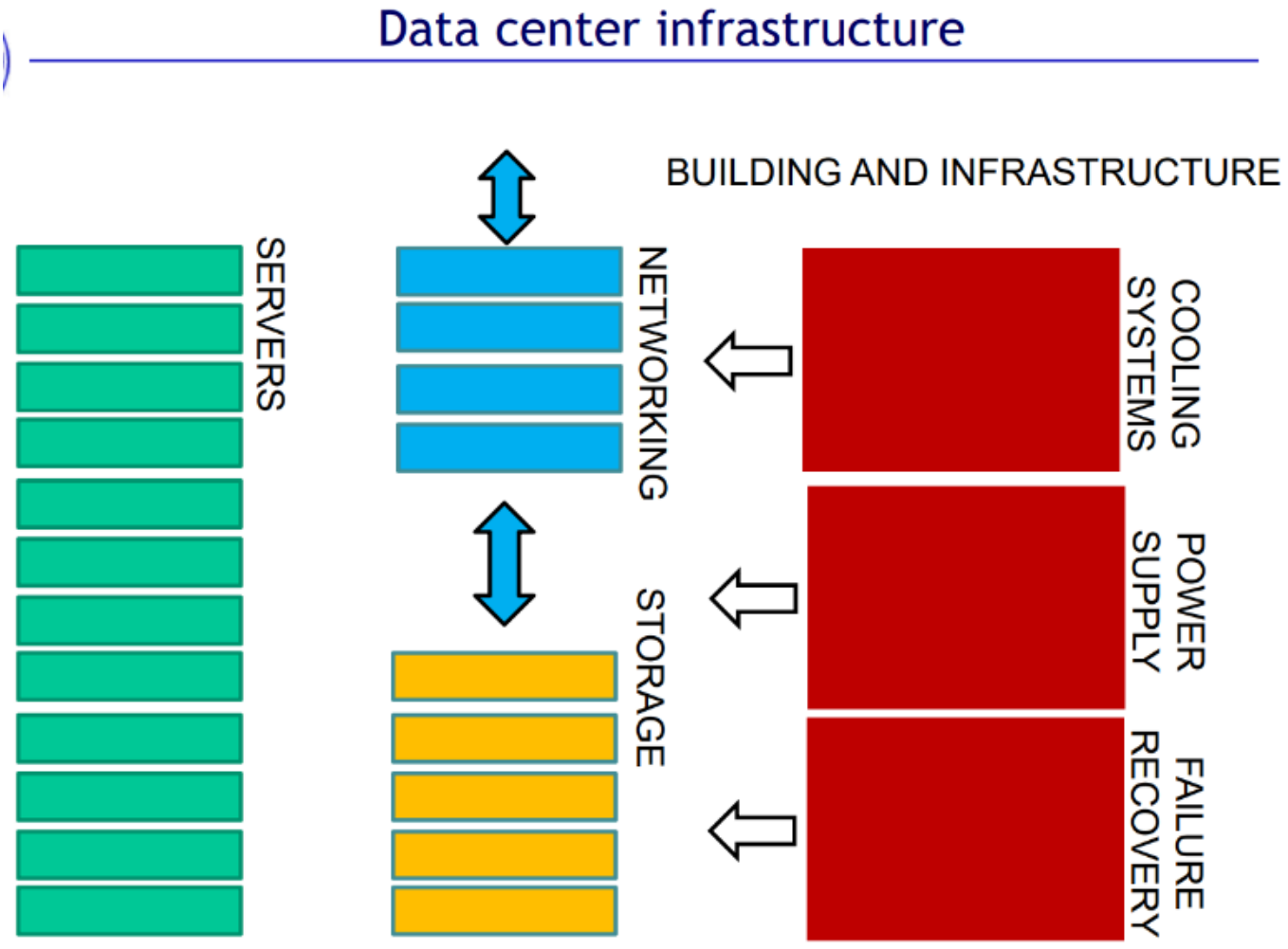
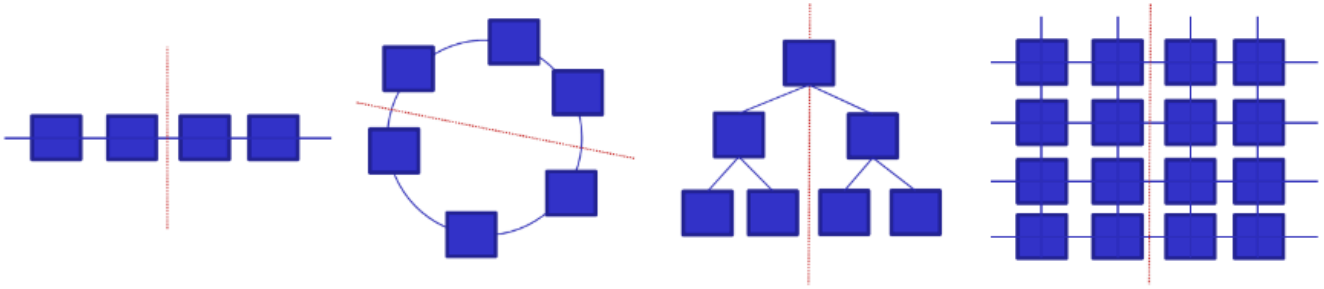
Networking

Various way to implement networking in data center: Switch-centric architectures, Server-centric and hybrid architectures.
At the start when computers become usable, everything was done inside the machine except some periferical devices. Minimal network demands, proprietary devices.
First network technology was applied to a LAN then, when it was needed to exchange data in long distance, it shifted to TCP/IP+Proprietary protocols.
The next step was the introducing of Web Applications and access from anywhere. Servers are broken in multiple units.
Then we passed to microservices where infrastructure moved to cloud providers, increases in server-to-server traffic, Datacenter are born.



In the networking part you have to increase the interfaces and the capacity behind the interfaces to utilise everything at the maximum efficiency.
All the traffic in the front panel has to be transferred somewhere(You have to upgrade the front and the back together).
BIsection Bandwidth: bandwidth across the narrowest line that equally divides the cluster into two parts. It characterizes network capacity since randomly communicating processors must send data across the middle of the network.



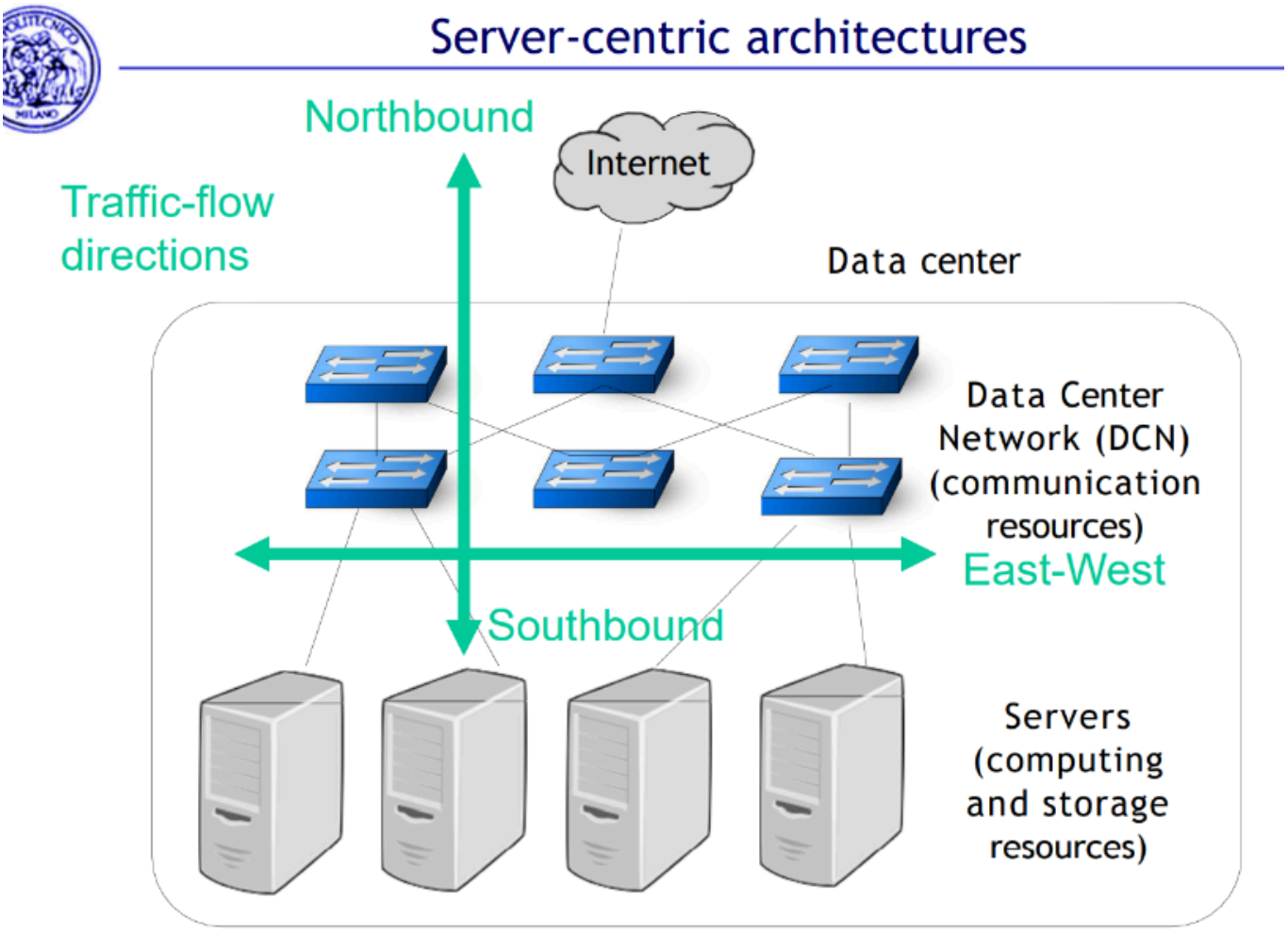
If we assume that every server needs to talk to every other server, we need to double not just leaf bandwidth but *bisection bandwidth*

Classes of DCN(Data Center Network)

DCNs can be classified into three main categories:

- Switch-centric architectures: Uses switches to perform packet forwarding
- Server-centric architecture: Uses servers with multiple Network Interface Cards (NICs) to act as switches in addition to performing other computational functions
- Hybrid architectures: Combine switches and servers for packet forwarding
  Creates their traffic when communicating but they will also handle the communication between the elements in the network. You can also virtualize switches having the capability of a switch but it is implemented in a specific CPU inside the network interface

Switch centric architecture



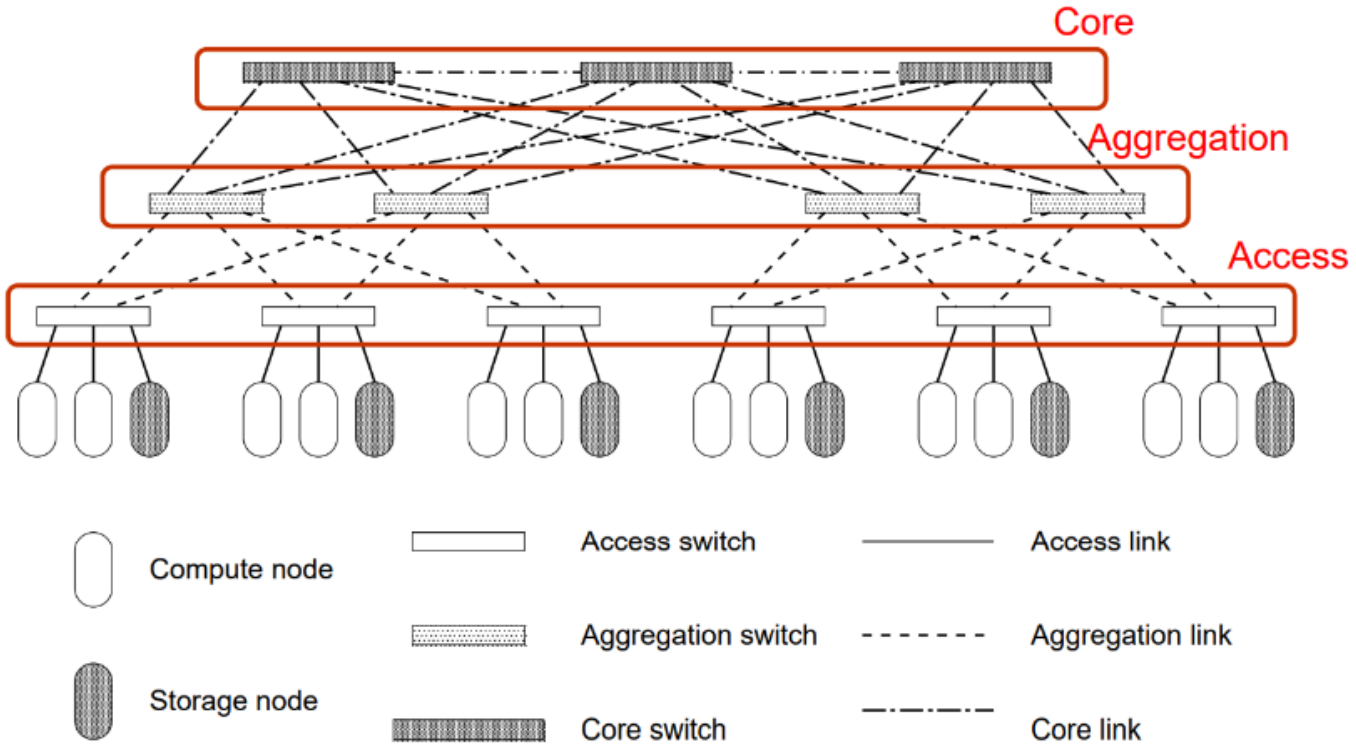
Two traffic flows direction:

- Northbound-Southbound: communication between server and internet
- East-West: traffic consistent, communication between data center. it also include all the network function virtualization. It can be of different type: unicast(point to point communication, VM migration, data backup, stream data processing), multicast(one-to-many communication, software update, data replication (≥ 3 copies per content) for reliability, OS image provision for VM), incast(many-to-one communication, reduce phase in MapReduce, merging tables in database)

Classic three tier architecture

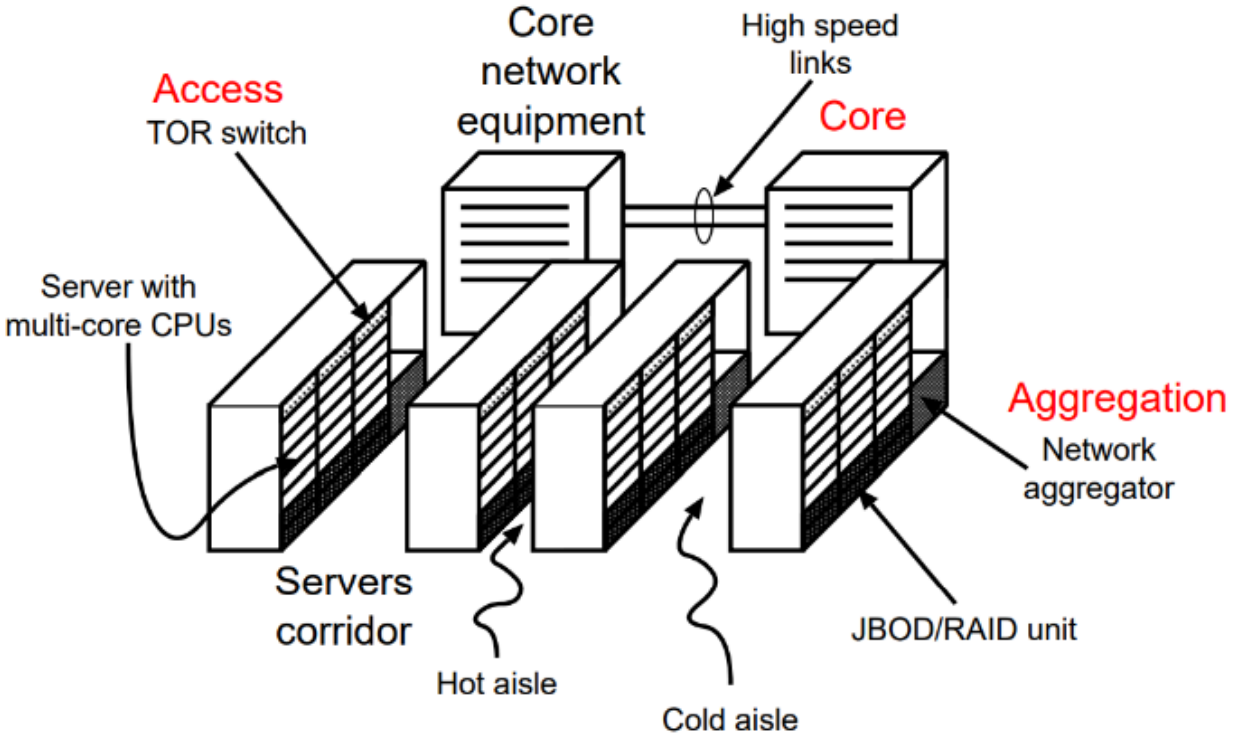
Three-Tier (or layer) "Classical" Network

- Three layer architecture configures the network in three different layers:

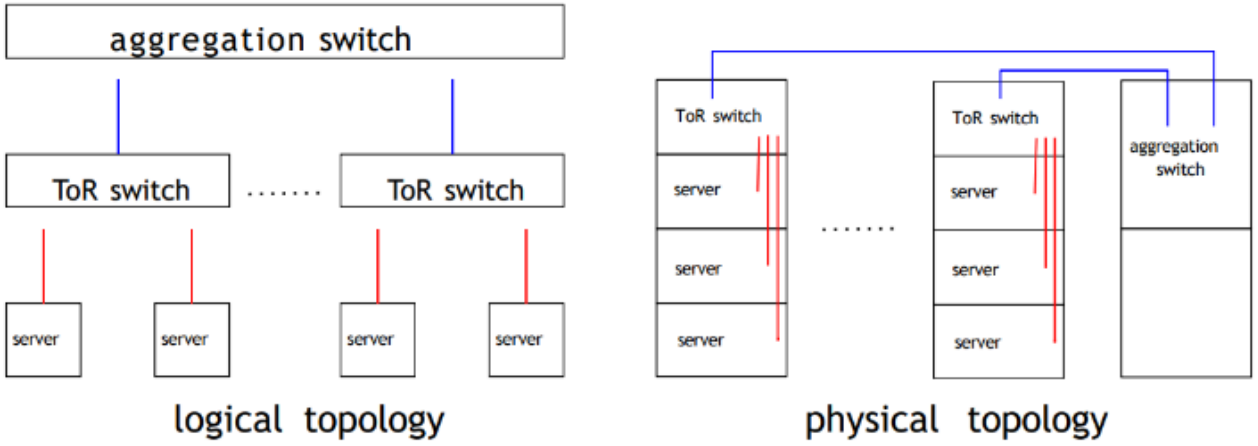


When you go to the top you start aggregating traffic so you need more capacity.
This architecture handles link between layers and between the same layer.
Each switches of a level is connected to multiple switches of the next layer.

Three layer architecture reflects the topology of the data center:



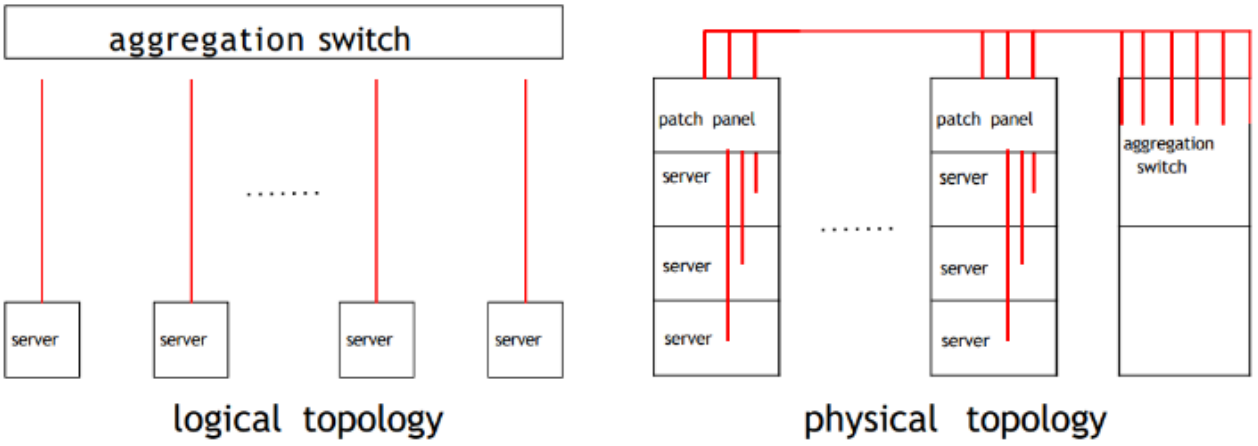
- ToR (Top-of-Rack) architecture
  - in a rack, all servers are connected to a ToR access switch
  - the servers and the ToR switch are colocated in the same rack
  - aggregation switches are in dedicated racks or in shared racks with othe ToR switches and servers
  - The number of cables is limited → simpler cabling. The number of ports per switch is also limited (lower costs)
  - Limited scalability, higher complexity for switch management (high number of switches)



Structure frequently used as the interconnection used is one for every racks and one for every switch. One alternative is to skip one layer to save on switches

EoR (End-of-Row) architecture

- Aggregation Switches are positioned one per corridor, at the end of a line of rack.
- servers in a racks are connected directly to the aggregation switch in another rack
- Aggregation switches must have a larger number of ports,
- more complex cabling, longer cables are required (higher costs)
- patch panel to connect the servers to the aggregation switch
- simpler switch management (lower number of switches)

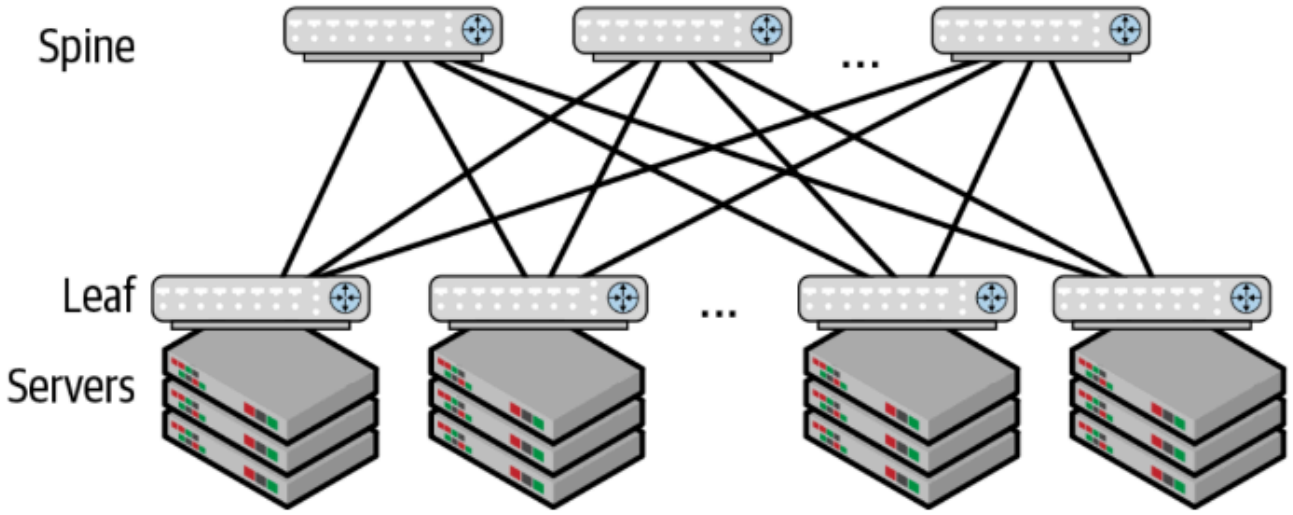


You need different type of switches for every layer, the cost of this architecture is very high as it requires very efficient switches. This is a problem as cost in term of acquisition, management, spare part stocks and energy consumption.

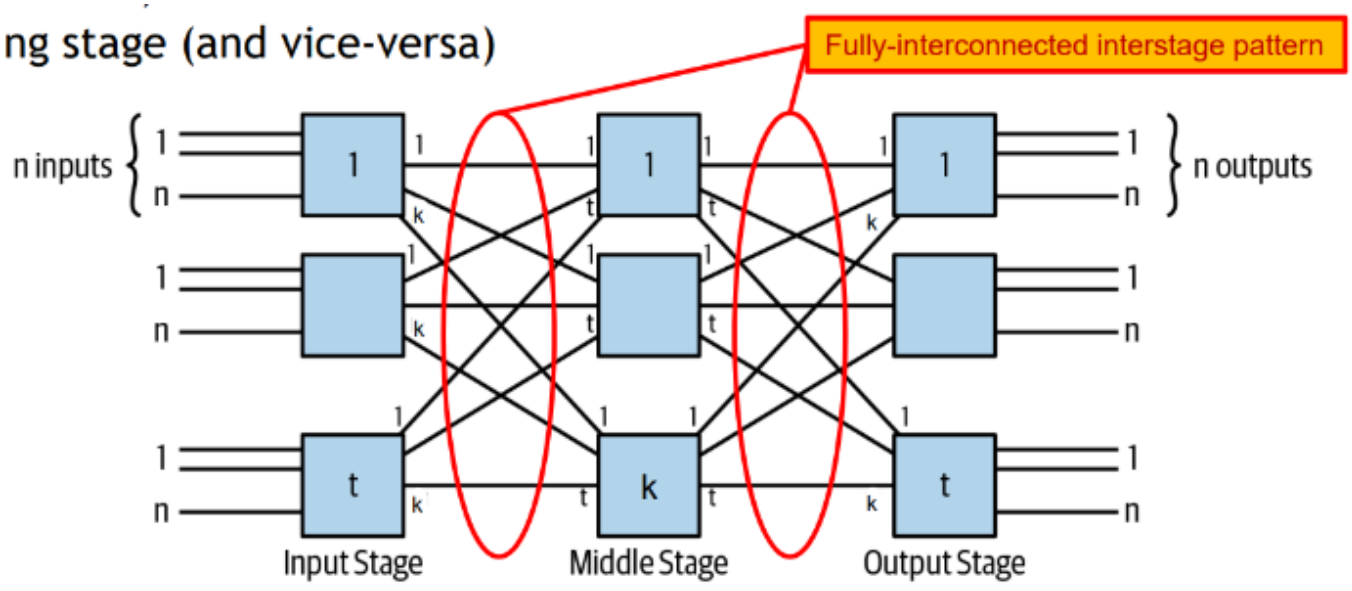
Leaf Spine Architecture

Leaf and Spine layer:

- Leaf layer is the equivalent of the Top of Rack layer
- Spine layer is the equivalent of the Core and Aggregation layer



No interconnection between the switches of the same layer to simplify the paths of the data.



The set of links is determined by the number of matrixs in the stage.
Let k be the number of middle stage switches(ensure that you have the bisectional bandwidth related to the number of server connected)
Let n be the number of input and outputs of switches of side stages(exploited by Leaf and Spine architecture)
If k ≥ n there is always a way to rearrange communications to free a path between any pair of

idle input/output

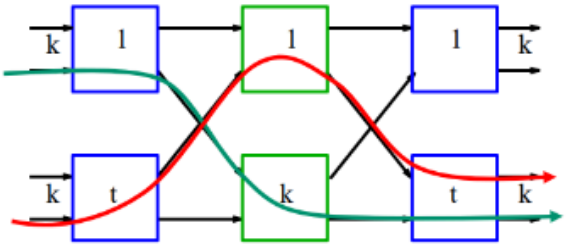
If k ≥ 2n - 1 there is always a free path between any pair of idle input/output

Notice that t is a free design parameter so the total number of input/output $N = t * n$ can scale freely (by increasing the size of middle-stage switches). But a DCN is a PACKET-SWITCHED network!!

If you always find a free path the architecture is called Freeblocking, if you find a path but it is used you can rearrange the connection and this architecture is called Moving Block architecture.

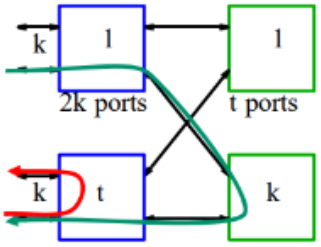
Clos topology (n = m = k)

- each switching module is unidirectional
  - k input + k output ports per module
- k matrices in the central stage
- t matrices in the side stages
- each path traverses 3 modules



Leaf and spine topology

- each switching module is bidirectional
  - Leaf: t switching modules with 2k bidirectional ports per module
  - Spine: k switching modules with t bidirectional ports per module
- each path traverses either 1 or 3 modules



In the folded architecture there are only two stages as the last one is collapsed on the middle one.

The advantages is also we can use a single type of switch, control protocol, interfaces and so on. This resolve the problem seen in the 3 tier model. Advantages also in routing as we can use protocol using multiple routing cost path exploiting all the bisectional bandwidth of the network.

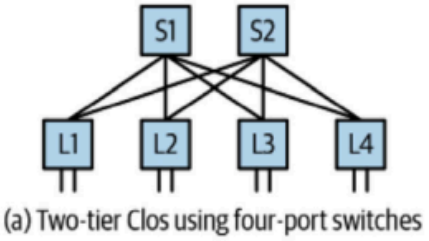
If we have a great number of servers we can reintroduce the 3-tier architecture but with some modification as using a Two tier Clos using four port switches. This model is a PoD based model aka the Fat Tree

An option: transform each spine-leaf group into a «pod» and add a super-spine tier

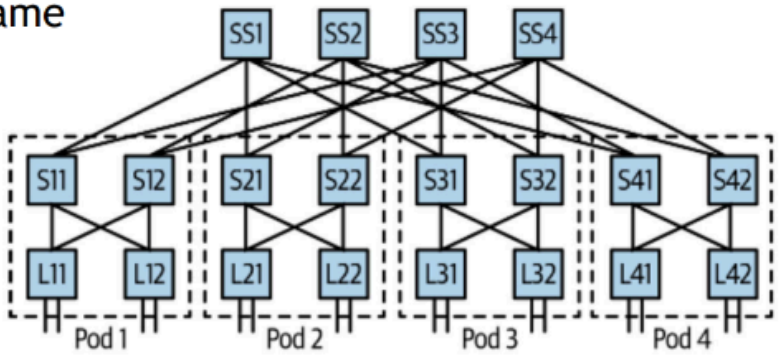
A highly scalable and cost-efficient DCN architecture that aims to maximize bisection bandwidth.

It can be built using commodity Gigabit Ethernet switches with the same number of ports.

Used by Microsoft, Amazon



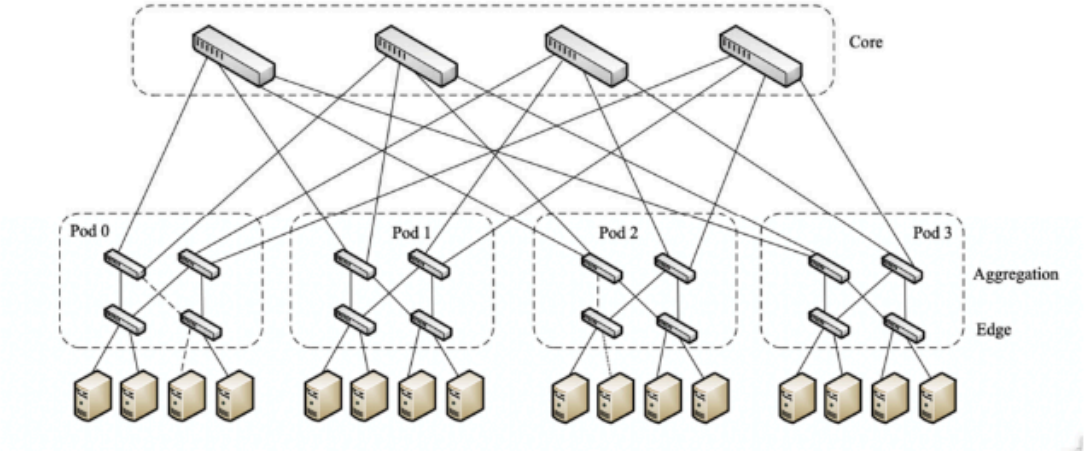
(a) Two-tier Clos using four-port switches



A PoD is a Point of Delivery, a module or group of network, compute, storage and application components that work together to deliver a network service. It is a repeatable pattern, its components increase the modularity, scalability and manageability of data. Full interconnectivity if you consider the pod, it is single connectivity if you consider the switch.

Fat Tree network

At the edge layer there are 2k PoDs each with k^2 servers. Each edge switch is directly connected to k servers in a pod and k aggregation switches. A fat tree network with 2k-port commodity switches can accomodate $2k^3$ servers in total. k^2 core switches with 2k-port each, each one connected to 2k pods. Each aggregation switch is connected to k core switches(Note the partial connectivity at switch level)



k = 2
4 pods
16 servers
20 switches

An example of a cost effective hierarchical fat tree based DCN architecture with high bisection bandwidth is VL2 Network. It uses three types of switches: intermediate, aggregation, and top-of-rack (ToR) switches. It uses $D_A/2$ intermediate switches D_I aggregation switches and

$D_A * D_I/2$ ToR switches. The number of servers in a VL2 network is $20(D_A * D_I)/2$. It uses a load-balancing technique called valiant load balancing (VLB).