ICT Infrastructure - Appunti

Francesco Lorenzoni

February 2024

Contents

Course info

Don't be shy to send multiple emails, prof. Cisternino receives many emails, and he known he can't reply to each one. He is okay to be contacted through teams using the symbol to "mention" him.

He designed the UniPi datacenters.

"Italy is more about the multiple micro businesses than the few existing industries"

Exam

The exam is **oral**.

Prof. Cisternino expects students to get the full picture, and understand key concepts, not to remember everything —which still wouldn't be bad ☺—.

Chapter 1

Introduction

Prof. Cisternino dropped a lot of measures in terms of Watts, Dollars, Gigabits and so on.

He mentioned with emphasis the problem of energy consumption. To give an idea, a single rack of a datacenter designed $\sim$ 10 years ago, absorbs up to $15kW$. The datacenter in *San Piero a Grado* is made up of 60 racks. It is not meant to provide the maximum energy possible for all racks simultaneously, but it still helps to get an idea of how things work in similar contexts.

1.1 Course map

1. Elements
    i. Datacenters
        (a) Power
        (b) Cooling
    ii. Cabling
    iii. Networking
    iv. Storage
    v. Compute
    vi. Virtualization
        (a) Hypervisor
        (b) Containers
2. Cloud
    i. Reference architecture
    ii. Resilience
    iii. Security
    iv. Legal aspects
        (a) GDPR
        (b) Security frameworks
    v. Procurement aspects
    vi. Operations
        i.e. Keep the system up and running while upgrading the system

Chapter 2

Datacenter

10 years ago datacenters were no more than a room with some computers, air conditioners and some plugs to power up the devices. Later on, customers started asking server vendors to include in the servers utilities to allow an *automated datacenter management*. Thus the trend moved towards **Software Defined Datacenter**, which currently is the only possible way to deploy a Datacenter.

An **Active Datacenter** allows for internet storage (?)

A Datacenter should be **future-proof**: servers may be replaced, but updating a whole datacenter is at least a 1-year project.

2.1 Structure

Racks are made of —$\sim$ 42— *units*.

Besides server theirselves, there is a **cooling system**. The first issue is the how to provide cool air. Then there is also how to define an evacuation plan, which must take into account dust.

However also the **floor** is not to be negliged.

- ◇ *Floating* floor or Ground floor
  - – *"A "floating floor" in a data center, also known as a "raised floor", is a type of construction used in data centers to create a void between the actual concrete floor and the floor tiles where the servers and other equipment are located12. This space is typically used for routing cables and for air circulation, which helps with cooling the equipment1."*[1]
- ◇ *Resistance* usually around $1\frac{ton}{m^2}$

For example, in San Piero A Grado, there was a power cabin receiveing current from three lines. Now the whole power management components are in a container outside the building placed close to the facility.

Cables are not super-resistant to current. A lot of current passing through a copper wire will *exhaust* both the wire and the components receiving such current; hence the current should also be balanced among different cables, to avoid exhausting some components before the others.

A UPS —first of all— stabilizes the output current.

In theory $1V * 1A = 1W$, but in reality, performing such conversion something gets lost, so we have

$$I * V * cos\phi = W$$

2.2 Power Management

Electric panels (aka *switchboards*) allow segmenting the power supply in the various zones of the datacenter.

PDUs stands for *Power Distribution Units*, and allow to distribute power for a server units in a rack. Typically, for each PDU there is another one, providing redundacy and thus resilience/robustness.

Definition 2.1 (PUE) *Power Usage Effectiveness **PUE** measures the efficiency of a Datacenter.*

$$PUE = \frac{Total\ energy}{ICT\ energy}$$

[1] ChatGPT 4.0 - Generated

The reason for improving Datacenter design is to lower the PUE; basically to save money, but also for "green-environment" concerns.

But *when* should PUE be measured?

The PUE in January is very different from the one in August, so generally it is calculated as the average of one year.

Note that a poorly designed datacenter placed in Siberia with $-20°$ may have a lower PUE than a datacenter in Italy, for instance.
In particulare geographical zones with high temperature variations over the year (e.g. in Italy the temperature variates from 40 to 50 Celsius degrees), are strongly unrecommended to build datacenters in. A counterintuitive example is the desert, where the temperature is very high during the day and very low during the night, but in general the temperature over the year is **stable**; allowing for defining physical processes exploiting such stability.
Also the oceans have a very stable temperature; not on the surface, but deep down it is very stable.

2.3 Cooling

Note that racks are always placed back-to-back, because the front requires cool air, and the back outputs hot air.

2.3.1 CRAC

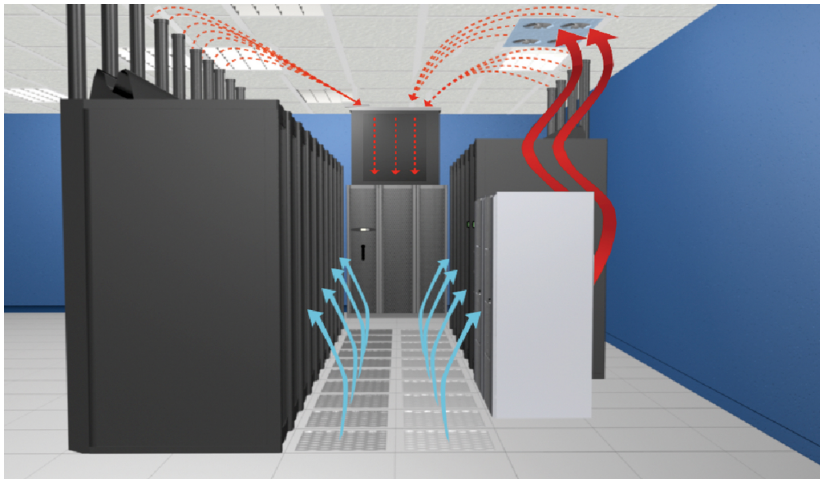

Figure 2.1: CRAC/CRAH cooling architecture

Chillers take hot air from above and push cool air in the bottom. Then air pushed under the floating floor wants to exit, and does so going through the grates placed in front of the racks. The racks suck the cool air in front and output hot air from the back.

There are two **drawbacks**:

1. It is difficult to confine and keep separated hot air and cool air. The mixup between the two leads to cooling inefficiency, thus energy and money waste
2. In case a rack has a workload heavier than others and thus requires more cooling air, the chiller must provide more cool air to all the racks in the same row; this makes this architecture particularly inefficient for datacenter which have heterogeneous workloads.

2.3.2 Inrow Cooling

The fan "towers" are called *inrow cooling*.

The first advantage is that it allows for heterogeneous cooling in the datacenter. Secondarily, a fan outputs hot air directly where another fan expects it to be. This allows to confine hot air and to avoid wasting energy in outputting air and sucking it.

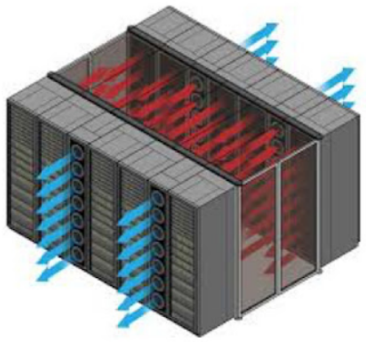

Figure 2.2: Inrow cooling architecture

2.3.3 Chilling water outside

Figure 2.2: Outside chillers

Outside of a datacenter there are chillers which cool down water which is then pumped into the datacenter, where it is used by CRAC/InRow chillers to cool down air.

It is important to ensure that the temperature does not heat up while travelling from the outside chillers to the datacenter, because it would mean wasting energy.

In SPG the outside chillers cool the water down to 18°, which seems high temperature, but in fact it is not: the datacenter is designed to work up to 26°.

The higher the "allowed" temperature is, the more is the energy saved.

Besides, **adiabatic** chillers —such as the ones in SPG— can use ***free cooling*** in case the outside temperature is lower than 18°[2], which basically exploits the lower outside temperature to *passively* chill water, without involving the compressor used in the standard cooling way.

Also **humidity** must be managed. An environment which is too dry leads to water condensation onto racks and plugs, possibly resulting in damage to devices and humans.

2.4 Redundancy for Resilience

Active-Passive means that aside from the active system, there is a mirrored one which is shut down waiting for failure and boots up *"just in case"*. This approach is usually not the ideal one, because the second system is very unlikely to be used and is costful. Besides, there are two critical issues with Active-Passive:

Cons

1. There is a non-negligible time interval where the switch from the active broken system and the passive one has to happen.
2. If when booted the backup system reveals itself to be flawed and not working, well... *very sad* ☹

[2] common case in winter and autumn

Active-active systems are usually better, because they also allow for load balancing. In case of SPG there are three cooling systems, and in case one breaks, the other two can keep working. Active-active costs even more, but it is the standard way to go.

2.5 Cooling CPU

High-end CPUs *heat up so much* that it has became *unreasonable* to cool them using air.

However, note that water conducts electricity, so a flaw in a waterpowered cooling system may lead to consistent damange and possible fires.

Oil instead doesn't conduct electricity, and there are some systems which are *submerged* in oil, but there are two drawbacks:

1. **Price**: oil is way more expensive than water
2. **Servicing**: it is impossible to maintain the system's hardware.

Distilled water is not conductive, but even not considering that distilling it is expensive, it is impossible to guarantee that is stays pure when travelling in pipes, chillers, and so on.

Most datacenters tend to have an hybrid approach to cooling, called **air-to-liquid**. The idea is simple: It is acceptable to use cool air to chill water, which is then used to chill the air by InRow coolers, which chills the liquid which chills the CPUs.

(Woa! We need a schema...)

This is *not* the most efficient approach.

A nice question would be, *"Canť we simply chill the liquid and send it directly onto the CPUs?"* **No** ☺.

- ◇ Required pressure is different
- ◇ Required temperatures do not match
- ◇ ... TODO

2.5.1 Spilling Pipes

Liquid cooling systems manufacturers allow customers to ensure that their pipes are not spilling by injecting in the pipes a known gas at a known pressure. The customer can measure the pressure when the product is shipped and check whether it is the expected one, and if not, send back the device.

Handling spills

Handling spills is an **open problem**. Theorically, the idea would be to check for pressure variations, but this is currently *impossible* to be done on each entrance of each rack. Too much actuaction and sensoring would be required.

Besides, in case a pipe is spilling, the operators must act *quickly*, before the water spills onto other racks and cause critical damage.

2.5.2 Chassis

Chassis are needed for various reasons:

- ◇ 2.4GHz is the frequency at which water in our cells resonates, and circuits generate electromagnetic fields, so it may be unsafe to directly expose humans to circuitry
- ◇ Act as Faraday cages
- ◇ TODO

Chapter 3

Cabling

Cabling is of paramount importance.

Prof. Cisternino learnt it "the hard way" when he performed the cabling of the first UniPi datacenter by himself

1. Maintenance
2. Cooling
    i. Cables may heat up
    ii. Cables may obstruct air flow
3. Determines which machines interact with each other (*fabric*)
4. Bandwidth
5. Not neglectable cost

3.1 Bandwidth

3.1.1 Storage implications

A standard datacenter has servers connected with 25Gbit links in both directories, summing up to 50Gbit total bandwidth. Current SSDs provide much more. 4 drives are enough to saturate a 100Gbit/s link.
We moved from a situation where the **bottleneck** were slow Hard Drives, to the current one where the bottleneck is the —network— **bandwidth**.
Recently the PCI 3.0, which lasted very long —providing $\sim$ 1000`Gbit/s`—, suddenly become unsufficient to handle the needed traffic.

Considering this, datacenters must be designed to allow *Terabytes* of data to be moved in east-west traffic.

*The **fabric** is the glue that makes the datacenter possible.*

Besides, a single server is *unable* to handle 10TBs of data and handling requests from 3000 users simultaneously. It is necessary to **distribute** the requests.

HDDs are still currently used for **cold storage**; CPUs will access data exclusively from SSDs, and sometimes the server is shipped with on board **full-flash storage**.
The difference in price between SSDs and HDDs becomes negligible since you pay for top CPU, top GPU, top RAM; furthermore, you can't waste —the high amount of— energy —consumed by such components— by waiting for a slow drive.

SSDs have a known write limit, but today, the usually last enough time: if you write the whole disk every day it will last for 5 years. Most-likely after five years you'd have to renew some components anyway, besides the failure is a predictable event.

3.1.2 Fabric and Cables

Optical

Electric current propagates at a speed $s = \sim 0.6c$. Hence **optical fiber** is —at least in theory?— faster.

Lasers are a coherent beam of equal fotons. It is possible to transfer energy through such fotons. Something resembling a laser is used for optical fibers.

Blu-Ray came out when scientists managed to create light using frequencies in the Blu area, which are the higher ones. Currently, the best and most expensive optical fibers exploit blu-lasers as source of light.

Note that with optical you always need 2 fibers, one sending and the other receiving. The two possible connectors are `SC` and `LC`. Sometimes the two ends of the cable are detachable so that the cables may be switched; this is useful because sometimes you may want to attach the TX cable on the RX plug and viceversa.

Figure 3.1: `SC` and `LC` connectors

Copper wires

In case of electricity there are many aspects to be considered. Interferences, cable diameter/size, length, and also the fact that if a `1` has been transmitted for some time, it takes longer to transmit a `0`, due to the *commutation* that must happen.

RJ45 is a standard physical interfaced for copper wires, which allows up to 1Gbit regularly. The `Cat 7` cables still use the RJ45 as connector and provide instead 10Gbit/s, but are very uncomfortable, they are so thick that they are difficult to bend.

It is estimated that there have been installed $70 \times 10^9 m$ of Ethernet cables, making them the most used.

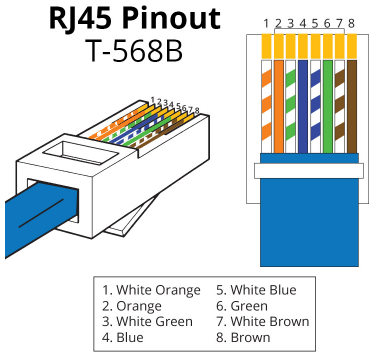

Figure 3.3: RJ45 - T568B

The aim of SFP is to decouple the optical transceivers from the server modules.

Is this correct?

They allow to go *optic-copper*, *copper-optic*, *optic-optic* and *copper-copper*.
SFP and GBIC (oldest one, now dead) pluggable modules acting as active transceivers for optical wiring using RJ45 connector.
A single cable having SFP ends costs about 100€. The cost ain't neglectable ☺.

Figure 3.4: SFP transceivers form factors

`SFP` ⟶ 1Gbit
`SFP+` ⟶ 10Gbit
`SFP28` ⟶ 25Gbit

Fun fact: ci sono 9 cavi USB-C e solo due portano informazioni video.

3.2 Cabling Issues

The key point is that it would be desirable for cabling to be reconfigurable.

There are things called *"Muffole"*, which are used for joining optical fiber cables, allowing for longer distances to be covered. They are designed to be underground.

Data traffic is always at least SFP+. Current standard is SFP28. Various SFP are typically compatible, the shape of the plug should stay the same. On switches there also some ports which are `QSFP+` or `QSFP28`, which allow up to `40` and `100Gbit/s` respecitively, and are used for north-south traffic.

The `Q` letter stands for *Quality*

Switches for datacenters should be **non-blocking**, meaning that no port has to wait for other ones —or any other thing— before transmitting, they can also transmit simultaneously.

In every datacenter it is *MANDATORY* to document the cabling.

Chapter 4

Networking

The two key aspects of a network are:

1. **Bandwidth** $\longrightarrow$ amount of data per second that can be moved through a specific connection
2. **Latency** $\longrightarrow$ is the amount of time required for trasmitting data, measured from the moment it is sent from the source to the one it is available to the source.

Latency —in a datacenter— to transmit data on the cable using *"pure ethernet"* is of the order of $0.5 \times 10^{-6} s(\mu s)$. If the TCP/IP stack is used (standard application case), latency is about $70 - 90 \mu s$.

Furthermore, current drives have reached speeds such that latency may act as bottleneck between them and the CPU.

Cable aggregation (e.g. aggregating 4 cables 10Gbit/s, providing 40Gbit/s total)can be performed only at a low —physical— level. Otherwise the TCP/IP stream will be associated to a single cable of the ones aggregated, resulting in less bandwidth.

4.1 Layers

Programmers usually do not care about anything under layer 3/4 traffic. However, in datacenters it is fundamental to understand how layer 2 works.

Also because in datacenters there are no routers doing the work for you; you are building the fabric in the first place.

Layer 2 is fundamental for 2 reasons:

1. East-west is Ethernet in the datacenter
2. All the dozens of protocols used in swithces are really used, so they are important.
3. MTU - Maximum Transmission Unit

4.1.1 Protocols inside switches

- ◇ LLDP Link Layer Discovery Protocol - Allows to reconstruct at least partially the functioning of the network.4
- ◇ DCBX Data Center Bridging Exchange - A meta-protocol so that two devices can agree on the configuration of a bunch of protocols, tipically related to storage/data
  e.g. "I need 50% percent of the bandwidth otherwise a can't work".
  It represents part of some kind of QoS for Ethernet
  .
- ◇ PFC Priority Flow Control
- ◇ ETS Enhanced Transmission Selection
- ◇ RSTP Rapid Spanning Tree Protocol - Uses BPDU packets to explore the graph of the network and compute the spanning tree of the network and detect the —malicious— cycles if any.

This just to recall that the switch is not a stupid thing! It is complex, fascinating, and deserves love; it's crucial to understand its functioning, also because its protocols occupy bandwidth.

4.2 Ethernet Topology

Typically nowdays the network is a **graph**, where internal nodes are switches or routers, and the leaves are servers.

The phyisical medium is no more shared, but conceptually the data link layer behaves as if it was.

On a switch, the only way to emulate a **shared bus**, is to ***"copy-paste"*** a frame onto multiple ports, losing the "identity" of frames. Packets in higher layers (IP?) have an ID, but frames don't, making it impossible to recognize whether a frame is a copy of another one or not. This approach makes **loops** a problem, because they disrupt performance by generating a packet storm.
The solution would be to ensure that the topology resembles a **tree**, instead of a graph. But, at the same time, a **fully connected graph** allows to have multiple routes for the same destination, possibly enhancing performance, reducing "hops" before reaching the destination.

4.2.1 `RSTP`

So... how can we leave the graph to be connected, but making it a tree from a logical point of view?

The answer is the `RSTP` protocol.

RSTP sends *probes* to understand whether there are loops and where are PCs located.

4.2.2 Spine Leaf architecture

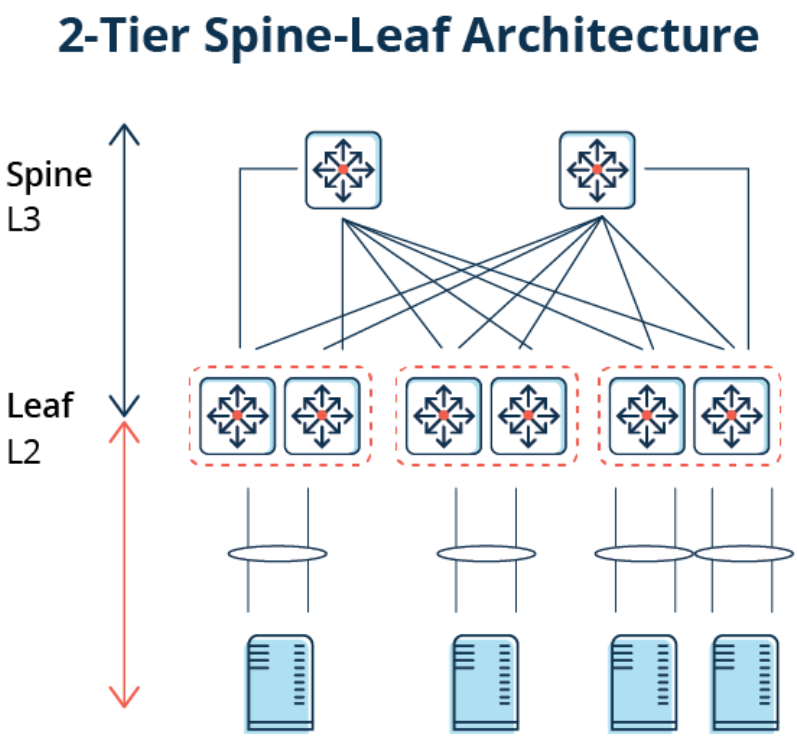

Figure 4.1: Spine-leaf architecture schema (from Arubanetworks.com)

A **spine-leaf** architecture is data center network topology that consists of two switching layers: a spine and leaf. It ensures traffic is always the same number of hops from its next destination, so latency is lower and predictable (fixed).

Capacity also improves because STP is no longer required. While STP enables redundant paths between two switches, only one can be active at any time. As a result, paths often become oversubscribed. Conversely, spine-leaf architectures rely on protocols such as *Equal-Cost Multipath* (`ECMP`) routing to load balance traffic across all available paths while still preventing network loops.

Spine-leaf allows *scale-out* opposed to *scale-up*, by adding additional spine switches, ultimately increasing capacity.

4.2.3 Full fat tree

In a —full?— **fat tree**, branches nearer the top of the hierarchy are "fatter" (thicker) than branches further down the hierarchy. In a telecommunications network, the branches are data links; the varied thickness (bandwidth) of the data links allows for more efficient and technology-specific use.

Full-fat tree is rarely needed.

4.3 Virtualization

With VLAN frames are extended by 4 bytes. Every switch nowdays automatically sets the `VLAN_ID` to `1`; if the field is not existent, it is appended, making an **tagged** an *untagged* frame.

Switches ensure that data cannot spill/leak from a VLAN to another. VLAN became largely of use when 10Gbit connection came out, because only 1Gbit was a too constrained bandwidth to be splitted into multiple VLANs.

VLAN are used to partition the traffic at data link layer without having to redo the fabric. They are particularly useful in cloud environments.

4.4 Network Administrator POV

The switch is split in two planes:

- **Control Plane**
  This plane is necessary to configure the data plane to make it behave according to our needs. Here there is an *OS*, which used to be proprietary with a functioning fitting a specific network configuration, but nowdays they are usually more configurable and may even be *open* OS.
  Dell's switches now have an *open* OS on board.
- **Data Plane**
  Here lies the chip responsible to perform all the data link operations required, runs protocols, handles VLANs, etc.
  OpenFlow allows us to manage the flow table inside of a switch.

The two planes are linked by a low-bandwidth PCIe.

It is possible to use a very fast and simple —reduced number of keystroke down to the strict necessary ones (e.g. `en` instead of `enable`)— CLI to program a switch. It is also possible to create a script file to be automatically executed by the switch at boot time.

Prof. Cisternino performed a demo of this in class.

Interestingly, the behaviour of the `netsh` command in Windows is very similar to the one of a switch.

4.5 SDN

Software Defined Networking is a new approach to networking that uses software-based controllers or application programming interfaces (APIs) to communicate with the underlying hardware infrastructure and direct traffic on the network.

The problem was that the network infrastructure was "ossified" and not programmable. SDN allows to program the network, and to make it more flexible and adaptable to the needs of the applications, without having to disrupt the existing infrastructure.

The key idea proposed in the OpenFlow article, which eventually became a standard, is to separate the control plane from the data plane, and to have a controller that can program through an API the data plane, where the **Flow Table** resides.

An interesting use of OpenFlow was implemented by a University and called Sandwich firewall, which consisted in routing the first part of the stream through a firewall and if the stream was not malicious, it was routed directly to the destination, otherwise it was dropped.

Latency-sensitive

Some workloads are called *latency-sensitive*, making the latency introduced by TCP-IP stack a problem.
Inside a datacenter nowadays the typical latency is sub microsecond.

InfiniBand is a fabric technology that allows to have a very low latency, and is used in HPC[1] environments. *OmniPath* is a technology that is similar to InfiniBand, but is more scalable.

Fibre Channel switches are used in storage area networks, and are used to connect storage to servers CPUs. They are used in datacenters, but not for networking.

[1] High performance computing

4.6 Storage

Data Loss

*"Storage is crucial because, if a switch fails, or a server fails, the service will be interrupted, but the data will still be there. If the storage fails, the **data will be lost**."* -Prof. Cisternino

Data is the most important of a system. Since data loss is **permanent**, the storage is completely different from computing or networking.

Historically the storage was the slowest part of the system, *ms* against *ns* of the CPU. Today, with SSDs, the gap is considerably reduced to *us*.

NVMe stands for *Non-Volatile Memory Express*, and is a protocol (*not a HW component!*) that allows to access the storage directly from the PCIe bus, without having to go through the SATA controller. This allows to have a much higher throughput, and a much lower latency.

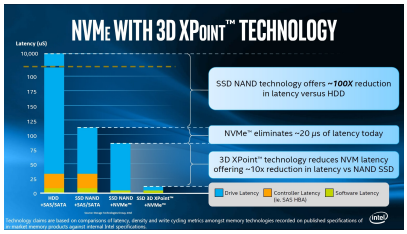

Figure 4.5: Storage types comparison

Optane was a technology developed by intel which is now end of life

SSDs were invented by Toshiba back in 1980, but they were not popular for almost 30 years, until they eventually became cost-effective. Sometimes extra size in SSDs is used for redundancy, to increase the lifespan of the disk e.g. on a 30TB disk, only 10TB are used, the rest is used for redundancy, extending x3 the lifespan of the disk.

Why would a 15TB disk be better than a 27TB disk?
Assume the same performance, and the same price.

It would be preferrable because it would take less time to extract all the data from the disk[2], since it is smaller.

However, large capacity drives are used for *cold storage*, where the data is not accessed frequently, speed is not a priority, and even if the data is accessed, only a portion of the disk is needed at a time; in case of failure and thus needing to retrieve an entire backup, the time taken to retrieve the data is not a priority, since this —hopefully— happens only "once".

4.7 SSDs - QLC and TLC

TLC stands for *Triple Level Cell*, and QLC stands for *Quad Level Cell*. The difference between the two is the number of bits stored in each cell. The more bits stored in each cell, the cheaper the disk is, but the slower it is. The more bits stored in each cell, the more difficult it is to read and write the data, and the more difficult it is to keep the data stored in the cell.

Generally QLC disks are used for cold storage, while TLC disks are used for hot storage. TLC in general is more reliable than QLC, has a longer lifespan and better performance, however they cost more.

4.8 Latency

A mechanical hard drive introduces 2.71% of latency when reading, for instance, 40MB of data. Optane can perform 416 accesses in the same time needed by a mechanical hard drive to perform 1 access. It looks like the latency in this latter case is neglegtible. Someone may be tempted to reduce the size of read/write operations and perform multiple smaller ones, since "it's free".

4.9 Checkpoints

It's unpractical for a system to go down after 5 months. For this reason it is necessary to have checkpoints, which are points in time where the system can be restored to. The system can be restored to the last checkpoint, and the

[2] i.e. taking advantage of the space provided

data that was written after the checkpoint can be re-applied. This is similar to what happens to applications on smartphones are closed and then re-opened, the application is restored to the last checkpoint.

4.10 RAID

RAID stands for *Redundant Array of Independent Disks*. It is a technology that allows to combine multiple disks into a single logical unit, in order to increase the performance, the reliability, or both. There are different levels of RAID, each with different characteristics.

Historically *Redundant Array of Inexpensive Disks*, because it was more common for disks to eventually fail, so RAID was the only countermeasure to this. Today, disks are more reliable, so RAID is used more for performance reasons.

In RAID, **XOR** is used to calculate the parity of the data. The parity is used to recover the data in case of a disk failure. The parity is calculated by XORing the data of the disks. The parity is stored on a separate disk, called the parity disk. The parity disk is used to recover the data in case of a disk failure.

4.11 Network sharing

SMB/CIFS is a protocol that allows to share files over the network. It is used by Windows, but it is also supported by Linux and MacOS. NFS is a protocol that allows to share files over the network, it is used by Linux and MacOS, but it is also supported by Windows.
NFS is faster than SMB, but it is also less secure. SMB is slower than NFS, but it is also more secure.

NAS instead are devices that are connected to the network, and that are used to store files. They are used to store files that are accessed by multiple users, and that need to be accessed from multiple devices. NAS devices are used to store files that are accessed by multiple users, and that need to be accessed from multiple devices. NAS devices are used to store files that are accessed by multiple users, and that need to be accessed from multiple devices.

HBA stands for *Host Bus Adapter*, and is a device that allows to connect a computer to a storage device. It is used to connect a computer to a storage device, and to allow the computer to access the storage device.

When we talk about **capacity**, there are two measures which we can refer to:

1. *Scale-**up***: adding more disks to the same server
2. *Scale-**out***: adding more servers to the same network

SAN stands for *Storage Area Network*, and is a network that is used to connect multiple storage devices to multiple servers. It is used to connect multiple storage devices to multiple servers, and to allow the servers to access the storage devices. It is used to connect multiple storage devices to multiple servers, and to allow the servers to access the storage devices.
SAN was, before SSDs, one of the datacenter pillars. Its architecture included a “Head” to which drives were attached, and the head was connected to the network. The head was used to manage the drives, and to allow the servers to access the drives.
When SSDs became popular, the head became a bottleneck, because it was not able to keep up with the speed of the SSDs(Recall that 4 SSDs are enough to saturate a 10Gbit link, See Sec. 3.1.1). For this reason, the head was removed, and the drives were connected directly to the network. This is called **DAS**.

4.11.1 Synchronization Software and its Price

The “storage guy” must ensure that there is no condition under which can happen data loss, because it is never an option. It is also important to have powerful **synchronization algorithms**, which must allow data to be copied and synchronized in multiple locations without disrupting performance and handling concurrency; such software is typically *costful*.

It is difficult nowdays to establish what is the “right” price for software. The shift from highly specialized and costful hardware to general hardware-plus-software, gave the software, which still a non-physical entity, increasingly more value, perhaps even too much.

4.12 Snapshots, Compression and other features

4.12.1 Storage Provisioning

Storage provisioning is the process of assigning storage to a server. It is used to assign storage to a server, and to allow the server to access the storage. To decide how much storage to assign to a server, usually the architect overbooks the storage, even beyond an estimation of how much storage it will need in the future.

Besides, available space may *decrease* over time. **Snapshots** are to blame in this case. Snapshots consists in saving the differences between the current state of the data and the previous state of the data. This allows to recover the *overwritten* data in case of a failure, but it also takes up space. The more snapshots are taken, the more space is taken up. The more space is taken up, the less space is available for the data.

4.12.2 Compression

Compression grants less space to be used, but it also requires more CPU power. However note that also allows to save some **bandwidth**, which we know to be critical. *Searching* in compressed data is not trivial, but there are tools to do it, such as the `FM-index`.

4.12.3 Pools and LUNs

Storage pools are used to combine multiple storage devices into a single logical unit, in order to increase performance and reliability. **Storage LUNs** define a storage partition and are used to assign storage —a portion of the pool— to a server, and to allow the server to access the storage, using ACLs.

`SCSI` is a protocol whose idea is to create a BUS of drives to be accessed together. It is an old protocol, designed in 1979 for small computers, but it is still used today, because it is very reliable. The key idea was for mutiple drives to share the same physical flat cable.
It had been "deprecated" in favor of `NVMe`, but it is still used today, because it is very reliable.

4.13 Hyperconverged Infrastructure

SAN started to create a sensible bottleneck, so designers started to "move drives towards the servers". **DAS** stands for *Direct Attached Storage*, and is a technology that allows to connect multiple storage devices to a single server, in order to increase the performance, the reliability, or both. The limitations is that you can only attach up to 2 or 3 drives to a server.

An idea came out to use the servers' internal drive to build a Storage Area Network, and this is called **VSAN**.

HCI stands for *Hyperconverged Infrastructure*, and is a technology that allows to combine multiple servers into a single logical unit, in order to increase the performance, the reliability, or both. The idea was born to allow a scale-out architecture, where you can add more servers to the network.

The **Hypervisor** is the software that allows to run multiple virtual machines on a single server. There should be some locality between the VM and the storage, because the VM should be able to access the storage quickly. `read` operations are always performed locally on local drives; `write` operations instead sometimes require to retrieve a remote piece of data.

4.13.1 Riak

Riak is a distributed database that is used to store data in multiple locations.

Recently it has been recognized that using general purpose hardware is no longer a feasible option.

Chapter 5

Computing

"In a server how many OSs are ran at the same time?"

2 in general, one is "Base Management Console"[1].

The BMC is a full OS running in a board which executes also when the server is off, but attached to a power supply. It is a component which allows you to manage the server as if you were phyisically handling the server.

Prof. Cisternino display iDRAC (Dell's BMC) in class. It is a web interface which allows you to manage the server, check its status, and even turn it on and off.
It is also possible to access a console, which is a virtual console, which allows you to interact with the server as if you were physically there with keyboard and mouse attached; such console also allows to install a new OS by uploading an `iso` and make the server boot from it as if it was attached to it.

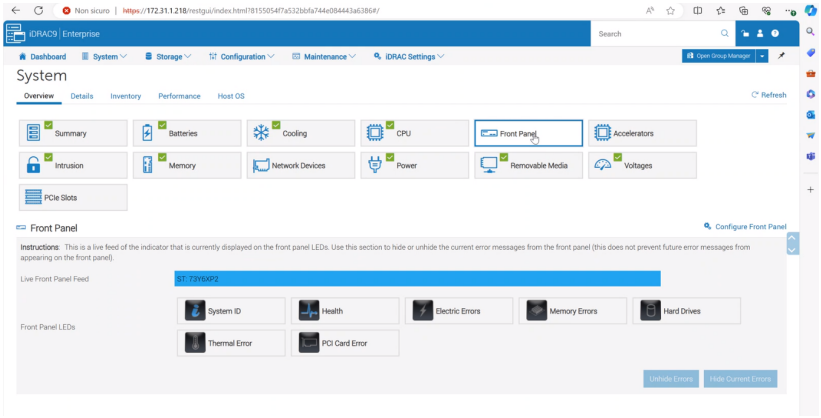

Figure 5.1: DEMO Interface displayed by prof cisternino in class
It is possible to remotely control and check the server's status.

Measuring Bus Speed

PCIe speed, as well as CPU speed is measured in GT/s, which stands for *Giga Transfers per second*. It is a measure of how many transfers can be performed in a second.

5.1 Knights Landing and high performance computing

Knights Landing processor architecture is a multi core architecture which is designed to be used in supercomputers. The memory had a super high bandwidth, to avoid bottlenecking the many cores inside, since such memory is shared among them.

NUMA is a technique used to avoid bottlenecking in multi core architectures. It is a technique which allows to have

[1] May have other names, but this is a common one, used by `Supermicro`

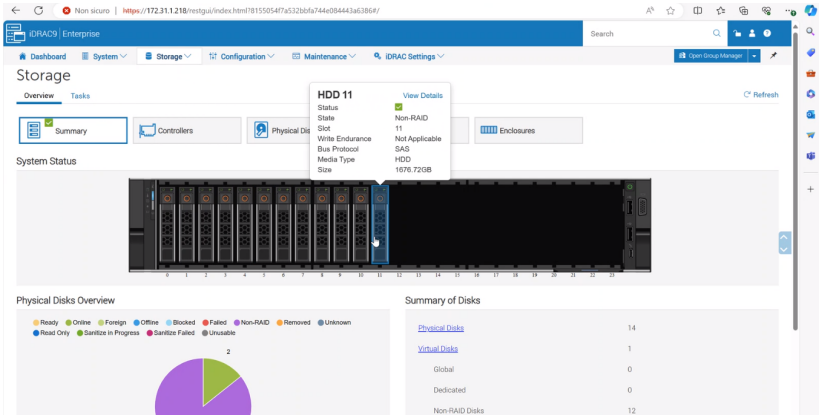

Figure 5.2: Monitoring storage
SAS bus protocol is used for storage devices.

multiple memory banks, each one connected to a subset of the cores. This way, each core can access its own memory bank without having to wait for the others to finish accessing the shared memory.

5.2 Rings

Bachelor's professors fooled us into thinking that CPUs have two operating modes, *user* and *kernel* mode. Sadly, this ain't true, it is an abstraction. In reality, CPUs have 4 **rings**, which are used to separate the different levels of privilege. The higher the ring, the higher the privilege level. The kernel runs in ring 0, while the user runs in ring 3.

Nowdays there are multiple units in the CPU, which are used to execute instructions, and there is a head unit which decides which instruction to execute next and on which unit.

Chiplets are a way to increase the number of cores in a CPU. They are small chips which are connected to the main CPU.