Computing Infrastructure: [Course Code]

[Sofia Martellozzo]

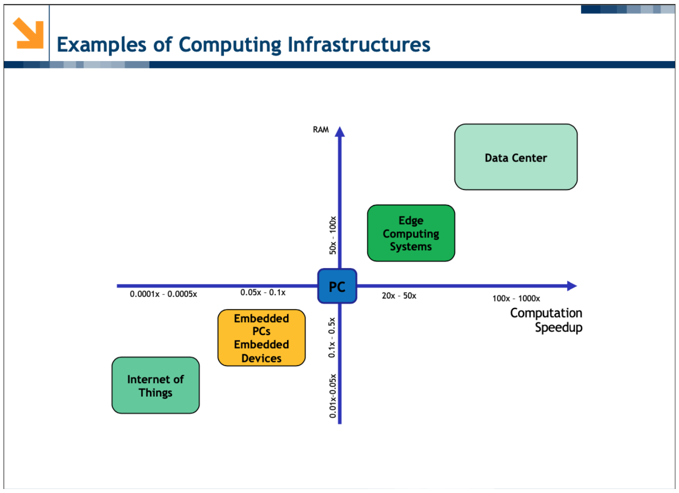
Academic Year 2021-2022

Contents

1 Computing Infrastructure

1.1 Introduction

Definition 1. Computing Infrastructure: *Technological infrastructure that provides hardware and software for computation to other systems and services.*



Examples of Computing Infrastructures

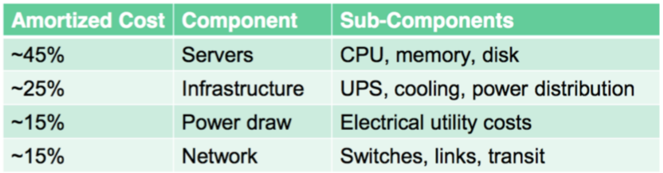
Figure 1: Computing Infrastructure

Advantages:

- Lower IT costs
- High performance
- Instant software updates
- "Unlimited" storage capacity
- Increased data reliability
- Universal document access
- Device Independence

Disadvantages:

- Require a constant Internet connection
- Do not work well with low-speed connections
- Hardware Features might be limited
- Privacy and security issues
- High Power Consumption (1% overall worldwide total energy consumption due to datacenters)
- Latency in making decision

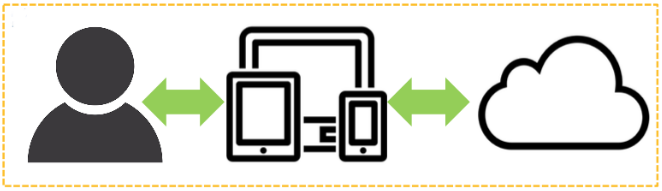


Amortized Cost	Component	Sub-Components
~45%	Servers	CPU, memory, disk
~25%	Infrastructure	UPS, cooling, power distribution
~15%	Power draw	Electrical utility costs
~15%	Network	Switches, links, transit

2 Data WareHouse

2.1 Introduction

In the last few decades, computing and storage have moved from PC- like clients to smaller, often mobile, devices, combined with large internet services.
Traditional enterprises are also shifting to Cloud computing.



User experience improvements

- Ease of management (no configuration or backups needed)
- Ubiquity of access

Advantages to vendors

- Software-as-a-service allows faster application development (easier to make changes and improvements)
- Improvements and fixes in the software are easier inside their data centers (instead of updating many millions of clients with peculiar hardware and software configurations)
- The hardware deployment is restricted to a few well-tested configurations.

Server-side computing allows

- Faster introduction of new hardware devices (e.g., HW accelerators or new hardware platforms)
- Many application services can run at a low cost per user.

Some workloads require so much computing capability that they are a more natural fit in datacenter (and not in client-side computing).
A couple of examples (Search services (web, images, and so on), Machine and Deep Learning (GPT-3).

2.2 From Data Centers to Warehouse-scale computers

Data centers = is a place in which there are many servers
Whareouse = is a type of Data Center, it works as a computer.
The trends toward server-side computing and widespread internet services created a new class of computing systems:

Definition 2. warehouse-scale computers (WSCs) *The massive scale of the software infrastructure, data repositories, and hardware platform.*

- *is an internet service (= service provided by inyternet)*
- *may consist of tens or more individual programs (= not a single program, a collection of them that together create/provide the service)*
- *such programs interact to implement complex end-user services such as email, search, maps or machine learning.*

Data centers are buildings where multiple servers and communication units are co-located because of their common environmental requirements and physical security needs, and for ease of maintenance.
In **Traditional Data Center**: typically host a large number of relatively small- or medium-sized applications, each applications is running on a dedicated hardware infrastructure that is de-coupled and protected from other systems in the same facility, applications tend not to communicate each other. Those data centers host hardware and software for multiple organizational units or even different companies.

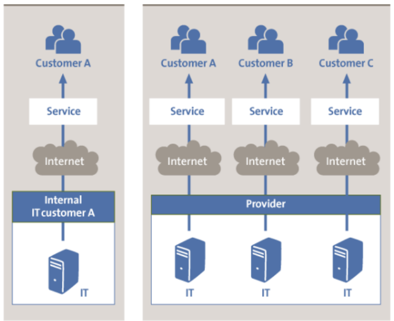


Figure 2: Traditional Data Center

WSCs belong to a single organization, use a relatively homogeneous hardware and system software platform (=¿ easier to manage and cheaper, but has limitations on functionalities), and share a common systems management layer (such as Google, Facebook, Alibaba, Amazon, Dropbox...).
(you have 1 services that you want to provide to a 'huge' amount of customers)
Run a smaller number of very large applications (or internet services).
The common resource management infrastructure allows significant deployment flexibility.
The requirements of:

- homogeneity
- single-organization control
- cost efficiency

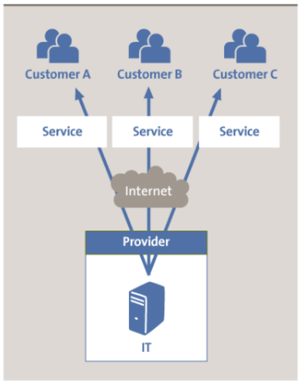


Figure 3: Warehouse-Scale Computers

Initially designed for online data-intensive web workloads, WSCs also now power public clouds computing systems (e.g., Amazon, Google, Microsoft). Such public clouds do run many small applications, like a traditional data center. (All of these applications rely on Virtual Machines (or Containers), and they access large, common services for block or database storage, load balancing, and so on, fitting very well with the WSC model).
These are not just a collection of servers:
The software running on these systems executes on clusters of hundreds to thousands of individual servers (far beyond

a single machine or a single rack)
+
The machine is itself this large cluster or aggregation of servers and needs to be considered as a single computing unit. (=> scale up in terms of performance)
Several datacenters: Multiple Data Center located far apart (placed near point of iterest) =¿ becomes important also the PRIVACY: data of a country must remain in it.
Multiple data centers are (often) replicas of the same service (to reduce user **latency** and improve serving **throughput**).
A request is typically fully processed within one data center.
Availability: Services provided through WSCs must guarantee high availability, typically aiming for at least 99.99% uptime (i.e., one-hour downtime per year). Achieving such fault-free operation is difficult when a large collection of hardware and system software is involved.
WSC workloads must be designed to gracefully tolerate large numbers of component faults with little or no impact on service level performance and availability.

2.3 Architectural Overview of WSCs

Hardware implementation of WSCs might differ significantly each other; However, the architectural organization of these systems is relatively stable.

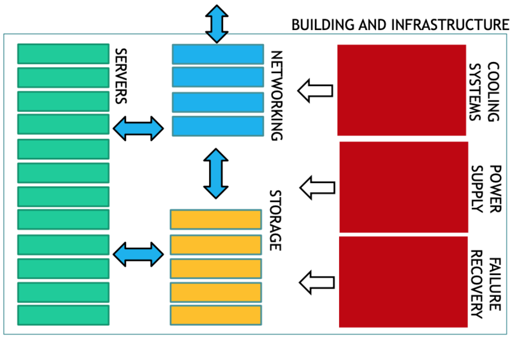


Figure 4: Warehouse-Scale Computers Overview

- Servers: the main processing equipment
- Storage: how and where to store the information
  - Disks and Flash SSDs are the building blocks of today's WSC storage systems.
  - These devices are connected to the data-center network and managed by sophisticated distributed systems

  Examples:

  - Direct Attached Storage (DAS)
  - Network Attached Storage (NAS)
  - Storage Area Networks (SAN)
  - RAID controllers
- Networking: providing internal and external connections
  - Communication equipment allows network interconnections among the devices.

  They can be:

  - Hubs
  - Routers
  - DNS or DHCP servers

  - Load balancers
  - Technology switches
  - Firewalls

- **Building and Infrastructure**:WSC has other important components related to power delivery, cooling, and building infrastructure that also need to be considered

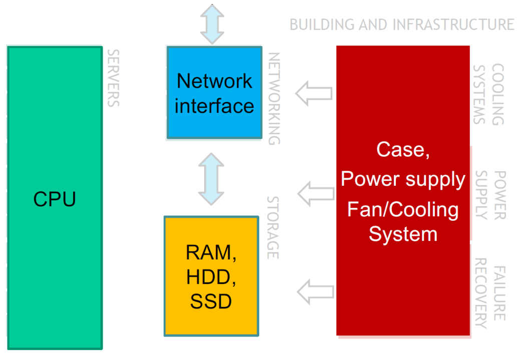


Figure 5

3 Server

3.1 Overview

Servers hosted in individual shelves are the basic building blocks of WSCs. They are interconnected by hierarchies of networks, and supported by the shared power and cooling infrastructure. Them are stored in the shelves (wraks) that are organized along corridors (the basic blocks of our data-center). All servers are connected to each other in the same wrak, but also (on a higher level) communicate through different wrack. They are like ordinary PC (from a LOGISTIC point of view), but with a form factor that allows to fit them into the racks. They have 3 type of shape:

- Rack (1U or more)
- Blade enclosure format
- Tower

They may differ in:

- Number and type of CPUs
- Available RAM Locally attached disks (HDD, SSD or not installed)
- Other special purpose devices (like GPUs, DSPs and coprocessors)

Servers are usually built in a tray or blade enclosure format, housing the motherboard, chipset, additional plug-in components.

3.1.1 The Motherboard

The motherboard provides sockets and plug-in slots to install CPUs, memory modules (DIMMs), local storage (such as Flash SSDs or HDDs), and network interface cards (NICs) to satisfy the range of resource requirements.

WSCs use a relatively homogeneous hardware and system software platform.

3.1.2 Chipset and additional components

- Number and type of CPUs:
  - From 1 to 8 CPU socket
  - Intel Xeon Family, AMD EPYC, etc..
- Available RAM
  - From 2 to 192 DIMM Slots
- Locally attached disks:
  - From 1 to 24 Drive Bays
  - HDD or SSD (see specific lecture)
  - SAS (higher performance but more expensive) or SATA (for entry level servers, usually cheaper)
- Other special purpose devices:
  - From 1 to 20 GPUs per node, or TPUs
  - NVIDIA Pascal, Volta, etc..
- Form factor:
  - Form 1U to 10U
  - Tower

We distinguish between rack, tower and blade.

3.1.3 Rack

Racks are special shelves that accommodate all the IT equipment and allow their interconnection. The rack are used to store many **Rack server**.
IT equipment must conform to specific sizes to fit into the rack shelves. Them has different hight, but same SIZE, that is **standardize** (so make them easier to design): server racks are measured in rack units "U's".
The advantages of using these racks is that it allows designers to stack up (one on top of the others) other electronic devices along with the servers.

A Rack is not only a physical structure:
(3) The rack is the shelf that holds tens of servers together.
(2-4) It handles shared power infrastructure, including power delivery, battery backup, and power conversion.
(2) There are 2 types of electronic delivery: it can provide/automatically select the server (can swich ON and OFF remotely the servers).
(4) It is used in case of problems with energy (power failure):

1. just swich off if a power failure occur, them cannot supply all the power demand (execution)
2. it provides 15 minutes of power, that is the time to swich on the power supply generator (source)

(3) The width and depth of racks vary across WSCs: some are classic 19-in wide, 48-in deep racks, while others can be wider or shallower.
(1) It is often convenient to connect the network cables at the top of the rack, such a rack-level switch is appropriately called a Top of Rack (TOR) switch.

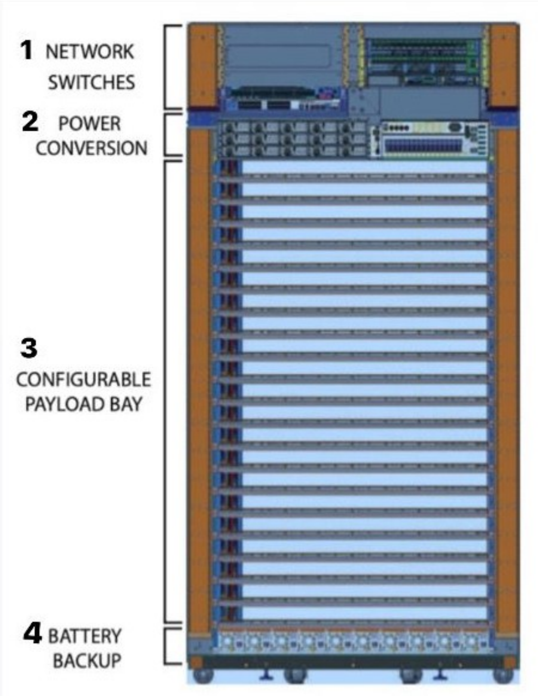


Figure 6: RACK Servers

Rack servers:
It is designed to be positioned in a bay, by vertically stacking servers one over the another along with other devices

Pros

- **Failure containment**: very little effort to identify, remove, and replace a malfunctioning server with another.
- **Simplified cable management**: easy and efficient to organize cables.
- **Cost-effective**: Computing power and efficiency at relatively lower costs.

Cons

- **Power usage**: Needs of additional cooling systems due to their high overall component density, thus consuming more power.
- **Maintenance**: Since multiple devices are placed in racks together, maintaining them gets considerably tough with the increasing number of racks.

3.1.4 Tower

It is the simplest but is not usually adopted. **Tower Servers** look and feel a lot like traditional tower PCs. Like everything they have their Pros and Cons.

Pros

- **Scalability and ease of upgrade**: customized and upgraded based on necessity.
- **Cost-effective**: Tower servers are probably the cheapest of all kinds of servers
- **Cools easily**: Since a tower server has a low overall component density, it cools down easily.

Cons

- **Consumes a lot of space**: These servers are difficult to manage physically.
- **Provides a basic level of performance**: a tower server is ideal for small businesses that have a limited number of clients.
- **Complicated cable management**: devices aren't easily routed together

3.1.5 Blade

Blade servers are the latest and the most advanced type of servers in the market. They can be termed as hybrid rack servers (like orizontal rack server but placed vertically), in which servers are placed inside blade enclosures, forming a blade system. The biggest advantage of blade servers is that these servers are the smallest types of servers available at this time and are great for conserving space. A blade system also meets the IEEE standard for rack units and each rack is measured in the units of “U’s”.

Pros

- **Load balancing and failover**: Thanks to its much simpler and slimmer infrastructure, load balancing among the servers and failover management tends to be much simpler.
- **Centralized management**: In a blade server, you can connect all the blades through a single interface, making the maintenance and monitoring easy.
- **Cabling**: Blade servers don’t involve the cumbersome tasks of setting up cabling. Although you still might have to deal with the cabling, it is near to negligible when compared to tower and rack servers.
- **Size and form-factor**: They are the smallest and the most compact servers, requiring minimal physical space.

Cons

- **Expensive configuration: Although upgrading the blade server is easy to handle and manage, the initial configuration or the setup might require heavy efforts in complex environments.**
- **HVAC: Blade servers are very powerful and come with high component density. Therefore, special accommodations have to be arranged for these servers in order to ensure they don’t get overheated. Heating, ventilation, and air conditioning systems must be managed well in the case of blade servers.**

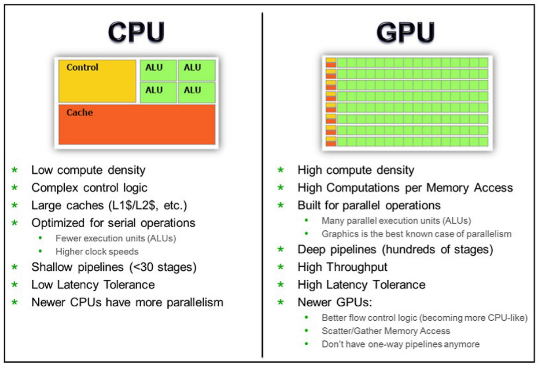
here the physical organization of a Data-Center!

Deep learning models began to appear and be widely adopted, enabling specialized hardware to power a broad spectrum of machine learning solutions. To satisfy the growing compute needs for deep learning, WSCs deploy specialized accelerator hardwares:

- GPU
- TPU
- FPGA

3.1.6 Graphical Processing Units (GPU)

Data-parallel computations: the same program is executed on many data elements in parallel. The scientific codes are mapped onto the matrix operations. High level languages (such as CUDA, OpenCL, OPENACC, OPENMP, SYCL) are required. Up to 1000x faster than CPU.



CPU

- Low compute density
- Complex control logic
- Large caches (L1$/L2$, etc.)
- Optimized for serial operations
  - Fewer execution units (ALUs)
  - Higher clock speeds
- Shallow pipelines (<30 stages)
- Low Latency Tolerance
- Newer CPUs have more parallelism

GPU

- High compute density
- High Computations per Memory Access
- Built for parallel operations
  - Many parallel execution units (ALUs)
  - Graphics is the best known case of parallelism
- Deep pipelines (hundreds of stages)
- High Throughput
- High Latency Tolerance
- Newer GPUs:
  - Better flow control logic (becoming more CPU-like)
  - Scatter/Gather Memory Access
  - Don’t have one-way pipelines anymore

Figure 7: CPU vs GPU

GPUs within the rack: PCI AND NVlink GPUs are configured with a CPU host connected to a PCIe-attached accelerator tray with multiple GPUs.
GPUs within the tray are connected using high-bandwidth interconnects such as NVlink. **NVLINK evolution and NVSwitch** In the A100 GPU, each NVLink lane supports a data rate of 50x 4 Gbit/s in each direction. The total number of NVLink lanes increases from six lanes in the V100 GPU to 12 lanes in the A100 GPU, now yielding 600 GB/s total

3.1.7 Tensor Processing Unit (TPU)

While suited to ML, GPUs are still relatively general purpose devices. In recent years, designers further specialized them to ML-specific hardware Custom-built integrated circuit developed specifically for machine learning and tailored for TensorFlow.
A Tensor is an n-dimensional matrix. This is the basic unit of operation in with TensorFlow.
TPUs are used for training and inference:

- TPUv1 is an inference-focused accelerator connected to the host CPU through PCIe links.
- Differently, TPUv2 and TPV3 focus training and inference

Each Tensor core has an array for matrix computations (MXU) and a connection to high bandwidth memory (HBM) to store parameters and intermediate values during computation. **TPUv2**
8 GiB of HBM for each TPU core, one MXU for each TPU core, 4 chips, 2 cores per chip.

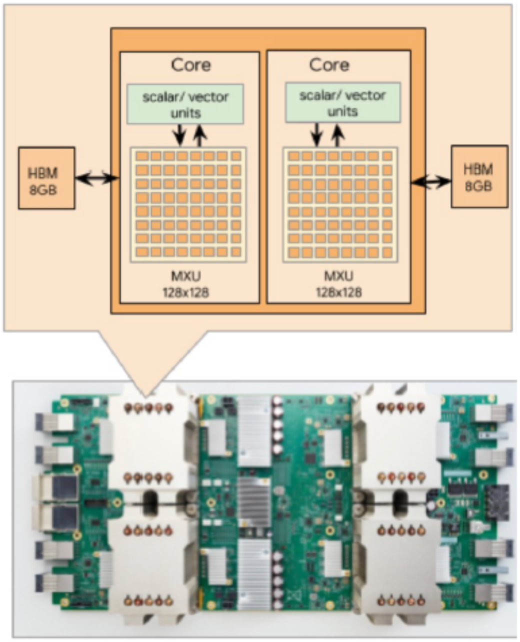


Figure 8: TPUv2 - 4 chips, 2 core per chip

In a rack multiple TPUv2 accelerator boards are connected through a custom high-bandwidth network to provide 11.5 petaflops of ML compute. The high bandwidth network enables fast parameter reconciliation with well-controlled tail latencies Up to 512 total TPU cores and 4 TB of total memory in a TPU Pod (64 units). **TPUv3**
TPUv3 is the first **liquid-cooled accelerator** in Google's data center. 2.5x faster than TPUv2. Such supercomputing-class computational power supports new ML capabilities (e.g., AutoML), and rapid neural architecture search. The v3 TPU Pod provides a maximum configuration of 256 devices for a total 2048 TPU v3 cores, 100 petaflops and 32 TB of TPU memory.

3.1.8 Field-Programmable Gate Array (FPGA)

Array of logic gates that can be programmed ("configured") in the field, i.e.,by the user of the device as opposed to the people who designed it.

Array of carefully designed, and interconnected digital subcircuits, that efficiently implement common functions offering very high levels of flexibility. The digital subcircuits are called configurable logic blocks (CLBs).
VHDL and Verilog are hardware description languages (HDLs), that allow to “describe” hardware. HDL code is more like a schematic that uses text to introduce components and create interconnections.
Microsoft deployed FPGAs inside its Datacenters.

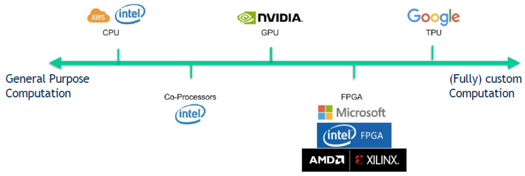


Figure 9: Overview

Advantages and Disadvantages

CPU Advantages: Easy to be programmed and support any programming framework.
Fast design space exploration and run your applications.
CPU Disadvantages: Most suited for simple AI models that do not take long to train and for small models with small training set.
GPU Advantages: Ideal for applications in which data need to be processed in parallel like the pixels of images or videos.
GPU Disadvantages: Programmed in languages like CUDA and OpenCL and therefore provide limited flexibility compared to CPUs.
TPU Advantages: Very fast at performing dense vector and matrix computations and are specialized on running very fast program based on Tensorflow.
TPU Disadvantages: For applications and models based on the TensorFlow. Lower flexibility compared to CPUs and GPUs.
FPGA Advantages: Higher performance, lower cost and lower power consumption compared to other options like CPUs and GPU.
FPGA Disadvantages: Programmed using OpenCL and High-level Synthesis (HLS).
Limited flexibility compared to other platforms.

Data-center architecture

The IT equipment is stored into corridors and organized into racks. Server Racks are NEVER BACK-to-BACK.
Corridors where servers are located are split into *cold aisle,* where the front panels of the equipment is reachable, and *warm aisle,* where the back connections are located.
Cold air flows from the front (cool aisle), cools down the equipment, and leave the room from the back (warm aisle).