Computing Infrastructure: [Course Code]

[Sofia Martellozzo]

Academic Year 2021-2022

Contents

1 Computing Infrastructure **3**

1.1 Introduction 3

2 Data WareHouse **4**

2.1 Introduction 4

2.2 From Data Centers to Warehouse-scale computers 4

2.3 Architectural Overview of WSCs 6

3 Server **7**

3.1 Overview 7

3.1.1 The Motherboard 7

3.1.2 Chipset and additional components 7

3.1.3 Rack 8

3.1.4 Tower 9

3.1.5 Blade 9

3.2 Data-center architecture 9

3.3 Hardware accelerators 9

3.3.1 Graphical Processing Units (GPU) 10

3.3.2 Tensor Processing Unit (TPU) 10

3.3.3 Field-Programmable Gate Array (FPGA) 11

3.3.4 Advantages and Disadvantages 11

4 Storage **12**

4.1 Hard Disk Drives 12

4.1.1 Read Write heads, basic characteristic 12

4.1.2 Other characteristics 12

4.2 Solid-state Storage Device 13

4.2.1 Internal Organization 13

4.2.2 SSD summary 16

4.3 HDD vs SSD 16

4.4 Hybrid solution 17

4.5 Storage system 17

4.5.1 DAS 17

4.5.2 NAS 18

4.5.3 SAN 18

5 Networking **19**

5.0.1 Architecture 19

5.0.2 High Performance Clusters 20

5.0.3 Network to support Virtualization 20

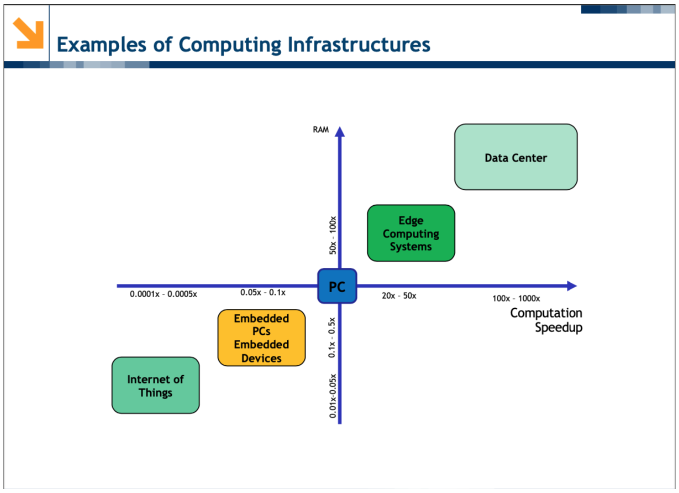
5.0.4 The interplay of storage and networking technology 20

5.0.5 Balanced design 21

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

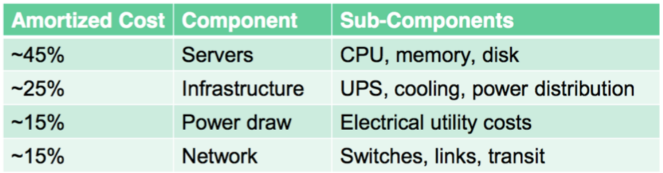
Figure 1: Computing Infrastructure

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

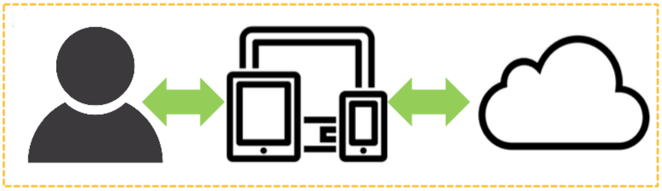


Amortized Cost	Component	Sub-Components
~45%	Servers	CPU, memory, disk
~25%	Infrastructure	UPS, cooling, power distribution
~15%	Power draw	Electrical utility costs
~15%	Network	Switches, links, transit

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.

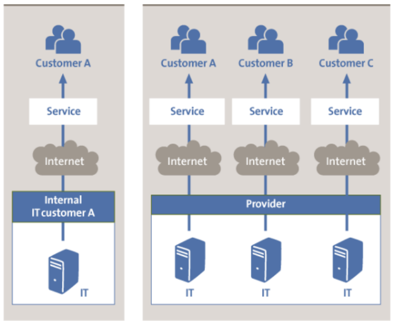


Figure 2: Traditional Data Center

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

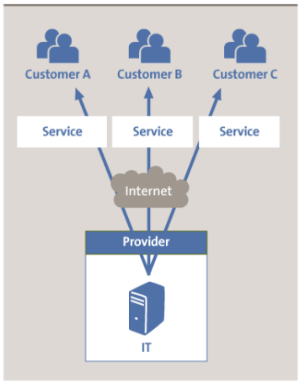


Figure 3: Warehouse-Scale Computers

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 data-centers: Multiple Data Center located far apart (placed near point of interest) =¿ 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.

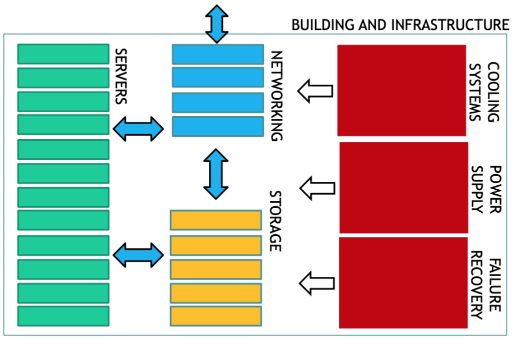


Figure 4: Warehouse-Scale Computers Overview

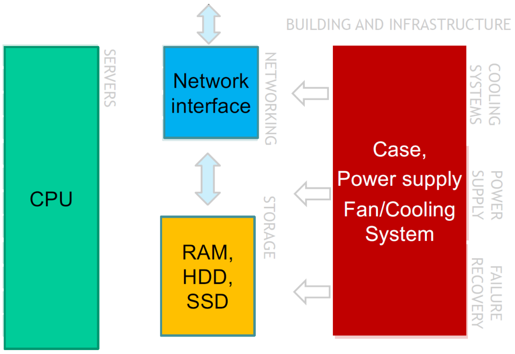


Figure 5

- **Servers**: the main processing equipment
- **Storage**: how and where to store the information
- **Networking**: providing internal and external connections
- **Building and Infrastructure**:WSC has other important components related to power delivery, cooling, and building infrastructure that also need to be considered

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. They are stored in shelves called wraks, that are organized along corridors. All servers are connected to each other in the same wrak, but also (on a higher level) communicate through different wrack. 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).

and 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.

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:

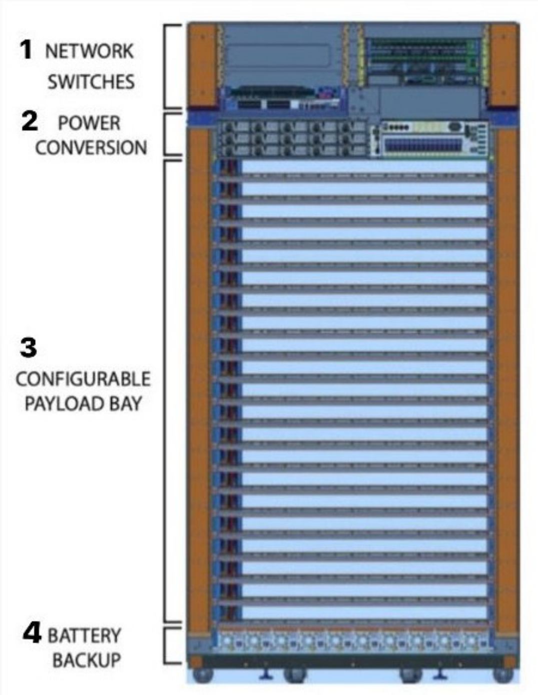


Figure 6: RACK Servers

(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):

- just swich off if a power failure occur, them cannot supply all the power demand (execution)

- 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.

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.

3.2 Data-center architecture

The IT equipment is stored into corridors and organized into racks (the goal is to maximize the number of racks =¿ max number of servers).
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.
cooling system: Cold air flows from the front (cool aside), cools down the equipment, and leave the room from the back (warm aide). There is an additional roof in order to do not waste cold air on the Back side (cold part of the corridor).

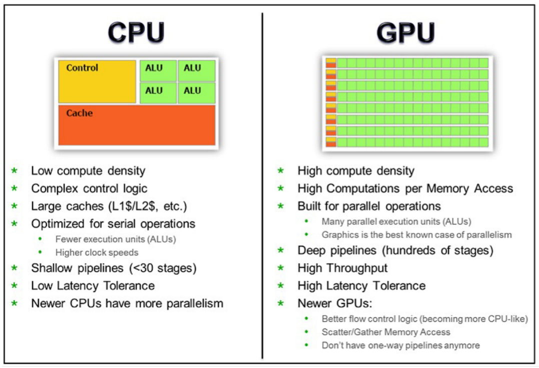
3.3 Hardware accelerators

Hardware accelerator are accurate particular applications that support specific high operation (of ML) with a lot of data. 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

3.3.1 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

The performance of such a synchronous system is limited by the slowest learner and slowest messages through the network. Since the communication phase is in the critical path, a high performance network can enable fast reconciliation of parameters across learners.
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

3.3.2 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.

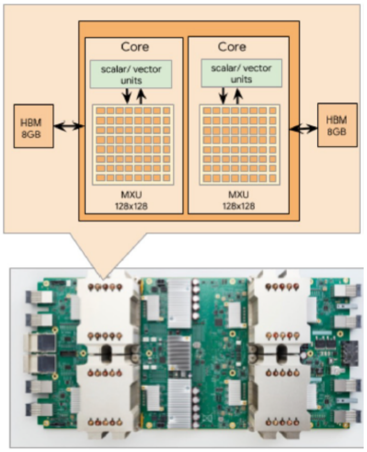


Figure 8: TPUv2 - 4 chips, 2 core per chip

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 is the first **liquid-cooled accelerator** in Google's data center (here there is no fresh air to cool it down, used fresh liquid). 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.
TPUv4 announced June 2021, used to support Google services (not yet available as a cloud service).One v4 TPU pod includes 4096 devices: About 2.7x faster than TPUv3 and same computing capacity as 10 millions of laptops.

3.3.3 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.

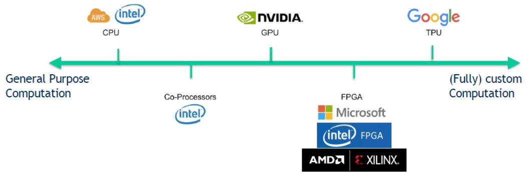


Figure 9: Overview

3.3.4 Advantages and Disadvantages

CPU

- **Advantages**:Easy to be programmed and support any programming framework.
Fast design space exploration and run your applications.
- **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.
- **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.
- **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.
- **Disadvantages**: Programmed using OpenCL and High-level Synthesis (HLS).
Limited flexibility compared to other platforms.

4 Storage

Nowadays machines generate data at an unprecedented rate

- 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

4.1 Hard Disk Drives

A hard disk drive (HDD) is a data storage using rotating disks (platters) coated with magnetic material. Data is read in a random-access manner, meaning individual blocks of data can be stored or retrieved in any order rather than sequentially.
An HDD consists of one or more rigid ("hard") rotating disks (platters) with magnetic heads arranged on a moving actuator arm to read and write data to the surfaces. The **Sector** is where you read and/or write; The organization of the data is in order to minimize the movement of the head (that increase the seek time): place the data that are related near! If the head continues to go forward and back increase the Latency!

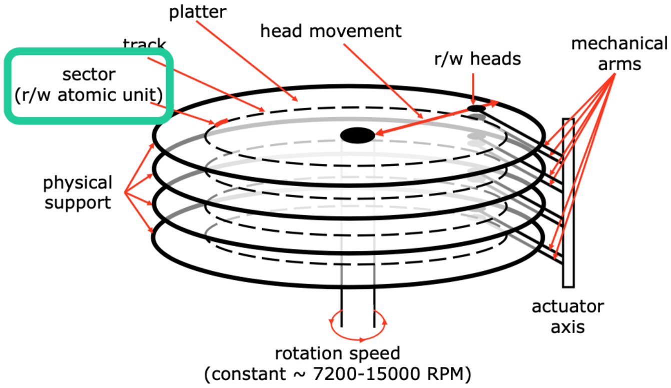


Figure 10: Hard Disk

4.1.1 Read Write heads, basic characteristic

- float on a film of air (tens of nanometers) above the platters
- one head for each magnetic platter
- cylinder: set of tracks with the same radius
- **seek time**: time required to reach the track that contains the data, 3÷14 ms

4.1.2 Other characteristics

- Diameter: about 9 cm (3,5÷2.5 in) - two surfaces
- Rotation speed: 7200÷15000 RPM round per minute
- Track density: 16,000 TPI (Track Per Inch)
- Sectors: 512 Byte (usually), but might be different (are numbered sequentially, have a header and an error correction code)
- Heads: can be parked close to the center or to the outer diameter (mobile drives)
- Disk buffer cache: embedded memory in a hard disk drive that has the function of a buffer between the disk and the computer (several MB)

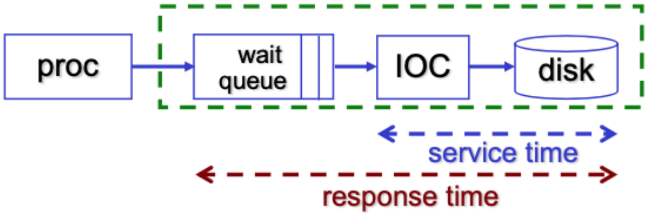


Figure 11: HDD service and response time

Service Time
s_{disk}: seek time + rotational latency + data transfer time + controller overhead

- **seek** time: head movement time ($\approx$ ms), is a function of the number of cylinders traversed
- **latency** time: time to wait for the sector ($\approx$ ms, $\frac{1}{2}$ round)
- **transfer** time: is a function of rotation speed, storing density, cylinder position ($\approx$ MB/sec)
- **controller overhead**: buffer management (data transfer) and interrupt sending time

no queue, average time to serve a single I/O request.
Response Time
service time + queue time, average time to serve an I/O request in working conditions

4.2 Solid-state Storage Device

No mechanical or moving parts like HDD
Built out of transistors (like memory and processors)
Retain information despite power loss unlike typical RAM
A controller is included in the device with one or more solid state memory components
It uses traditional hard disk drive (HDD) interfaces (protocol and physical connectors) and form factors
Higher performance than HDD
It stores Bit, based on Transistors (which compose them):

- Single-level cell (SLC) : single bit per cell
- Multi-level cell (MLC) cell : two bits per cell
- Triple-level cell (TLC) : three bits per cell
- QLC, PLC...

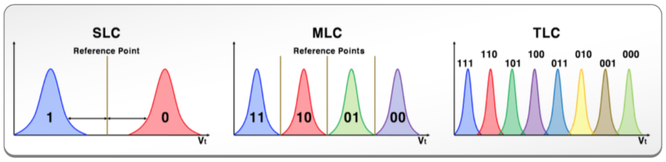


Figure 12: SSD

4.2.1 Internal Organization

NAND flash is organized into Pages and Blocks:
A **page** contains multiple logical block (e.g. 512B-4KB) addresses (LBAs), a **block** typically consists of multiple pages (e.g., 64) with a total capacity of around 128-256KB; Block/Page terminology in the SSD context can clash with previous use.
Blocks (or Erase Block): smallest unit that can be erased. It consists of multiple pages and can be cleaned using the ERASE command.
Pages : smallest unit that can be read/written. It is a sub-unit of an erase block and consists of the number of bytes which can be read/written in a single operations through the READ or PROGRAM commands.
Pages can be in three states:

- Dirty (or INVALID): they contain data, but this data is no longer in use (or never used)

- Empty (or EREASED): they do not contain data
- In use (or VALID): the page contains data that can be actually read

Only empty pages can be written
Only dirty pages can be erased, but this must be done at the block level (all the pages in the block must be dirty or empty)
It is meaningful to read only pages in the "in use" ("valid") state
If no empty page exists, some dirty page must be erased:

- If no block containing just dirty or empty pages exists, then special procedures should be followed to gather empty pages over the disk
- To erase the value in flash memory the original voltage must be reset to neutral before a new voltage can be applied, known as write amplification

Remark: we can write and read a single page of data from a SSD but we have to delete an entire block to release it

WRITE AMPLIFICATION: the actual amount of information physically written to the storage media is a multiple of the logical amount intended to be written.

Wear out: breakdown of the oxide layer within the floating-gate transistors of NAND flash memory.
The erasing process hits the flash cell with a relatively large charge of electrical energy.
Each time a block is erased:

- the large electrical charge actually degrades the silicon material
- after enough write-erase cycles, the electrical properties of the flash cell begin to break down and the cell becomes unreliable

Flash Translation Layer = layer that matches the file system and the SSD.

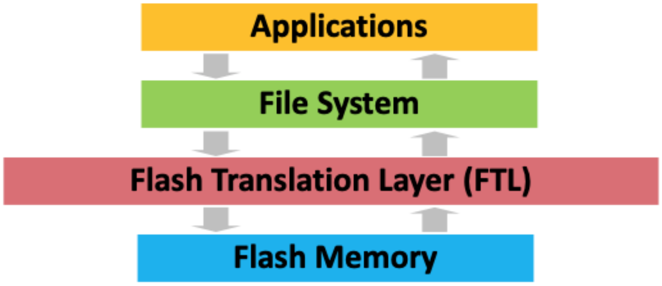


Figure 13: Flash Translation Layer

Direct mapping between Logical to Physical pages is not feasible
FTL is an SSD component that make the SSD "look as HDD":

1. Data Allocation and Address translation
   - Efficient to reduce Write Amplification effects
   - Program pages within an erased block in order (from low to high pages): Log-Structured FTL
2. Garbage collection: Reuse of pages with old data (Dirty/Invalid)
3. Wear leveling: FTL should try to spread writes across the blocks of the flash ensuring that all of the blocks of the device wear out at roughly the same time

Garbage collection
When an existing page is updated à old data becomes obsolete!
Old version of data are called garbage and (sooner or later) garbage pages must be reclaimed for new writes to take place.
Garbage Collection is the process of finding garbage blocks and reclaiming them: Simple process for fully garbage blocks or more complex for partial cases. I.e. Basic steps:

1. Find a suitable partial block

2. Copy non-garbage pages
3. Reclaim (erase) the entire block for writing

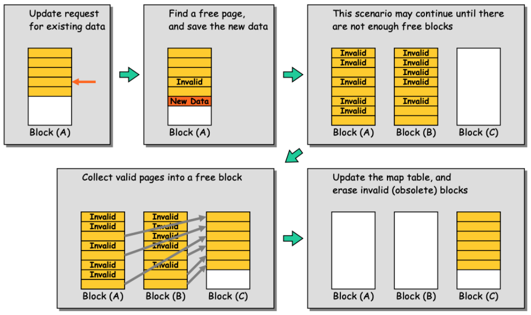
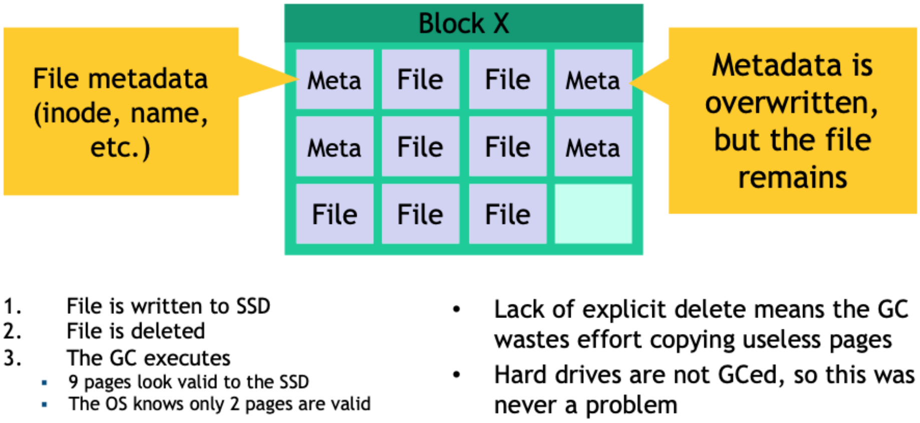

Update request for existing data
Block (A)
Find a free page, and save the new data
Invalid
New Data
Block (A)
This scenario may continue until there are not enough free blocks
Invalid
Block (A)
Block (B)
Block (C)
Collect valid pages into a free block
Block (A)
Block (B)
Block (C)
Update the map table, and erase invalid (obsolete) blocks
Block (A)
Block (B)
Block (C)

Figure 14: Garbage collection (Example)

Garbage collection is expensive: Require reading and rewriting of live data and the ideal garbage collection is reclamation of a block that consists of only dead pages.
The **8cost** of Garbage Collection depends on the amount of data blocks that have to be migrated.
Solutions to alleviate the problem:

- Overprovision the device by adding extra flash capacity: Cleaning can be delayed
- Run the Garbage Collection in the background using less busy periods for the disk

When performing background GC the SSD assumes to know which pages are invalid
Problem: most file systems don't actually delete data (E.g., on Linux, the "delete" function is unlink() and it removes the file meta-data, but not the file itself)


Block X
File metadata (inode, name, etc.)
Meta
File
File
Meta
Meta
File
File
Meta
File
File
File
Metadata is overwritten, but the file remains

1. File is written to SSD
2. File is deleted
3. The GC executes
   - 9 pages look valid to the SSD
   - The OS knows only 2 pages are valid

- Lack of explicit delete means the GC wastes effort copying useless pages
- Hard drives are not GCed, so this was never a problem

Figure 15: Garbage collection (Example)

New SATA command TRIM (SCSI – UNMAP): The OS tells the SSD that specific LBAs are invalid and may be GCed.
OS support for TRIM (Win 7, OSX Snow Leopard, Linux 2.6.33, Android 4.3).

The size of page-level mapping table is too large: With a 1TB SSD with a 4byte entry per 4KB page, 1GB of is needed for mapping.
Some approaches to reduce the costs of mapping:

- Block-based mapping: Coarser grain approach.
  Mapping at block granularity, to reduce the size of a mapping table. Small write problem: the FTL must read a large amount of live data from the old block and copy them into a new one.

- Hybrid mapping : Multiple tables.
  FTL maintains two tables:

  – Log blocks: page mapped
  – Data blocks: block-mapped

  When looking for a particular logical block, the FTL will consult the page mapping table and block mapping table in order

- Page mapping plus caching : Exploiting Data Locality.
  Basic idea is to cache the active part of the page-mapped FTL, f a given workload only accesses a small set of pages, the translations of those pages will be stored in the FTL memory. High performance without high memory cost if the cache can contain the necessary working set. Cache miss overhead. exists

Wear Leveling
Erase/Write cycle is limited in Flash memory:
- Skewness in the EW cycles shortens the life of the SSD
- All blocks should wear out at roughly the same time
Log-Structured approach and garbage collection helps in spreading writes. However, a block may consist of cold data

- the FTL must periodically read all the live data out of such blocks and re-write it elsewhere
- Wear leveling increases the write amplification of the SSD and decreases performance (Simple Policy: Each Flash Block has EW cycle counter, Maintain —Max(EW cycle) – Min(EW cycle)—< e)

4.2.2 SSD summary

- They cost more than the conventional HDD
- Flash memory can be written only a limited number of times (wear):
  have a shorter lifetime
  error correcting codes
  over-provisioning (add some spare capacity)
- Different read/write speed
  Write amplification
- Write performance degrades of one order of magnitude after the first writing
- Often the controller become the real bottleneck to the transfer rate
- SSD are not affected by data-locality and must not be defragmented (actually, defragmentation may damage the disks)
- Flash Translation Layer is one of the key components
  - Data Allocation
  - Address Translation
  - Garbage Collection
  - Wear Leveling

4.3 HDD vs SSD

Unrecoverable Bit Error Ratio (UBER): A metric for the rate of occurrence of data errors, equal to the number of data errors per bits read
Endurance rating : Terabytes Written (TBW is the total amount of data that can be written into an SSD before it is likely to fail). The number of terabytes that may be written to the SSD while still meeting the requirements

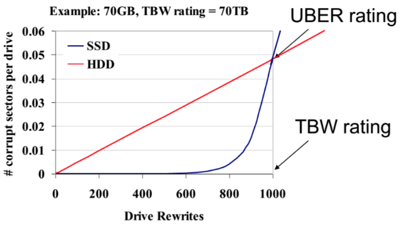


Figure 16: Garbage collection (Example)

Memory cells can accept data recording between 3,000 and 100,000 during its lifetime: once the limit value is exceeded, the cell ”forgets” any new data

A typical TBW for a 250 GB SSD is between 60 and 150 Terabytes of data written to the drive; This means that in order to overcome, for example, a TBW of 70 Terabytes, a user should write 190 GB every day for a year or fill his SSD on a daily basis for two thirds with new files for a whole year

It is difficult to comment on the duration of SSDs: Dell, in an old study (2011), spoke of an estimated duration between three months and ten years explaining, however, that there are so many factors (temperature and workload) that may depend on the life of an SSD that is very difficult to make predictions.

4.4 Hybrid solution

HDD + SSD

- Some large storage servers use SSD as a cache for several HDD. Some mainboards of the latest generation have the same feature: they combine a small SSD with a large HDD to have a faster disk.
- Some HDD manufacturers produce Solid State Hybrid Disks (SSHD) that combine a small SDD with a large HDD in a single unit.

4.5 Storage system

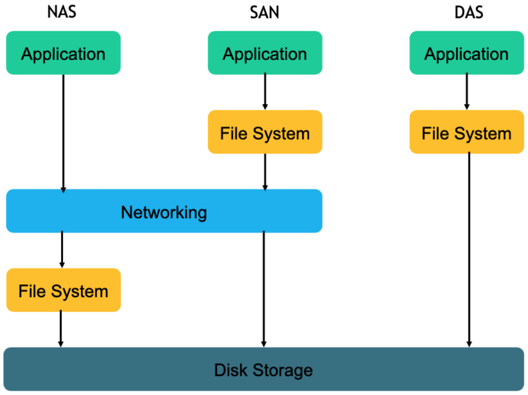


Figure 17: NAS SAN DAS

Definition 3. *A* **Direct Attached Storage** *(DAS) is a storage system directly attached to a server or workstation. They are visible as disks/volumes by the client OS.*

Definition 4. *A* **Network Attached Storage** *(NAS) is a computer connected to a network that provides only file-based data storage services (e.g., FTP, Network File System and SAMBA) to other devices on the network and is visible as File Server to the client OS*

Definition 5. Storage Area Networks *(SAN) are remote storage units that are connected to a server using a specific networking technology (e.g., Fiber Channel) and are visible as disks/volumes by the client OS Direct Attached Storage (DAS) is a storage system directly*

4.5.1 DAS

DAS is a storage system directly attached to a server or workstation
The term is used to differentiate non-networked storage from SAN and NAS (that will be described later).
Main features:

- limited scalability (for ex. if I want 100 server connected, I have to multiply x100 the capacity)
- complex management
- to read files in other machines, the “file sharing” protocol of the OS must be used (on application laver, if I have one Windows and one Linux they must be able to communicate)

Internal and external:

- DAS does not necessary mean “internal drives” (it could have an external hard disk)
- All the external disks, connected with a point-to-point protocol to a PC can be considered as DAS

4.5.2 NAS

A NAS unit is a computer connected to a network that provides only file-based data storage services to other devices on the network.
NAS systems contain one or more hard disks, often organized into logical redundant storage containers or RAID.
Provide file-access services to the hosts connected to a TCP/IP network though Networked File Systems/SAMBA.
Each NAS element has its own IP address.
Good scalability (incrementing the devices in each NAS element or incrementing the number of NAS elements).

NAS vs DAS
The key differences between direct-attached storage (DAS) and NAS are:

- DAS is simply an extension of an existing server and is not necessarily networked
- NAS is designed as an easy and self-contained solution for sharing files over the network

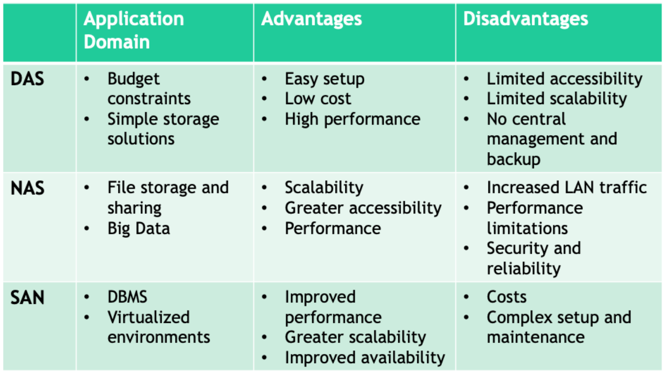
Comparing NAS with local (non-networked) DAS, the *performance* of NAS depends mainly on the speed of and congestion on the network

4.5.3 SAN

Storage Area Networks, are remote storage units that are connected to a PC/server using a specific networking technology.
SANs have a special network devoted to the accesses to storage devices.
Two distinct networks (one TCP/IP and one dedicated network, e.g., Fber Channer).
High scalability (simply increasing the storage devices connected to the SAN network).

NAS vs SAN

- NAS provides both storage and a file system
- This is often contrasted with SAN which provides only block-based storage and leaves file system concerns on the ”client” side
- One way to loosely conceptualize the difference between a NAS and a SAN is that:
  - NAS appears to the client OS (operating system) as a file server (the client can map network drives to shares on that server)
  - a disk available through a SAN still appears to the client OS as a disk: it will be visible in the disks and volumes management utilities (along with client’s local disks), and available to be formatted with a file system
- NAS is used for low-volume access to a large amount of storage by many users
- SAN is the solution for petabytes (1012) of storage and multiple, simultaneous access to files, such as streaming audio/video



	Application Domain	Advantages	Disadvantages
DAS	• Budget constraints • Simple storage solutions	• Easy setup • Low cost • High performance	• Limited accessibility • Limited scalability • No central management and backup
NAS	• File storage and sharing • Big Data	• Scalability • Greater accessibility • Performance	• Increased LAN traffic • Performance limitations • Security and reliability
SAN	• DBMS • Virtualized environments	• Improved performance • Greater scalability • Improved availability	• Costs • Complex setup and maintenance

Figure 18: NAS vs SAN vs DAS

5 Networking

Communication equipment allows network interconnections among the devices.
They can be:

- Hubs
- Routers
- DNS or DHCP servers
- Load balancers
- Technology switches
- Firewalls

The performance of servers increases over time, the demand for inter-server bandwidth naturally increases as well!
We can double the aggregate compute capacity or the aggregate storage simply by doubling the number of compute or storage elements, but how?
Networking has no straightforward horizontal scaling solution.
Doubling leaf bandwidth is easy: with twice as many servers, we'll have twice as many network ports and thus twice as much bandwidth.

But if we assume that every server needs to talk to every other server, we need to deal with **bisection bandwidth**.

Definition 6. *The bandwidth across the narrowest line that equally divides the cluster into two parts. Characterizes network capacity since randomly communicating processors must send data across the "middle" of the network*

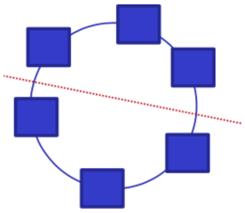


Figure 19: bisection bandwidth

5.0.1 Architecture

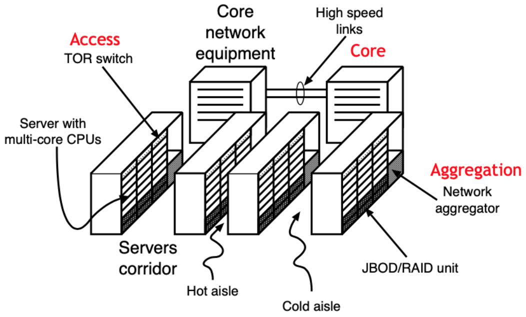


Figure 20: Data-center network architectures

Switches at the access layer can be put into two positions:

- *Top-Of-the-Rack* (TOR): **Access** switches are put at the top of each rack. The number of cables is limited. The number of ports per switch is also limited (lower costs). However, the scalability is also limited.
- *End-Of-the-Line* (EOL): Switches are positioned one per corridor, at the end of a line of rack. **Aggregation** switches must have a larger number of ports, and longer cables are required (higher costs). However, the system can scale to have a larger number of machines.

Three layer architecture configures the network in three different layers:

- core
- aggregation
- access

Bandwidth can be increased by increasing the switches at the core and aggregation layers, and by using routing protocols such as Equal Cost Multiple Path (ECMP) that equally shares the traffic among different routes.

This solution is very simple, but can be very expensive in large data-centers since:
• Upper layers require faster network equipments. (For example: 1 GB Ethernet at the access layer, 10 GB Ethernet at the aggregation layer, 25 GB Optical connections at the core layer)
• The cost in term of acquisition and energy consumption can be very high
Another benefit of SANs: a way to tackle network scalability, offload some traffic to a special-purpose network connecting servers to storage units

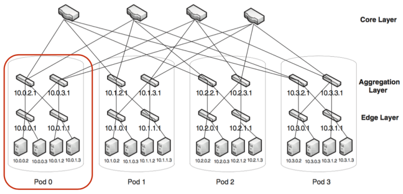


Figure 21: Fat-tree topologies

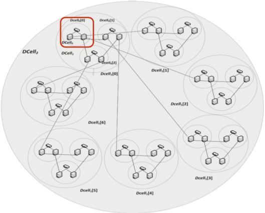


Figure 22: D-Cell topology

Fat-tree topologies use a larger number of slower speed switches and connections. In particular, nodes are divided into pods characterized by the same number of nodes and switches.

D-Cell topology, defines the network in recursive way. Cells are organized in levels. Switches connects nodes at the lower level.Some nodes belong to different cells: they perform routing among them to create a higher level cell.

5.0.2 High Performance Clusters

High-Performance Computing (HPC) supercomputer clusters often have a much lower ratio of computation to network bandwidth.

Applications (e.g., weather simulations) distribute their data across RAM in all nodes, and nodes need to update neighboring nodes after performing relatively few floating-point computations.
Traditional HPC systems have used proprietary interconnects with leading-edge link bandwidths, much lower latencies, and some form of a global address space (where the network is integrated with CPU caches and virtual addresses)

High **throughputs** but much more expensive solutions.

5.0.3 Network to support Virtualization

Connection endpoints (i.e., IP address/port combinations) can move from one physical machine to another
Typical networking hardware as well as network management software don't anticipate such moves and in fact often explicitly assume that they're not possible
(All machines in a given rack have IP addresses in a common subnet, which simplifies administration and minimizes the number of required forwarding table entries routing tables)
Solution: the cluster manager that decides the placement of computations also updates the network state through programmable Network control planes (Software Defined Networks)

5.0.4 The interplay of storage and networking technology

The success of WSC distributed storage systems can be partially attributed to the evolution of data center networking fabrics:*disk locality is no longer relevant in intra-data center computations.*
This observation enables:

- simplifications in the design of distributed disk-based storage systems
- utilization improvements

since any disk byte in a WSC facility can, in principle, be utilized by any task regardless of their relative locality

5.0.5 Balanced design

Computer architects are trained to solve the problem of finding the right combination of performance and capacity from the various building blocks that make up a WSC

The right building blocks are apparent only when one considers the entire WSC system.
It is important to characterize the kinds of workloads that will execute on the system with respect to their consumption of various resources.
Keeping in mind three important considerations:

1. Smart programmers may be able to restructure their algorithms to better match a more inexpensive design alternative
2. The most cost-efficient and balanced configuration for the hardware may be a match with the combined resource requirements of multiple workloads
3. Fungible resources tend to be more efficiently used

System balance: **Storage Hierarchy**

Example

We assume a system with 5,000 servers, each with 256 GB of DRAM, one 4 TB SSD, and eight 10 TB disk drives.
Each group of 40 servers is connected through a 40-Gbps link to a rack-level switch (TOR)
Each rack has an additional 10-Gbps uplink bandwidth per machine for connecting the rack to the cluster-level switch (AGGREGATION)
Network latency numbers assume a TCP/IP transport, and networking bandwidth values assume that each server is using its fair share of the available cluster-level bandwidth.
For disks, typical commodity disk drive (SA T A) latencies and transfer rates are considered.

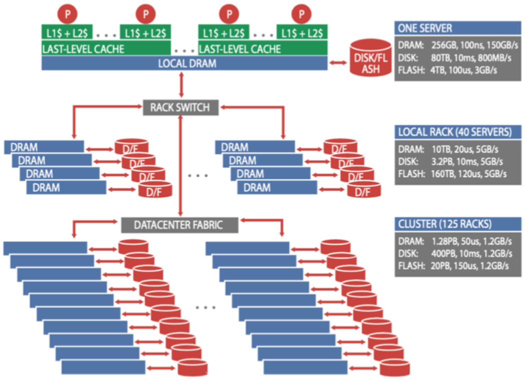


Figure 23: Storage Hierarchy

A large application that requires servers in many racks to operate must deal effectively with large discrepancies in latency, bandwidth, and capacity.

These discrepancies are much larger than those seen on a single machine, making it more difficult to program a WSC:

- A key challenge for architects of WSCs is to smooth out these discrepancies in a cost-efficient manner
- A key challenge for software architects is to build SW infrastructure and services that hide this complexity

Three specific comments about SSDs:

1. Much faster than HDDs
2. Demand a high bandwidth
3. In the worst case, writes to flash can be several orders of magnitude slower than reads