# Computing Infrastructures - Notes

260236

May 2024

# Preface

Every theory section in these notes has been taken from two sources:

- The Datacenter as a Computer: Designing Warehouse-Scale Machines, Third Edition. [1]
- Quantitative System Performance: Computer System Analysis Using Queueing Network Models. [2]

About:

GitHub repository

# Contents

# 1 Introduction: definition of Data Center and Computing Infrastructure

There's no single definition of a Data Center, but it can be summarized as follows.

**Definition 1**

**Data Center**s are buildings where multiple servers and communication gear are co-located because of their common environmental requirements and physical security needs, and for ease of maintenance. [1]

**Definition 2**

A **Computing Infrastructure** (or IT Infrastructure) is a technological infrastructure that provides hardware and software for computation to other systems and services.

Traditional data centres have the following characteristics:

- **Host a large number** of relatively small or medium sized **applications**;
- Each **application is running on a dedicated HW 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**.

# 2 Hardware Infrastructures

## 2.1 System-level

### 2.1.1 Computing Infrastructures and Data Center Architectures

**Overview of Computing Infrastructures**

A number of computing infrastructures exist:

- **Cloud** offers virtualized computing, storage and network resources with highly-elastic capacity.
- **Edge Servers** are on-premises hardware resources that perform more compute-intensive data processing.

  In other words, an edge server is a piece of hardware that performs data computation at the end (or "edge") of a network. Like a regular server, an edge server can provide compute, networking, and storage functions.[1]
- **IoT** and **AI-enabled Edge Sensors** are hardware devices where the data acquisition and partial processing can be performed at the edge of the network.

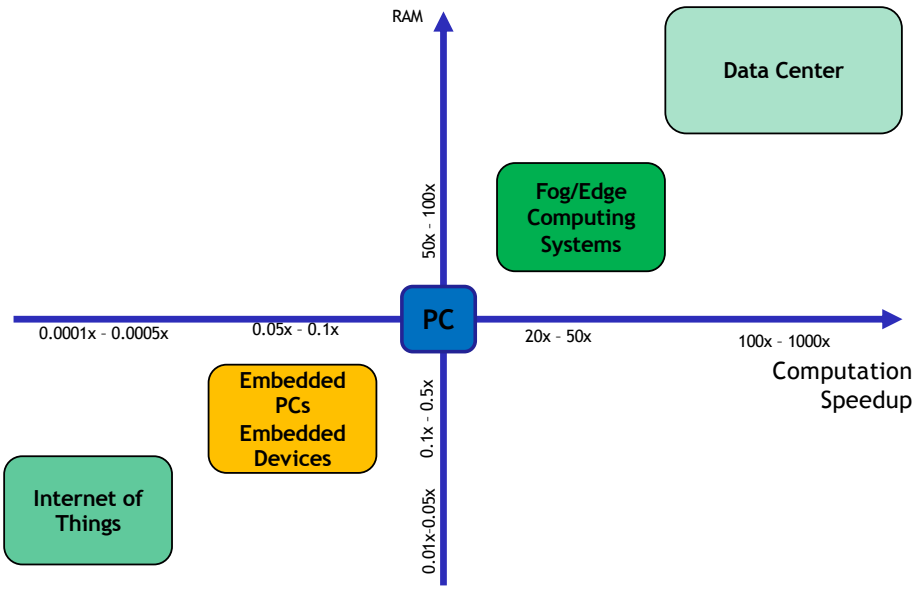
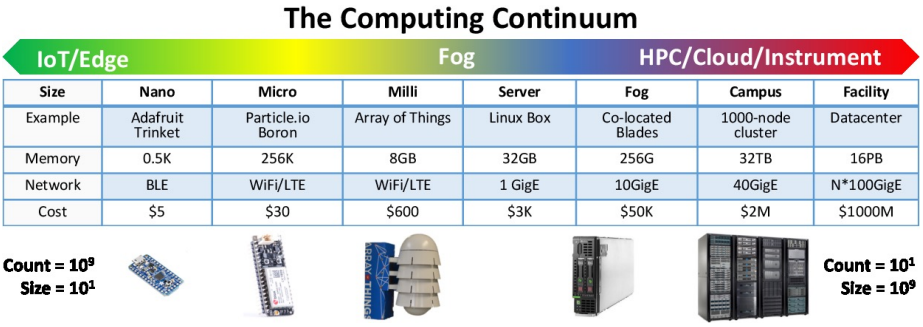



Figure 1: An **example** of Computing Infrastructures. [4]

The **Computing Continuum**, a novel paradigm that extends beyond the current silos of cloud and edge computing, can enable the seamless and dynamic deployment of applications across diverse infrastructures. [3]

[1] More info here.




| Size | Nano | Micro | Milli | Server | Fog | Campus | Facility |
|---|---|---|---|---|---|---|---|
| Example | Adafruit Trinket | Particle.io Boron | Array of Things | Linux Box | Co-located Blades | 1000-node cluster | Datacenter |
| Memory | 0.5K | 256K | 8GB | 32GB | 256G | 32TB | 16PB |
| Network | BLE | WiFi/LTE | WiFi/LTE | 1 GigE | 10GigE | 40GigE | N*100GigE |
| Cost | $5 | $30 | $600 | $3K | $50K | $2M | $1000M |

Figure 2: The Computing Continuum. [4]

In the following pages, we analyze the computing infrastructures mentioned in the previous example.

---

## Data Centers

The definition of a Data Centers can be found on page 4.

### ✔ Data Centers Advantages

- **Lower IT costs.**
- **High Performance.**
- **Instant software updates.**
- **"Unlimited" storage capacity.**
- **Increased data reliability.**
- **Universal data access.**
- **Device Independence.**

### 👎 Data Centers 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.**
- **Latency in taking decision.**

## Internet-of-Things (IoT)

An **Internet of Things (IoT)** **device is any everyday object embedded with sensors, software, and internet connectivity**.

This allows to collect and exchange data with other devices and systems, typically over the internet, with limited need of process and store data.

Some **examples** are Arduino, STM32, ESP32, Particle Argon.

### ✔ Internet-of-Things Advantages

- **Highly Pervasive**.
- **Wireless connection**.
- **Battery Powered**.
- **Low costs**.
- **Sensing and actuating**.

### 👎 Internet-of-Things Disadvantages

- **Low computing ability**.
- **Constraints on energy**.
- **Constraints on memory (RAM/FLASH)**.
- **Difficulties in programming**.

---

## Embedded (System) PCs

An **Embedded System** is a computer system, a combination of a computer processor, computer memory, and input/output peripheral devices, that has a dedicated function within a larger mechanical or electronic system.

A few **examples**: Odroid, Raspberry, jetson nano, Google Coral.

### ✔ Embedded System Advantages

- **Persuasive computing**.
- **High performance unit**.
- **Availability of development boards**.
- **Programmed as PC**.
- **Large community**.

### 👎 Embedded System Disadvantages

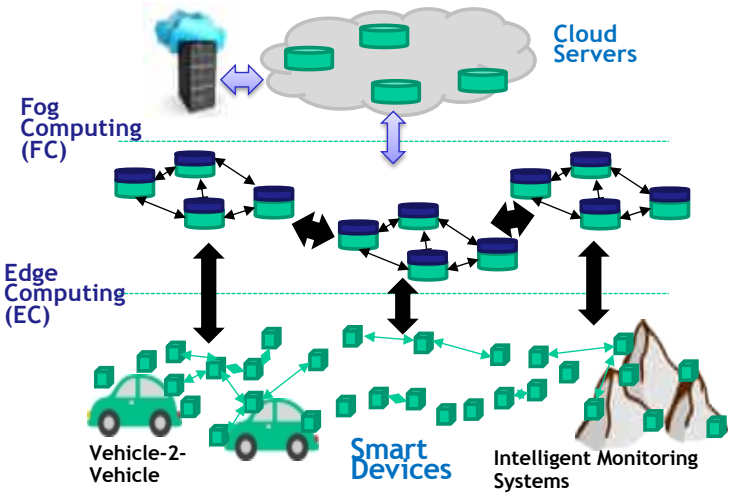
- **Pretty high power consumption**.
- **(Some) Hardware design has to be done**.

## Edge/Fog Computing Systems

The key **difference** between **Fog Computing** and **Edge Computing** is associated with the location **where the data is processed**:

- In **edge computing**, the data is processed closest to the sensors.
- In **fog computing**, the computing is moved to processors linked to a local area network (IoT gateway).

Edge computing places the intelligence in the connected devices themselves, whereas, fog computing puts in the local area network.




### ✔ Fog/Edge Advantages

- **High computational capacity**.
- **Distributed computing**.
- **Privacy and security**.
- **Reduced Latency in making a decision**.

### 👎 Fog/Edge Disadvantages

- **Require a power connection**.
- **Require connection with the Cloud**.

| Feature | Edge Computing | Fog Computing |
|---|---|---|
| **Location** | Directly on device or nearby device. | Intermediary devices between edge and cloud. |
| **Processing Power** | Limited due to device constraints, sending data to central server for analysis. | More powerful than edge devices. However, sending data to a central server for analysis. |
| **Primary Function** | Real-time decision-making, low latency. However, central server analyzing combined data and sending only relevant information further. | Pre-process and aggregate data, reduce bandwidth usage. However, central server analyzing combined data and sending only relevant information further. |
| **Advantages** | Low latency, reduced reliance on cloud, security for sensitive data. | Bandwidth efficiency, lower cloud costs, complex analysis capabilities. |
| **Disadvantages** | Limited processing power, single device focus. | Increased complexity, additional infrastructure cost. |

Table 1: Differences between Edge and Fog Computing Systems.

## The Datacenter as a Computer

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

The **advantages** of this migration are:

- **User-side**:
  - **Ease of management** (no configuration or backups needed);
  - The **availability of the service is everywhere**, but we need connectivity.
- **Vendors-side**:
  - SaaS (Software-as-a-Service) allows **faster application development** (more accessible to make changes and improvements);
  - **Improvements and fixes** in the software are **more straightforward 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**:
  - **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**.

Finally, another advantage is that **some workloads require so much computing capability that they are a more natural fit in the datacenter** (and not in client-side computing). For example, the search services (web, images, and so on) or the Machine and Deep Learning.

## Warehouse-Scale Computers

The trends toward server-side computing and widespread internet services created a new class of computing systems: **Warehouse-Scale Computers**.

**Definition 1**

**Warehouse-Scale Computers (WSCs)** is intended to draw attention to the most distinctive feature of these machines: **the massive scale of their software infrastructure, data repositories and hardware platform**.

### What is a *program* at a WSC?

In Warehouse-Scale Computing **the program is an internet service**, which may **consist of tens or more individual programs that interact to implement complex end-user services** such as *email*, *search*, or *maps*. These programs might be implemented and maintained by different teams of engineers, perhaps even across organizational, geographic, and company boundaries.

### Difference between WSCs and Data Centers

WSCs currently power the services offered by companies such as Google, Amazon, Microsoft, and others. The main difference from traditional data centers (see more on page 4) is that **WSCs belong to a single organization, use a relatively homogeneous hardware and system software platform, and share a common systems management layer**. In contrast with the typical data center that belongs to multiple organizational units or even different companies, use dedicated HW infrastructure in order to run a large number of applications (more details on page 4).

### How is the WSC organized?

The **software on WSCs**, such as Gmail, runs on a scale far beyond a single machine or rack: it **runs on clusters of hundreds to thousands of individual servers**. Therefore, the machine, the computer, is itself this **large cluster or aggregation of servers** and must be **considered a single computing unit**.

Most importantly, WSCs run fewer vast applications (internet services). An **advantage** is that the **shared resource management infrastructure allows significant deployment flexibility**. Finally, the requirements of:

- **Homogeneity**
- **Single-Organization Control**
- **Cost Efficiency**

Motivate designers to take new approaches to constructing and operating these systems.

## Multiple Data Centers

Sometimes the data centers are **located far apart**. **Multiple data centers** are (often) **replicas of the same service**:

- To *reduce user latency*
- To *improve service throughput*

Typically, a request is fully processed within one data center.

The world is divided into **Geographic Areas (GAs)**. Each Area is defined by Geo-political boundaries (or country borders). Also, there are at least two computing regions in each geographical Area.

The **Computing Regions (CRs)** are the smallest geographic unit of the infrastructure from the customer's perspective. Multiple Data centers within the same region are not exposed to customers.

However, they are defined by a latency-defined perimeter, typically less than 2ms for round-trip latency. Finally, they're located hundreds of miles apart, with considerations for different flood zones, etc. It is too far from synchronous replication but suitable for disaster recovery.

The **Availability Zones (AZs)** are finer-grain **locations within a single computing region**. They allow customers to run mission-critical applications with high availability and fault tolerance to Data Center failures. Because there are fault-isolated locations with redundant power, cooling, and networking (they are different from the concept of the Availability Set).

This hierarchical structure ensures efficient data management and compliance with local data laws while optimizing network performance through strategically placing data centers.

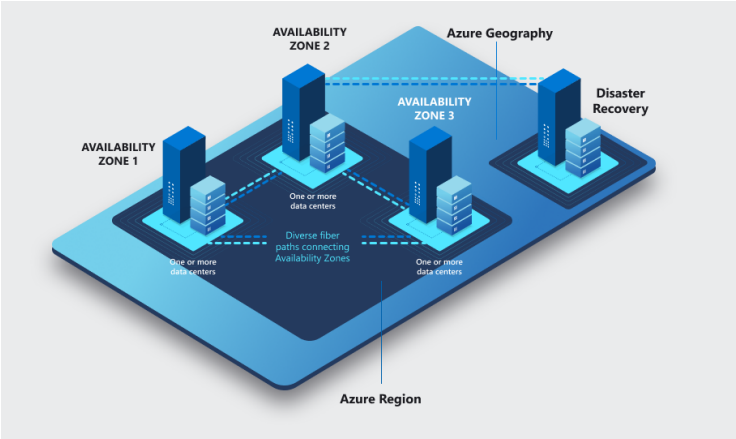



Figure 3: Example of Azure Availability Zones.

## Warehouse-Scale Computing / Data Centers Availability

The services provided through WSCs (or DCs) **must guarantee high availability**, typically aiming for at least 99.99% uptime (e.g. one hour of downtime per year).

Some **examples**:

- 99,90% on single instance VMs with premium storage for a more accessible lift and shift;
- 99,95% VM uptime SLA for Availability Sets (AS) to protect for failures within a data center;
- 99,99% VM uptime SLA through Availability Zones.

Such fault-free operation is more accessible when an extensive 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!**

---

## Architectural overview of Warehouse-Scale Computing

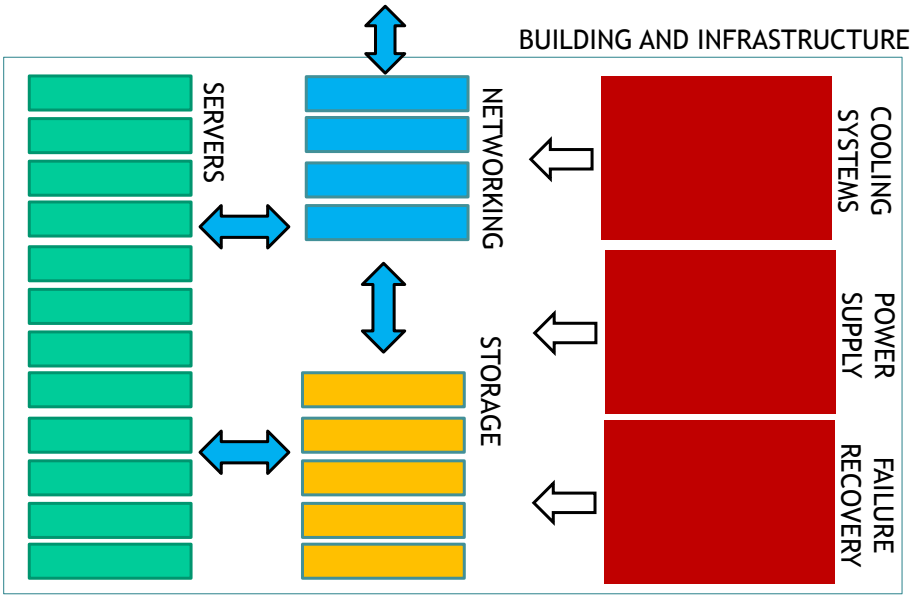



Figure 4: Architectural overview of Warehouse-Scale Computing.

- **Server**. Servers are the **leading processing equipment**: different types according to CPUs, RAM, local storage, accelerators, and form factor. The servers are **hosted on individual shelves** and are the **basic building blocks of Data Centers and Warehouse-Scale Computers**. They are interconnected by hierarchies of networks and supported by the shared power and cooling infrastructure.

- **Storage**. Disks, flash SSDs, and Tapes 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**.

  Some **examples**:

  - Direct Attached Storage (DAS)
  - Network Attached Storage (NAS)
  - Storage Area Networks (SAN)
  - RAID controllers

- **Networking**. The **Data Center Network (DCN) enables efficient data transfer and interaction between various components**. The data processing ecosystem within the DCs needs to reach the DC services from outside. Communication equipment includes switches, Routers, cables, DNS or DHCP servers, Load balancers, Firewalls, etc.

- **Building and Infrastructure**. WSC has other essential components related to power delivery, cooling, and building infrastructure that must be considered. Some interesting numbers:

  - Data Centers with up to 110 football-pitch size.
  - 2-100s MW power consumption (100k houses), and the largest in the world is 650 MW.

# References


[1] L.A. Barroso, U. Hölzle, and P. Ranganathan. *The Datacenter as a Computer: Designing Warehouse-Scale Machines, Third Edition*. Synthesis Lectures on Computer Architecture. Springer International Publishing, 2022.

[2] E.D. Lazowska. *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*. Prentice-Hall, 1984.

[3] Jacopo Marino and Fulvio Risso. Is the computing continuum already here?, 2023.

[4] Gianluca Palermo. Lesson 1, computing infrastructures. Slides from the HPC-E master's degree course on Politecnico di Milano, 2024.

# Index