CS 4530: Fundamentals of Software Engineering

Module 10.1 Distributed Systems: Goals and Challenges

Adeel Bhutta, Mitch Wand

Khoury College of Computer Sciences

Learning Goals for this Lesson

- At the end of this lesson you should be able to
  - List and define 5 goals of using distributed systems
  - List 4 major challenges inherent in distributed systems

Distributed Systems Goals

- Scalability
- Performance
- Latency
- Availability
- Fault Tolerance

Think of these as **benefits** of going from Monolithic to multi-server system!

“Distributed Systems for Fun and Profit”, Takada

Distributed Systems Goals

- **Scalability**
- Performance
- Latency
- Availability
- Fault Tolerance

"the ability of a system, network, or process, to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth."

"Distributed Systems for Fun and Profit", Takada

Distributed Systems Allow Horizontal Scaling

- “Vertical” scaling: add more resources to existing server
  - Faster CPUs, more CPU cores, more RAM, more storage
  - Becomes ineffective : Clock speed plateaus; difficult to write applications that utilize 256 CPU cores (though adding 2TB RAM to a server *can* often help)
- “Horizontal” scaling: add more servers
  - Rely on “commodity” servers rather than state-of-the-art hardware
  - Allows for dynamic addition of resources as needed by load

Distributed Systems Goals

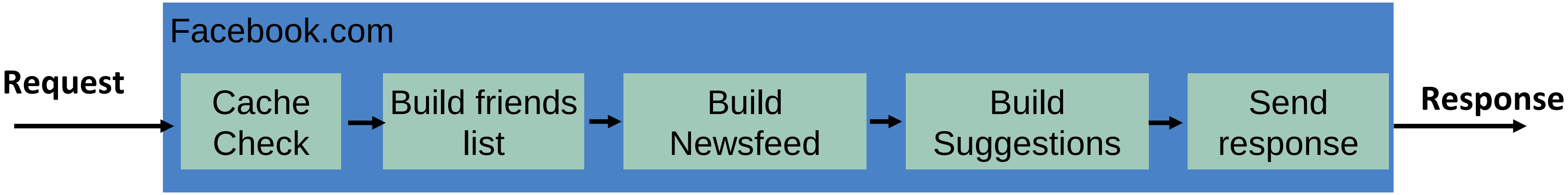
- Scalability
- **Performance**
- Latency
- Availability
- Fault Tolerance

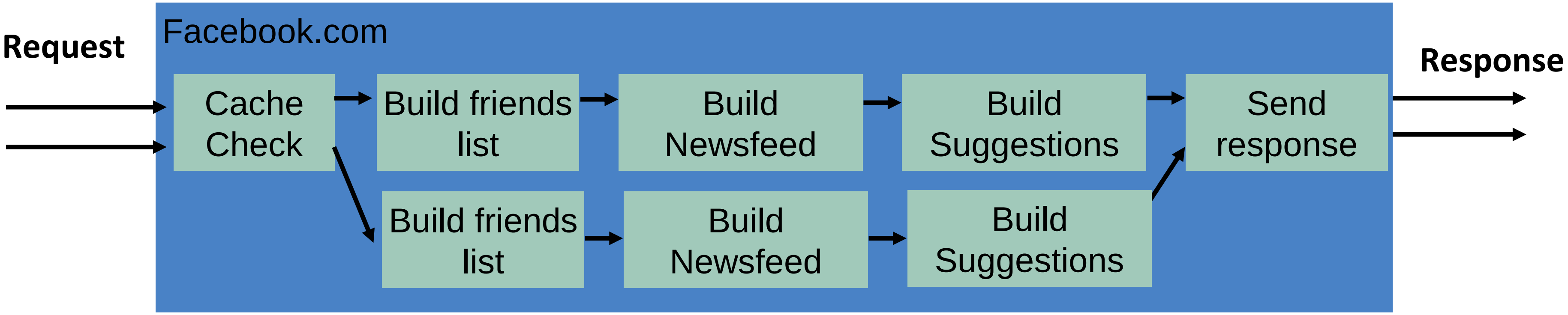
"The amount of useful work accomplished by a computer system compared to the time and resources used."

"Distributed Systems for Fun and Profit", Takada

Multiple Servers Can Improve Throughput With Concurrency

Throughput: total requests that can be processed per unit-time





Distributed Systems Goals

- Scalability
- Performance
- **Latency**
- Availability
- Fault Tolerance

The time between a user making a request and receiving their response.

In a monolithic-server system, we can work on improving response time.

In a multi-server system, we can select a server that is closer to the user.

[“Distributed Systems for Fun and Profit”, Takada]

Reduce latency by distributing data

- Move or replicate the data
  - Decrease transmission time
  - Avoid bottlenecks

[“Distributed Systems for Fun and Profit”, Takada]

Distributed Systems Goals

- Scalability
- Performance
- Latency
- **Availability**
- Fault Tolerance

"the proportion of time a system is in a functioning condition."

Availability = uptime / (uptime + downtime).

Often measured in "nines"

Availability %	Downtime/year
90%	>1 month
99%	< 4 days
99.9%	< 9 hours
99.99%	<1 hour
99.999%	5 minutes
99.9999%	31 seconds

"Distributed Systems for Fun and Profit", Takada

Distributed Systems can improve availability by replicating servers

- A single-server system is either up or down.
- If you have many servers, the probability that *some* server is down increases
- BUT: the probability that **all** servers are down decreases (exponentially!)

replication helps exponentially!

Distributed Systems Goals

- Scalability
- Performance
- Latency
- Availability
- **Fault Tolerance**

"ability of a system to behave in a well-defined manner once faults occur"

A distributed system can respond to faults because it has redundancy

"Distributed Systems for Fun and Profit", Takada

Design to expect faults

- "Define what faults you expect and then design a system or an algorithm that is tolerant of them. You can't tolerate faults you haven't considered."

A distributed system can be agile. We expect that there will be failures, and our system can be designed to handle them.

What kind of faults?

Disks fail

Networking fails

Power supplies fail

Security breached

Power goes out

Datacenter goes offline

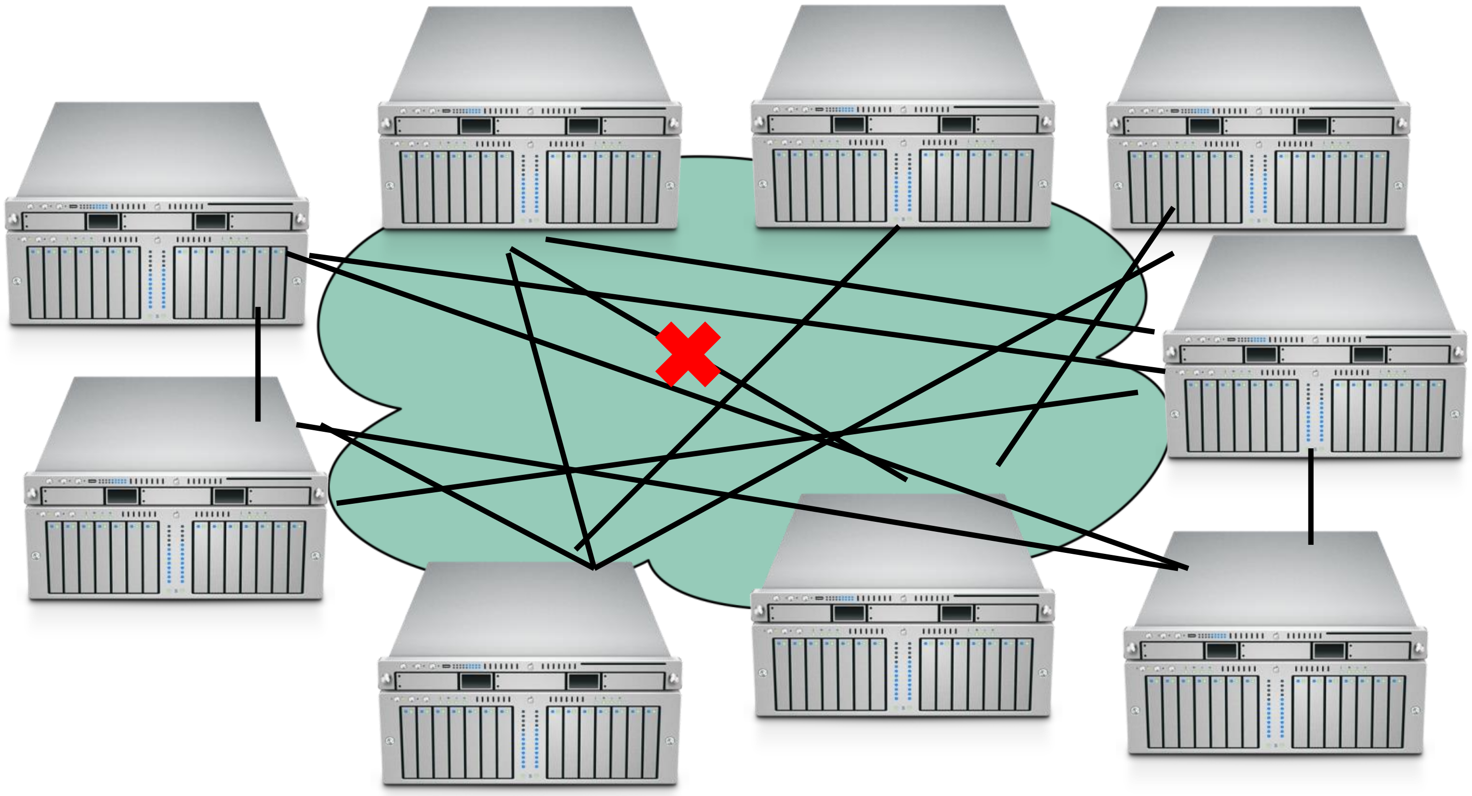
"Distributed Systems for Fun and Profit", Takada

Distributed Systems Challenges

- Reaping benefits from a distributed system isn’t always easy! Like everything else in life, it comes with challenges

More machines means more links that might fail.

- Number of nodes + distance between them



Networks introduce delays

- Cannot expect network to be a perfect analog for communication within a single computer because:
  - Speed of light (1 foot/nanosecond)
  - Communication links exist in uncontrolled/hostile environments
  - Communication links may be bandwidth limited (tough to reach even 100MB/sec)
- In contrast to a single computer, where:
  - Distances are measured in mm, not feet
  - Physical concerns can be addressed all at once
  - Bandwidth is plentiful (easily GB/sec)

We rely on other administrators, who are not infallible

2024 CrowdStrike-related IT outages

Article Talk

From Wikipedia, the free encyclopedia

On 19 July 2024, American cybersecurity company CrowdStrike distributed a faulty update to its Falcon Sensor security software that caused widespread problems with Microsoft Windows computers running the software. As a result, roughly 8.5 million systems crashed and were unable to properly restart[1] in what has been called the largest outage in the history of information technology[2] and "historic in scale".[3]

The outage disrupted daily life, businesses, and governments around the world. Many industries were affected—airlines, airports, banks, hotels, hospitals, manufacturing, stock markets, broadcasting, gas stations, retail stores, and more—as were governmental services, such as emergency services and websites.[4][5] The worldwide financial damage has been estimated to be at least US$10 billion.[6]

Within hours, the error was discovered and a fix was released,[7] but because many affected computers had to be fixed manually,[8] outages continued to linger on many services.[9][10]

2024 CrowdStrike-related IT outages

Multiple blue screens of death caused by a faulty software update on baggage carousels at LaGuardia Airport, New York City

Date	19 July 2024; 6 months ago
Location	Worldwide
Type	IT outage, computer crash
Cause	Faulty CrowdStrike software update
Outcome	~8.5 million Microsoft Windows operating systems crash worldwide, causing global disruption of critical services

… often leading to common-fault failures

2024 CrowdStrike-related IT outages

Article Talk

From Wikipedia, the free encyclopedia

On 19 July 2024, American cybersecurity company CrowdStrike distributed a faulty update to its Falcon Sensor security software that caused widespread problems with Microsoft Windows computers running the software. As a result, roughly 8.5 million systems crashed and were unable to properly restart[1] in what has been called the largest outage in the history of information technology[2] and "historic in scale".[3]

The outage disrupted daily life, businesses, and governments around the world. Many industries were affected—airlines, airports, banks, hotels, hospitals, manufacturing, stock markets, broadcasting, gas stations, retail stores, and more—as were governmental services, such as emergency services and websites.[4][5] The worldwide financial damage has been estimated to be at least US$10 billion.[6]

Within hours, the error was discovered and a fix was released,[7] but because many affected computers had to be fixed manually,[8] outages continued to linger on many services.[9][10]

2024 CrowdStrike-related IT outages

Multiple blue screens of death caused by a faulty software update on baggage carousels at LaGuardia Airport, New York City

Date	19 July 2024; 6 months ago
Location	Worldwide
Type	IT outage, computer crash
Cause	Faulty CrowdStrike software update
Outcome	~8.5 million Microsoft Windows operating systems crash worldwide, causing global disruption of critical services

Learning Goals for this Lesson

- You should now be able to
  - List and define 5 goals of using distributed systems
  - List 4 major challenges inherent in distributed systems