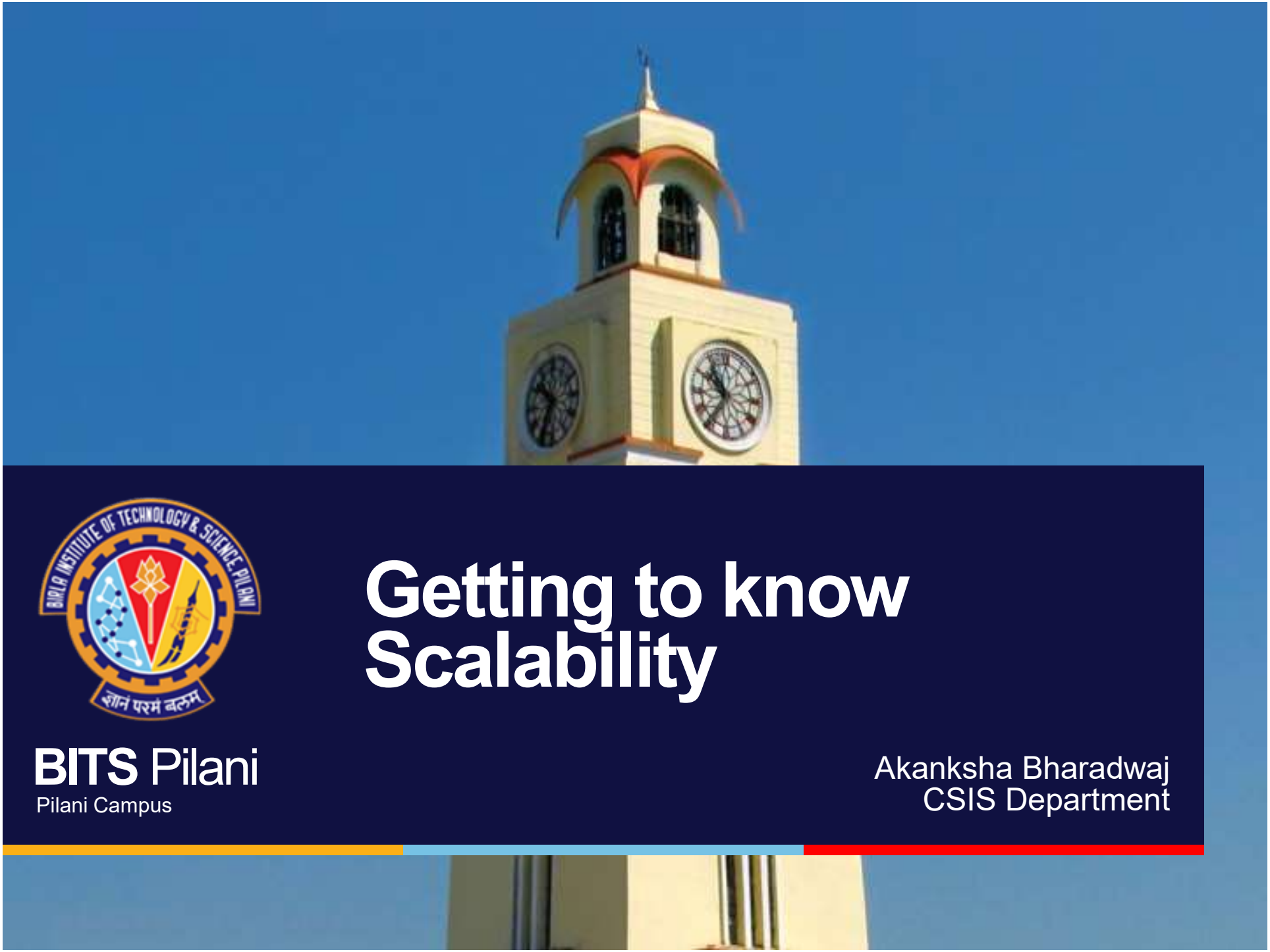

Getting to know Scalability

BITS Pilani
Pilani Campus

Akanksha Bharadwaj
CSIS Department

innovate achieve lead

BITS Pilani
Pilani Campus

SE ZG583, Scalable Services
Lecture No. 1

innovate achieve lead

BITS Pilani
Pilani Campus

Introduction to Performance, Consistency and availability

CAP Theorem

- The **CAP theorem** states that a distributed system can only guarantee two out of these three characteristics: Consistency, Availability, and Partition Tolerance.

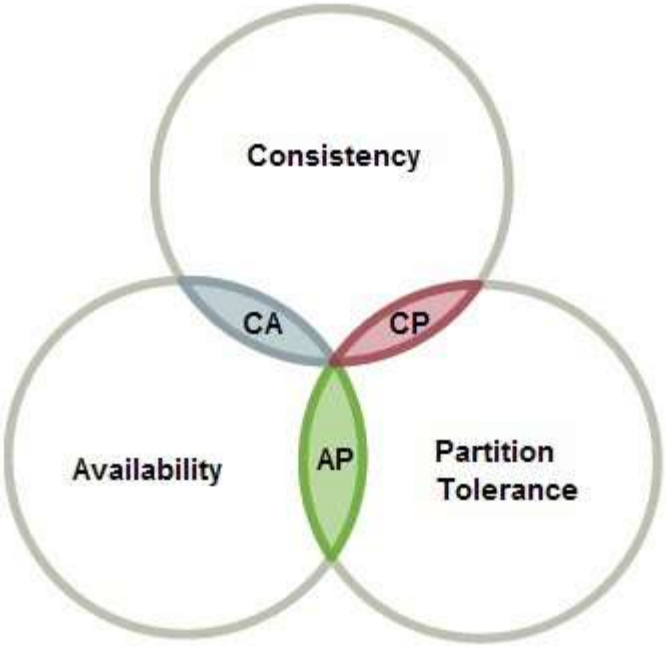

Image: Towardsdatascience

CAP Theorem

Consistency

Availability

Partition tolerance

Eventual Vs Strong Consistency

- Eventual consistency is a consistency model that enables the data store to be highly available.
- Strong Consistency simply means that all the nodes across the world should contain the same value for an entity at any point in time.

Performance

- The topmost reason for performance concerns is that the tasks we set our systems to perform have become much more complex over a period of time

Availability of a system

- Availability refers to a property of software that it is there and ready to carry out its task when you need it to be.
- Here are some of the key resources you can implement to make high availability possible:
  - Use multiple application servers
  - Spread out physically
  - Backup system

  .....

What is scalability?

- Scalability of an architecture refers to the fact that it can scale up to meet increased work loads.
- Types of Scalability
  - Vertical Scalability
  - Horizontal Scalability

Need for scalable architectures

Monolithic Architecture

- Monolith means composed all in one piece.
- Traditionally, applications were built on a monolithic architecture, a model in which all modules in an application are interconnected in a single, self-contained unit.
- They’re typically complex applications that encompass several tightly coupled functions.
- When all functionality in a system had to be deployed together, we consider it a **monolith**.

Advantages of Monolith

- Simplicity
- Network latency and security

Disadvantages of Monolith

- Scalability
- Slow development
- Long deployment cycle

Principles of Scalability

- Avoid single point of failure
- Scale horizontally, not vertically
- API
- Cache
- Maintenance and automation
- Asynchronous

All these mainly target three areas **Availability, Performance, and Reliability**

Guidelines for Building Highly Scalable Systems

- Avoid shared resources as they might become a bottleneck
- Avoid slow services
- Scaling Data tier is tricky
- Cache is the key
- Monitoring is important

Architecture's scalability requirements

- How important are the scalability requirements?
- Identify the scalability requirements early in the software life cycle so that that it allows the architectural framework to become sound enough as the development proceeds.
- System scalability criteria could include the ability to accommodate
  - Increasing number of users,
  - Increasing number of transactions per millisecond,
  - Increase in the amount of data

Challenges for Scalability

- Centralized approach
- Synchronous communication
- Cost

innovate achieve lead

BITS Pilani

Pilani Campus

Case Study

YouTube case Study

- YouTube is a video sharing website which uses the Client/server architecture.
- The NetScaler is implemented in the front of the web servers.
- YouTube uses the Apache with mod_fastcgi as the web servers.
- Python is used in the YouTube

Video Servers in YouTube

- In the video streaming, the bandwidth, hardware, and power consumption are three important issues
- YouTube applies cluster method to solve the consumption problem.
- YouTube also uses different strategies to deal with most popular videos and less popular videos.

Databases in YouTube

- Started with MySQL
- A shard architecture is designed to solve the replication problem of MySQL

Self Study

Example of scalable architecture:

https://www.youtube.com/watch?v=VHELcOe1gy0

References

- Textbooks and reference books mentioned in the handout
- https://groups.csail.mit.edu/tds/papers/Gilbert/Brewer2.pdf
- http://highscalability.com/youtube-architecture