Challenges in Distributed Systems

Y. V. Lokeswari AP / CSE

Reference: George Coulouris, Jean Dollimore and Tim Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education, 2012

Challenges in DS

- Heterogeneity
- Openness
- Security
- Scalability
- Failure Handling
- Concurrency
- Transparency
- Quality of Service

Challenges in DS

1. Heterogeneity

- Internet enables users to access services and run applications over a heterogeneous collection of computers and networks.

- Networks.
- Computer hardware.
- Operating systems.
- Programming languages.
- Implementations by different developers.

Middleware

- A Software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, operating systems and programming languages. Examples are
- **CORBA** provides remote object invocation, which allows an object in a program running on one computer to invoke a method of an object in a program running on another computer.
- **Java RMI.**

Challenges in DS

1. Heterogeneity

Heterogeneity and mobile code

- Mobile code is used to refer to program code that can be transferred from one computer to another and run at the destination – Example **Java applets**
- **Problem**: Code suitable for running on one computer is not necessarily suitable for running on another computer (different ISA)
- **Solution** : virtual machine approach provides a way of making code executable on a variety of host computers
- Compiler for a particular language generates code for a virtual machine instead of a particular hardware order code
- **Example**: Java compiler produces code for a Java virtual machine

Challenges in DS

2. Openness

- Characteristic that determines **whether** the **system** can be **extended** and **re-implemented** in various ways.
- **Openness in DS**: The **degree** to which **new resource-sharing** services can be **added and be made available** for use by client programs.
- Open systems are characterized by the fact that their **key interfaces** are **published**.
- Open distributed systems are based on the provision of a **uniform communication mechanism** and published interfaces for access to shared resources.
- Open distributed systems can be constructed from **heterogeneous** hardware and software, possibly from different vendors. But the **conformance** of each component to the **published standard** must be carefully tested and verified

Challenges in DS

3. Security

- **Confidentiality** (protection against disclosure to unauthorized individuals).
- **Integrity** (protection against alteration or corruption).
- **Availability** (protection against interference with the means to access the resources).
- A doctor might request access to hospital patient data or send additions to that data. **Solution** : Authentication.
- In electronic commerce and banking, users send their credit card numbers across the Internet. **Solution** : Encryption of messages
- **Denial of service attacks**:
  - Bombarding the service with such a large number of pointless requests that the serious users are unable to use it
- **Security of mobile code**
  - Running a executable code from email attachment may display an interesting picture but in reality it may access local resources

Challenges in DS

4. Scalability

- A system is described as scalable if it will **remain effective** when there is a significant **increase** in the **number** of **resources** and the **number** of **users**
- Controlling the cost of physical resources
- Controlling the performance loss
- Preventing software resources running out
- Avoiding performance bottlenecks
  - Internet addresses held by the Domain Name System, which is used mainly to look up DNS names such as www.amazon.com.
  - Name table kept in single master file.

Challenges in DS

5. Failure Handling

- **Faults** occurring in hardware or software, programs may produce **incorrect results** or may **stop** before they have completed the intended computation.
- Detecting failures
- Masking failures
  - Messages can be retransmitted
  - File data can be written to a pair of disks
- Tolerating failures (Web server informs the user about the problem)
- Recovery from failures (Rollbacks)
- Redundancy (Replicate routes, name table of DNS, DBs)
- Availability (if failure occurs, start server process on another computer)

Challenges in DS

6. Concurrency

- Several **clients** will attempt to access a **shared resource** at the **same time**.
- **Process** that manages a shared resource could **take one client request** at a time.
- But this approach **limits throughput**.
- Any **object** that represents a shared resource in a distributed system must be **responsible** for ensuring that it **operates correctly**.
- Its operations must be **synchronized** in such a way that its data remains **consistent**
- Achieved through **semaphores**.

Challenges in DS

7. Transparency

- **Concealment** from the **user** and the **application programmer** of the **separation** of **components** in a distributed system.

Access transparency: enables local and remote resources to be accessed using identical operations.

Location transparency: enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).

Concurrency transparency: enables several processes to operate concurrently using shared resources without interference between them.

Replication transparency: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Failure transparency: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

Mobility transparency: allows the movement of resources and clients within a system without affecting the operation of users or programs.

Performance transparency: allows the system to be reconfigured to improve performance as loads vary.

Scaling transparency: allows the system and applications to expand in scale without change to the system structure or the application algorithms.

Challenges in DS

8. Quality of Service (QoS)

- **Reliability**
- **Security**
- **Performance**.
- **Adaptability** (to meet changing system configurations and resource availability)

Summary of Challenges

- Heterogeneity
- Openness
- Security
- Scalability
- Failure Handling
- Concurrency
- Transparency
- Quality of Service

Thank You