Design Issues of Distributed Systems

By Estudies4you

Design Issues of Distributed Systems

Designing issues of DS

- Heterogeneity
- Openness
- Security
- Synchronization
- Absence of global clock
- Partial failures
- Scalability
- transparency

Heterogeneity:

- The distributed system contains many different kinds of hardware and software working together in cooperative fashion to solve problems.
- There may be many different representations of data in the system this might include different representations for integers, byte streams, floating point numbers and character sets.
- There may be many different instructions sets. An application compiled for one instruction set cannot be easily run on a computer with another instruction set unless an instruction set interpreter is provided
- Components in the distributed system have different capabilities like faster clock cycles, larger memory capacity, bigger disk farms, printers and other peripherals and different services

High Degree of node heterogeneity:

- High-performance parallel systems (multiprocessors as well as multicomputer)
- High-end PCs and workstations (servers)
- Simple network computers (offer users only network access)
- Mobile computers (palmtops, laptops)
- Multimedia workstations

High degree of network heterogeneity:

- Local area gigabit networks
- Wireless connections
- Long-haul, high-latency connections

Observation: Ideally, a distributed system must hide these differences

Openness:

- The openness of a computer system is the characteristic that determines whether the system can be extended and reimplemented in various ways
- The challenge to designers is to tackle the complexity of distributed systems consisting of many components engineered by different people
- 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

Security

- Shared data must be protected
    - o Privacy - avoid unintentional disclosure of private data
    - o Security – data is not revealed to unauthorized parties
    - o Integrity – protect data and system state from corruption
- Denial of service attacks – put significant load on the system, prevent users from accessing it

Security in detail concerned in the following areas:

· Authentication, Authorization/Access control: are the means to identify the right user and user right

· Critical Infrastructure Protection: CIP is the protection of information systems for critical infrastructures including telecommunications, energy, financial services, manufacturing, water, transportation, health care and emergency services sectors

· Distributed Trust and Policy Management: designed to address the authorization needs for the next-generation distributed systems. A trust management system is a term coined to refer to a unified framework for the speciation of security policies, the representation of credentials, and the evaluation and enforcement of policy compliances

· Multicasting security and IPR Protection: defines the common architecture for multicast security(MSEC) key management protocols to support a variety of application, transport, network layer security protocol and the intellectual property rights

· Multimedia Security: is intended to provide an advanced multimedia application course with its focus on security. two major areas of concern- to ensure secure uses of multimedia data and to use multimedia data for security applications

Ø Object security (OMG/CORBA security, EJB Security, DCOM/COM Security)

Ø Privacy: is it a purely political or moral issue?

· Risk analysis, Assessment, Management: A security policy framework is necessary to support the security infrastructure required for the secure movement of sensitive information across and within national boundaries

Synchronization

· Concurrent cooperating tasks need to synchronize

- o When accessing shared data

- o When performing a common task
- · Synchronization must be done correctly to prevent data corruption:
  - o Example: two account owner; one deposits the money, the other one withdraws; they act concurrently
  - o How to ensure the bank account is in “correct” state after these actions?
- · Synchronization implies communication
- · Communication can take a long time
- · Excessive synchronization can limit effectiveness and scalability of distribute system

Absence of Global Clock

- Cooperating task need to agree on the order of events
- Each task its own notion of time
- Clocks cannot be perfectly synchronized
- How to determine which even occurred first?

Example:
Bank account, starting balance = $100
Client at bank machine A makes a deposit of $150
Client at bank machine B makes a withdrawal of $100
Which event happened first?
Should the bank charge the overdraft fee?

Partial Failures

- Detection of failures - may be impossible
- Has a component crashed? Or is it just show?
- Is the network down? Or is it just slow?
- If it’s slow – how long should we wait?
- Handling of failures
- Re-transmission
- Tolerance for failures
- Roll back partially completed task
- Redundancy against failures
- Duplicate network routes
- Replicated databases

Scalability

- Does the system remain effective as of grows?
- As you add more components:
- More synchronization
- More communication à the system runs slowly.
- Avoiding performance bottlenecks:
- Everyone is waiting for a single shared resource
- In a centrally coordinated system, everyone waits for the co-coordinator

Transparency
Distributed systems designers must hide the complexity of the systems as much as they

can. Adding abstraction layer is particularly useful in distributed systems.
Example: While users hit search in google.com, they never notice that their query goes through a complex process before Google shows them a result

- Concealing the heterogeneous and distributed nature of the system so that it appears to the user like one system

Transparency categories
Access: access local and remote resources using identical operations (NFS or Samba-mounted file systems)
Location: access without knowledge of location of a resource (URL's, e-mail)
Concurrency: allow several processes to operate concurrently using shared resources in a consistent fashion (two users simultaneously accessing the bank account)
Transparency categories
Mobility: allow resources to move around
Performance: adaption of the system to varying load situations without the user noticing it
Scaling: allow system and applications to expand without need to change structure of applications or algorithms

Also read : Advantages of Distributed Systems over Centralized Systems

Also read : Design Issues of Distributed Systems

Also read : Examples of Distributed Systems

Also read : Introduction to the Distributed Systems

Tags Distributed Systems

Share :

OLDER
Examples of Distributed Systems

NEWER
Advantages of Distributed Systems over Centralized Systems

SIDEBAR POSTS

- How to Remove/Terminate Staff from Minority Educational Institutions in Telangana/AP November 12, 2022 NEW
- Backward Search Method- Strips Planning Systems March 20, 2022
- The Sussman Anamoly- Strips Planning Systems March 20, 2022
- Plans with Run-Time Conditionals-Strips Planning Systems March 19, 2022
- Recursive STRIPS-Strips Planning Systems March 19, 2022
- Strips Planning Systems in Artificial Intelligence March 19, 2022

Show More

SOCIAL PLUGIN

POPULAR JOBS

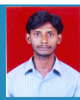
- OOAD MULTIPLE CHOICE QUESTIONS UNITWISE February 10, 2019
- OOAD MCQ: Basic Behavioral Modeling II April 11, 2019
- CLASS AND OBJECT DIAGRAMS: OOAD MCQ February 17, 2019
- BASIC STRUCTUREAL MODELING: OOAD MCQ February 16, 2019
- ADVANCED STRUCTURAL MODELING: OOAD MCQ February 17, 2019

STUDY MATERIAL

» Arithmetic » Artificial Intelligence » C++ » C++ MCQ » Compiler Design » COMPUTER PROGRAMMING IN C » Education Act » G.O.Ms.No.467 » OOAD MCQ » OOAD Notes » Web Technologies

ABOUT US

V KIRAN KUMAR



FOLLOW US