Distributed Systems

COMP90015 2018 Semester 1
Tutorial 02

Q1. Define and briefly explain each of the following distributed system challenges:

- Heterogeneity
- Openness
- Security
- Scalability
- Failure Handling
- Concurrency
- Transparency

Heterogeneity

- Where?
    - Networks
    - Computer hardware
    - Operating systems
    - Programming Languages
    - Implementation by different developers
- How to deal with it?
    - Standard protocols
    - Adhering to articulated APIs, message formats, and data types
    - Middleware
    - Portable code

Heterogeneity and mobile code

Mobile code is sent from one computer to the anther to run at the destination (e.g. Java applets):

- Code that is compiled to run in one OS does not run in another:
  - Different hardware
  - Different OS versions
- Virtual Machine approach provides a way of making code executable on any hardware – compiler produces code that is interpreted by the virtual machine
- Cross-platform compilation and code portability is another way, that compiles source code to multiple targets.

Q1. Define and briefly explain each of the following distributed system challenges:

- Heterogeneity
- **Openness**
- Security
- Scalability
- Failure Handling
- Concurrency
- Transparency

Openness

Openness refers to the ability of extend the system in different ways by adding hardware or software resources. Following are some approaches to address openness:

- Publishing key interfaces
- Allowing a uniform communication mechanism to communicate over the published interfaces
- Ensuring all implementations adhere to the published standards

Q1. Define and briefly explain each of the following distributed system challenges:

- Heterogeneity
- Openness
- **Security**
- Scalability
- Failure Handling
- Concurrency
- Transparency

Security

- There has three aspects of security:
  - Confidentiality (protection against disclosure to unauthorized individuals)
  - Integrity (protection against alteration and corruption)
  - Availability (protection against interference with the means of access)
- Security Mechanisms:
  - Encryption (e.g. Blowfish, RSA)
  - Authentication (e.g. passwords, public key authentication)
  - Authorization (e.g. access control lists)

Q1. Define and briefly explain each of the following distributed system challenges:

- Heterogeneity
- Openness
- Security
- **Scalability**
- Failure Handling
- Concurrency
- Transparency

Scalability

A system is considered to be scalable if it can handle the growth of the number of users.

- Scalability Challenges:
  - Cost of physical resources
  - Controlling the performance loss
  - Resources should not run out
  - Avoiding Performance bottlenecks

Q1. Define and briefly explain each of the following distributed system challenges:

- Heterogeneity
- Openness
- Security
- Scalability
- **Failure Handling**
- Concurrency
- Transparency

Failure Handling

- **Detecting**: some types of failures can be detected, e.g. checksums can be used to detect corrupted data in a message, and some kinds of failure are hard to be certain about, e.g. the failure of a remote server
- **Masking**: some failures that have been detected can be hidden or made less severe, e.g. timeout and message retransmission
- **Tolerating**: it is sometimes impractical to try and handle every failure that occurs, sometimes it is better to tolerate them, e.g. failure is reported back to the user, e.g. web server not being available, try again later, or video is rendered with errors
- **Recovery**: failure sometimes leads to corrupted data and software can be designed so that it can recover the original state after failure, e.g. implementing a roll back mechanism.
- **Redundancy**: services can be made to tolerate failures using redundant components, e.g. multiple servers that provide the same service, as so called fail over.

Q1. Define and briefly explain each of the following distributed system challenges:

- Heterogeneity
- Openness
- Security
- Scalability
- Failure Handling
- **Concurrency**
- Transparency

Concurrency

- Multiple clients can access the same resource at the same time, in some cases for updates
- One approach to handling concurrency is making access sequential - slows down the system
- Semaphores supported by the operating system is a well accepted mechanism to handle concurrency

Q1. Define and briefly explain each of the following distributed system challenges:

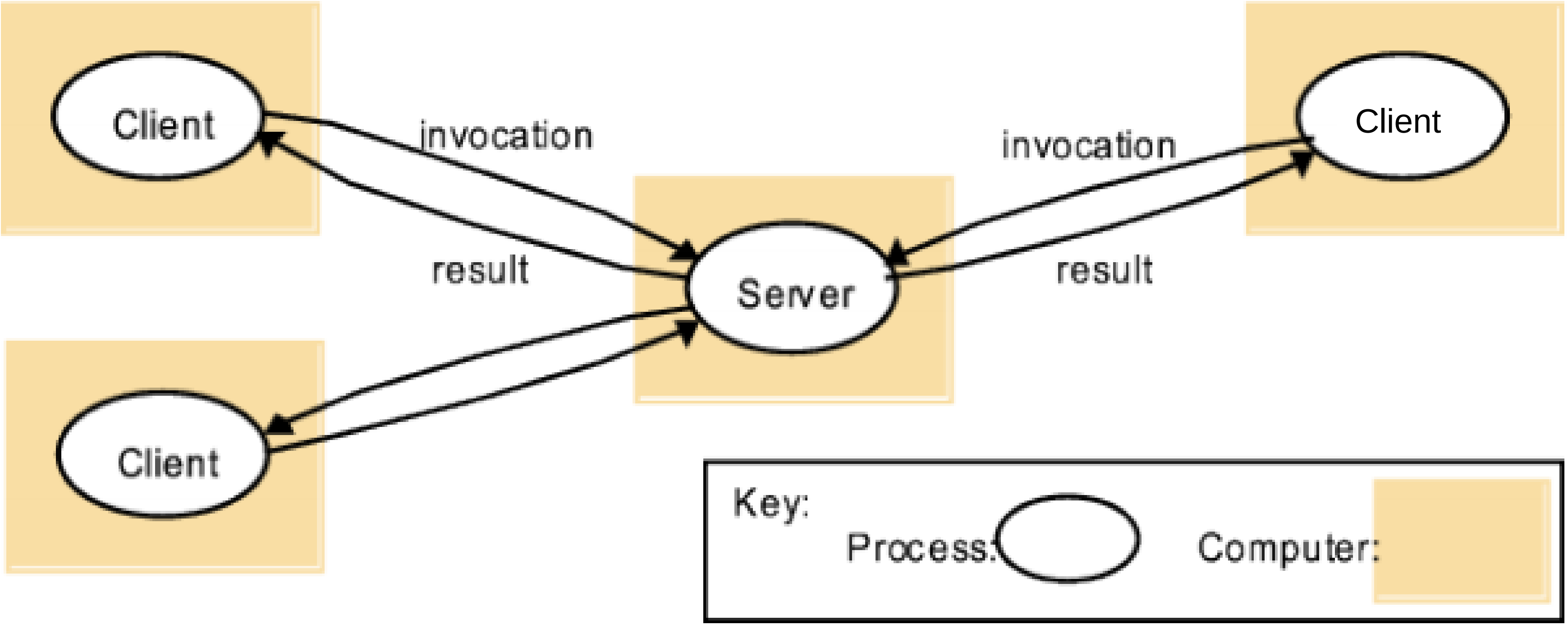
- Heterogeneity
- Openness
- Security
- Scalability
- Failure Handling
- Concurrency
- **Transparency**

Transparency

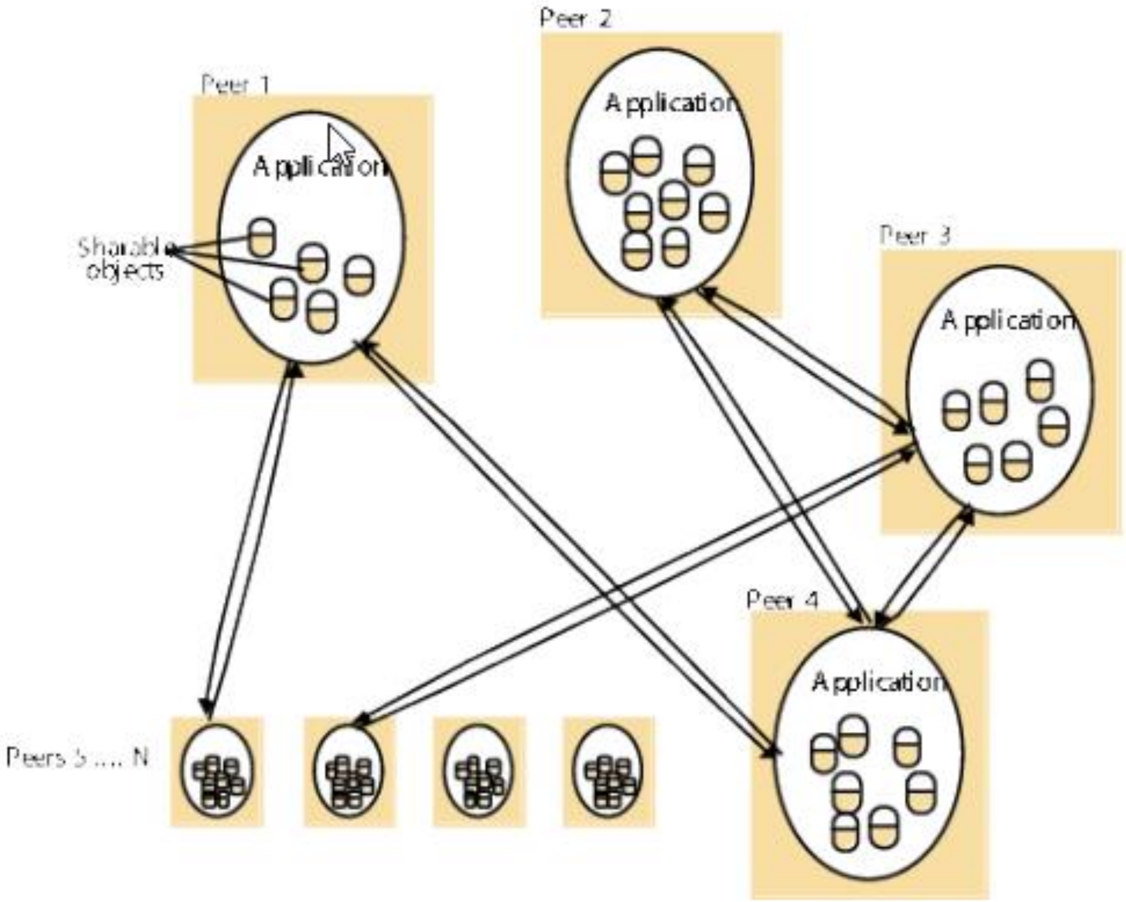
- Access transparency
    - Access to local or remote resources is identical
- Location transparency
    - Access without knowledge of location
- Failure transparency
    - Tasks can be completed despite failures
- And many more

Q2. Briefly explain the difference between a client-server architecture and a peer-to-peer architecture.

Client-server



Peer-to-Peer


Peer 1
Application
Sharable objects
Peer 2
Application
Peer 3
Application
Peer 4
Application
Peers 5 .... N

Article Discussion

- Debugging Distributed Systems
  - What is the paper about?
  - What are the challenges in debugging?
  - What are the existing approaches for debugging?
  - How to virtualise and understand distributed-system executions?

What is the paper about?

- Analyse key features and debugging challenges that differentiate distributed systems from other kinds of software
- Present promising tools and ongoing research on this front.

What are the challenges in debugging?

- Heterogeneity
- Concurrency
- Distributed state
- Partial failures

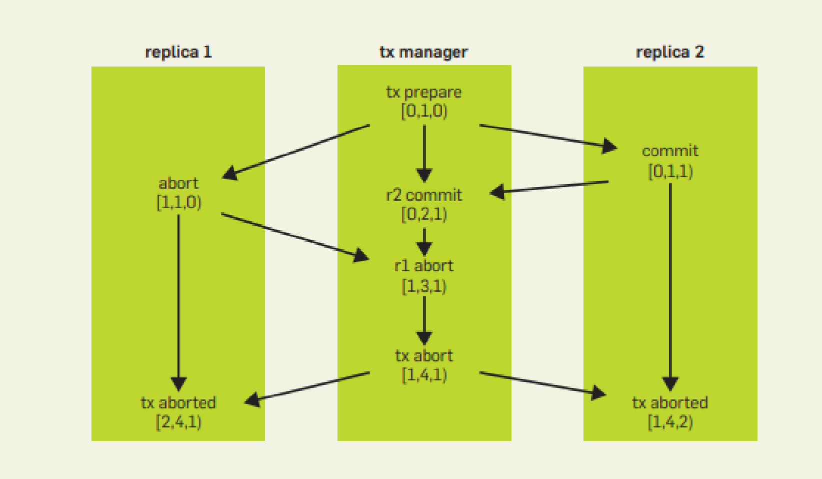
Existing Approaches

- Testing
- Model checking
- Theorem proving
- Record and replay
- Tracing
- Log analysis
- Visualization

Promising tool - ShiViz

- A visualisation tool that help developers understand the distributed system execution as interactive time-space diagrams that explicitly capture distributed ordering of messages and events in the system.
- Keyword search and structured search operations
- Compare multiple executions
- Group execution into clusters

Figure 1. Time-space diagram of an execution with three nodes.



Questions?