Introduction to Distributed Systems

February 9, 2020

Summary

What is a distributed system?

Examples

Why distribution?

Challenges

Further Reading

Summary

What is a distributed system?

Examples

Why distribution?

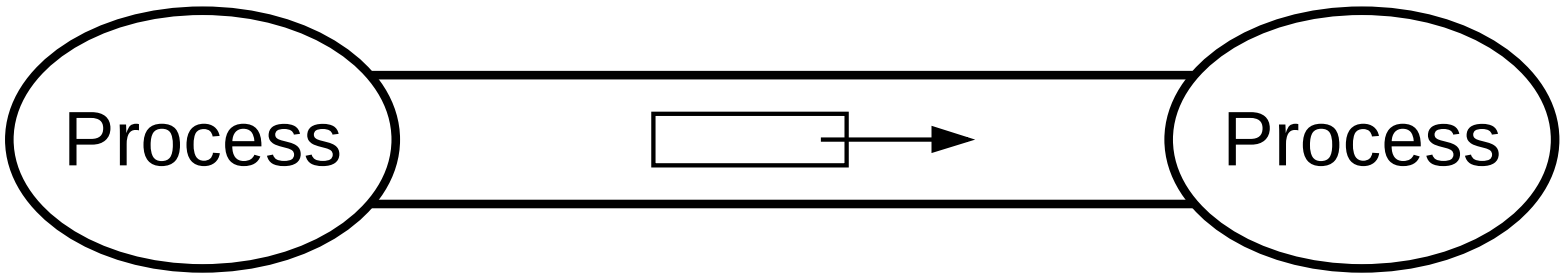
Challenges

Further Reading

Distributed System

Definition A distributed system consists of a **collection of** distinct **processes** which are spatially separated and **which communicate with one another by exchanging messages**. (L. Lamport, "Time, Clocks and the Order of Events in a Distributed System", CACM)

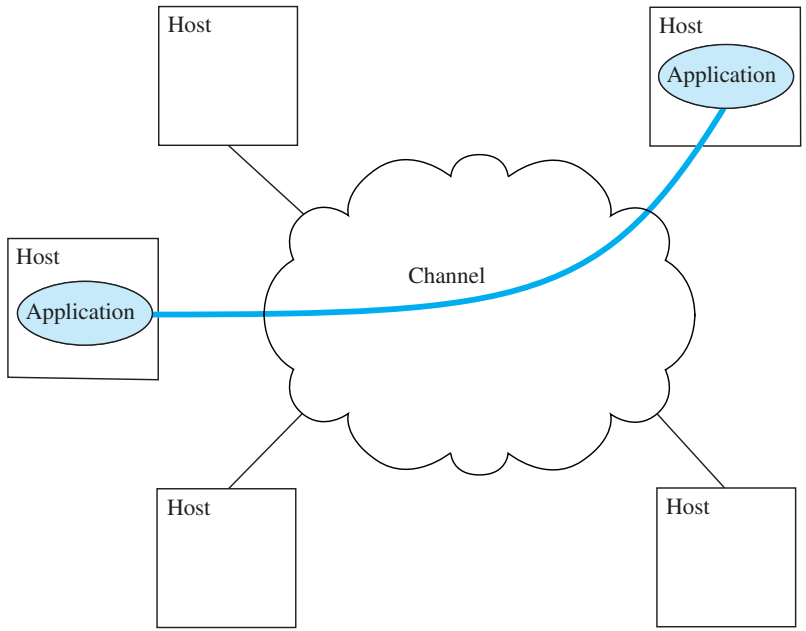
- "A system is distributed if the message transmission delay is not negligible compared to the time between events in a single process."



Message based communication

Message a sequence of bits

- Whose format and meaning are specified by a *communication a protocol*
- That is transported from its source to its destination by a **communications network**



Summary

Examples?

Other (?) Distributed Systems/Applications.

- Web and Internet
- Google's search service
- Google's voice-to-text service
- *Email service*
- *Peer-to-peer* applications, such as Bittorrent
- FEUP's file system
- Telecommunication networks
- ATM networks (SIBS)
- Home automation (IoT)
- Factory automation (Industry 4.0)
- Fly-by-wire, drive-by-wire. (Autonomous driving)

Summary

Potencial Advantages

- Sharing of resources
- Access to remote resources
- Performance
  - Can use multiple computers to solve a problem
- Scalability:
  - Load (no. of users/request rate)
  - Geographical;
  - Administrative
- Fault tolerance
  - Reliability
  - Availability

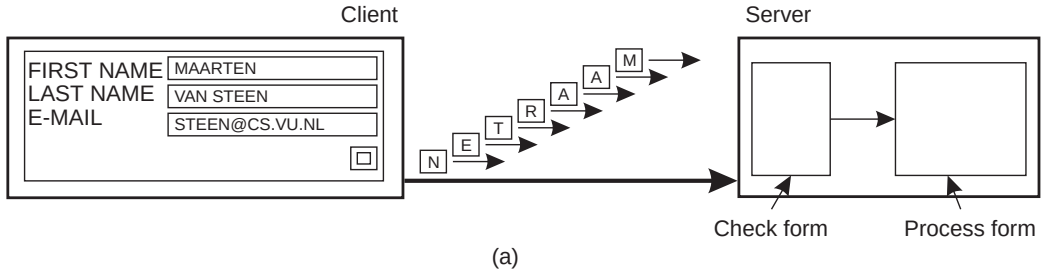
Scalability: Challenges

- Centralization
  - processing;
  - data;
  - algorithms.
- Synchronous comunication
- Security and (lack of) trust

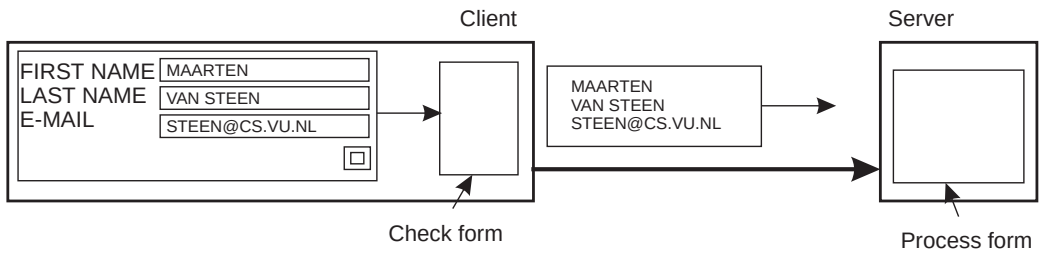
Scalability: Some Techniques (1/2)

Distribuition

processing:

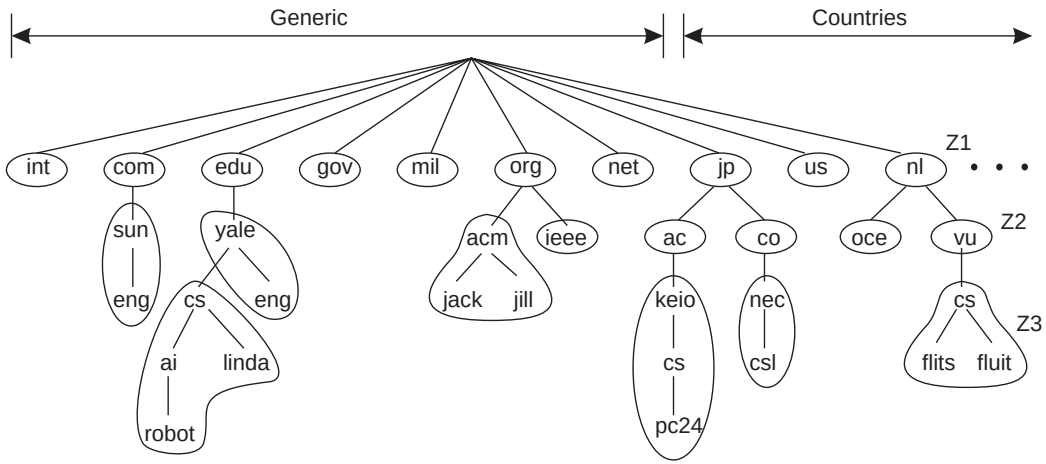


(a)



(b)

data (partitioning):



Scalability: Some Techniques (2/2)

- Distributed (decentralized) algorithms:
  - System global state is unknown (relativity)
    - Can use only information locally available
  - Correctness must be ensured even in the presence of faults
  - No single physical clock
- Asynchronous communication
- Replication and *caches*:
  - \+ reduces communication latency;
  - \+ allow distributed processing;
  - \- raises consistency problemss

Summary

Challenges

- Partial failures
  - Some components may fail, while others continue to operate correctly
- IPC latency
  - IPC across the network has a larger and unpredictable latency, which usually cannot be bounded
- No global time
- No shared physical memory and distinct address spaces
  - Pointers are meaningful only in the context of the respective address space
- Heterogeneity
  - Has several facets
- Lack of security and trust

Summary

Further Reading

- *Distributed Systems, 2nd Ed.*, Chapter 1
- Michael Schroeder (et. al.) State-of-the-Art Distributed System: Computing with BOB
  - Nice "vision" from leading distributed system's researchers of DEC's SRC around 1990
  - Read only Sections 1 and 2
- Jim Waldo, et. al, A Note on Distributed Computing
  - Somewhat language-oriented, by people who designed Java RMI