COMP90015 Distributed Systems

Introduction

School of Computing and Informati

2022 Semester II

1. Subject administration

2. A Computer System Basis
   - Physical Model
   - Process Model

3. Distri
   - De
   - Motivation
   - Consequences
   - Case Studies
   - Commercial distributed systems

4. Summary

Staff

- Aaron H
  aharw
- Moha
- Jonath
- Tharindu Bandara, Tutor
- Yiwen Zeng, Tutor

Assessment

Projects will be group work with groups of size 2. You may work alone if you wish ho … ogramming language i … . Project 1 details wi …

1. Projec … k 4 and due around Week 8
2. Project 2, 20%, software and written report, starting ar … e around Week 12
3. Final Exam, 60%, online

Academic Integrity

All Univer cademic
integrity cr
assessme

- https://academicintegrity.unimelb.e

Traditional Course Overview

This course was originally developed from Coulouris, Dollimore and Kindberg, *Distributed Systems: Concepts and Design*, Edition 5, © Addison-Wesley 2012, with emphasis on the following chapters:

Chapt

Chapt

Chapt

Chapter 5 Remote Invocation

Chapter 6 Indirect Communication

Chapter 7 Operating System Support

Chapter 11 Security

Chapter 12 Distributed File Systems

Chapter 13 Name Services

2021 Semester 2 Course Overview

A tentative plan for the semester:

Week	Date	Lecture	Tutorial
1	25th Jul.	Motivation, Challenges	
2	1st Aug.	IPC, Data Representation	Tutorial 1
3	8th		
4			
5	22		
6	29th Aug.	Indirect Communication Paradigms	Tutorial 5
7	5th Sep.	Encrypting Communication	Tutorial 6
8	12th Sep.	Digital Signatures, Certificates, SSL/TLS	
9	19th Sep.	Distributed File Systems	Tutorial 8
-	26th Sep.	Non-teaching Week	
10	3rd Oct.	Name Services	Tutorial 9
11	10th Oct.	TBA	Tutorial 10
12	17th Oct.	TBA	Tutorial 11

Learning Outcomes

- Familiarity with distributed system terminology and fundamental concepts
- Develo ... ues
- Know a r ... es
- Unders ... ate their approp
- Develop skills in distributed system design and progra ... tation
- Develop skills in small group work and written communi

Computer Hardware

In this subject we are mainly interested in distributed systems based on *computer systems*, with a broad applicability:

- Machi … ously depend … ice", "cluster node", …
- Physic … r tablet), but technically not much difference – we can move our desktop if we want.
- Hardware devices/specifications:
  - CPUs/cores (clock rate, cache), RAM (capacity, latency)
  - Connectivity: ethernet, wireless, cellular, bluetooth, USB (…
  - Storage: SSD, HDD (capacity, latency, throughput, read/ …
  - Graphics: GPU (resolution, refresh rate)
  - Peripheral devices: keyboard, monitor, mouse, printer, webcam, microphone, etc

Computer Software

We are mainly interested in the higher layers of software from the Application/User software down to the Operating System.

- Applic
- Middle)

- Operat
  - Kern
  - Device drivers
  - Modules/services
- Hypervisor (for virtualization)

- Bootloaders, EFI (Extensible Firmware Interface) BI put System)
- ROM (Read Only Memory) and firmware, POST (Power On Self Test)

Platform

A platform is a *layer* of functionality or API across a number of machines, by de facto being a consistent hardware and OS specification combined, but sometimes being a *middleware* layer above the OS primarily to hide differences in OS and hardware on different machines.

- POSIX or Portable Operating System Interface is the standard OS API adopted by most Unix-based OSes. It was defined in 1988 by the IEEE Computer Societ
- The ma OS becam
- The Lin ices such as routers are "mostly" POSIX-compliant.
- Cygwin/MinGW provide a mostly POSIX-complia ntime environment for Microsoft Windows. Conversely, W ty layer for UNIX-like operating systems to run programs writt Windows.

- The Windows API or WinAPI is the API for the Microsoft Windows OS, from Microsoft Corporation.
- The Java Virtual Machine executes programs, usually written in Java, which compile to Java bytecode. The JVM provides a consistent API across many different Hardware/OS platforms, which is one aspect which lead to the popularity of the Java programming language.

Virtualization and Emulation

Even commodity OS and computer hardware provides feature rich virtualization and emulation support that can be of interest when studying distributed systems. However the details are often quite technical and OS spe

- Virtual at there appear erspective (i.e. logi

  - Virtual memory: paging to disk
  - Virtual Cores: time sharing physical cores
  - OS Virtualization: Jails, Containers, Zones
  - Virtual Machine (VM): all devices are virtualized.
    - Hypervisors
    - run an OS in the VM

- Emulation: create the appearance of a device existing, even though it does not:
  - Mobile device emulation for testing and debugging on hardware that is not physically available
  - Network device emulation: switches and routers (Mininet)
  - OS emulation

Computer Networks I

Besides the computer system, the computer network is an essential aspect for our study of distributed systems:

- We are almost exclusively interested in networks that support the TCP/UDP/IP protocol stack, although sometimes we use other network protocols specific to a given technology:
  - Blue ... P)
  - Cellu ... ith Inter
- Releva

  - Wireless Base Stations: provide a way to connect to the network using radio waves anywhere from 900MHz to 60GHz, sometimes called WLA
  - Switches and Routers: provide a way to connect to the network u Ethernet at least 1Gbps but also at 10, 25 and 40Gbps

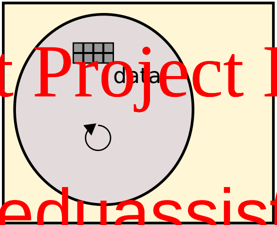
- The well known *Internet*, arising in the 1980's, has b global computer network and communications platf
  - Internet Service Providers (ISPs) provide access which among other things provides one or more unique *Internet Addresses*:
    - represent end-points on the *public* Internet,
    - devices with public Internet addresses can be directly communicated with by every other device connected to the Internet.

Computer Networks II

  - No one country owns or controls the entire Internet, although some governing bodies such as ICANN (The Internet Corporation for Assigned Names and Numbers) have traditionally controlled some essential aspects, namely the Internet Addresses and Internet Domain Names.
  - There are lessor known alternatives arising to ICANN's governance, e.g. Yeti DNS (Chi sponsored).

- Organi Interne tion:
  - Priva rivate network.
  - Local Area Networks (LANs): the smallest unit of network management where typically all of the machines on the LAN are subject to the same network poli
  - Wide Area Networks (WANs): multiple LANs connected to ing network policies across an organization.
  - Network Address Translation (NAT): provide an ability for nect to the public Internet via a machine (typically a router) that has bot rnet Address and a private network address.
  - Virtual Private Network (VPN): connecting two or more private networks via a secure connection over the public Internet .

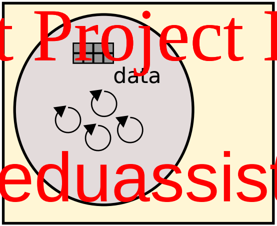
Single process, single thread



- Smallest OS encapsulation:
  - Every application requires at least one process
  - Primarily encapsulates processor and memory resources
  - System calls to access devices on the machine
- One machine
- Single thread:
  - one core
  - one address space
  - no concurrency control

Not a distributed system.

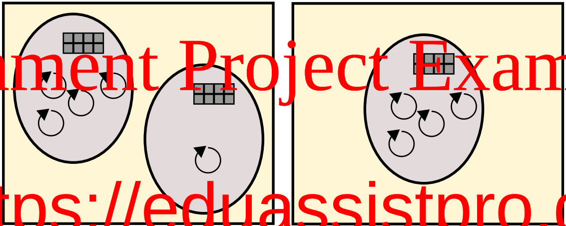
Single process, multiple threads



- One machine, one process
- Multiple threads:
  - multiple cores
  - one address space shared among threads
  - concurrency control required

Not a distributed system – limited to a single machine.

Multiple processes, single/multi-threaded



- One or more machines – each process can run on a different m
- Each process has its own address space
- *Inter-process communication* (IPC) is required:
  - Shared memory, pipes and file-based communication if pro ine
  - TCP/UDP/IP for processes on the same or different machines
- Each process may have one or more threads:
  - One or more cores per process
  - Concurrency control within processes and across processes

Now do we have a distributed system?

What is a Distributed System?

Distributed System definitions – many and varying:

- "A system in which hardware or software components located at networked compu ... ing messa
- "A colle ... single cohere

Key aspects:

- a number of components
- communication between the components (implied
- synergy; achieve more than the simple sum of individual c

Discussion questions

Question (1): If we have a number of floor cleaning robots, *that never communicate*, but that independently go through a process of cleaning the same floor … n to each othe … leaned or can be skip … ly cleaning the floor faster than a single independent robot could do so, is this an example of a distributed system? Why or why not?

Question (2): Multiple threads can be seen as wor … a problem. So why don't we call a multi-threaded, single p … application as a distributed system?

Why would we build a Distributed System? I

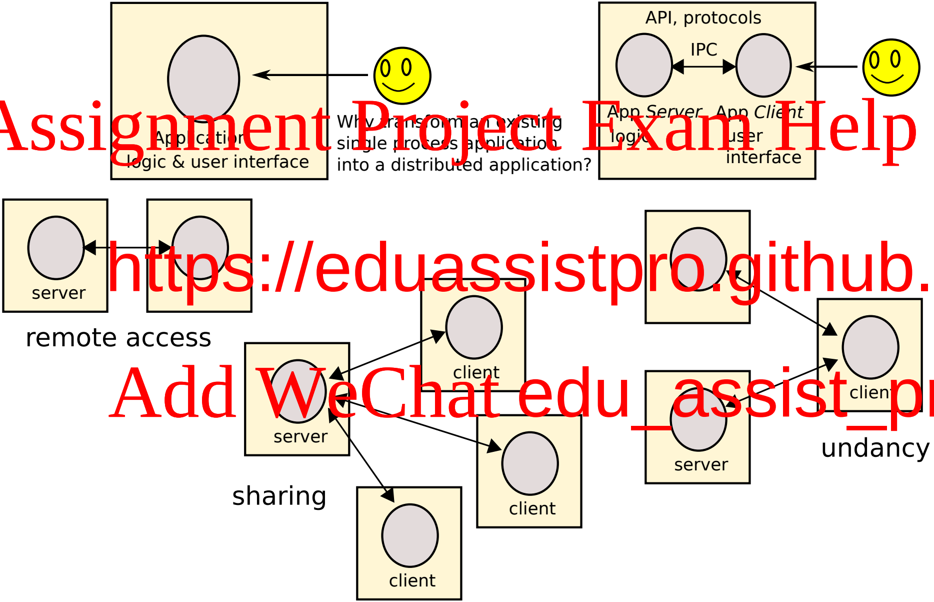
Lots of reasons, many of which you would take for granted in today's world:

- Comm
  - users
  - multi conf
- Remot
  - user does not need to travel to a given computer to use it, but can acce
  - can use resources on the other side of the Earth, can control a rover o nds successfully)
- Resource sharing:
  - single resource can be shared among multiple users: significant cost saving and increased utilization

Why would we build a Distributed System? II

- Reliability:
  - incre tinue to oper
- Availa
  - less do line on remaining hardware

- Scalability:
  - using more resources to provide greater capacity than any sing
  - ideally having N resources provides N times the cap


Application
logic & user interface
Why transform an existing
single process application
into a distributed application?
API, protocols
IPC
App Server
logic
App Client
user
interface
server
remote access
client
server
client
client
sharing
server
client
undancy

Consequences of Distributed Systems I

The benefits that distributed systems bring are not free:

- *Communication complexity*. Communication is not free, and is largely an overhe … o take place. Comm … ication errors can occu … significant periods
- *Concurrency*. In a Distributed System computers perf … autonomously and communicate with other comput … sing when needed. Services provided by distributed system … by multiple users simultaneously. Distributed system d … s into consideration and implement appropriate techniqu … rency.

Consequences of Distributed Systems II

- *No global clock*. Clocks on individual computers operate independently. Since compu accuracy with whi s. Therefore, distrib a clock and imple

- *Independent failures*. Some components of the syste others are still running. Failures of participating compu t known to others immediately.

Interprocess Communication

APIs, protocols and semantics

IPC significantly increases the design and implementation effort over a single process application. IPC generally requires:

- an API w mmunicating proces
  - GetR
  - Retr
  - Stor

- *communication semantics* that define what the pr
  - "all API calls block until the operation has successfully comple n exception is thrown in which case the operation outcome is und

- a *communication protocol* which defines how th communication and exchange data to implement the A e the semantics, e.g.:
  - Transmission Control Protocol (TCP) connections and JSON data objects

Distributed system challenges I

- Heterogeneity – the parts of the system are not consistent (homogenous)
  - Hardware, OS and networking can differ across different components.
  - Processes can be written in different languages, and by different developers.

- Openness – the system can be built upon or accessed by third-party developers, through public APIs and protocols
  - Whe
  - APIs s

- Securit
  - Com ich is a significant security risk.
  - System components may become controlled or corrupted b
  - System components may become targets of denial-of-servi
  - Systems that execute third-party (untrusted) code/scrip

- Scalability - increasing the number of system compone overheads
  - For N system components, overheads may grow geometrically, e.g. proportional to N^2.
  - For a distributed system operation to be scalable we would like the operation's overheads to grow no faster than $\log N$.
  - Avoiding all bottlenecks in a design is non-trivial.

Distributed system challenges II

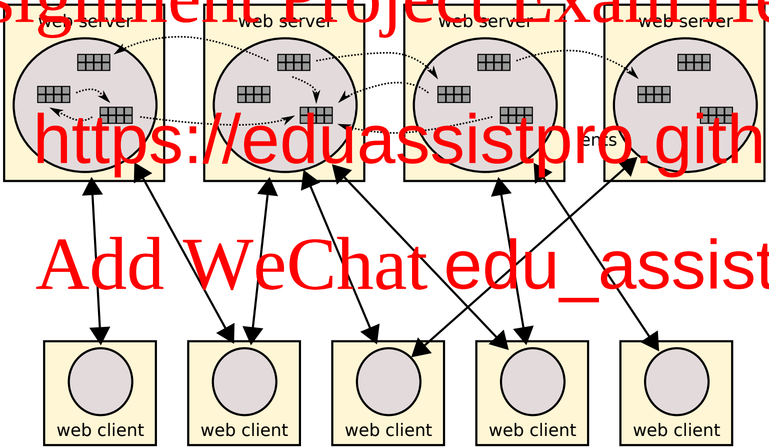
- Failure Handling - system components fail independently and the communication network can fail as well
  - How can we determine whether a remote component has failed or is simply taking a longer time than expected to respond?
  - How do we know if data has been lost or if some operations were completed or partly completed or not completed at all?
  - What do we do?
    - T
    - A
    - T ... leted: *erro* *re*

- Concur ... ency control requirements
  - data needs to be consistent across the system
  - outcomes need to be deterministic
- Transparency - hiding aspects of the system from the hig ... cation

  - Access transparency - APIs for accessing remote resources
  - Location transparency - location of data and resources
  - Concurrency transparency - consistency of data
  - Replication transparency - replication of data and operations
  - Failure transparency - system components and network communication can fail
  - Mobility transparency - system components change their IP address
  - Performance transparency - overheads such as communication delays
  - Scaling transparency - increasing the number of resources

World Wide Web

The World Wide Web is based on Web Servers and Web Clients with documents that embed the location of other documents.



Discussion questions

The WWW began as a simple system for exchanging documents at the European centre for nuclear research (CERN), Switzerland, in 1989. Documents are logically organized using a *hypertext* structure and are connected through *links*. The three main standard technological components used in the web are:

- Hypter … cifying the conten
- Unifor … stored on the web.
- HyperText Transfer Protocol (HTTP) - This is the prot transferring resources between web servers and client

Question (3): Which distributed system chall address well and which ones does it not address well? Explain your reasoning. **Question (4):** What other aspects of the WWW do you think has lead to its immense success in becomming a defacto platform for most of our distributed applications today?

Discussion questions

Questio

you comm ... ely
multi-pla ... ation etc)
and discus ... at you
think these applications face, based on your own experi ... m;
try to relate your discussion to concepts given in this lectu

Summary

- This subject is concerned with computer systems and computer networks as the und
- Distrib tly taken for grante

- Comm r distributed systems.
- There are many challenges associated with the design an tion of effective distributed systems.