UNIVERSITY of STIRLING

Concurrent and Distributed Systems
CSCU9V5

Andrea Bracciali

abb@cs.stir.ac.uk

4B86

Two main topics

- Concurrency
- Distributed systems

… a taster ->

Concurrency

B=200

100!

Balance?

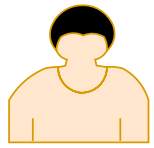
200!

Authorized!

newBalance =
200 -100 = 100

B=100

Concurrency



B=200

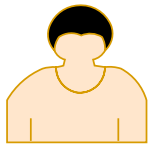
100!

100!

?

… you tell me ->

Concurrency



B=100

100!

100!

homework!

… ready for the next topic! ->
(distributed systems)

Distributed systems (I)

Please, clap you hands all together when ready

anytime from now

Distributed systems (II)

You can only exchange information with your four neighbours (north south east west).

Please, count how many you are in the class.

(everyone must have the same final count)

Two main topics

INFORMALLY SPEAKING !

- Concurrency

  managing processes within the same computer

- Distributed systems

  managing processes across different computers

CANVAS OVERVIEW

Let us move to canvas

Introduction to the course

- Some Logistics
- Course Contents
- Course Layout
- Resources for private study
- Introduction / Motivation to the subject

Some Logistics

- **Dr. Andrea Bracciali**, abb@cs.stir.ac.uk

- **Lectures:**
  - Monday at 17:00 in LTB3
  - Tuesdays at 09:00 in 2A13

Some Logistics

- **Laboratories**
  - The first lab is today at 11am
  - Tuesdays: **G1** 11.00 - 13.00, 4X5, 4X8
    G2 14.00 - 16.00, 4X5, 4X8
  - Thursdays: 12.00 - 14.00, 4X5, 4X8
  - Two lab sessions – one lab task (sheet)
  - Checkpoints (**20%**)
- **Assignments** (Compulsory)
  - Class Test: tbc ~October (**40%**)
  - Assignment: tbc ~November (**40%**)

Course Contents

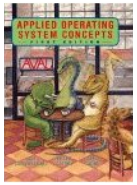
- **Lectures**
  - Previous modules :
    - programming Java
    - basic OS principles (threads) (CSC9V4)
    - principles of communication (CSC9W6)
  - Now :
    - insight into Process and Thread management
    - concurrent systems
    - distributed systems
- **Laboratories**
  - Focus not so much Java, more Concurrency & Distribution
  - Practical experience on taught (lecture) material

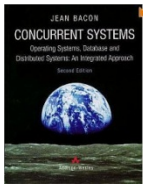
Course Layout

- Introduction to lecture series (1 lecture - today)
- Process and Thread Management (2 lectures)
  - Basic process model
  - Introduction of the problem
- Concurrency (7 lectures)
  - Means of communication
  - Critical sections, Synchronisation
  - Scheduling
- Distributed Systems (8 lectures)
  - Client – Server model
  - RPC principles, Java for distributed systems

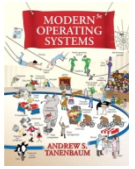
Resources for Private Study

- Internet: lectures & lab sheets will be available in pdf format
- Books





  - Applied Operating Systems Concepts, A. Silberschatz, P. Galvin, G. Gagne, John Wiley & Sons
  - Concurrent Systems, J. Bacon, Addison-Wesley
  - Modern Operating Systems, A.S. Tanenbaum, Prentice-Hall,
  - Distributed Systems, A.S. Tanenbaum, M. v. Steen, Prentice-Hall



- Emails are welcome
  - abb@cs.stir.ac.uk

Concurrent & Distributed Systems

- Concurrent System
  - Several activities are happening at the same time

  Two activities are concurrent if, at any given time, each is between its starting and finishing point. (Bacon)
- Distributed Systems
  - Special case of concurrent systems

  A distributed system is a collection of independent computers that appear to the user of the system as a single computer. (Tanenbaum)
  - Two aspects:
    - Hardware: autonomous machines
    - Software: hardware appears as single machine to the user

Classification of Concurrent Systems

- Inherently Concurrent Systems
  - Real-time Systems
  - Operating Systems

- Potentially Concurrent Systems
  - Large amount of computing (graphics applications)
  - Large amount of data to be processed (simulations)
  - Real-time requirement for the result (voice processing)
  - Hardware is available to run applications in parallel
  - **Big data** problems

Example: Real-time Systems

- Timing constraints dictated by the environment of a system
- System has to respond to external events within a certain time
- Examples: - aircraft systems
  - hospital patient monitoring
- Software monitors & controls aspects of the environment of the system

Example: Operating Systems

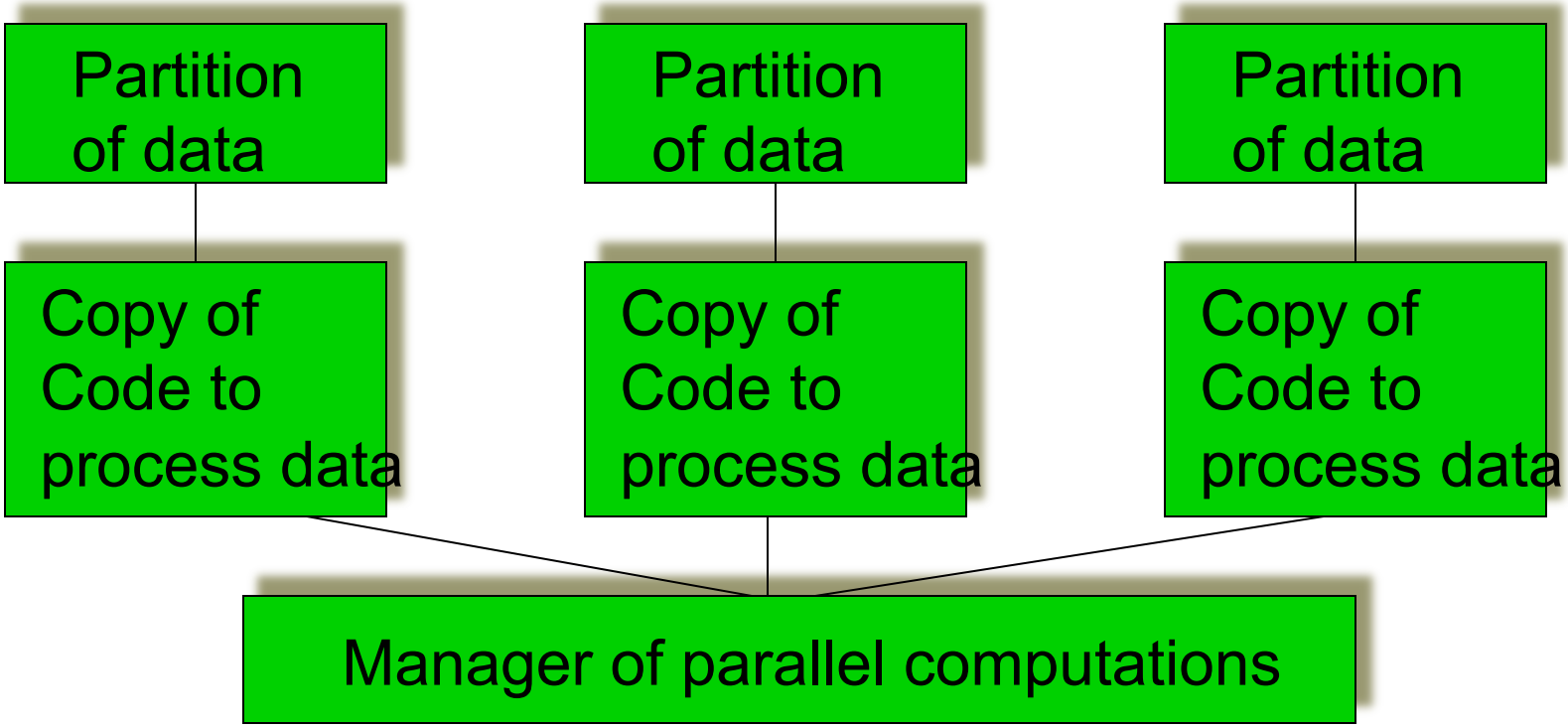
- Single user and multi-user systems
- Devices are slow compared to the main processor
  - OS attends to the devices when necessary
  - Other tasks are performed while devices are busy
- User want to carry out a number of tasks in parallel (lengthy computation + reading email)
- Running programs of different users in parallel
- Preemptive and non-preemptive scheduling
- Distributed Operating Systems

Examples of Potential Concurrency

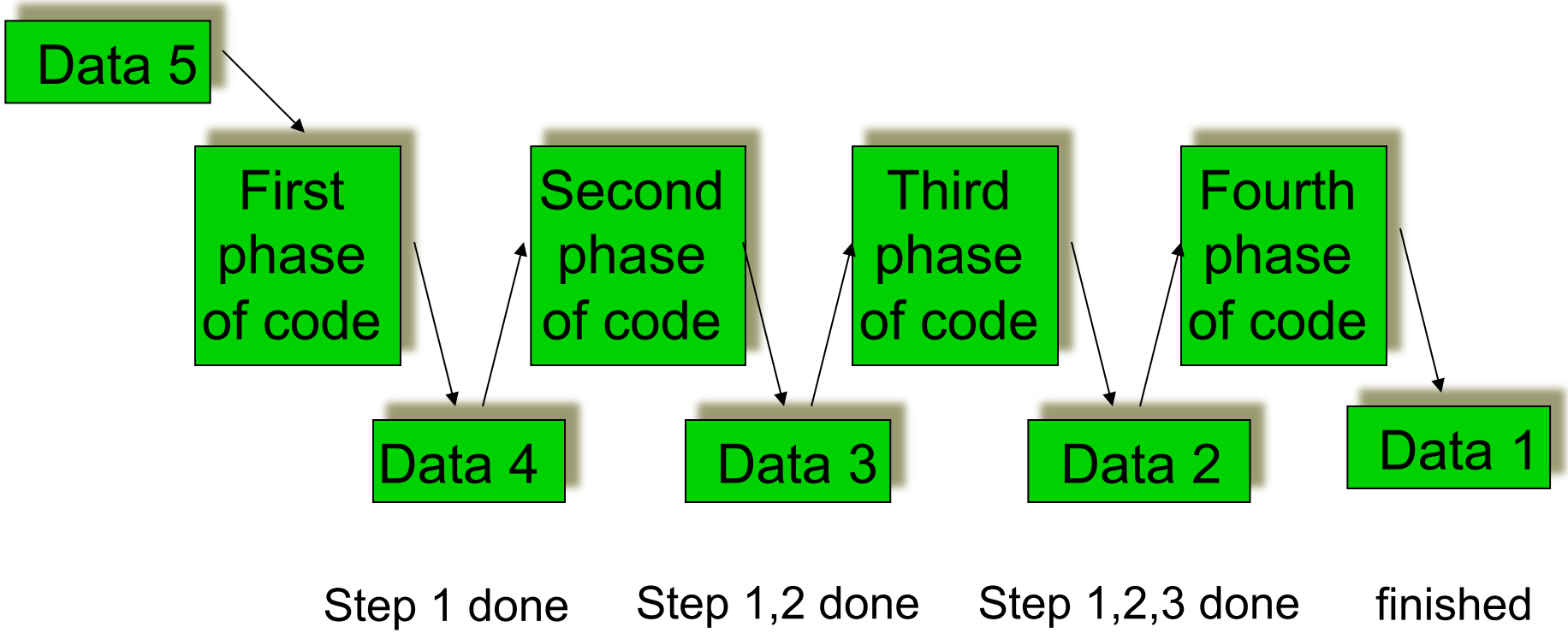
- Examples in this category would benefit from concurrency
- Example cooking recipe
  - Can be ‘executed’ sequentially
  - Some steps can be carried out simultaneously
  - Ingredients for the next step need to be ready

- How is concurrency introduced
  - Various models!
  - Partition data, replicate code (SIMD - Flynn's taxonomy)
  - Different code on different data (MIMD)
  - Pipeline

Partition data, replicate code (Bacon)



Pipeline


Data 5
First phase of code
Second phase of code
Third phase of code
Fourth phase of code
Data 4
Data 3
Data 2
Data 1
Step 1 done
Step 1,2 done
Step 1,2,3 done
finished

Benefits of Distributed Systems

Economics	Data Sharing
Speed	Device Sharing
Inherent Distribution	Communication
Reliability	Flexibility
Incremental Growth	Transparency

Challenges of Distributed Systems

Expensive Software	Communications Delay
Scalability/Overhead	Inconsistent State
Security	No Global Time
Independent Failure Nodes	Heterogeneity

Summary

- Future lectures will be on *concurrent* **and** *distributed* systems
  - Manage concurrent activities in a centralised and distributed environment
  - Use Operating Systems as example
    - How do OS cope with concurrent activities?
    - What do OS offer to support the development of concurrent software?
- Laboratories will involve building simple concurrent and distributed systems
  - Apply the theory & use Java language