CS3210 – Parallel Computing Course Admin

Lecture 0

Overview

- Teaching Staff

- Course Overview:
  - Objectives
  - Topics & Lecture Schedule

- Course Structure:
  - Tutorial & Lab
  - Assessment

- References

Teaching Staff – **Lecturer**

- Dr. Cristina Carbunaru
- Contact Information:

  Email: **ccris@comp.nus.edu.sg**

  - ❑ Comments / suggestions welcome
  - ❑ Email in advance to book a consultation slot

Teaching Assistants

Name	Email
Sriram Sami	sriramsami@nus.edu.sg
Richard Willie	e0550368@u.nus.edu
Yap Zhi Heng	yap.zhh@u.nus.edu
Liew Zhao Wei	zhaoweiliew@gmail.com
Simon Julian Lauw	simonjulianl@u.nus.edu
Vimuth Mendis	e0550551@u.nus.edu
Theodore Leebrant	theodoreleebrant@u.nus.edu
Rohit Rajesh Bhat	rajeshbhatrohit@gmail.com
Seow Alex	seowalex@u.nus.edu
Koh Hong Po	kohhongpo@u.nus.edu

Course Objectives

- Provide an introduction to the field of parallel computing with **hands-on** parallel programming experience on **real parallel machines**

- Four major parts:
  1. Parallel architectures
  2. Parallel computation models
  3. Parallel algorithm design and programming
  4. Performance of parallel programs

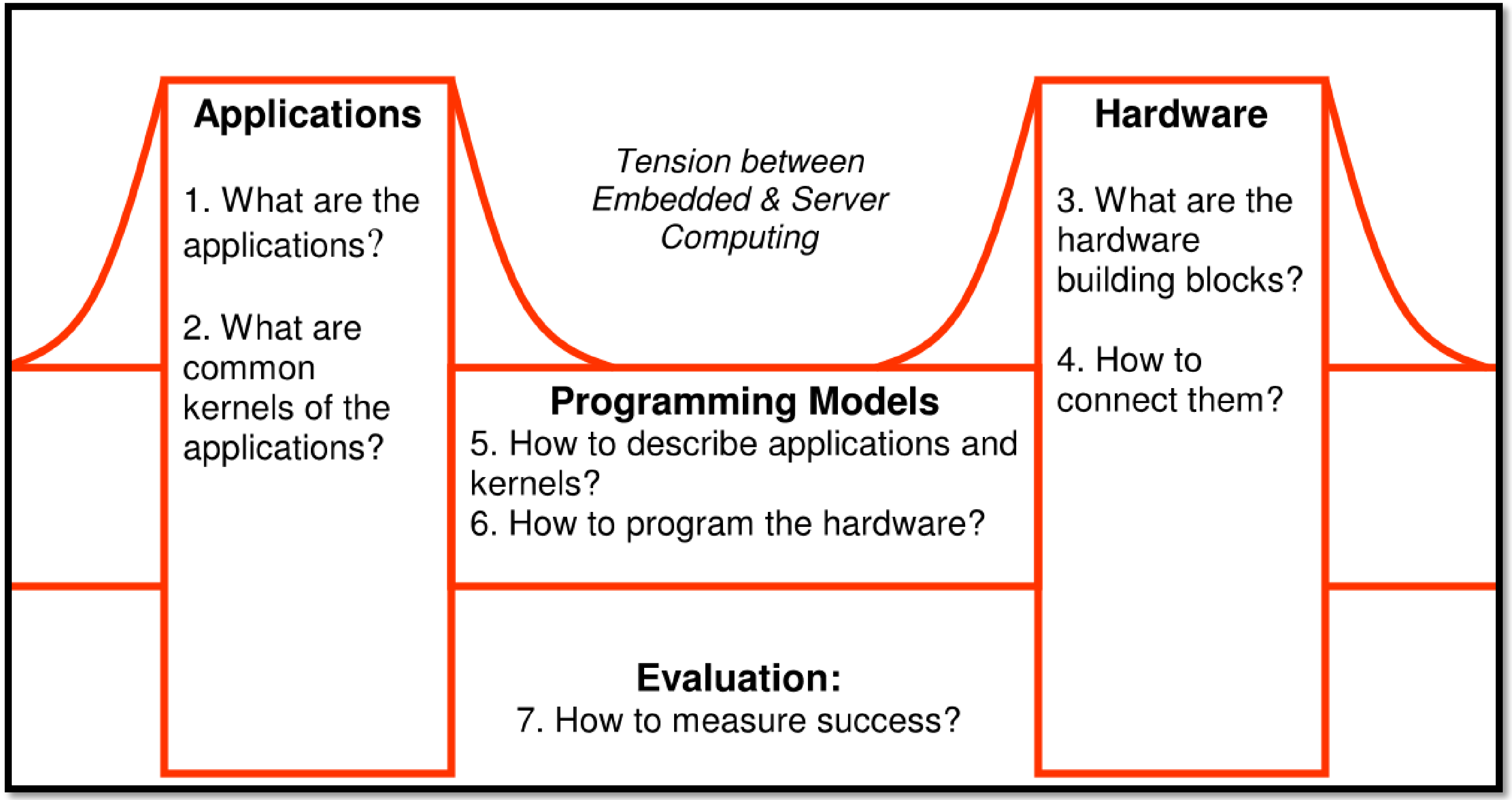
Not-so-formal Objectives

- Designing and writing scalable parallel programs
  - Parallel thinking and writing code in popular parallel programming languages
  - Emphasis on performance
- Parallel computer hardware implementation: how parallel computers work
  - Characteristics of the machine really matter
- Efficiency of running parallel programs on parallel hardware
  - Efficient != fast: running faster on a parallel computer, it does not mean it is using the hardware efficiently

Topics in Lectures

- Background on parallelism (from OS)
  - L2: Processes and threads
- Architecture
  - L3: Processor and memory organization
  - L7: Cache coherence and memory consistency
  - L11: Interconnection networks
- Parallel computation models
  - L4: Shared-memory programming models
  - L6: Data parallel models (GPGPU)
  - L9, L10: Distributed-memory programming models
- Performance and scalability of parallel programs
  - L5: Performance of parallel systems
  - L8: Performance instrumentation
- New trends
  - L12: Energy efficient computing

Overview



Applications
1. What are the applications?
2. What are common kernels of the applications?
Tension between Embedded & Server Computing
Hardware
3. What are the hardware building blocks?
4. How to connect them?
Programming Models
5. How to describe applications and kernels?
6. How to program the hardware?
Evaluation:
7. How to measure success?

Picture taken from "The Landscape of Parallel Computing Research: A View from Berkeley" (2008)

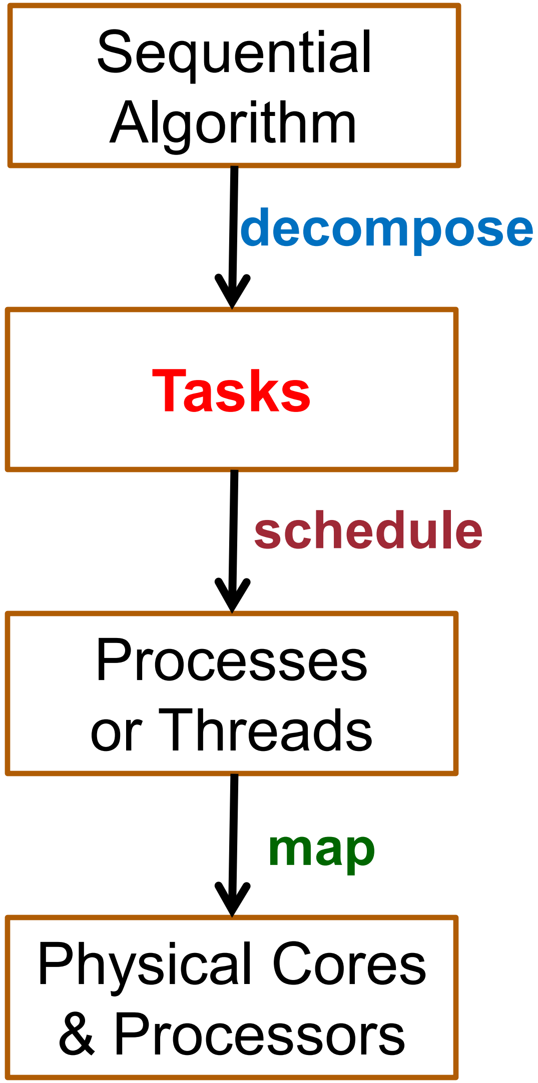
Program Parallelization: Steps

- 3 main steps:

1. **Decomposition** of the computations

2. **Scheduling** (assignment of tasks to processes (or threads))

3. **Mapping** of processes (or threads) to physical processors (or cores)



High-level Structure of the Module

- L1-L7: **Shared-memory models**
  - Architecture, memory consistency, programming
  - OpenMP and CUDA
- L8-L11: **Distributed memory models**
  - Architecture, interconnects, programming
  - MPI
- with an emphasis on **Parallel performance**

In-person Teaching

- All lectures are **recorded**
  - Recordings published in Canvas
  - No attendance taking at lecture
- Tutorial and labs: in-person, in various tutorial rooms
  - Attendance is taken
  - Lab sheets submissions

Course Structure – Tutorial & Lab

- Tutorials and Labs in different weeks:
  - Starting with Lab 1 in Week 3, Tutorial 1 in Week 4, …., etc
    - Register on EduRec for a tutorial slot
  - Each session is 2 hours long
    - 2nd hour is open session
  - Parallel & Distributed Computing Lab, Com 1, #B1-02
    - Classes are conducted in other tutorial rooms

Activities - Assessment

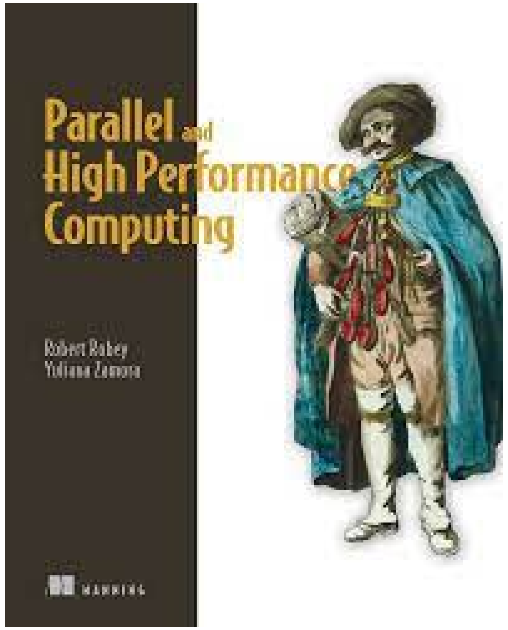
- Continuous Assessment (60%)
  - Lab (6%) – 3 lab sheets
  - Tutorial (4%) – attendance and participation
  - Programming Assignments (35%)
    - C/C++
    - Work in pairs or individually
    - Teammates might be registered to different tutorial slots
  - 4 Quizzes (15%)
    - Weeks 5, 8, 10,12
    - **Due on Sun, 8pm**
- Exam (40%) – Wed, 29 Nov, 9am
  - Open book, **in-person**

Schedule

Week	Lecture (Mon, 2pm) – SR1 (COM1-206)	Tutorial (Mon-Thu)	Lab (Mon-Thu)	Deadlines
1	L0: Course Admin L1: Introduction			
2	L2: Processes, Thread & Synchronization			
3	L3: Processor & Memory Organization		L1: Processes & Threads	
4	L4: Parallel Programming Models – I	T1: Parallel Computer Architecture		Wed, 2pm – Lab 1 due
5	L5: Performance of Parallel Systems		L2: Shared-memory Programming, Performance Instrumentation	Sun, 8pm – Quiz 1 due
6	L6: GPU Programming	T2: Performance of Parallel Systems		Wed, 2pm – Lab 2 due
Recess				Wed, 2pm – Assignment 1
7	L7: Cache Coherence & Memory Consistency		L3: CUDA programming	
8	L8: Performance Instrumentation	T3: Memory Consistency & CUDA Programming		Sun, 8pm – Quiz 2 due
9	L9: Parallel Programming Models – II	T4: Shared-memory Problems		
10	L10: Message-passing Programming		L4: Introduction to Distributed-memory Programming	Mon, 2pm - Assignment 2 Sun, 8pm – Quiz 3 due
11	L11: Interconnection Networks		L5: Message-passing Programming with MPI	Wed, 2pm – Lab 4 due
12	L12: Energy-efficient Computing and Cloud Computing L13: Summary & Recap	T5: MPI		Sun, 8pm – Quiz 4 due
13	PH: No lecture		L6: Benchmarks & performance	Fri, 2pm – Assignment 3 (optional) Sun, 2pm – Lab 6 due
Exam	29 Nov, 9am - Exam In-person, open book			Wed, 29 Nov, 9am - Exam

Reference book

- Robert Robey, Yuliana Zamora, ”Parallel and High-Performance Computing”, Manning Publications



Supplementary Reference Book

- Parallel Programming for Multicore and Cluster Systems
  - by Thomas Rauber and Gudula Rünger
  - 1st Edition, Springer-Verlag, 2010
  - Ebook - NUS Digital Library - http://linc.nus.edu.sg/record=b2974382

- 2nd Edition (2013) available online