F21DP2 - Distributed and Parallel Technology
High-Level Parallel and Distributed Technologies
Learning Objectives
Phil Trinder

Multithreaded Computation
Understand the varieties of, and motivations for, multithreaded computation:
- concurrent
- parallel
- distributed
- mobile
Understand the additional coordination aspect of multithreaded computation,
Understand the levels at which coordination can be specified.

Haskell and GpH
Be able to read and write simple pure and stateful Haskell programs.
Know the definition of referential transparency, and it’s significance for programming languages.
Be able to read and write GpH programs, using existing and developing new evaluation strategies
Be able to apply common parallel paradigms, e.g.
- Divide and conquer (D&C)
- Data parallel
- Pipeline

High-level Parallel Program Development
Understand, and be able to apply the GpH development methodology, including effective use of profiling tools at each stage.
Be able to tune parallel performance by adjusting key coordination aspects with appropriate techniques, e.g.
- Thread granularity, e.g. D&C thresholds, chunking data parallelism.
- Reducing communication, e.g. chunking data parallelism
MultiCore Programming
To be aware of multi- and many-core architectures and their implications
To be able to read, write, control, and critique simple
• data parallel
• control parallel
OpenMP programs

Introducing Distribution
Understand the varieties of, and motivations for, distributed systems:
• Computational
• High Throughput
• Distributed Working
• Distributed Databases
Understand the additional functionality provided by distributed systems:
• Reliability
• Availability
• Scalability

Understand the distinction between large and small-scale distribution
Understand the additional concepts in distributed systems.
Be aware of common distributed paradigms:
• Client/Server
• Request Broker
• Publish/Subscribe

Socket-Based Distribution
Understand the Socket API including datagrams and streams
Be able to read and write simple Java Socket programs.
Be able to critique sockets as a distributed programming paradigm.
RPC/RMI
Understand the Remote Procedure Call concept.
Understand remote objects in object-oriented languages.
Be able to read and write simple Java/RMI programs.
Be able to critique Java/RMI as a distributed programming paradigm.

Grid Computing
Understand the Grid concept and common varieties:
- Distributed Supercomputing
- High-Throughput
- On-Demand
- Data-Intensive
- Collaborative
Be aware of the Grid layers and the Globus toolkit.

Mobile Computation
Understand the motivation for mobile computation, and its distinction from mobile computing.
To be able to critique mobile computation as a model of distributed computing.