Liverpool John Moores University

Title:	PARALLEL ALGORITHMS
Status:	Definitive
Code:	**6121COMP** (121296)
Version Start Date:	01-08-2021
Owning School/Faculty:	Computer Science and Mathematics
Teaching School/Faculty:	Computer Science and Mathematics

Team	Leader
Somasundaram Ravindran	Y

Academic Level:	FHEQ6	**Credit Value:**	20	**Total Delivered Hours:**	57
Total Learning Hours:	200	**Private Study:**	143		

Delivery Options
Course typically offered: Semester 2

Component	Contact Hours
Lecture	33
Seminar	22

Grading Basis: 40 %

Assessment Details

Category	Short Description	Description	Weighting (%)	Exam Duration
Report	AS1	Design and analyse algorithms.	40	
Exam	AS2	Examination	60	2

Aims

The aim of this module is to introduce the computational aspects of parallel and distributed computing and apply new techniques, methods and results from the rapidly-developing field of algorithms.

Learning Outcomes

After completing the module the student should be able to:

1	Justify a variety of advanced algorithmic techniques to solve problems
2	Construct algorithms for execution in parallel and distributed settings.
3	Analyse parallel algorithms for correctness and performance.
4	Apply appropriate distributed and/or parallel algorithms to practical situations by taking into account tractable and intractable problems.

Learning Outcomes of Assessments

The assessment item list is assessed via the learning outcomes listed:

Algorithm Analysis	2	3
Exam	1	4

Outline Syllabus

• Models of Parallel & Distributed computation: PRAM, BSP, Distributed Objects, Message Passing
• Amdahl's Law and Gustafson's Law
• Parallel Decomposition techniques (Task-based decomposition, Data-parallel decomposition)
• Basic techniques: Prefix sums and doubling
• Graph algorithms: Parallel computation on trees, paths, spanning trees and connected components.
• Theoretical and physical topologies for parallel and distributed computing
• Communication and coordination / synchronisation
• P-complete, NC problems.
• Distributed models and algorithms: termination, failure tolerance and distributed search.
• Parallelism vs Concurrency.
• Shared resource access; synchronisation, mutual exclusion, atomicity
• Races and Deadlocks

Learning Activities

Lectures are supported by seminars.

Notes

The module combines many of the concepts taught over the course of the Computer Science Programme such as complexity theory and algorithm design, and provides important skills for work with large applications since these usually must be implemented on a parallel or distributed system, due to their memory space and

speed requirements.