

**Module Description, available in: EN, FR**

## *Parallel and distributed computing*

**General Information****Number of ECTS Credits**

3

**Module code**

TSM\_ProgAlg

**Valid for academic year**

2019-2020

**Last modification**

2018-11-02

**Responsible of module**

Pierre Kuonen (HES-SO, pierre.kuonen@hefr.ch)

**Explanations regarding the language definitions for each location:**

- Instruction is given in the language defined below for each location/each time the module is held.
- Documentation is available in the languages defined below. Where documents are in several languages, the percentage distribution is shown (100% = all the documentation).
- The examination is available 100% in the languages shown for each location/each time it is held.

	Berne	Lausanne	Lugano	Zurich
<b>Instruction</b>		X F 100%		X E 100%
<b>Documentation</b>		X F 10% X E 90%		X E 100%
<b>Examination</b>		X F 100%		X E 100%

**Module Category**

TSM Technical/scientific specialization module

**Lessons**

2 lecture periods and 1 tutorial period per week

**Entry level competences****Prerequisites, previous knowledge**

- Procedural and object oriented programming
- Software engineering (UML or other)
- Basic notions of algorithms and complexity
- Basic notions of concurrent programming (Threads)

## Brief course description of module objectives and content

The objective of this module is to present an overview of parallel and distributed computing and related algorithms. The first part of the course will be dedicated to the architectures of parallel and distributed infrastructures, the different theoretical models for these architectures and the different programming models and tools for programming such architectures. The second part will be dedicated to the study of a number of classical parallel and distributed algorithms. This course includes practical work to train the student in the use of parallel and distributed computing.

## Aims, content, methods

### Learning objectives and acquired competencies

At the end of the course the student knows:

- The most common parallel and distributed hardware infrastructures
- The different ways to model and efficiently program these architectures
- How to choose the proper parallel or distributed algorithm to write an application for solving a specific problem on a specific architecture
- How to efficiently program this application
- How to assess the performance of this application

### Contents of module with emphasis on teaching content

Introduction

- Different architectures of parallel and distributed infrastructures
- Communications models and communication costs
- Performance metrics for parallel and distributed systems
- Scalability of parallel and distributed systems

Heterogeneous shared memory systems

- Architecture of widely used multi-core systems
- Parallel programming models (OpenMP)

Distributed memory systems

- Communication operations and their costs
- Message passing paradigm (MPI)
- Distributed object paradigm

Parallel and distributed algorithms

- Asymptotic analysis of parallel programs
- Decomposition techniques
- Mapping techniques for load balancing
- Matrix-vector and matrix-matrix multiplication
- Parallel and distributed sorting algorithms
- Parallel and distributed Graph algorithms

### Teaching and learning methods

This course involves theoretical presentations and practical exercises or laboratories. Some of the exercises or laboratories are programming exercises that can be done at home by accessing a parallel and distributed infrastructure made available through the internet.

### Literature

- A. Introduction to Parallel Computing, Zbigniew J. Czech, Cambridge University Press, 2017
- B. An Introduction to Parallel Programming, 1st edition, Peter Pacheco, Morgan Kaufmann Publishers Inc, 2011

## Assessment

### Certification requirements

Module uses certification requirements

### Certification requirements for final examinations (conditions for attestation)

Some exercises or laboratories could be mandatory.

### Basic principle for exams

**As a rule, all the standard final exams for modules and also all repetition exams are to be in written form**

### Standard final exam for a module and written repetition exam

Kind of exam

written

**Duration of exam**

120 minutes

**Permissible aids**

*Aids permitted as specified below:*

**Permissible electronic aids**

No electronic aids permitted

**Other permissible aids**

A handwritten summary of a fixed number of pages given by the lecturer.

**Special case: Repetition exam as oral exam**

**Kind of exam**

oral

**Duration of exam**

30 minutes

**Permissible aids**

No aids permitted

Description du module, disponible en: EN, FR

## Calcul parallèle et distribué

### Informations générales

Nombre de crédits ECTS

3

Code du module

TSM\_ProgAlg

Valable pour l'année académique

2019-2020

Dernière modification

2018-11-02

Nom du/de la responsable de module

Pierre Kuonen (HES-SO, pierre.kuonen@hefr.ch)

Explication des définitions de langue par lieu :

- Les cours se dérouleront dans la langue définie ci-dessous par lieu/exécution.
- Les documents sont disponibles dans les langues définies ci-dessous. Pour le multilinguisme, voir la répartition en pourcentage (100% = documents complets)
- L'examen est disponible à 100% dans chaque langue sélectionnée pour chaque lieu/exécution.

	Berne	Lausanne	Lugano	Zurich
<b>Leçons</b>		X F 100%		X E 100%
<b>Documentation</b>		X F 10%	X E 90%	X E 100%
<b>Examen</b>		X F 100%		X E 100%

Catégorie de module

TSM approfondissement technico-scientifique

Leçons

2 leçons et 1 leçon de pratique par semaine

### Compétences préalables

Connaissances préalables, compétences initiales

- Programmation procédurale et orientée-objet
- Ingénierie logicielle (UML ou autre)
- Notions de base en algorithmique et complexité
- Notions de base en programmation concurrente (processus)

## Brève description du contenu et des objectifs

L'objectif de ce module est de présenter un aperçu de l'informatique parallèle et distribuée ainsi que des algorithmes y afférant. La première partie du cours est consacrée aux architectures parallèles et distribuées, les différents modèles théoriques de ces structures et les différents modèles et outils pour programmer ces architectures. La deuxième partie sera dévolue à l'étude de certains algorithmes parallèles et distribués classiques. Ce cours comprend une partie pratique pour familiariser les étudiants à l'usage de l'informatique parallèle et distribuée.

## Objectifs, contenus, méthodes

### Objectifs d'apprentissage, compétences à acquérir

Au terme du cours, l'étudiant:

- connaît les infrastructures matérielles parallèles et distribuées les plus courantes;
- connaît les différentes méthodes de modélisation de ces architectures;
- sait choisir entre algorithme parallèle ou distribué selon le cas pour écrire une application qui résout un problème spécifique sur une architecture donnée;
- est capable de programmer efficacement cette application;
- sait comment évaluer la performance de cette application.

### Contenu des modules avec pondération du contenu des cours

Introduction

- architectures parallèles et distribuées
- modèles et coûts de communication
- métriques et indicateurs de performance
- scalabilité de systèmes parallèles et distribués

Systèmes parallèles hétérogènes

- architecture des infrastructures multicore
- modèles et outils de programmation ( OpenMP)

Systèmes distribués

- opérations de communication et coûts
- transmission de message (MPI)
- programmation objet distribués

Algorithmes parallèles

- analyse asymptotique des coûts des algorithmes parallèles et distribués
- techniques de décomposition
- techniques de dispatching et répartition de charge
- produit matrice-vecteur et matrice-matrice
- algorithmes de tri parallèle et distribué
- algorithmes pour les graphes

### Méthodes d'enseignement et d'apprentissage

Ce cours contient des présentations théoriques ainsi que des exercices pratiques ou laboratoires. Une partie des exercices sont des exercices de programmation qui peuvent être réalisés à la maison en accédant à une infrastructure parallèle et distribuée rendue accessible via internet.

### Bibliografie

- A. Introduction to Parallel Computing, Zbigniew J. Czech, Cambridge University Press, 2017
- B. An Introduction to Parallel Programming, 1st edition, Peter Pacheco , Morgan Kaufmann Publishers Inc, 2011

## Evaluation

### Conditions d'admission

Le module utilise les conditions d'admission

### Conditions d'admission à l'examen de fin de module (exigences du certificat)

Certains exercices ou laboratoires peuvent être obligatoires.

### Principe pour les examens

**En règle générale, tous les examens de fin de module réguliers et les examens de rattrapage sont organisés sous la forme écrite**

### Examen de fin de module régulier et examen écrit de répétition

Type de l'examen

écrit

**Durée de l'examen**

120 minutes

**Aides autorisés**

*Les aides suivantes sont autorisées:*

**Aides électroniques autorisées**

Aucune aide électronique autorisée

**Autres aides autorisées**

Un résumé manuscrit d'un nombre fixe de pages indiqué par l'enseignant.

#### Cas spécial: examen de répétition oral

**Type de l'examen**

oral

**Durée de l'examen**

30 minutes

**Aides autorisés**

Sans aides