### Summary

The course introduces parallel programming models, algorithms, and data structures, map-reduce frameworks and their use for data analysis, as well as shared-memory concurrency.

### Content

See https://lara.epfl.ch/w/parcon17:top
Parallel programming & execution models
Functional parallelism
Data-level parallelism
Threads and fork/join parallelism
Synchronization
Threads and Shared Memory in Java
Futures
Large-Scale Parallel programming using Apache Spark

### Keywords

Parallelism, threads, synchronization, locks, memory models.

### Learning Prerequisites

**Required courses**
- Functional programming (CS-210)
- Algorithms (CS-250)
- Computer Architecture (CS-208)

**Recommended courses**
System oriented programming (CS-207)

### Important concepts to start the course

Functional programming and functional data structures
Algorithms and data structures

### Learning Outcomes

By the end of the course, the student must be able to:
- Construct parallel software.
• Perform tuning parallel software.

Teaching methods
Ex cathedra, labs, exercices

Assessment methods
Programming assignments (30%); final exam in August (70%)

Resources
Notes/Handbook
Lecture notes, copies of the slides