PARALLEL COMPUTING

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT I WIEN GRAZ

623.714 Selected Topics in Distributed Multimedia Systems - November 2018

Parallel computing enables the use of computer simulation of complex systems, also called the third pillar of science. Many fields of knowledge, such as medicine, weather forecast, automobile industry, among many others, resort to computer simulation for new developments. Additionally, parallel computing is becoming the current computing paradigm as hardware tends to multi-processing units. The common desktop and laptop are today built with a multicore processor that collectively has more processing power, than a single core processor, but cores are individually less powerful. Accelerators, such as GPUs, are also becoming a commodity, which allows the common user to have access to high-performance machines. Programmers will have to deal with multiprocessor architectures in order to use effectively the machines of today and of the future.

There are several approaches to program a multicore machine, being the shared memory model the straight step to implement a parallel version of a given sequential code. The well known library pthreads allows the implementation of this model but with a level of detail that is not always desirable. To overcome this burden, OpenMP emerged as a reliable and efficient library to develop shared memory programs on multicore processors. For GPUs there are special languages, such as CUDA and OpenCL, that follow also a shared memory approach.

The distributed memory model is another approach to obtain a parallel version of a sequential code, and is suitable when more than one processor cooperates to run a given application. MPI is the most popular library to implement this model, which is based on the exchange of messages among processors.

Ass.-Prof. Jorge Barbosa
Universidade do Porto (FEUP)
https://web.fe.up.pt/~jbarbosa/en/

The objective of this course is to provide an extensive overview of the field of parallel computing, with emphasis on the use of OpenMP and MPI to develop parallel programs for multicores, and an introduction to GPU programming using CUDA and OpenCL.

At the end of this course, participants should be able to:

- Understand the fundamentals of parallel computing
- Analyze a problem and identify the adequate parallelization model
- Write message-passing and shared memory programs
- Design parallel solutions for new problems
- Use computational models to estimate applications performance

Timetable for the course:

November 8-10, 2018
November 22-24, 2018

LOGIN @ AAU.AT for detailed information and registration!