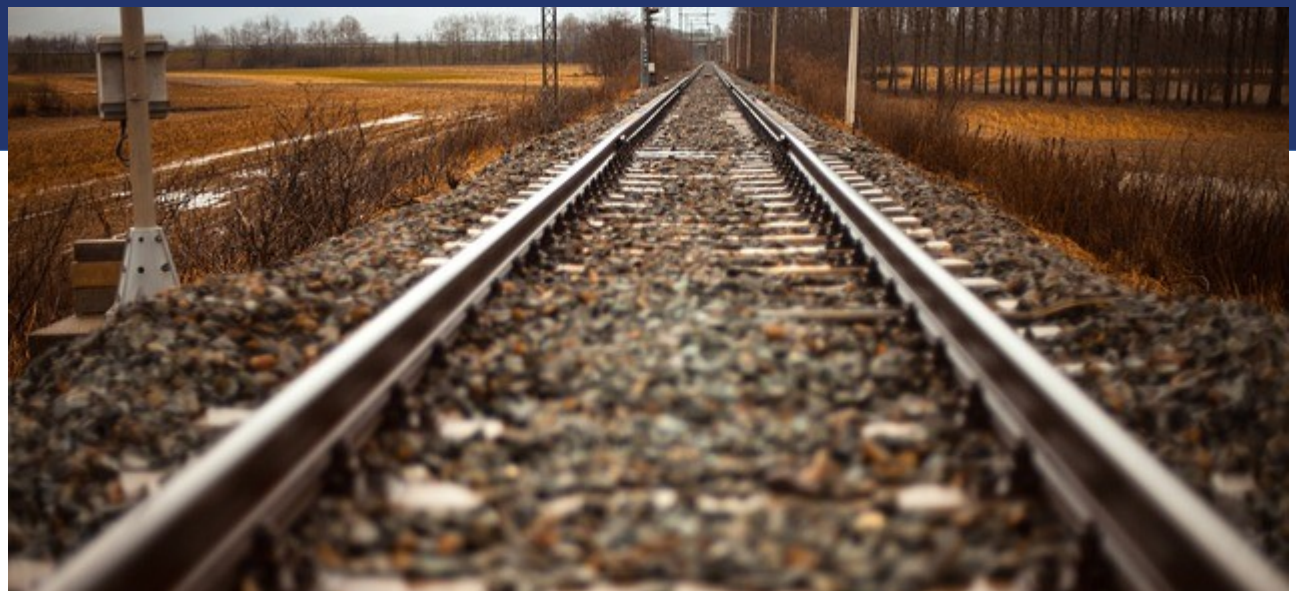
Parallel programming

Introduction

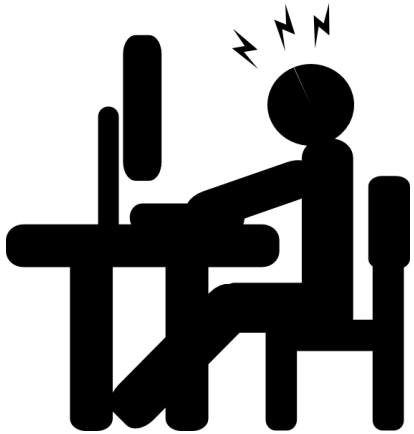


FAKULTA
ELEKTROTECHNICKÁ
ČVUT V PRAZE

What is the aim of the labs?

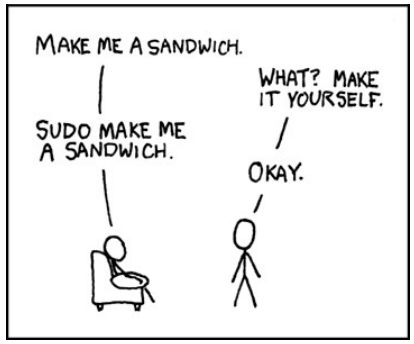
- To get the feel for parallel programming
  1) Understand what makes the parallelisation **complicated**
  2) Which **problems** can occur during the paralellisation
  3) What can be a **bottleneck**
  4) How to think about **algorithms** from the paralellisation point of view

- To get basic skills in common parallel programming frameworks
  1) for Multicore processors – C++11 threads, OpenMP
  2) for Computer clusters – OpenMPI

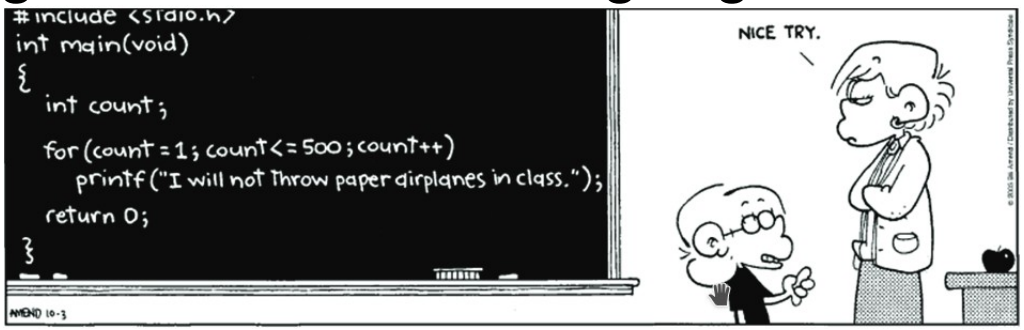


What this course requires?

- Basic skills with Linux – shell, ssh, etc.



- Knowledge of C and C++ language



- Analytical thinking and opened mind

Web

- Course page https://cw.fel.cvut.cz/wiki/courses/b4m35pag/start
  - Plan of the labs, grading

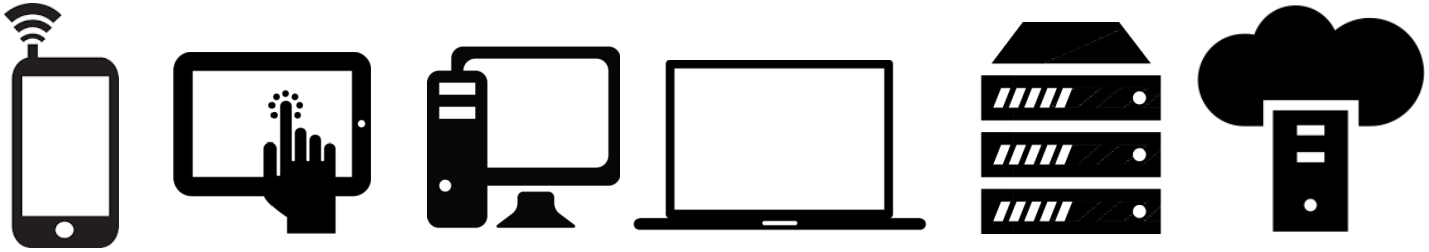
Parallel programming – the first cut

No questions?

Let's start with our business!

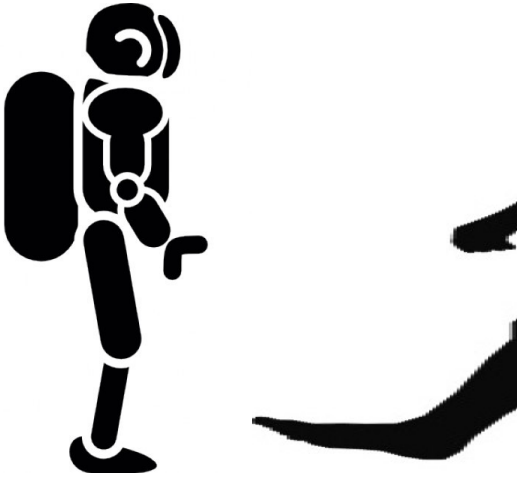
Why should you care about it?

- Parallel computing is a dominant player in scientific and cluster computing. Why?
  - Moore law is reaching its limits
    - Increase in transistor density is limited
    - Memory access time has not been reduced at a rate comparable with processing speed
- How to get out of this trap?
  - Most promising approach is to have multiple cores on a single processor.
  - Parallel computing can be found at many devices today:



Ok; However, It should be task for compiler and not for me!!!

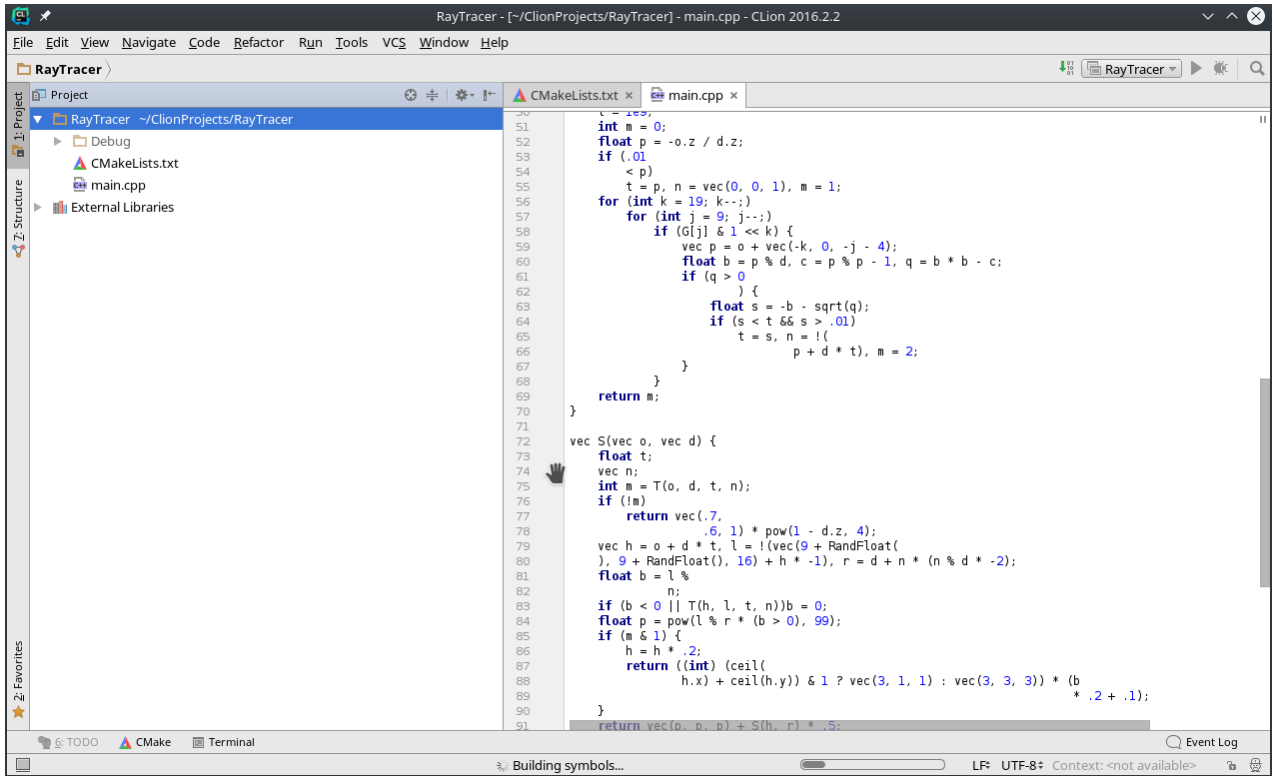
- Yes, compiler can help you, but without your guidance, it is not able pass all the way to the successful result.
  - Parallel programs often look very different than sequential ones.
  - An efficient parallel implementation of a serial program may not be obtained by simply parallelizing each step.
  - Rather, the best parallelization may be obtained by stepping back and devising an entirely new algorithm.



That was theory and now something practical!

Clion IDE

- Download - https://download.cvut.cz, JetBrains
- Licence server - https://turnkey.felk.cvut.cz/



Debugging and Profiling

- **Intel Parallel Studio** - a software development suite that facilitates native code development on Windows, macOS and Linux in C++/C and Fortran for parallel computing.
  - Intel C++ Compiler with OpenMP
  - Intel Fortran Compiler with OpenMP
  - IDE plug-in integration with Visual Studio, Eclipse and Xcode
  - Debugging and profiling tools
  - Free for students: https://software.intel.com/en-us/qualify-for-free-software/student

Debugging and Profiling

- **Intel VTun Amplifier** - performance profiling tool for C, C++, and Fortran code. It can identify where in the code time is being spent in both serial and threaded applications.
- https://www.youtube.com/watch?v=rRtef997xww&t=20s

- **Intel Inspector** - memory and threading error debugger for C, C++, and Fortran.
- https://www.youtube.com/watch?v=JM603bqPKaU&t=205s

Debugging and Profiling

- **Intel Trace Analyzer and Collector** - tool for understanding MPI application behavior, finding bottlenecks, improving performance.

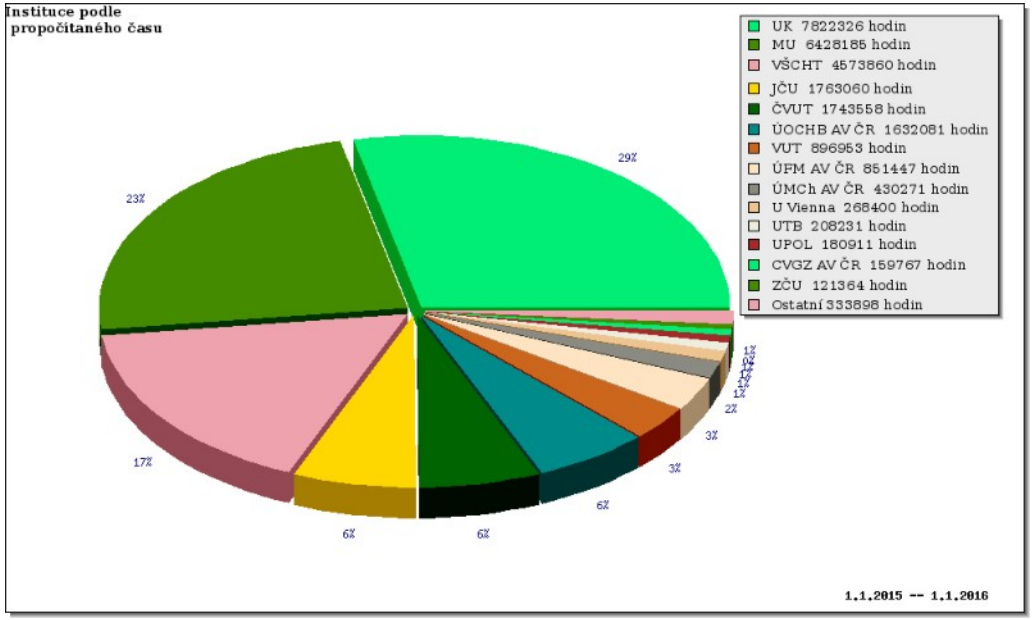
- **Intel Advisor** - is a SIMD vectorization optimization and shared memory threading assistance tool for C, C++ and Fortran.
- https://www.youtube.com/watch?v=MsWq2gwrrgU

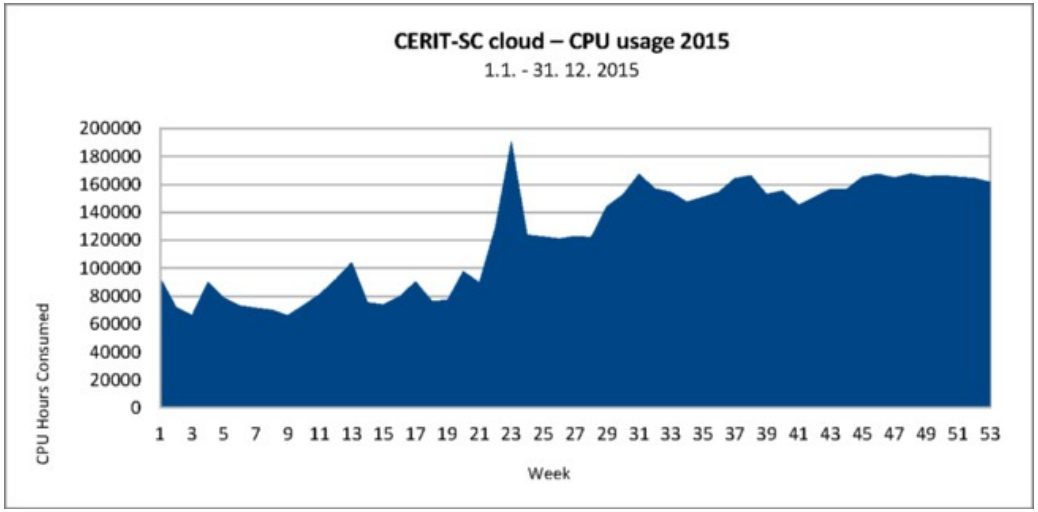
Debugging and Profiling

- **Valgrind** - a tool suite providing a number of debugging and profiling tools
  - **Memcheck** is a memory error detector. It helps you make your programs, particularly those written in C and C++, more correct.
  - **Cachegrind** is a cache and branch-prediction profiler. It helps you make your programs run faster.
  - **Callgrind** is a call-graph generating cache profiler. It has some overlap with Cachegrind, but also gathers some information that Cachegrind does not.
  - **Helgrind** is a thread error detector. It helps you make your multi-threaded programs more correct.

Metacentrum system

- operates and manages **distributed computing** infrastructure consisting of computing and storage resources owned by **CESNET**
- MetaCentrum membership is free for researchers and students of academic institutions in the Czech Republic





MetaCentrum – Sign up

- How to sign up

That was nice, wasn't it?

Thank you for your attention...