Programming Tools for Distributed and Parallel Systems (SPD)

- Teacher name: Massimo Coppola, massimo.coppola@isti.cnr.it, 050 621 2992

- When, how much : 6 credits, 2nd semester
- Exam mode: lab project + written report + oral (includes project discussion)
- Pre-requisites: HPC; SPM is strongly suggested (possibly during the same term)
- Area: Computer Science
- Course home page https://didawiki.cli.di.unipi.it

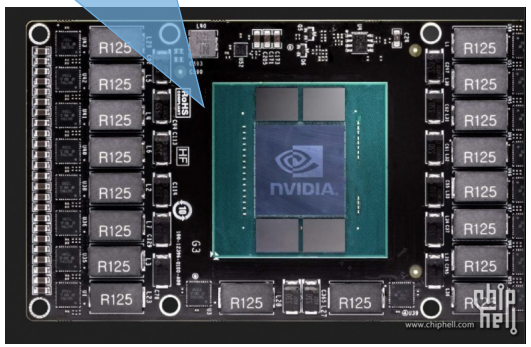
The course relies on knowledge about parallel skeletons, their performance models and techniques to exploit them in the design and evaluation of parallel software.

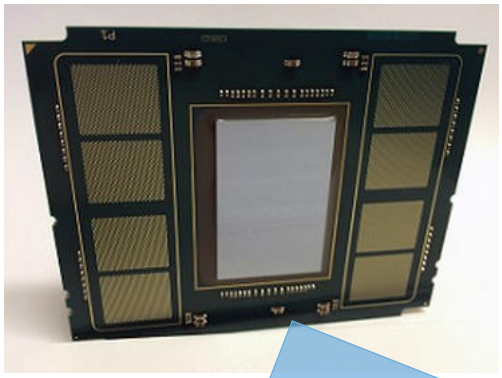
The course presents a selection of parallel and distributed programming languages and frameworks, covering parallelism exploitation at different scales.

We address exploitation of parallelism via software at different architectural levels, targeting distributed systems, shared-memory/multicore CPUs and GPUs

A changing landscape where parallelism is pervasive

Nvidia Pascal GPU architecture
2560 shader cores on-chip





Intel's Xeon Phi Knight's Landing
72-cores with 4 HW threads/core

RISC-V FPGA CPUs up
to 1680 cores/board

Fujitsu K supercomputer
705000 SPARC VIIIfx cores
post-K supercomputer in 2021
will use custom ARMV8 CPUs

Home made cluster
of 120 Raspberry PI
(ARM 32 bit core)

Syllabus

Parallel tools & platforms for HPC and large scalable systems. Lessons + lab time

- MPI – Message Passing Interface standard
  - Message passing standard, linked library with support for multiple languages
- TBB – Intel Thread Building Blocks library
  - C++ template library for shared memory multi-thread programming
  - Multi core CPUs and multiprocessor systems
- OpenCL – High-level, portable standard to exploit many-core on-chip parallelism
  - Multithread, high-memory bandwidth algorithms with streaming/regular access patterns
  - Targets graphic units (GPUs), CPU vectorization, APUs, FPGA devices …
- Other frameworks
  - Assist, BSP/Map&Reduce based (Spark / Graphx, Hama)
- Some application examples for laboratory time (change from year to year):
  Data Mining, Graph and Optimization Algorithms, Stream Data Processing

Some potential topics for Master Thesis or Research fellowships

- Clouds, Cloud-Federations and *Edge* / *Fog* computing:
  - Dynamical System Modeling, Resource Brokering, Scheduling Optimization strategies
    - Hierarchical and skeleton-based programming frameworks and performance models
    - Genetic programming, (mixed integer) linear programming, other optimization approaches to brokering and autonomic/adaptive resource management
    - Container-based and VM-based application composition, deployment and elastic scalability
    - High-performance implementation of authorization mechanisms for data security and privacy: Scalable policy evaluation and enforcing mechanisms at the hypervisor, cloud and/or federation manager levels as well as on edge devices
- Multicore CPU/GPU design and deployment on FPGA
- High-performance computing applications
  - HPC for Data Mining, Stream Mining, Data Analysis and Machine Learning
  - Applications to HealthCare
  - Application of stream and Big-data Analysis for Clouds