

Parallelisation in HPC

# Intro to parallel computing

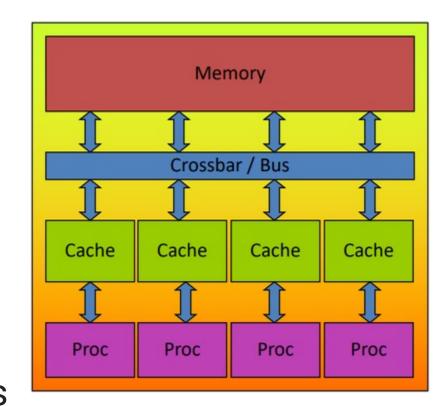
Main objective: reduce Time To Results

Currently, two main concepts:

- Shared memory: parallel threads, auto-parallelization, vectorization
- Distributed memory: parallel tasks (MPI), offloading (using a GPU)

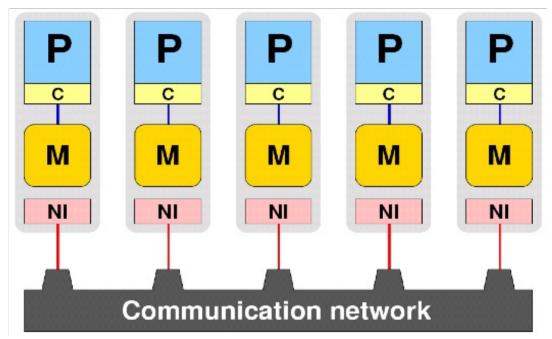
## Shared memory

- 1) Parallel threads (threading)
- Relies mostly on OpenMP
- All processors/cores access a shared main memory and employ multiple threads
- 2) Vectorization (SLURM job arrays)
- Multiple instances of the same program execute on multiple data files simultaneously
- Can be done via job arrays, best results with parallel filesystem



#### Distributed memory

- 1) Parallel tasks
- Runs tasks across multiple nodes on a cluster
- Requires inter-node communication
- MPI standard (implementations: OpenMPI, Intel MPI, mpich)
- 2) Offloading
- The use of an accelerator like a general purpose GPU
- CUDA, OpenACC, OpenMP target directive



## Practice 1: SLURM job arrays



https://help.accre.vanderbilt.edu/index.php?title=Parallel Processing and Job Arrays

#### Practice 2: Parallelization with MPI



https://ulhpc-tutorials.readthedocs.io/en/latest/bio/basics/

#### References

#### Web references:

- [UCLA2019] The ParFlow Hydrologic Model: HPC Highlights and Lessons Learned retrieved from <a href="https://www.slideserve.com">https://www.slideserve.com</a> on 30.08.2023
- [HPCW2023] Parallel Programming retrieved from <a href="https://hpc-wiki.info/hpc/Parallel\_Programming">https://hpc-wiki.info/hpc/Parallel\_Programming</a> on 30.08.2023