# MARKO MLADENOVIC

# MONTE CARLO CODE PARALLELIZATION

Scientific Computing Laboratory
Institute of Physics Belgrade

University of Belgrade
Department of Nanoelectronics
and Photonics

# DAY 1: OPTIMIZATION AND PROFILING

**Optimization:**
- O2, -O3, -fast, -unroll
- No significant improvement

**Profiling:**
- gprof, valgrind
- conclusion: code should be parallelized

# DAY 2: OpenMP IMPLEMENTATION

**1.**

- Several Monte Carlo steps at the same time on different threads
- Chosen molecules should be independent
- Trivial problem
- **#pragma omp parallel private(tid,…)**

**2.**

- Energy calculation function parallelization
- Trivial problem
- **#pragma omp parallel for private(rxi, ryi, rzi, rxij, ryij, rzij, vij, wij) reduction(+ : vv, ww)**

# DAY 2: OpenMP IMPLEMENTATION

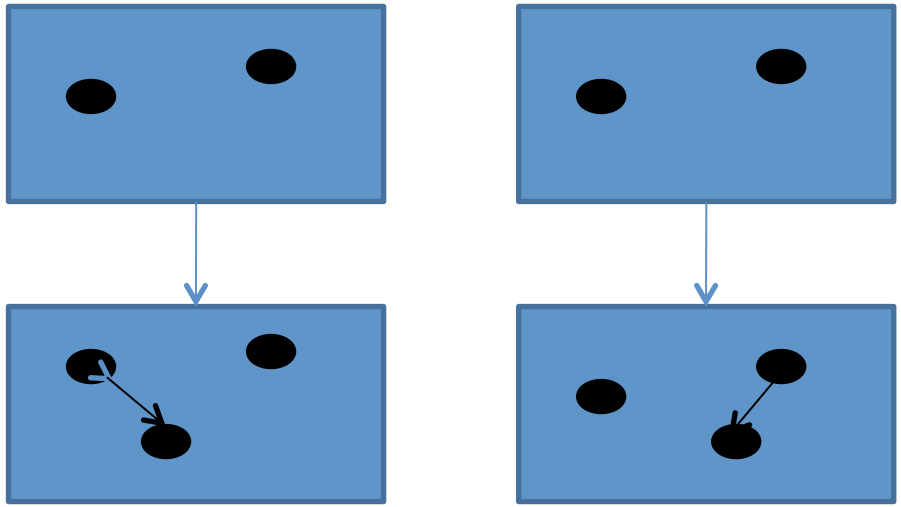
-icc -openmp -i-static naphtalene2.cpp -o naphtalene.ex

```
#!/bin/bash
#PBS -q hpsee
#PBS -l nodes=1:ppn=8
#PBS -l walltime=00:10:00
#PBS -e ${PBS_JOBID}.err
#PBS -o ${PBS_JOBID}.out

cd $PBS_O_WORKDIR

export OMP_NUM_THREADS=8
chmod +x naphtalene.ex
./naphtalene.ex > tmp
```

# DAY 3: MPI IMPLEMENTATION

**1.**

- Several Monte Carlo steps at the same time on different processors
- Chosen molecules should be independent
- Non-trivial problem



**MPI_Allreduce(change, change_all, 1 , MPI_DOUBLE, MPI_SUM, MPI_COMM_WORLD);**

# DAY 3: MPI IMPLEMENTATION

```
mpic++ naphtalene2.cpp -o naphtalene.ex  -lm
```

```
#! /bin/bash
#PBS -q hpsee
#PBS -l nodes=2:ppn=1
#PBS -l walltime=12:00:00
#PBS -e ${PBS_JOBID}.err
#PBS -o ${PBS_JOBID}.out

cd $PBS_O_WORKDIR
cat $PBS_NODEFILE

${MPI_MPICH2_MPIEXEC} --comm=pmi ./naphtalene.ex > tmp
```

# DAY 4: TESTING

-correct results
-parallel  Monte Carlo will reach equilibrium configuration faster than serial (less steps needed)
-OpenMP implementation is faster the MPI (due to additional variables and MPI_Allreduce routine used in MPI implementation)
-energy calculation routine speed-up: 7/2 = 3.5, 8 threads (can be increased)