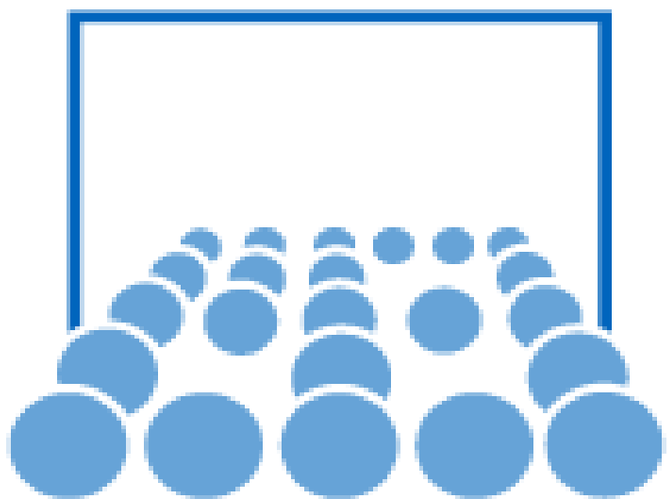
PSE Molekulardynamik

OpenMP

Philipp Neumann, Nikola Tchipev

18.12.2015



Outline

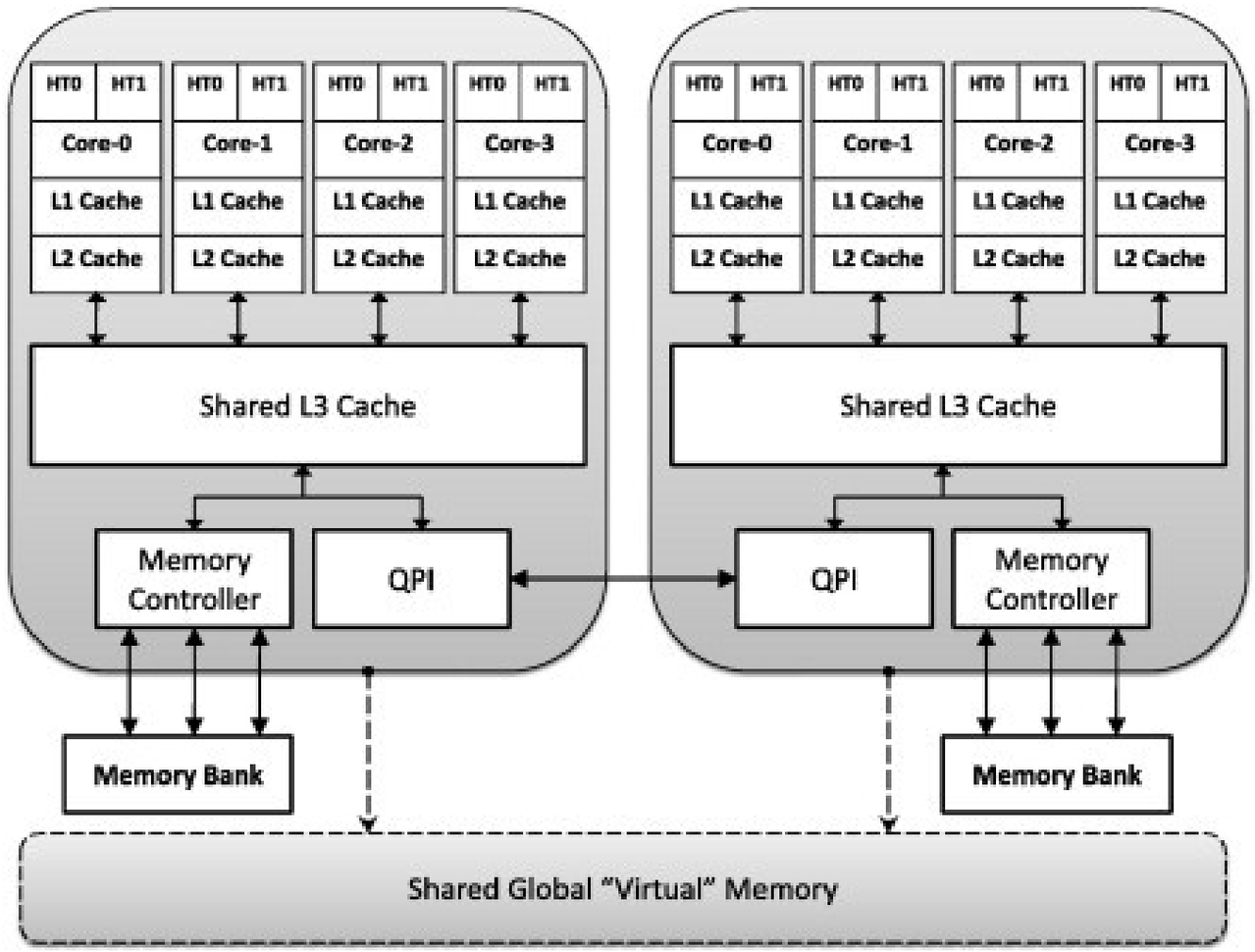
- Schedule
- Presentations: Worksheet 4
- Multicore Architectures
- OpenMP
- Membrane, Crystallization

Schedule

16.10.2015	WS1
30.10.2015	WS2/Review WS1
13.11.2015	WS3/Review WS2
04.12.2015	WS4/ Review WS3
18.12.2015	WS5/Review WS4
22.01.2016	Review WS5

Presentations: Worksheet 4

Multicore Architectures: Nehalem EP



Çatalyüreka et. al, Graph coloring algorithms for multi-core and massively multithreaded architectures
Parallel Computing
Volume 38, Issues 10–11, October–November 2012, Pages 576–59

Revision: MAC Cluster

- http://www.mac.tum.de/wiki/index.php/MAC_Cluster
- Read usage instructions carefully!
- Here: MAC-Cluster AMD-BDZ / Intel SNB Partition

	MAC-Cluster SNB Partition	MAC-Cluster BDZ Partition
Architecture	Intel SandyBridge-EP Xeon E5-2670 (2.6 GHz)	AMD Opteron 6274 (2.2 GHz)
#Nodes	28	19
#Processor cores	448	1216
Aggregate peak performance (Tflop/s)	9.3	10.7
Aggregate memory (TByte)	3.5	4.9

OpenMP: Overview

- Shared memory parallelization
- Define parallel regions that are executed by each thread
- Work flow:
  - Define parallel regions and parallelization via pragmas
  - Compile your code:

    ```
    g++ -fopenmp -lgomp my_main.cpp -o main
    ```
  - Run your simulation with, e.g., N=8, threads:

    ```
    export OMP_NUM_THREADS=8; ./my_main
    ```

    Alternative: use `omp_set_num_threads(8);`

OpenMP - Pragmas

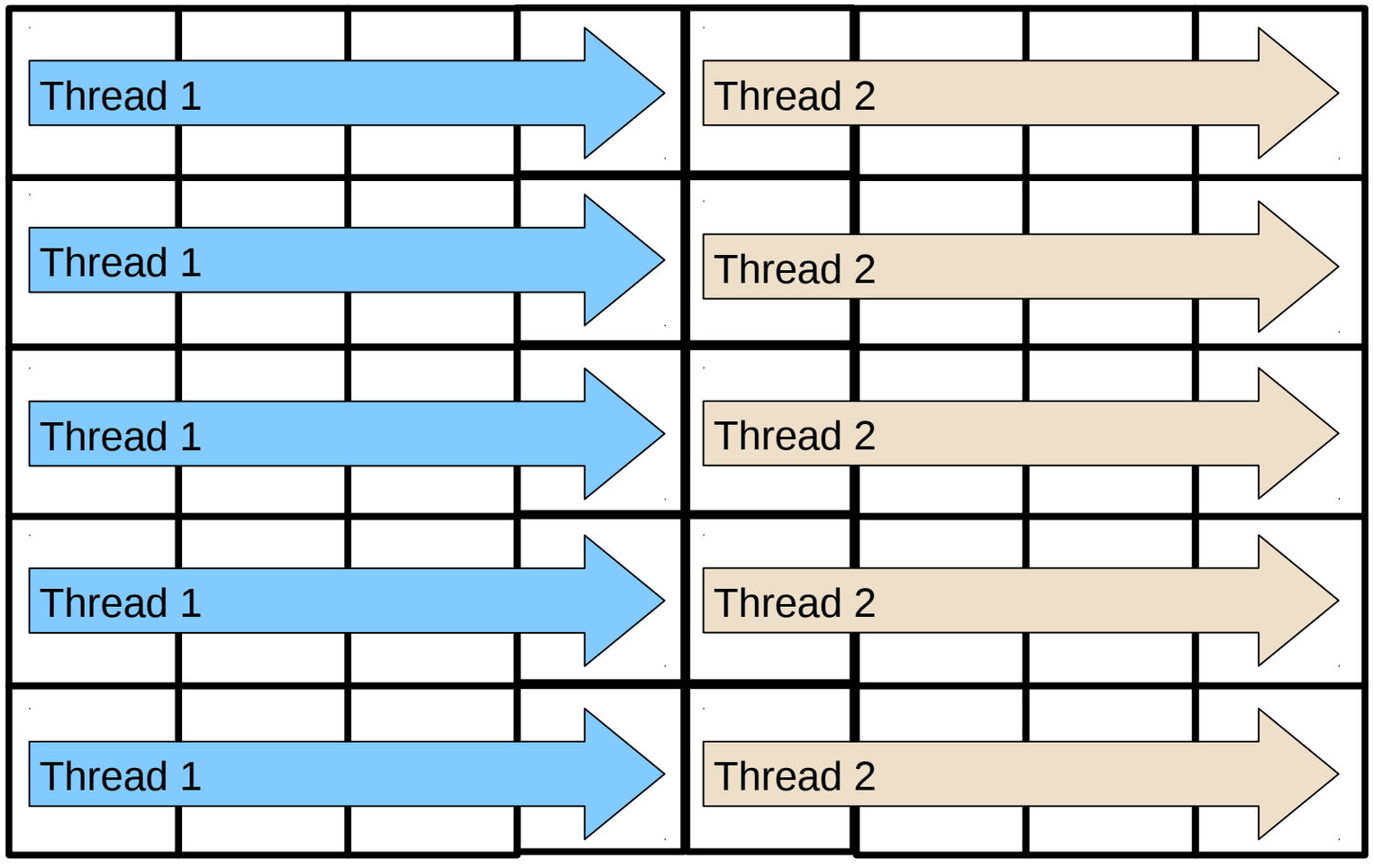
- `#pragma omp parallel`
  → defines parallel region
  → each thread executes this region
- `#pragma omp parallel for`
  `for (int i=0; i < N; i++){ … }`
  → splits loop on all threads
- `#pragma omp private(myVar)`
  `for (int i=0; i < N; i++){ myVar=foo(); … }`
  → variable myVar is allocated separately for each thread
- `#pragma omp critical`
  → lets only one thread enter the following region at a time
  → other threads have to wait
- And much more: dynamic vs. static scheduling, reduce-operations, ...

OpenMP – References, Tasks, and Hints

- References
  - https://computing.llnl.gov/tutorials/openMP/
  - http://openmp.org
  - Hager, Wellein: Introduction to High Performance Computing for Scientists and Engineers. CRC Press, ISBN 978-1439811924, 356 pages, July 2010
  - http://icl.cs.utk.edu/papi/

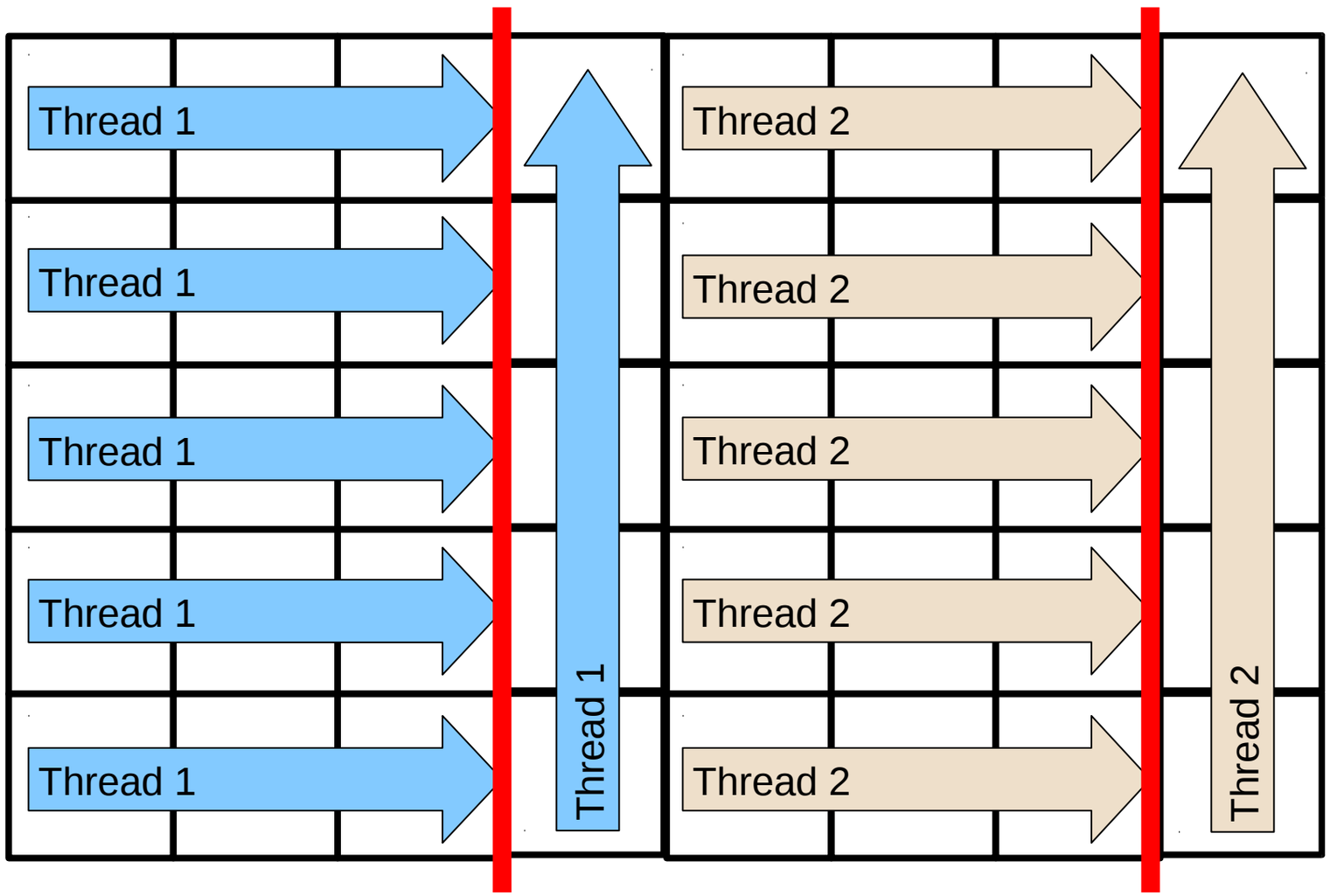
- Make yourself familiar with given hardware and shared memory parallelization
- Parallelize compute intensive parts first
  - Here: Loop for force calculation
- (CC)NUMA: Allocate memory in parallel

OpenMP for Linked Cell Algorithm



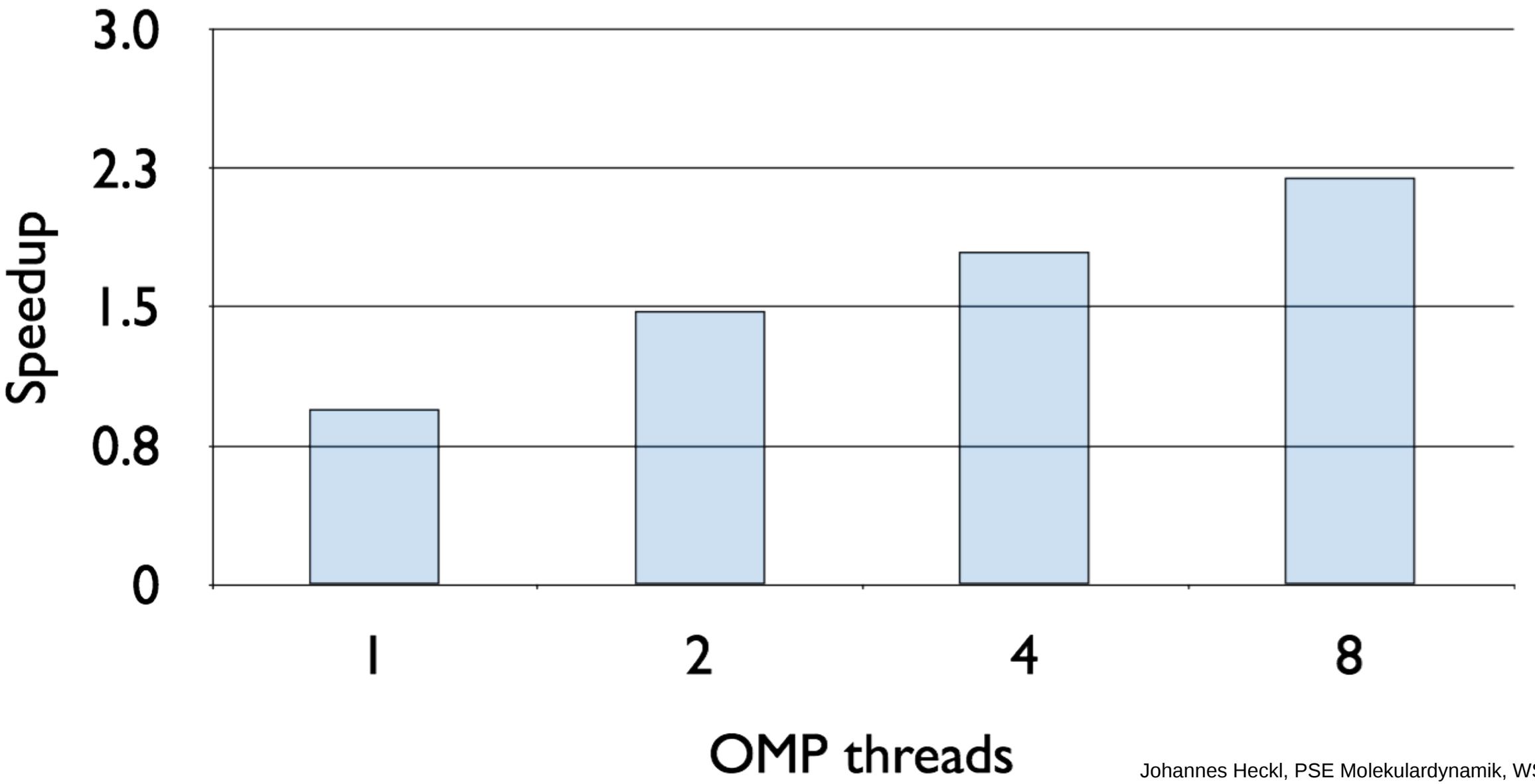
Problem: Synchronisation for boundary cells between Thread 1 and 2, due to Newton 3.

OpenMP for Linked Cell Algorithm



Possible solution: - Compute inner cells
- Synchronize all threads
- Compute boundary cells

OpenMP: Molecular Dynamics - 2011/2012



Johannes Heckl, PSE Molekulardynamik, WS11/12