Masterpraktikum Scientific Computing – High Performance Computing

SWE Project Presentation and Results

Gaurav Kukreja

Evangelos Drossos

21.01.2014

Optimizations Performed

- Instrumentation with Scalasca
- Vectorization with Intel Cilk
- OpenMP optimization
- MPI optimization

Instrumentation

- Instrumented different variants of the code to get a better understanding of performance issues and improvement potential

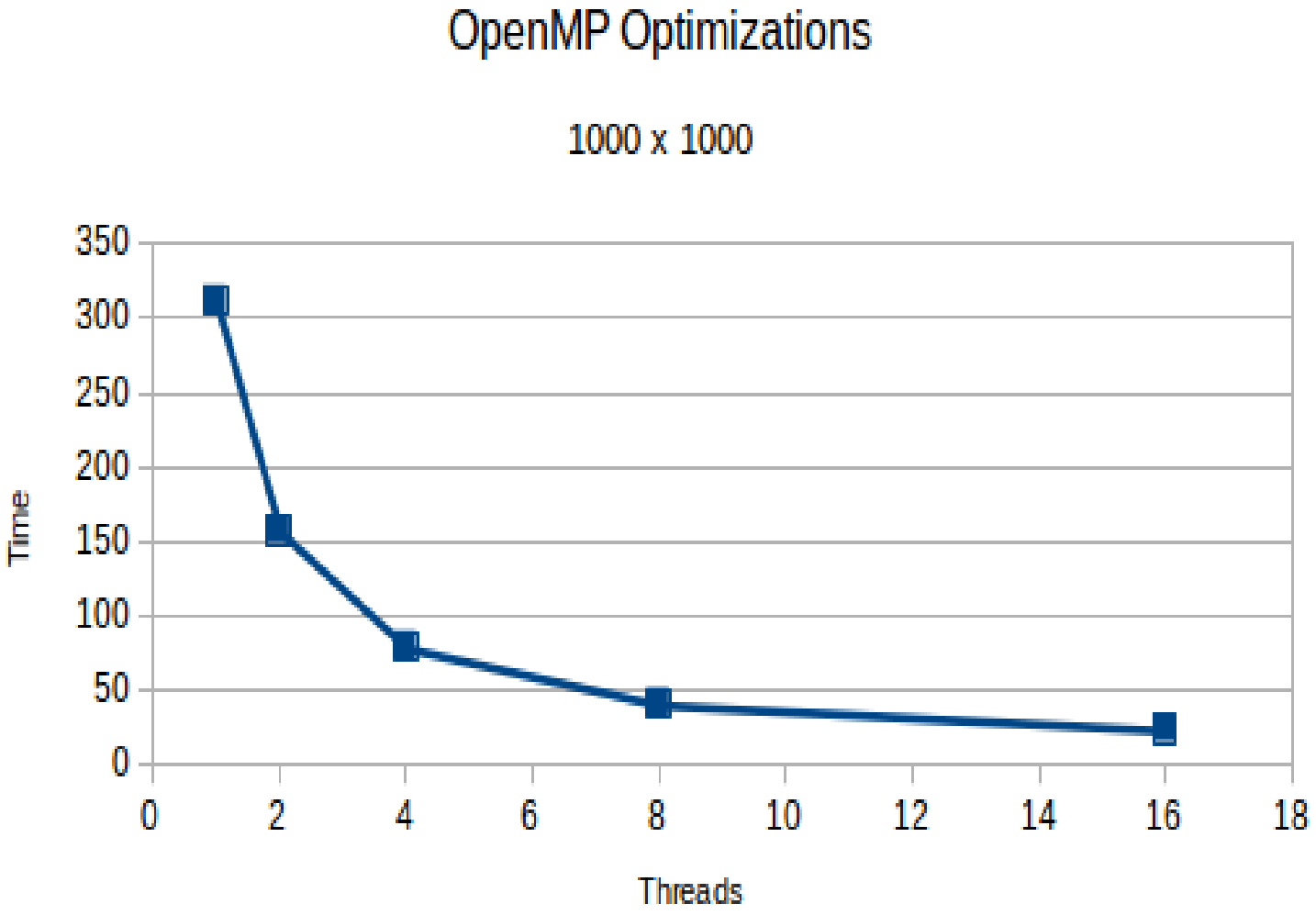
- We analyzed the code for every round of optimizations we performed until reaching the final status (Cilk + OpenMP + MPI)

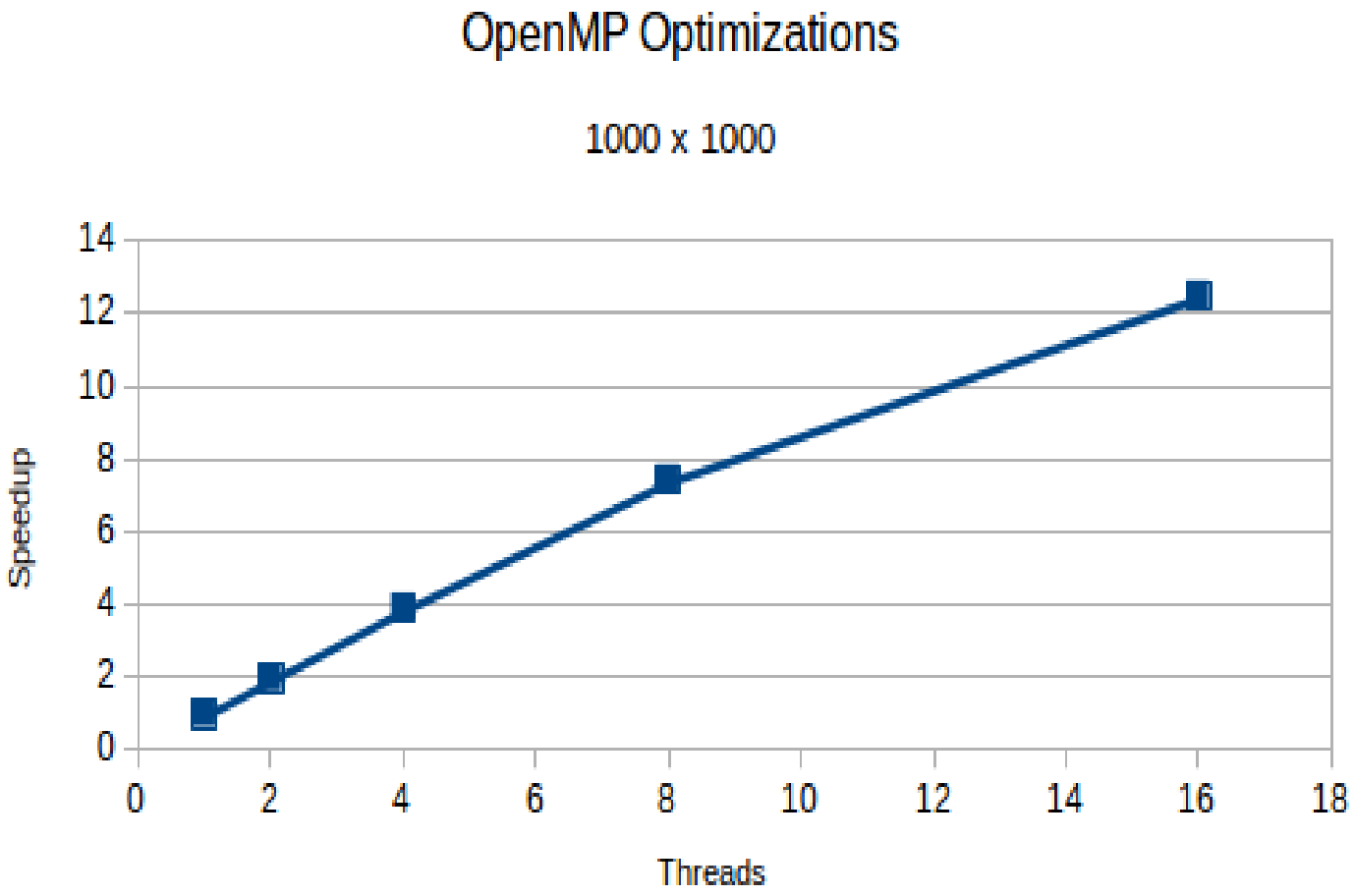
Vectorization

- Vectorization using Intel Cilk
- Idea: compute net updates for a number of cells at each iteration and use Intel Cilk in the computeNetUpdates function. Assume that compiler will efficiently vectorize
- After testing we determined an optimal vector length of 8

OpenMP

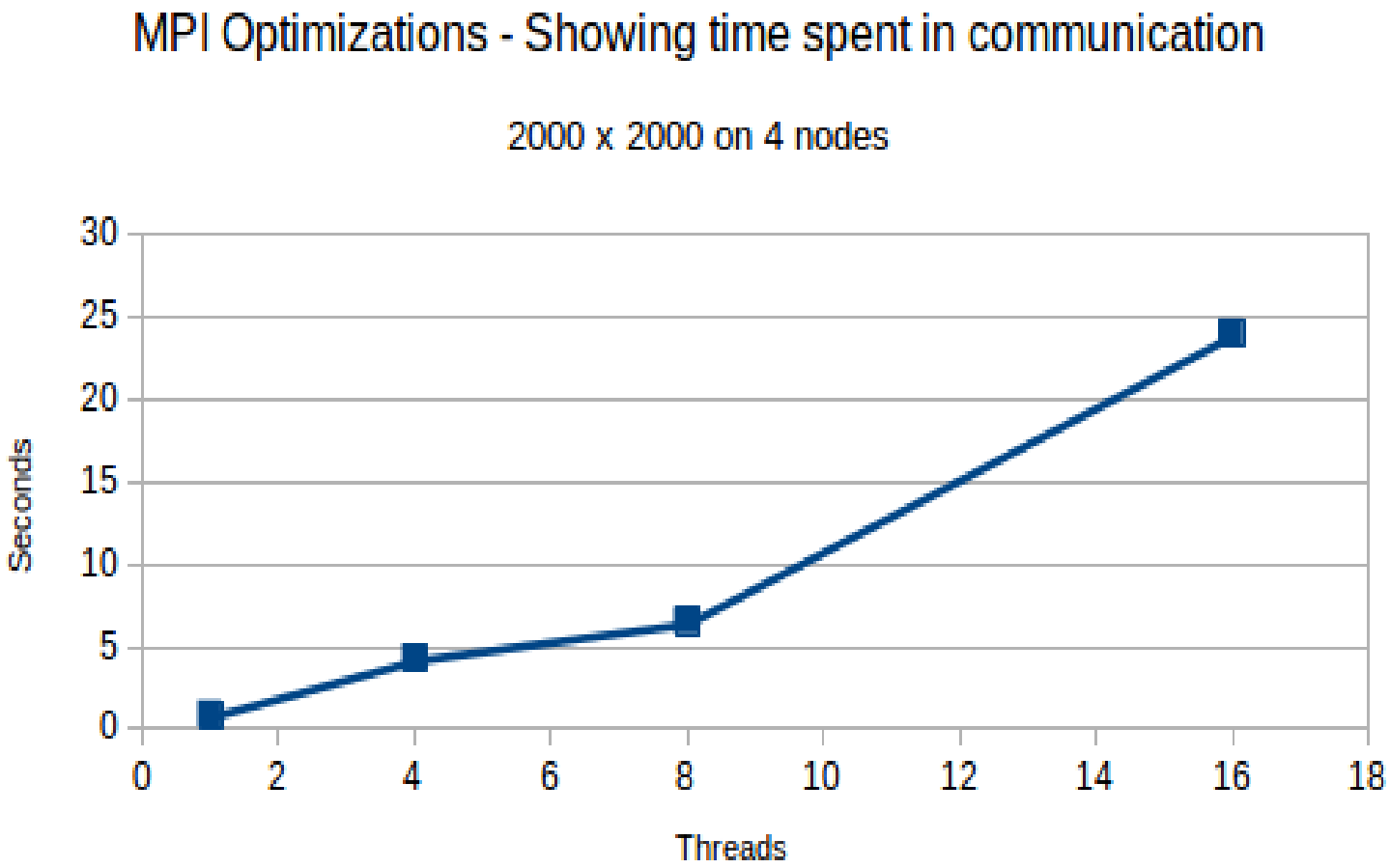
- Some OpenMP optimization was already present in the original code
- Fusion of loops performing the calculations for the vertical and horizontal edge updates
- We used OMP threads on the cores on one node to utilize shared L2 cache

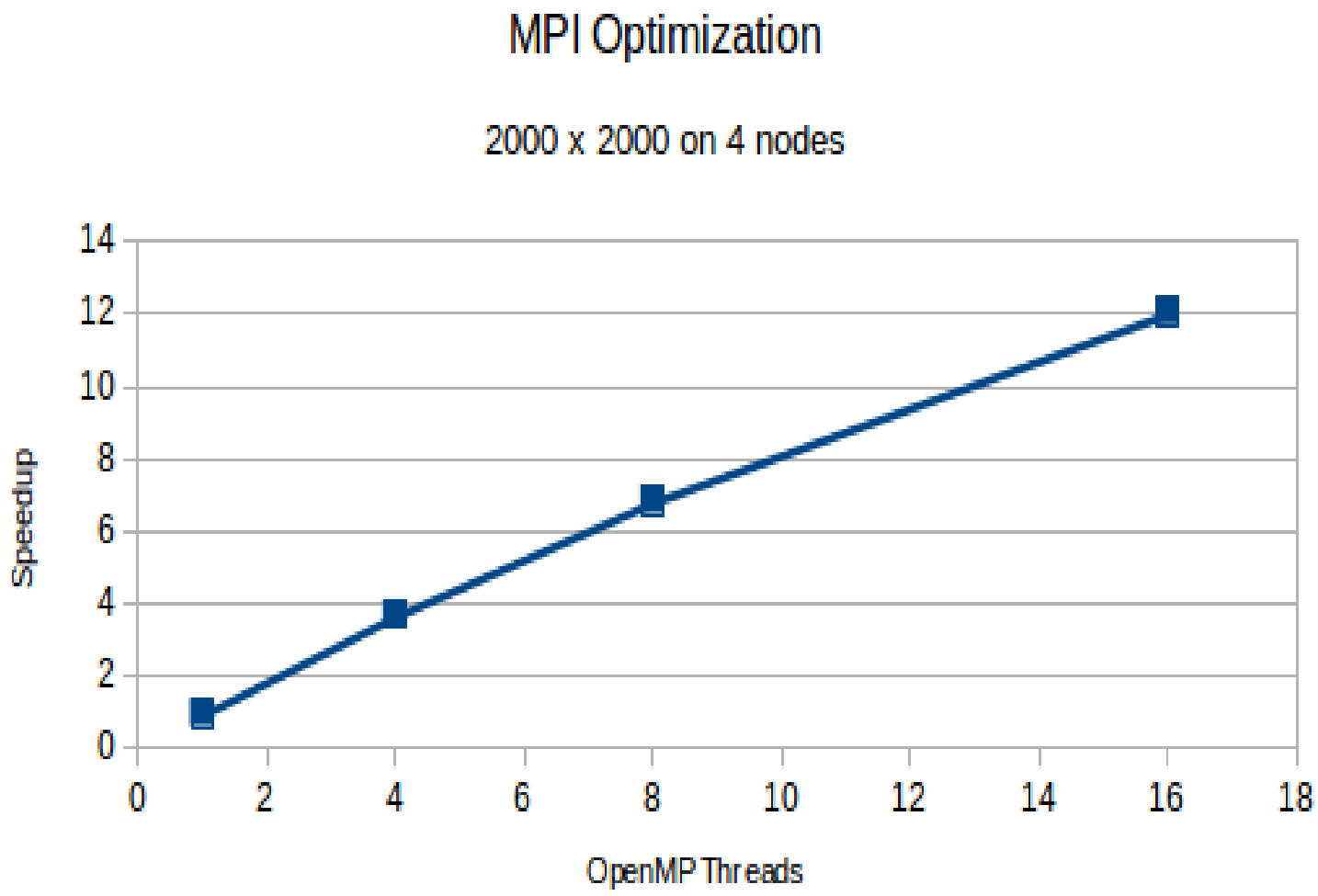

OpenMP Optimizations
1000 x 1000
Time
350
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
Threads


OpenMP Optimizations
1000 x 1000
Speedup
Threads
0
2
4
6
8
10
12
14
16
18

MPI

- Some MPI code already present in original code
- Goal: overlap communication with computations
- Slight improvement in performance
- Instrumentation showed that the time needed for communication is insignificant compared to the time for


MPI Optimizations - Showing time spent in communication
2000 x 2000 on 4 nodes
Seconds
30
25
20
15
10
5
0
0
2
4
6
8
10
12
14
16
18
Threads


MPI Optimization
2000 x 2000 on 4 nodes
Speedup
0
2
4
6
8
10
12
14
0
2
4
6
8
10
12
14
16
18
OpenMP Threads