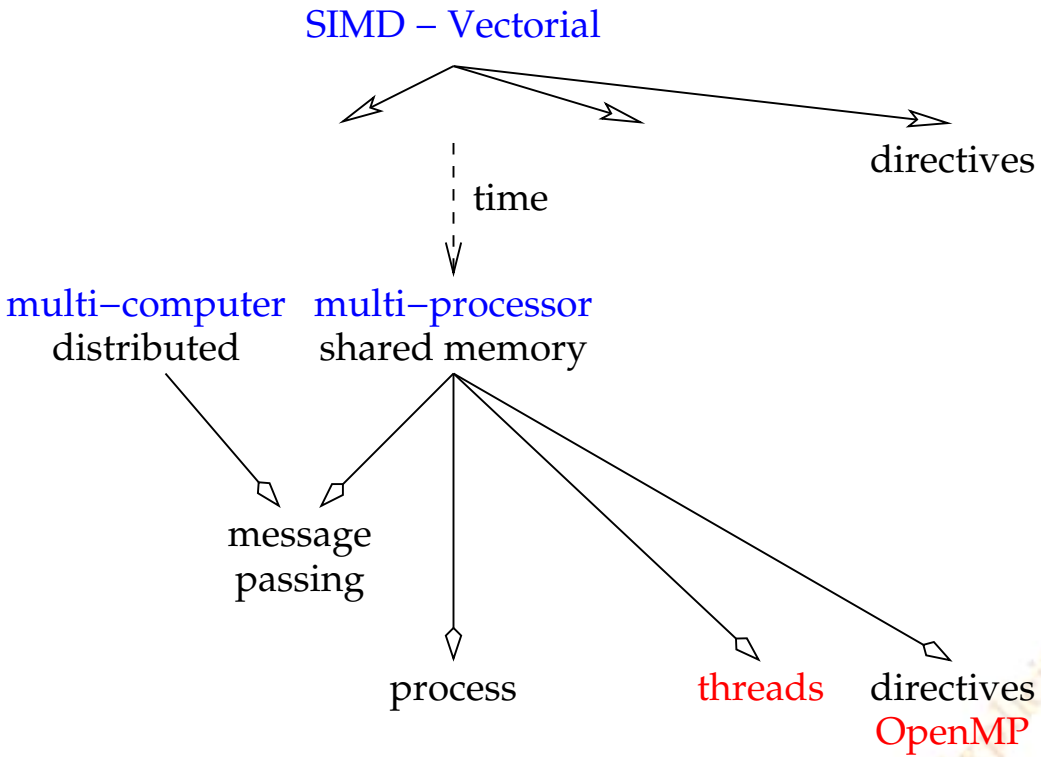
Comparison of OpenMP and Classical Multi-Threading Parallelization for Regular and Irregular Algorithms

Eugen Dedu, Stéphane Vialle, Claude Timsit

Supélec, France

May 18, 2000

Context


SIMD – Vectorial
directives
time
multi–computer
distributed
multi–processor
shared memory
message
passing
process
threads
directives
OpenMP

OpenMP code

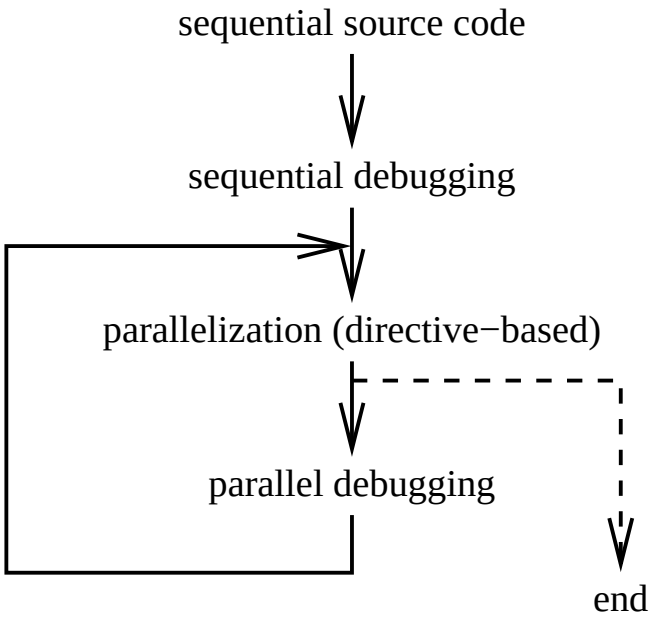
- works at implementation level, not algorithmic level!
- simplicity to learn and use: parallelization of a loop

```
1 // Classic threads version
2 // int nb_threads, my_tid, size, first, last;
3 size = last_index / nb_threads; // number of indexes assigned to every thread
4 first = my_tid * size;
5 last = (my_tid + 1 == nb_threads) ? first + size : last_index;
6 for (i=first ; i<last ; i++)
7   array[i ] = ...;
```

```
1 // OpenMP version
2 // automatic loop decomposition
3 #pragma omp parallel for
4 for (i =0; i<last_index ; i++)
5   array[i ] = ...;
```

Writing parallel code in OpenMP

☞incremental parallelization



Main characteristics of OpenMP

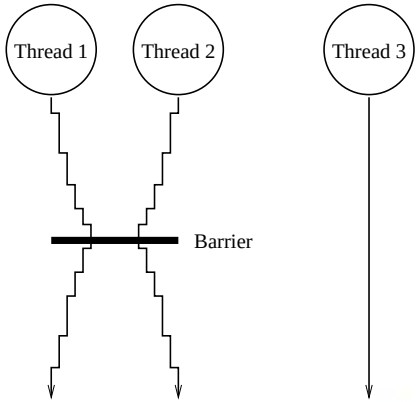
- easy to learn and to use (higher level than threads)
- incremental parallelization
- automatical computing of the number of threads
- identical sequential and parallel code sources
- portable
- efficient

- private variables
- reduction (tree-based)
- critical regions

Features of threads

☞lower level than OpenMP

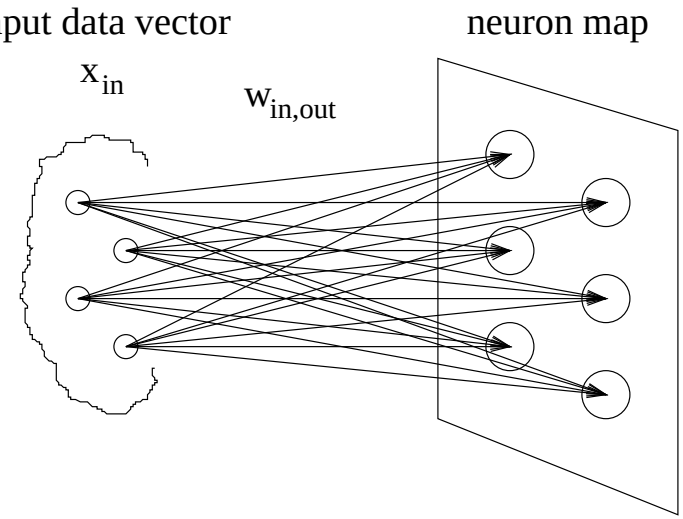
Pthreads	OpenMP
3 types of mutexes	1 type
semaphores	–
expressiveness: high	medium
✗work with groups of threads : easy	possible sometimes



✗barriers, mutual exclusions bound to groups of threads

✓OpenMP more appropriate for data parallelism than code parallelism

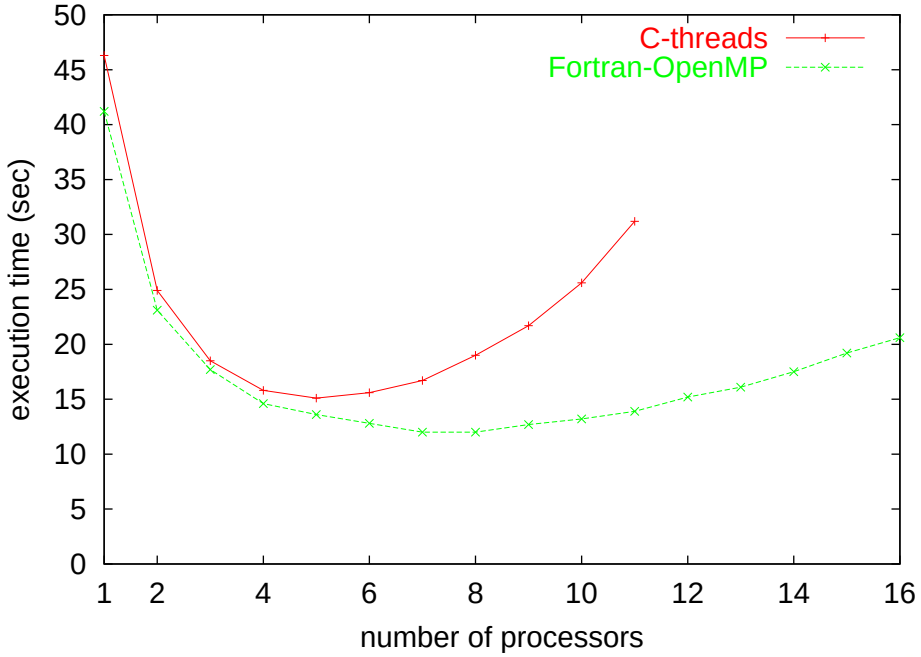
Kohonen map



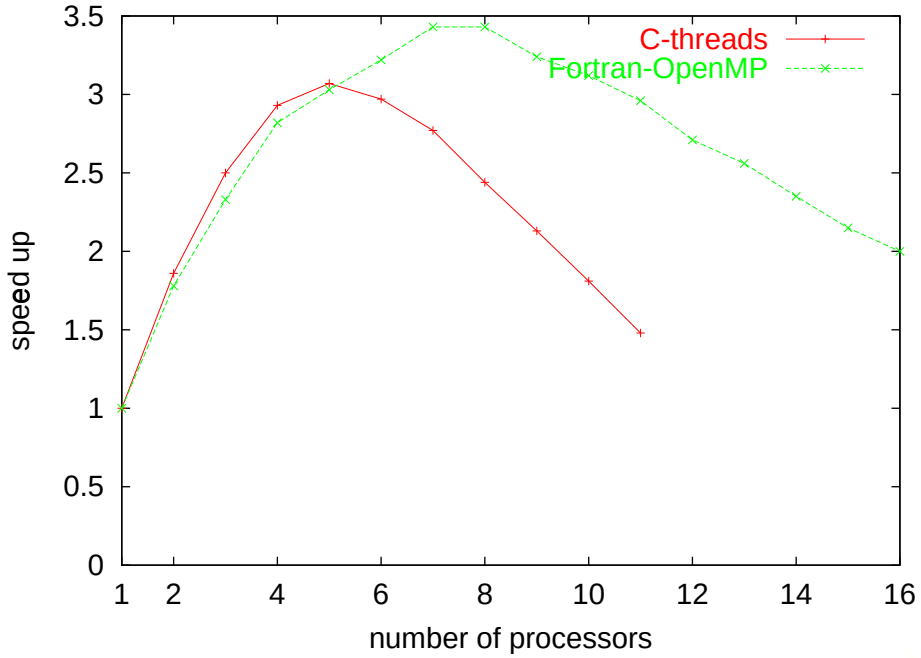
$$d_{out} = \sum_{in} (w_{in,out} - x_{in})^2$$

$$w_{in,out} = w_{in,out} + \eta(x_{in} - w_{in,out})$$

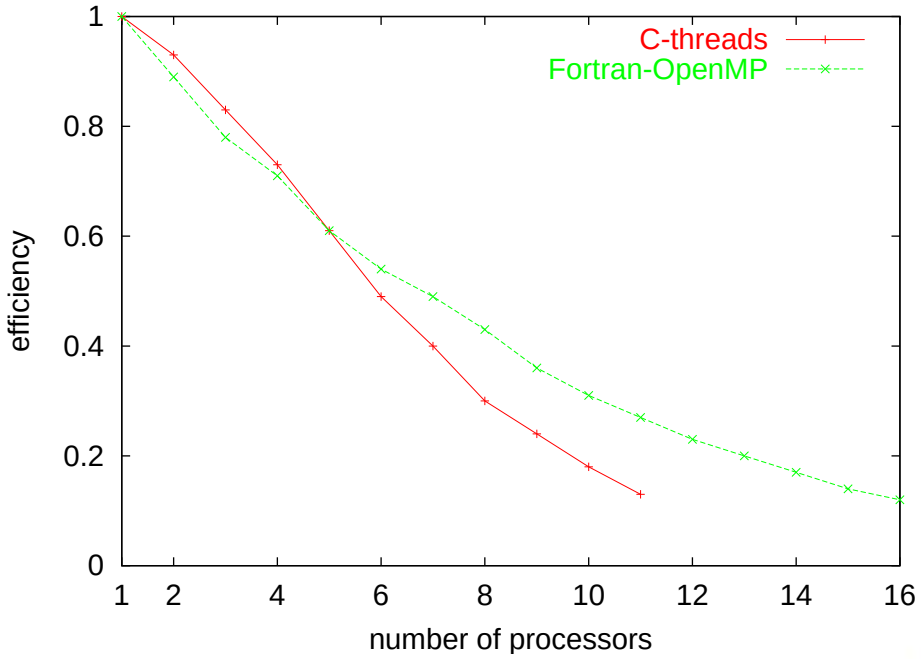
Kohonen implementation performance (texec)



Kohonen implementation performance (speed up)

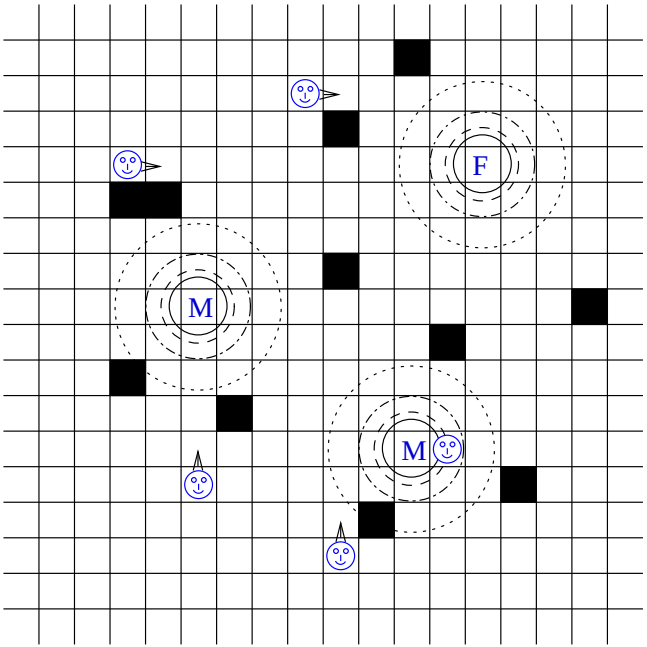


Kohonen implementation performance (efficiency)



Situated multi-agent system

Agent : environment, perceptions, actions, goal.



Agent

M Mine

F Factory

Obstacle

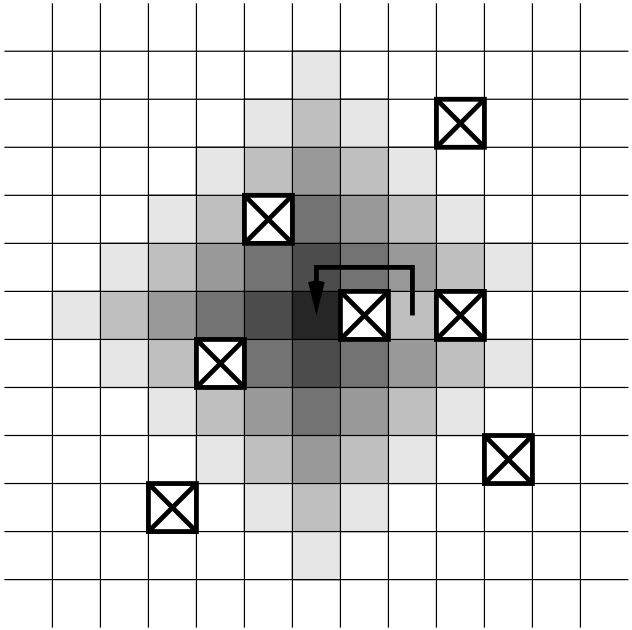
Situated multi-agent system

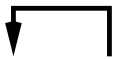
Difficult to parallelize, because of:

- migration of agents ⧫ load balancing, data localization
- dynamic environment
  - propagation of fields ⧫ synchronization, load balancing
- different behaviours of the agents ⧫ load balancing

⧫ influences on load balancing, cache performance …

Wave-propagation algorithm





Example of increasing potential way, avoiding obstacles

Source

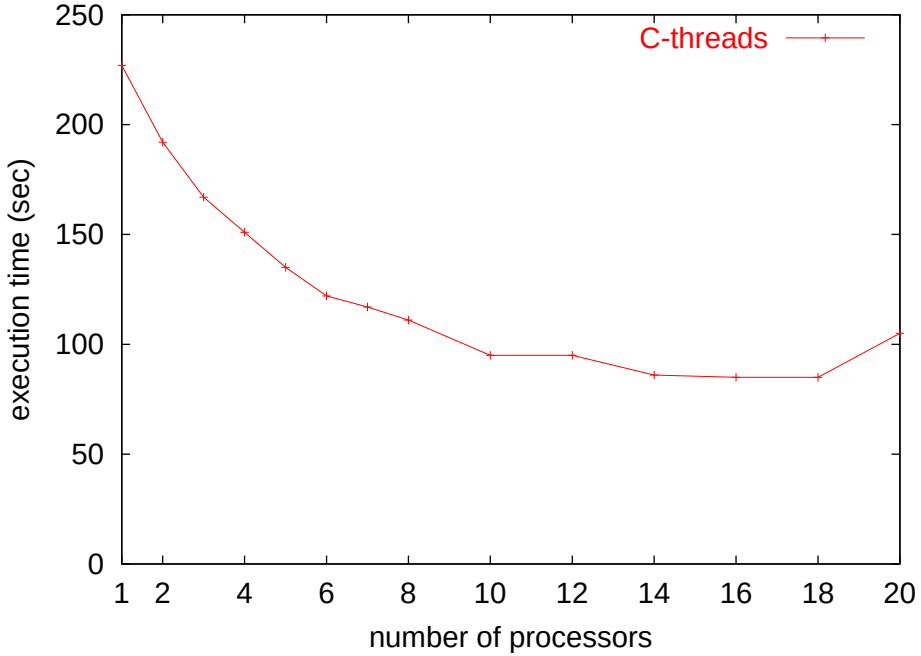
Obstacle

OpenMP problems

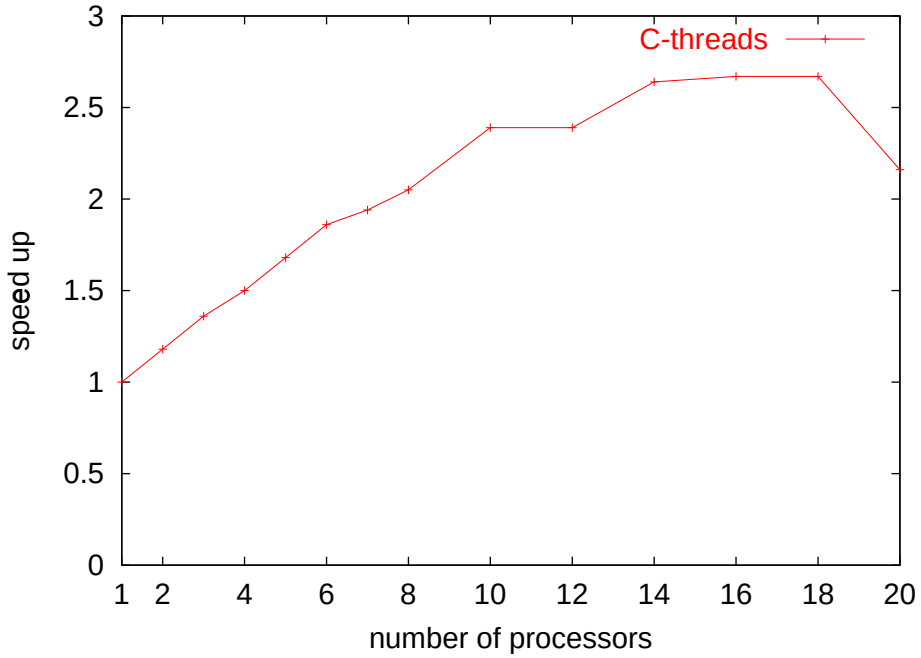
- find manually the index in REDUCTION clause

- static domain partitioning
- equal number of threads
- research of the optimal number of threads

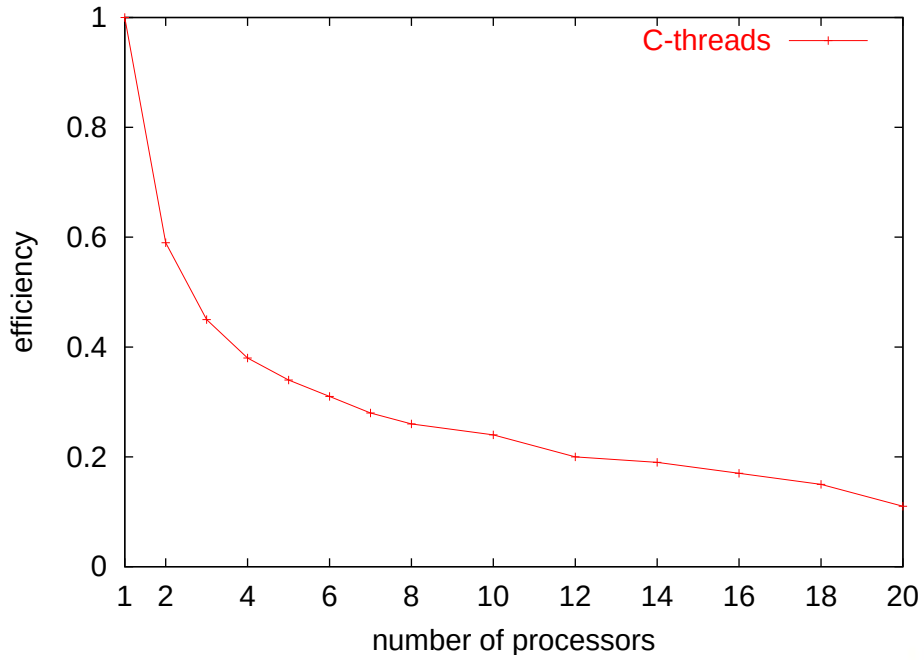
SMAS implementation performance (texec)



SMAS implementation performance (speed up)



SMAS implementation performance (efficiency)



Comparison OpenMP/threads, regular/irregular application

Implementation	Regular algorithm: Kohonen map		Irregular algorithm: SMAS	
	C-threads	Fortran-OpenMP	C-threads	C-OpenMP
Maximum speed-up	3	3.5	2.7	–
Optimal number of threads	5	7–8	16–18	–
Parallelization complexity	medium	easy	high	–
Development time	2 weeks	1 week	5 weeks	> 5 weeks
Number of code source lines	450	400	950	850

Conclusions

- a regular and an irregular application
- implemented in OpenMP and threads

- *execution* times comparable for regular applications
- *development* times better in OpenMP for regular applications
- irregular application and higher level of OpenMP:
  - difficulty in programming
  - even utilization of non-OpenMP fonctions

Our advise

Type of application	*Appropriate method*	*Reason*
regular	OpenMP	rapid development and execution
irregular	threads	better control (expressiveness)

Bibliography

- Leonardo Dagum and Ramesh Menon. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Computational Science & Engineering*, January–March, 1998.
- Jeff Fier (documentation SGI). Performance Tuning Optimization for Origin2000 and Onyx. `http://techpubs.sgi.com/library/manuals/3000/007-3511-001/html/O2000Tuning.0.html`
- Specifications, information...: `http://www.openmp.org`