Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC 2013) - La Cité des Sciences et de l'Industrie, Paris, France, October 27-31, 2013

Parallelization of TWOPORFLOW, a Cartesian Grid based Two-Phase Porous Media Code for Transient Thermo-hydraulic Simulations

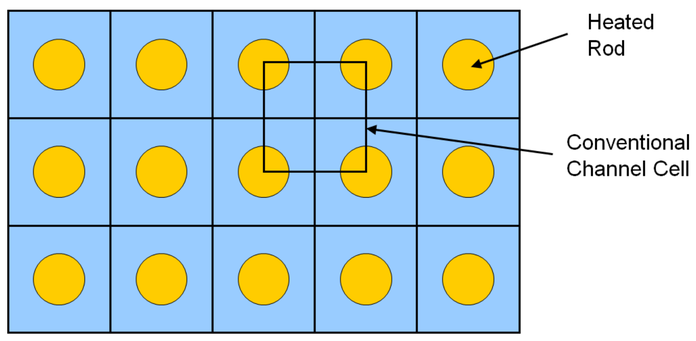
Reactor Physics and Dynamics Group (RPD)

Javier Jiménez *, Nico Trost, Uwe Imke, Victor Sanchez

Karlsruhe Institute of Technology (KIT), Institute for Neutron Physics and Reactor Technology (INR), Germany
Hermann-von-Helmholtz-Platz 1, Geb. 521, 76344 Eggenstein-Leopoldshafen
*Corresponding Author, Email: Javier Jimenez, javier.jimenez@kit.edu

TWOPORFLOW CODE

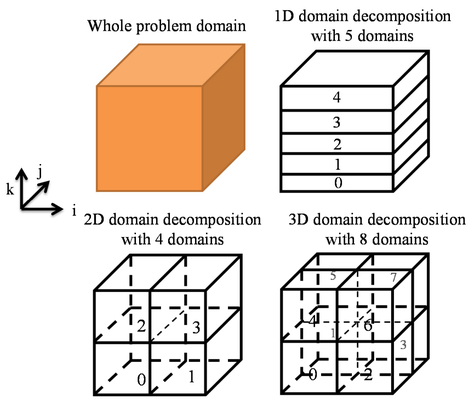
- TWOPORFLOW (TPF) is a thermo-hydraulic code based on a porous media approach to simulate single- and two-phase flow including boiling.
- Coarse Cartesian grids are used to obtain volume-averaged parameters. TPF features a 3D transient solution of the mass, momentum and energy conservation equations for two inter-penetrating fluids with a semi-implicit continuous Eulerian type solver.
- The application domain of TWOPORFLOW includes the flow in standard and structured porous media such as micro-channels, spent fuel pools or reactor cores of nuclear power plants.
- The IAPWS-97 formulation of the water and steam thermodynamic properties as well as new models for describing physical phenomena were recently implemented in TPF.
- TPF has been parallelized by using the OpenMP standard combined with a domain decomposition methodology.
- Both parallelization approaches can be enabled at the same time leading to a highly scalable and flexible execution of TWOPORFLOW code.



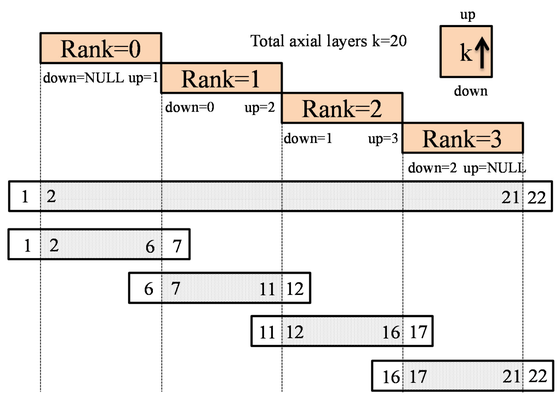
DOMAIN DECOMPOSITION IMPLEMENTATION

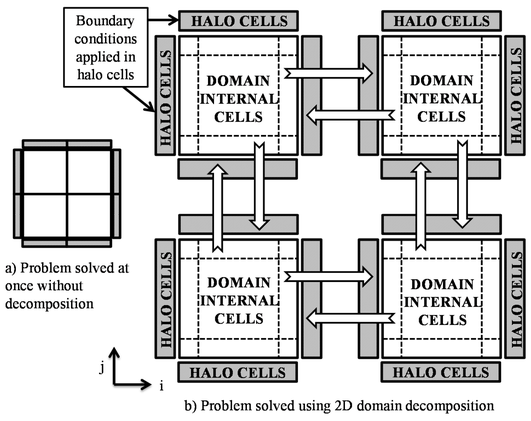
- A domain decomposition method was implemented in order to take benefit from distributed memory systems.
- It is based on splitting the initial Cartesian problem into smaller sub-problems which are distributed over different processors and solved simultaneously.
- The processors need to communicate between each other during the solution process as the information on the sub-problems interfaces have to be updated consequently. This communication is implemented using the MPI standard.

Examples of possible domain decompositions in several dimensions



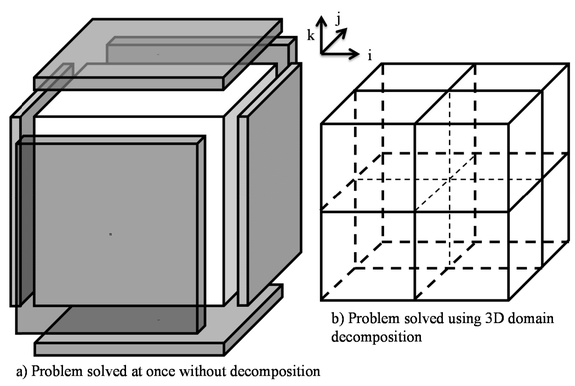
Example of a 1D domain decomposition using 4 processors (from rank 0 to 3)





a) Problem solved at once without decomposition

b) Problem solved using 2D domain decomposition

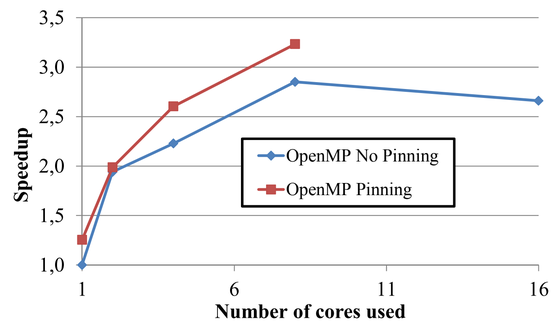


a) Problem solved at once without decomposition

b) Problem solved using 3D domain decomposition

OpenMP IMPLEMENTATION

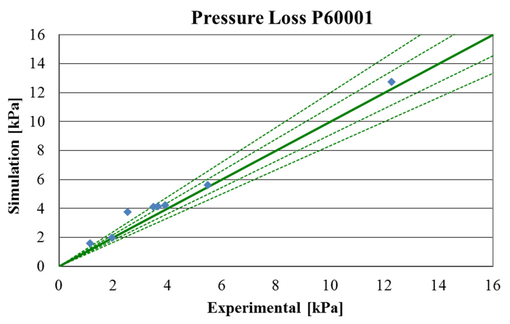
- OpenMP pragmas were implemented in the routines in charge of inter-phase momentum-, energy- and mass-coupling.
- The result of the profiling analysis was crucial in order to parallelize the loops which have a higher fraction of execution time while avoiding the performance penalization coming from the fork and join model as much as possible .
- After some investigations on the source code structure, the steam and water property related subroutine calls were identified to be completely independent of each other. This is because most property computations depend only on temperature and pressure and can therefore be computed in parallel.
- The performance of codes parallelized with the OpenMP API may be significantly improved by pinning the OpenMP threads to a specific CPU on dual socket machines.
- For the best performance, it is necessary to load data mainly from local memory, seen from the threads point of view. This implies the exclusive cache utilization by our application. In this work, Likwid-pin has been employed.
- The speedup measurements were performed using the case Nr. 1071-55 of the NUPEC BFBT benchmark.



# of cores	No Pinning		Pinning	
	Time [s]	Speedup	Time [s]	Speedup
1	856.88	1.000	683.23	1.254
2	441.51	1.941	431.15	1.987
4	384.61	2.228	329.23	2.603
8	300.57	2.851	265.04	3.233
16	322.13	2.660	-	-

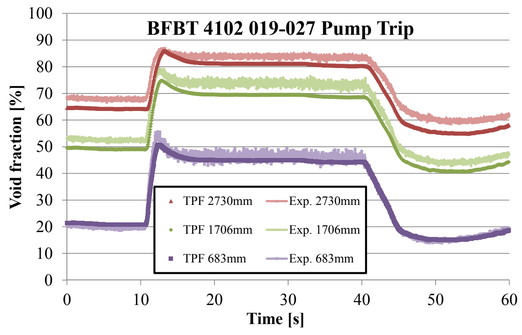
OVERALL PERFORMANCE OPTIMIZATION AND RESULTS

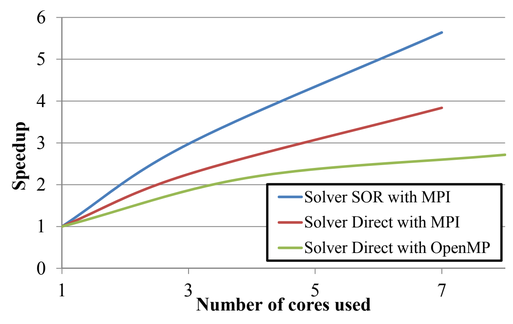
- A parallelized tool not always implies to obtain a net speedup above 1.0.
- Time consumption by communication overhead is a crucial aspect in parallel programming and should be kept always as minimal as possible.
- Communication overhead typically arises from communication transfer (barriers, synchronization points, data transfer, packing, unpacking) and memory management within the cores (thread fork and join).
- A saturation of the applications scalability will be always existent and can be achieved or not, depending on the settings for a specific execution.
- To illustrate the saturation, we took case P60001 from the BFBT Benchmark single-phase pressure drop measurement series. For the fuel assembly geometry, a very coarse mesh representation was selected.



# of cores	MPI		OMP No Pinning		OMP Pinning	
	Time [s]	S	Time [s]	S	Time [s]	S
1	23.6	1.064	25.0	1.002	24.6	1.019
2	17.9	1.398	18.6	1.346	17.8	1.409
4	14.7	1.702	15.4	1.627	14.9	1.682
8	16.4	1.532	18.3	1.255	13.9	1.805
16	-	-	20.0	1.254	20.0	1.251

- In the BFBT Phase I Exercise II Transient experiments, a pump trip is simulated for a whole BWR bundle. The average void fraction is measured at three different elevations (683, 1706 and 2730 mm from the bottom). The serial runtime is 4h 37min.





CONCLUSIONS

- The performance of TPF was investigated and optimized.
- The IAPWS-97 water and steam property equations have been decoupled and parallelized using OpenMP. The same happened with the fuel rod heating model.
- The pinning of each OpenMP thread to a specific core during execution time has been proven to be a key factor to improve the performance gain by using more efficiently the CPU cache.
- A domain decomposition method has been implemented using MPI.
- The performance of this method has been proven to be higher for computationally intensive cases.

ACKNOWLEDGMENTS

This work has been carried out at the Institute for Neutron Physics and Reactor Technology (INR) of the Karlsruhe Institute of Technology (KIT). The authors would like to thank the Program Nuclear Safety Research of KIT for the financial support of the research topic "multi-physics methods for LWR".

Abstract #A051