Karlsruher Institut für Technologie

Neutron Physics and Reactor Technology Institute

“Recent developments in TWOPORFLOW, a Two-phase Porous Media Code for Transient Thermo-hydraulic Simulations”

Javier Jimenez (javier.jimenez@kit.edu)
N. Trost, U. Imke, and V. Sánchez

Reactor Physics and Dynamics Group (RPD)

Introduction

- To ensure safety at reasonable cost by optimized designs, **high fidelity simulation tools** are being developed and continuously improved.

- Due to constantly increasing computational power, thermal-hydraulic subchannel codes coupled with time dependent neutron transport codes are the option of choice for accurate predictions of **local safety parameters**.

- Nevertheless, **high fidelity simulations** are not mature enough. Why?
  - The long computing times are still not acceptable for industrial applications.
  - They are under development and therefore not well validated.
  - The Industry is still using legacy codes for the safety assessment of reactor cores as they are well validated and fulfill their needs for licensing.

- **Fast running** codes with the **best physical models** are needed for high fidelity coupled thermal hydraulic / neutron kinetic solutions.

Introduction (2)

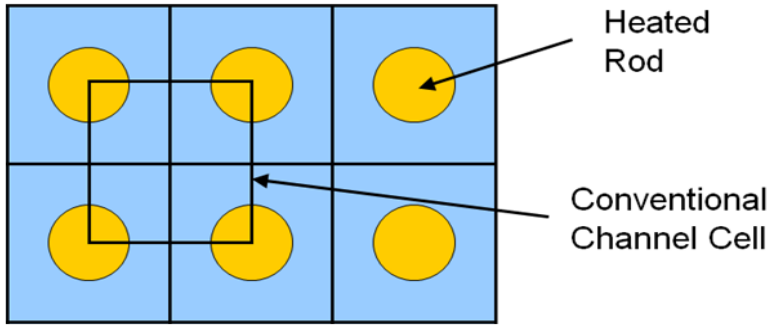
- At KIT, different subchannel codes such as **SUBCHANFLOW** and **TWOPORFLOW** are being improved, validated and coupled with different neutron kinetics solutions.

- A brief summary of the investigations devoted to the enhancement of the code numerics and informatics structure will be presented and discussed.

- By some examples, the gain on code speed-up will be demonstrated and finally an outlook of further activities concentrated on the code improvements will be given.

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.



TWOPORFLOW Code (2)

- Water and steam properties: IAPWS-97 formulation
- Flow regime dependent heat transfer models including boiling
- Several models for wall friction and vapor-liquid friction
- Iterative BiCGSTAB pressure solver for large problems
- Gap conductance model for fuel-cladding gap
- Implicit Continuous Eulerian method (ICE)

- TPF has been parallelized by using the OpenMP standard combined with a domain decomposition methodology based on OpenMPI message passing interface library.
- Both parallelization approaches can be enabled at the same time leading to a highly scalable and flexible execution of TWOPORFLOW code.

OpenMP Implementation in TPF

- 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.

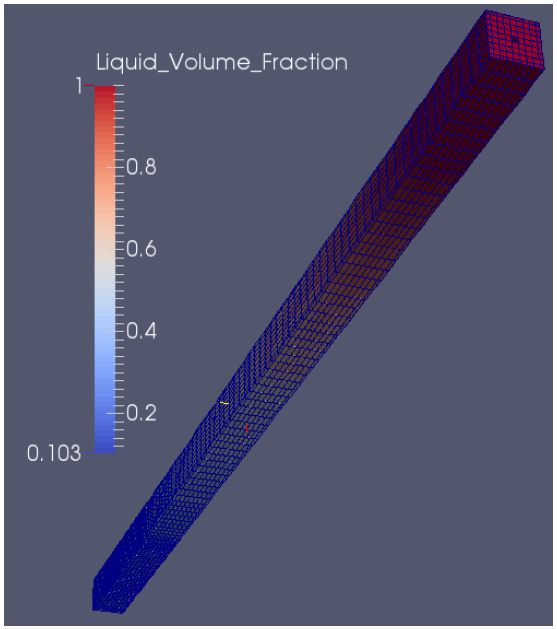
- The steam and water property related subroutine calls were identified to be completely independent of each other and can therefore be computed in parallel.

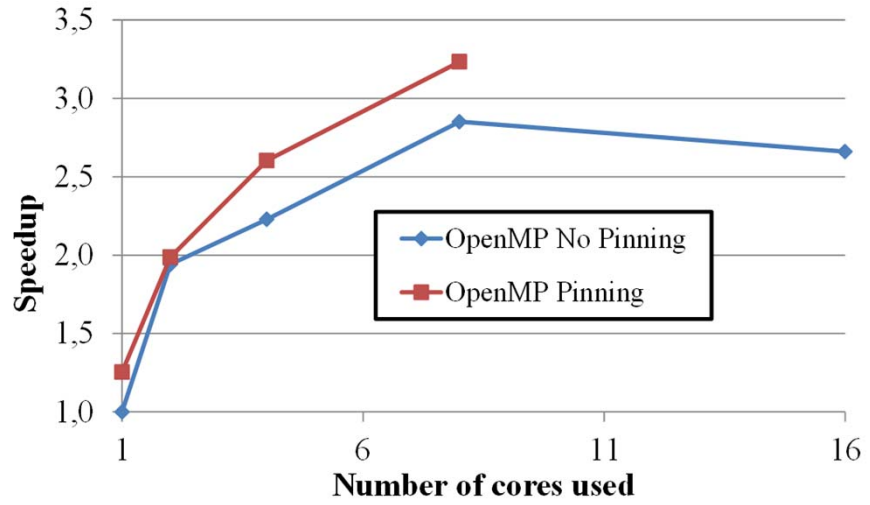
- The performance of codes parallelized with the OpenMP API may be significantly improved by pinning the OpenMP threads to a specific CPU.

- 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.

OpenMP Implementation in TPF

- The speedup measurements were performed using the case Nr. 1071-55 of the NUPEC BFBT benchmark.
- The runs were done in a dual socket Intel Xeon DP E5620 CPU featuring 8 physical cores (2 threads per core)
  - 8x8x1020 mesh points

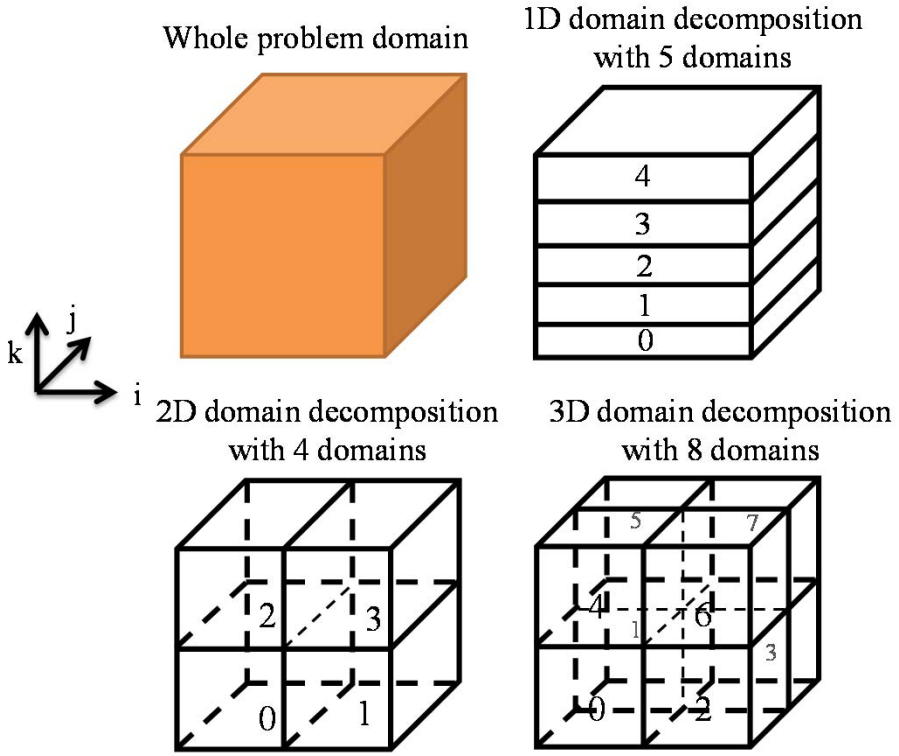


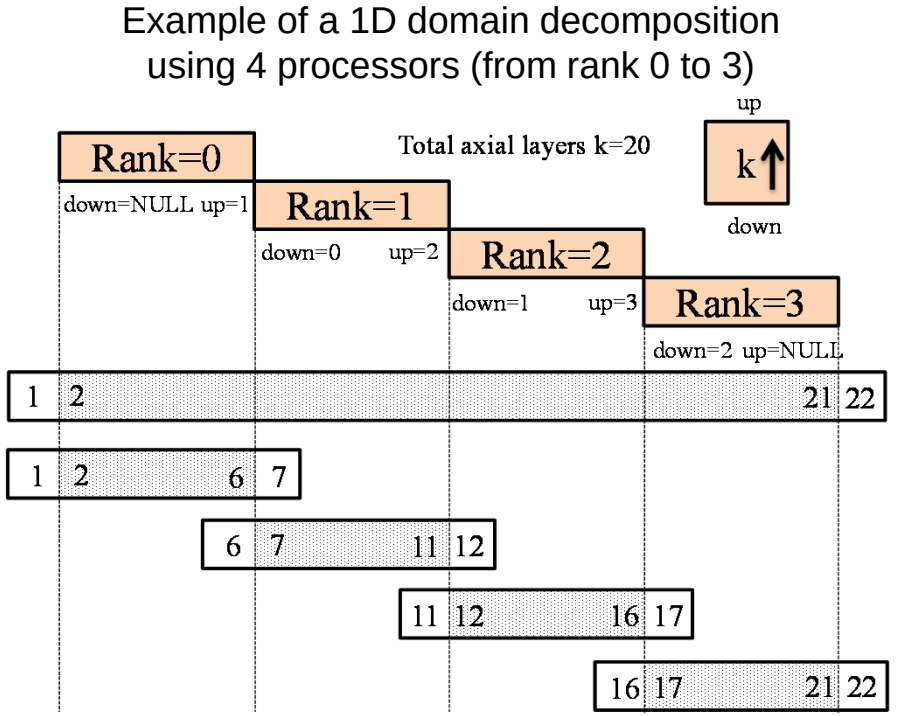


# of cores	No Pinning		Pinning	
	Time [s]	Speedup	Time [s]	Speedup
1	856.88	1.000	683.23	1.254
2	441.51	1.941	431.15	1.987
4	384.61	2.228	329.23	2.603
8	300.57	2.851	265.04	3.233
16	322.13	2.660	-	-

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.
- This communication is implemented using the MPI standard.

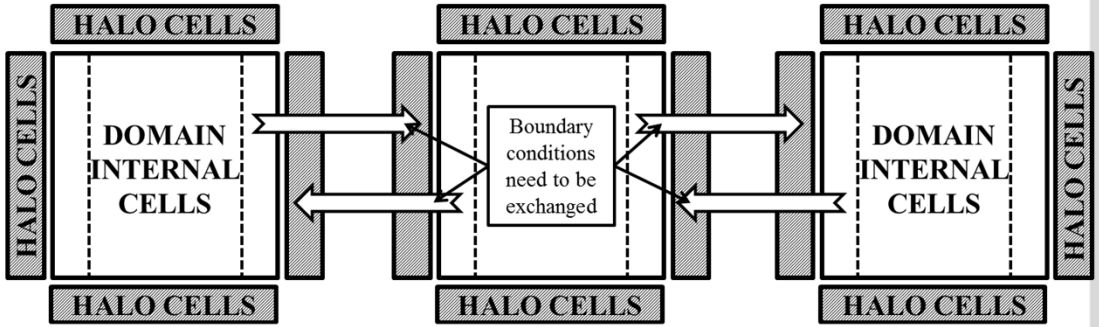




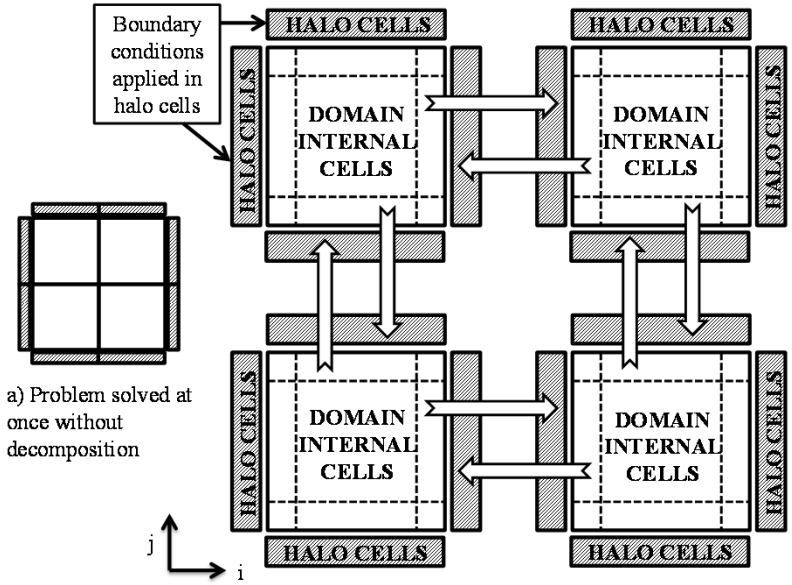
Domain Decomposition Implementation

- The processors need to communicate between each other during the solution process

1D Domain Decomposition



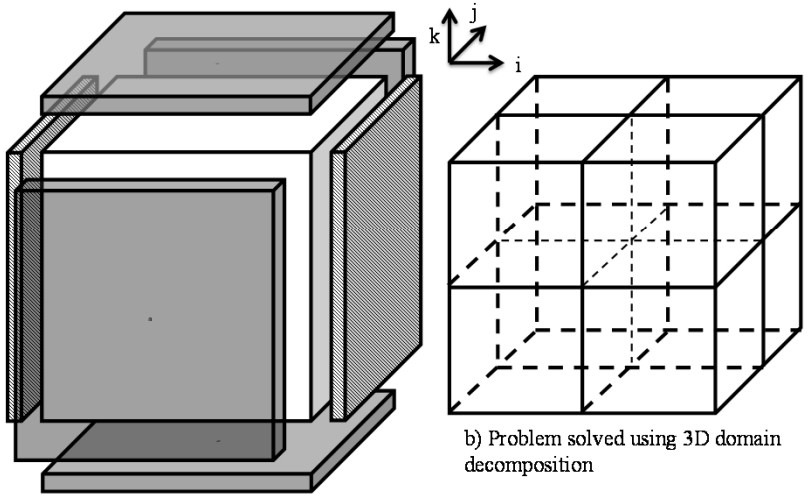
b) Problem solved using 1D domain decomposition



a) Problem solved at once without decomposition

b) Problem solved using 2D domain decomposition

2D Domain Decomposition



a) Problem solved at once without decomposition

b) Problem solved using 3D domain decomposition

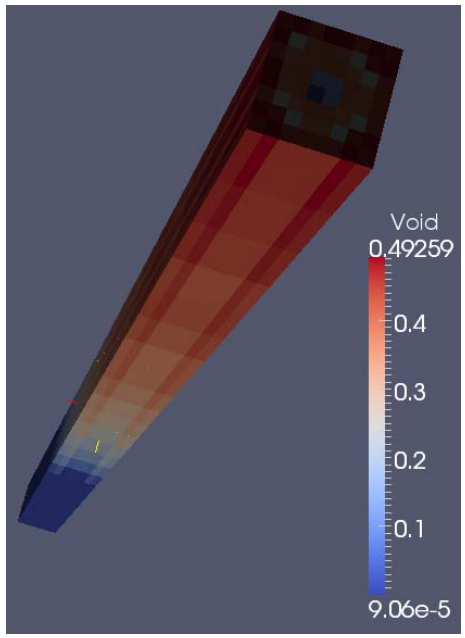
3D Domain Decomposition

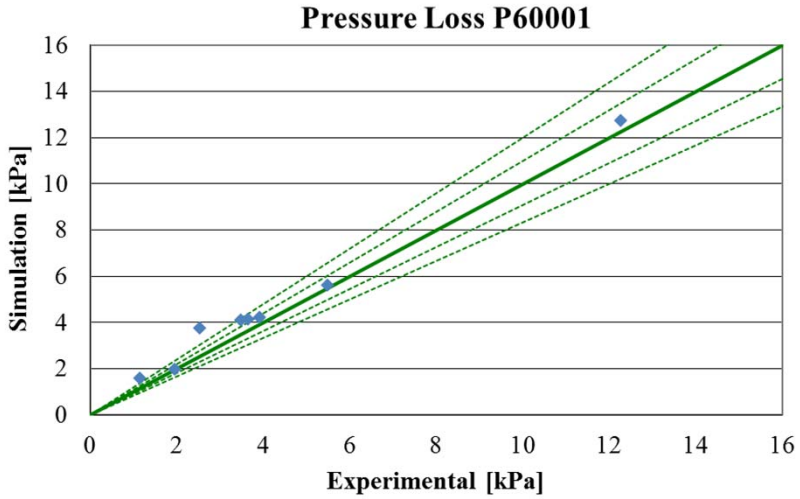
Overall performance optimization and results

- In a parallelized tool one cannot expect/observe to half runtime by doubling the computation cores
- Time consumption by communication overhead is a crucial aspect in parallel programming and should be kept always as minimal as possible.
- There is not only communication overhead, also operations coming from the standard (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.
  - Load balance problem
  - Enough data to be distributed and computed in parallel.

Overall performance optimization and results

- To illustrate the **saturation**, we took the case P60001 from the BFBT Benchmark single-phase pressure drop measurement series. For the fuel assembly geometry, a very coarse mesh representation was selected.
  - 8x8x50 mesh points, **very coarse** mesh.

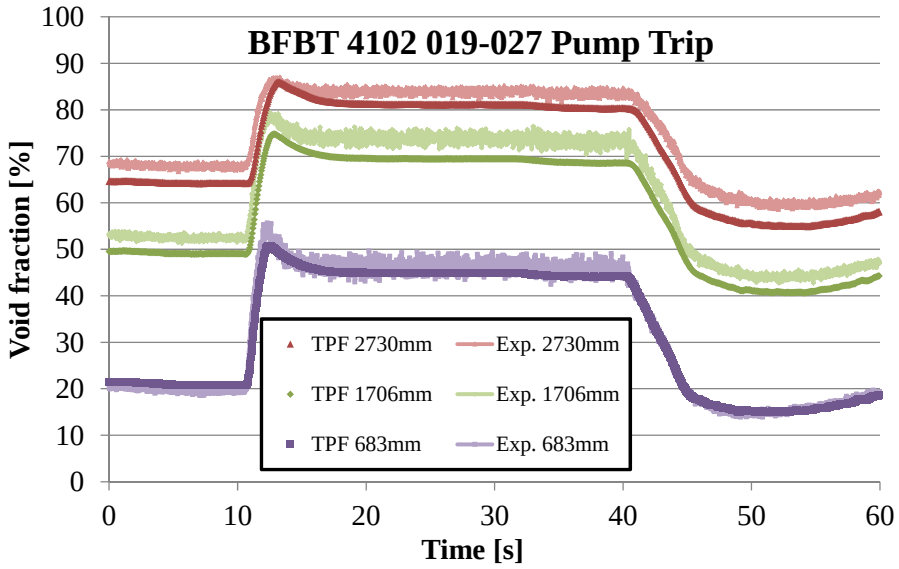


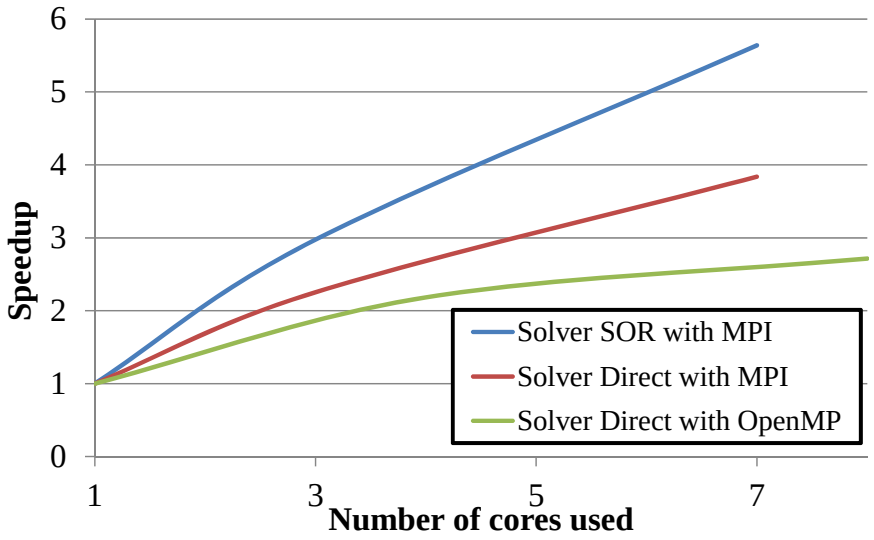


# of cores	MPI		OMP No Pinning		OMP Pinning	
	Time [s]	S	Time [s]	S	Time [s]	S
1	23.6	1.064	25.0	1.002	24.6	1.019
2	17.9	1.398	18.6	1.346	17.8	1.409
4	14.7	1.702	15.4	1.627	14.9	1.682
8	16.4	1.532	18.3	1.255	13.9	1.805
16	-	-	20.0	1.254	20.0	1.251

Overall performance optimization and results

- 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 and Outlook

- 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.

Future Plans

- Further improvements of TPF are foreseen on the following areas:
  - Improvement of user-friendlyness:
    - Development of pre- and post-processing tools for automation and visualization.
  - Further validation of the physical models.
  - Coupling with neutronic codes for multi-physics applications.

Acknowledgements

- 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*".

Karlsruher Institut für Technologie

Neutron Physics and Reactor Technology Institute

“Recent developments in TWOPORFLOW, a Two-phase Porous Media Code for Transient Thermo-hydraulic Simulations”

Javier Jimenez (javier.jimenez@kit.edu)
N. Trost, U. Imke, and V. Sánchez

Reactor Physics and Dynamics Group (RPD)