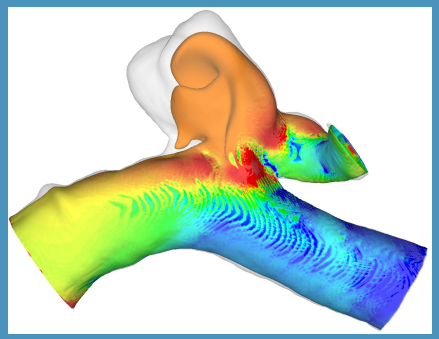
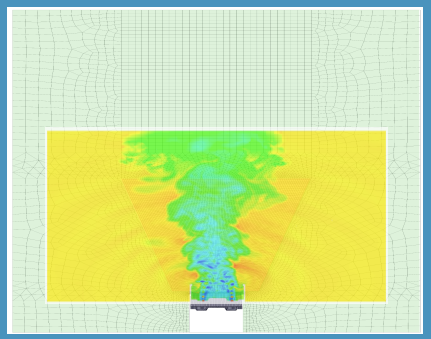
German Research School
for Simulation Sciences

A fully distributed CFD framework for massively parallel systems





J. Zudrop, H. Klimach, M. Hasert, K. Masilamani, S. Roller
j.zudrop@grs-sim.de

JÜLICH
FORSCHUNGSZENTRUM

RWTH AACHEN
UNIVERSITY

Motivation

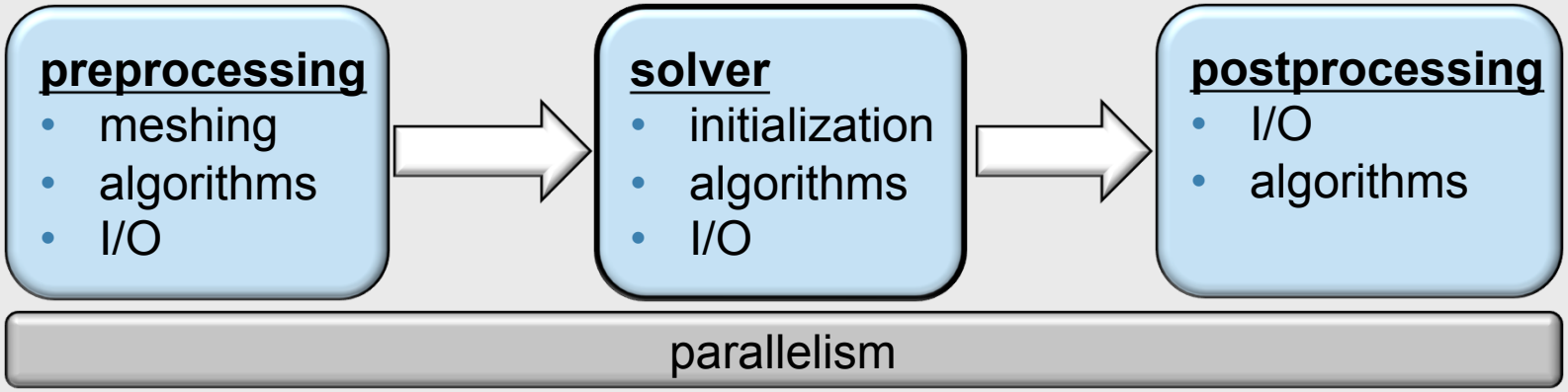
- CFD extremly important for industry and science
  - Greengineering
- Fluid dynamics involves enormous amount of scales
  - Large computational ressources required

Content

- Octree introduction
- Octrees and massive parallelism
- CFD on octrees
- Performance analysis
- Outlook and conclusion

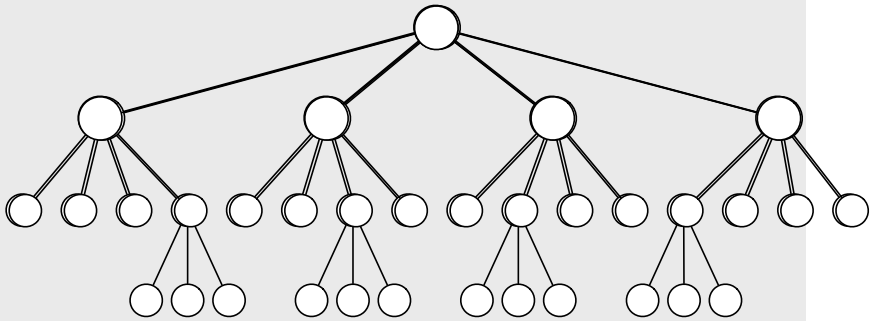
Introduction

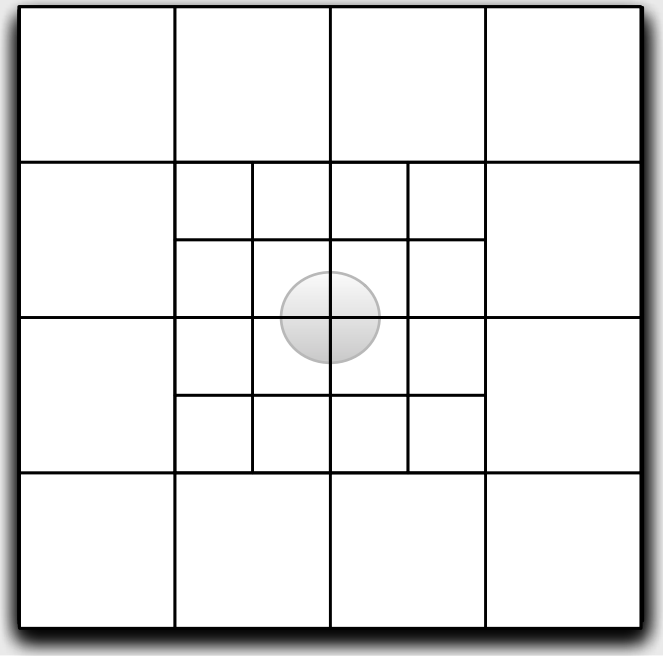
- Problems with CFD: there are many tools, but ...
  - They solve parts of the problem, but not the complete process ...
  - Bottlenecks avoid usability
- Solution: end-to-end parallelism
  - **The toolchain is important**
  - Our approach: octree based framework!
  - On top: mesh based CFD solver



Octree

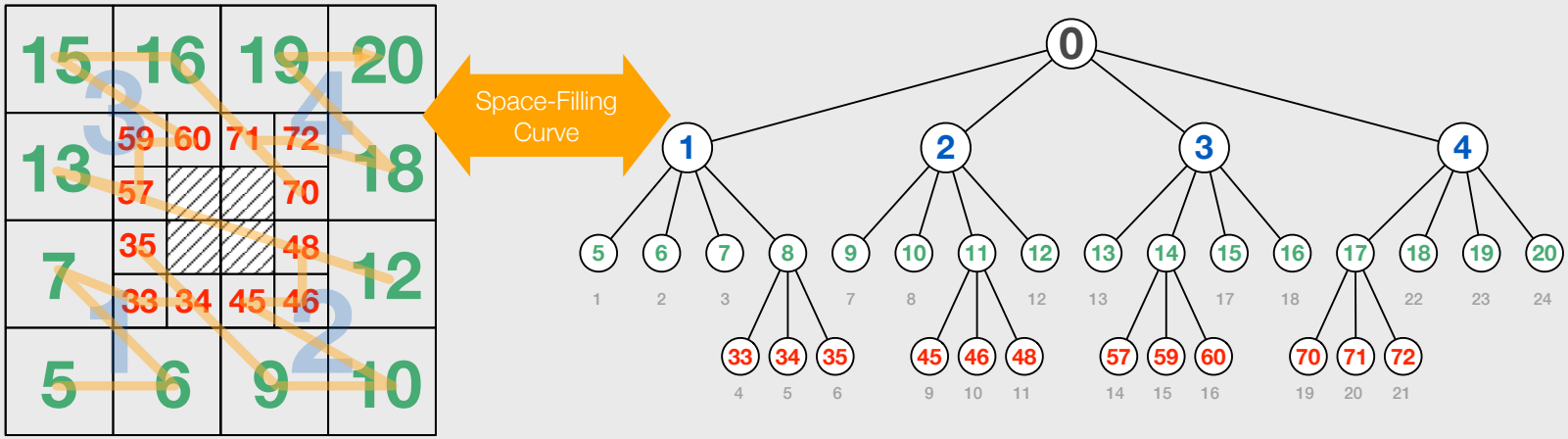
- Recursive definition
- Tree like topology
  - Root
  - Parent
  - Children
- Local refinement
  - Refine towards obstacles
  - Idea: store only fluids (sparse)





Octrees and SFC

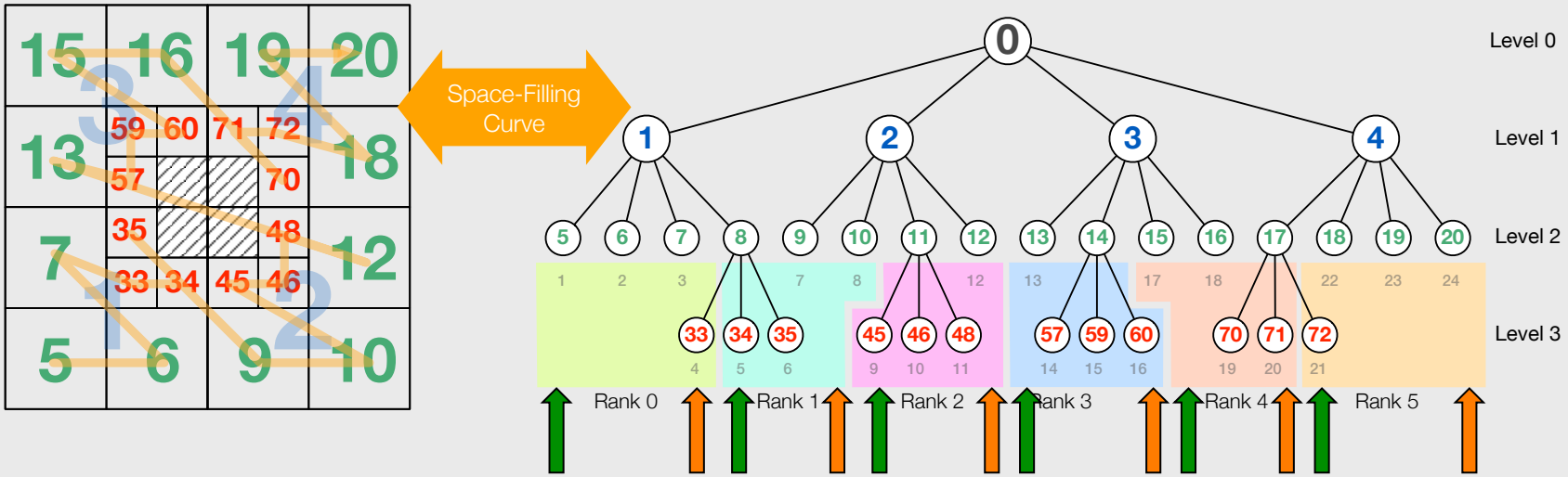
- Linearization of octree by SFC, important for ...
  - Domain decomposition
  - Parallel I/O
- Definition of unique identifier: TREE ID
- Neighbor identification (inherently existing)



Octrees and massive parallelism

Use linearization of tree

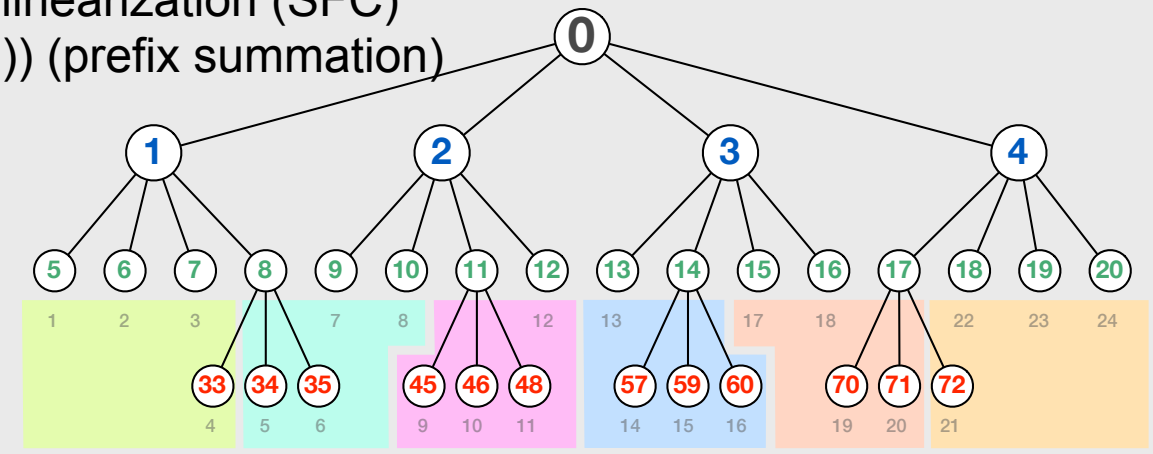
- Domain decomposition: store first and last TREE ID
  - Easy identification of neighbor location!
  - Local process!
- Memory complexity is O(p), but still cheap ...



German Research School
for Simulation Sciences

Parallel I/O in an octree framework

- Goal: end-to-end parallelism
- I/O as an example
  - Mesh data and mesh related data (e.g. restarting)
  - Parallel initialization
- Take advantage of linearization (SFC)
- Complexity O(log(p)) (prefix summation)



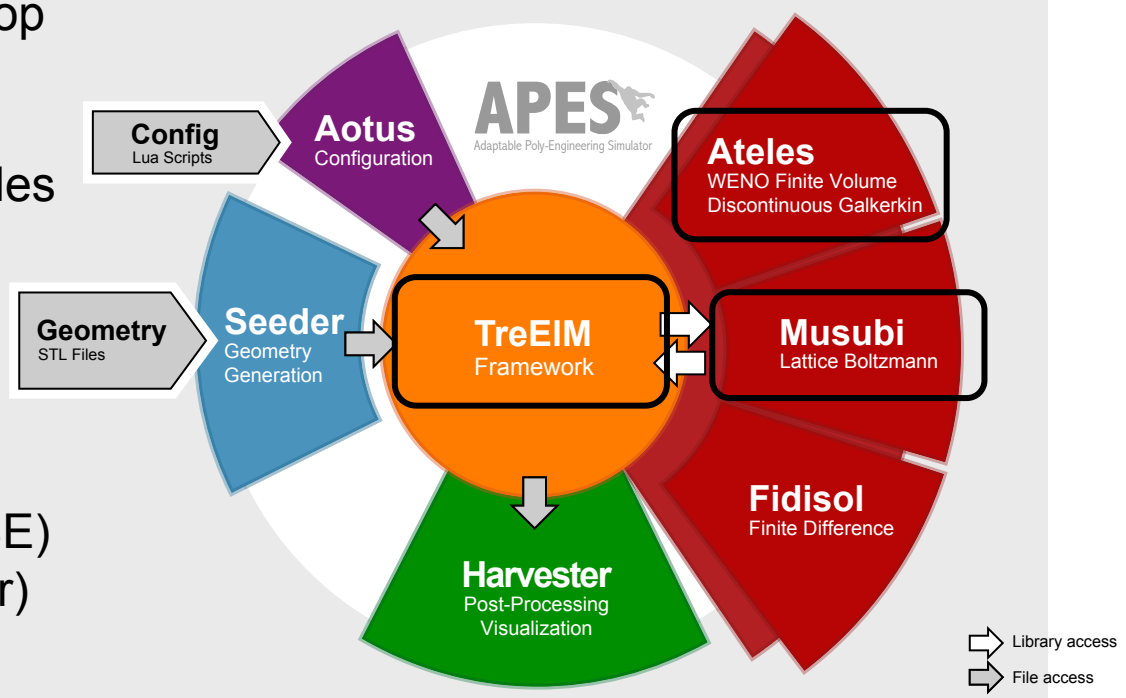
treelDlist: Representation on Disk

5	6	7	33	34	35	9	10	45	46	48	12	13	57	59	60	15	16	70	71	72	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

JÜLICH FORSCHUNGSZENTRUM
RWTH AACHEN UNIVERSITY

CFD in an octree

- Our approach: CFD on top of an octree
- End-to-end framework
- Octree framework provides
  - Communication
  - Datastructures
- Algorithmic approach
  - Act levelwise
- Two CFD solvers
  - LBM (incompr. NSE)
  - FVM (compr. Euler)
  - Other solvers are possible



Performance analysis

- Investigate performance for LBM and FVM (weak + strong scaling)
- Testcase: periodic cube
  - Do not consider local refinement
  - Domain size: c=8^k
  - Processes: p=8^l
  - Can be physically meaningful, too

→ Performance map (interpretation)

- Performance measures

$$MCUPS = \frac{cell\,updates}{(time)\cdot(\#\,cells)}$$

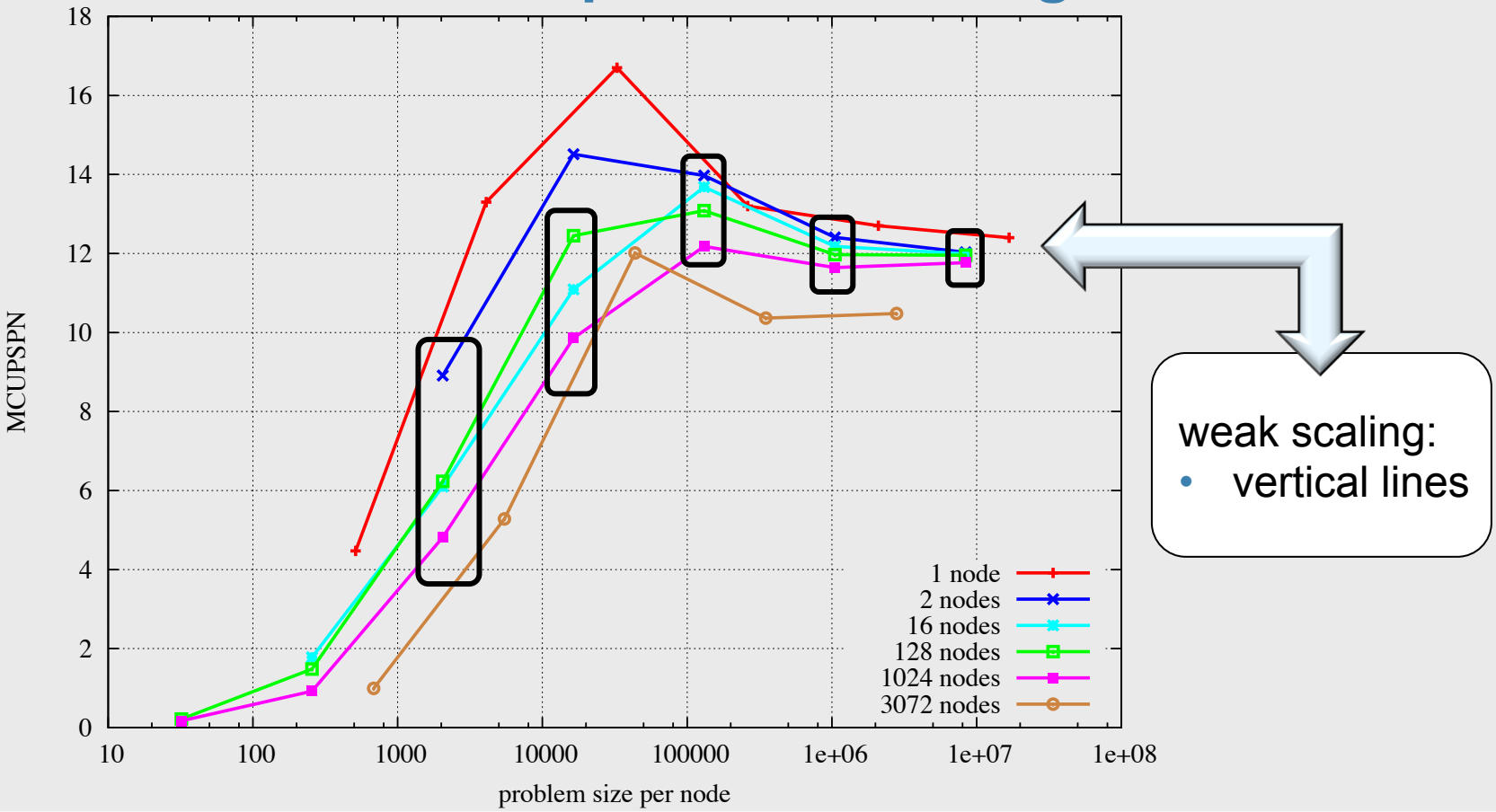
$$MCUPSPN = \frac{MCUPS}{\#\,nodes}$$

→ ideal weak scaling

→ ideal strong scaling

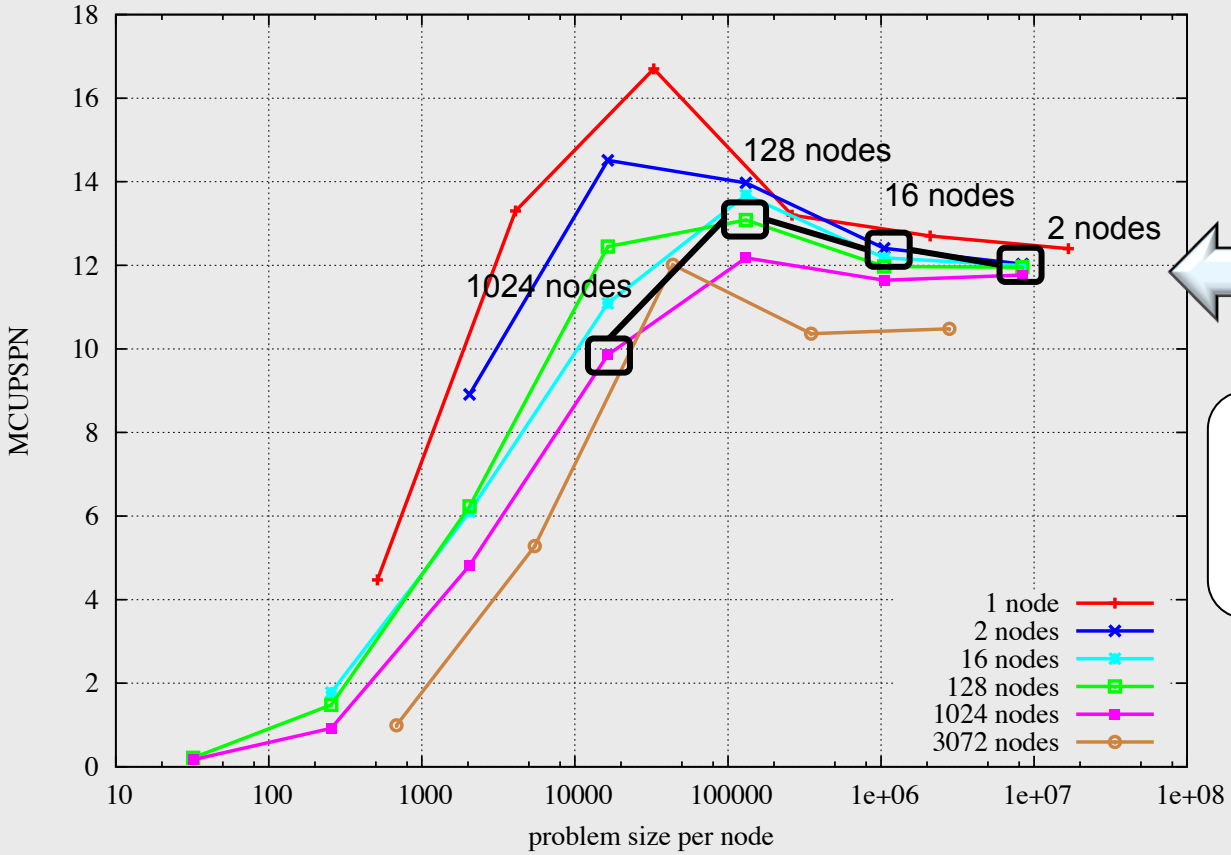
$$MCUPSPN = const$$

Performance map – weak scaling



weak scaling:
- vertical lines

Performance map – strong scaling



Strong scaling:

- Cores: multiply by 8
- Cells/node: divide by 8

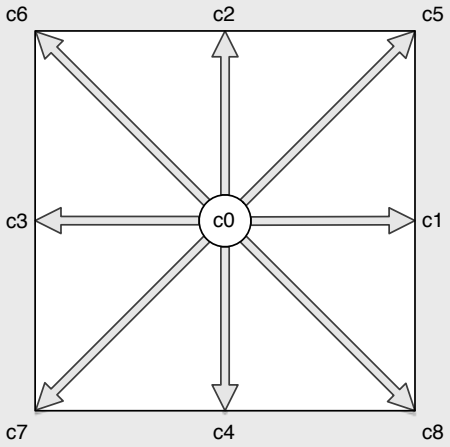
German Research School for Simulation Sciences

Finite volume method (FVM)

- Compressible Euler equation
- Approximation by polynomials
- Discontinuous at interface
  - Numerical flux function
  - Here: MUSCL + HLLE

Lattice Boltzmann method (LBM)

- Incompressible Navier-Stokes equation
- Statistical approximation (Boltzmann)
- Streaming
- Collision



$$\partial_t u + \nabla \cdot F(u) = 0$$

discr.

face / cell based

$$\partial_t \int u\psi \, dV = \int F(u) \cdot \nabla\psi \, dV - \int F^*(u)\psi d\vec{S}$$

$$f_i(x+1,t+1) = f_i(x,t) + \frac{1}{\tau}\left(f_i^{eq}(x,t) - f_i(x,t)\right)$$

JÜLICH FORSCHUNGSZENTRUM

RWTH AACHEN UNIVERSITY

German Research School
for Simulation Sciences

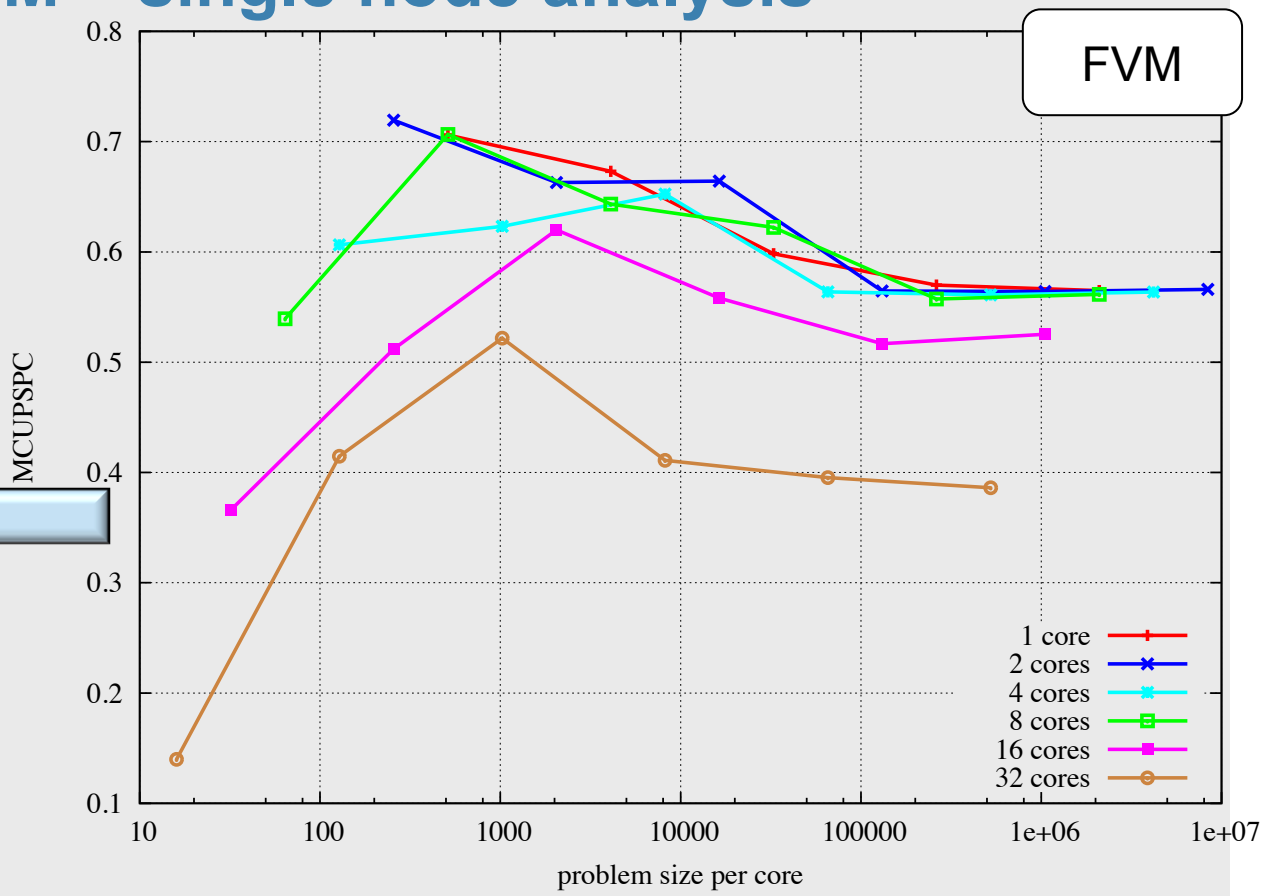
FVM and LBM – single node analysis

Shared hardware features (Interlagos)

- FPU
- Cache
- Mem. controler

More cores, better performance

Intranode analysis with full nodes



JÜLICH FORSCHUNGSZENTRUM

RWTH AACHEN UNIVERSITY

German Research School
for Simulation Sciences

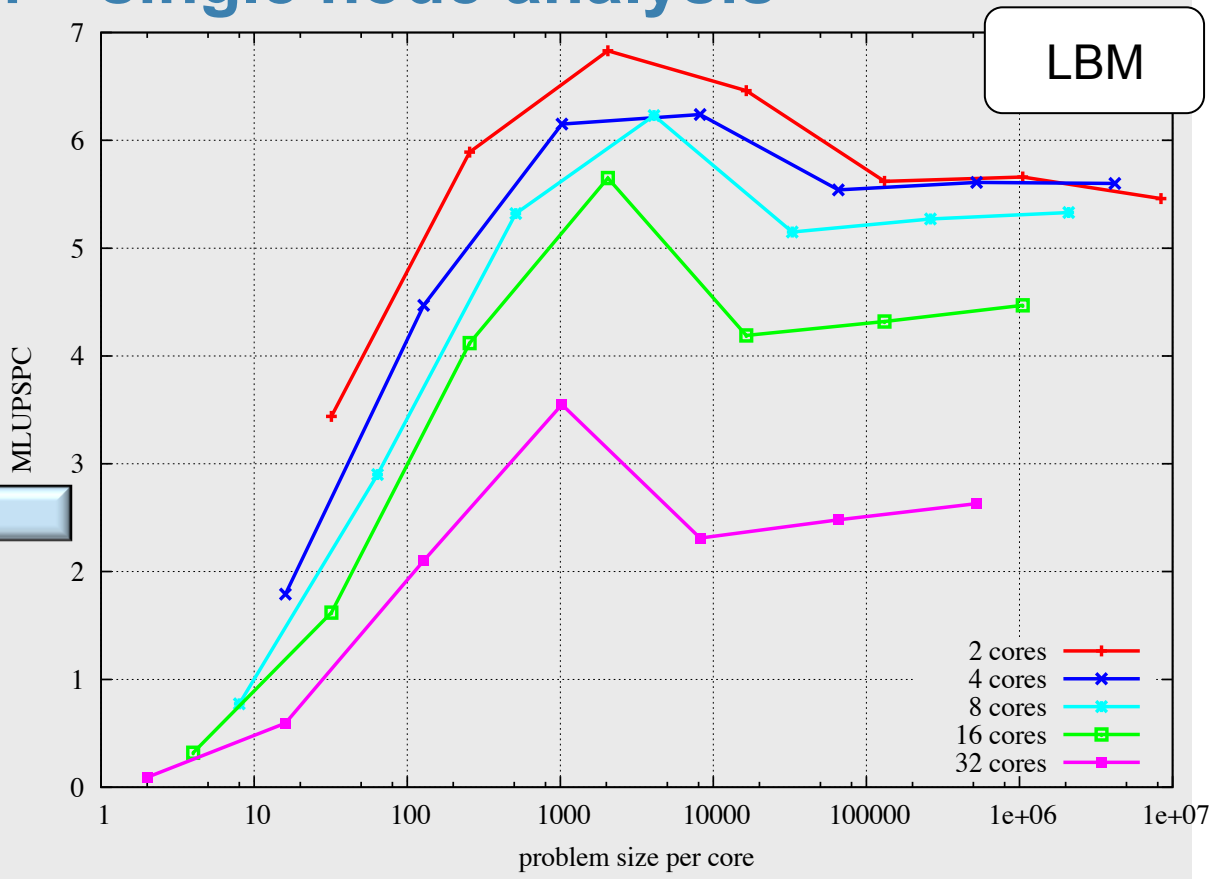
FVM and LBM – single node analysis

Shared hardware features (Interlagos)

- FPU
- Cache
- Mem. controler

More cores, better performance

Intranode analysis with full nodes



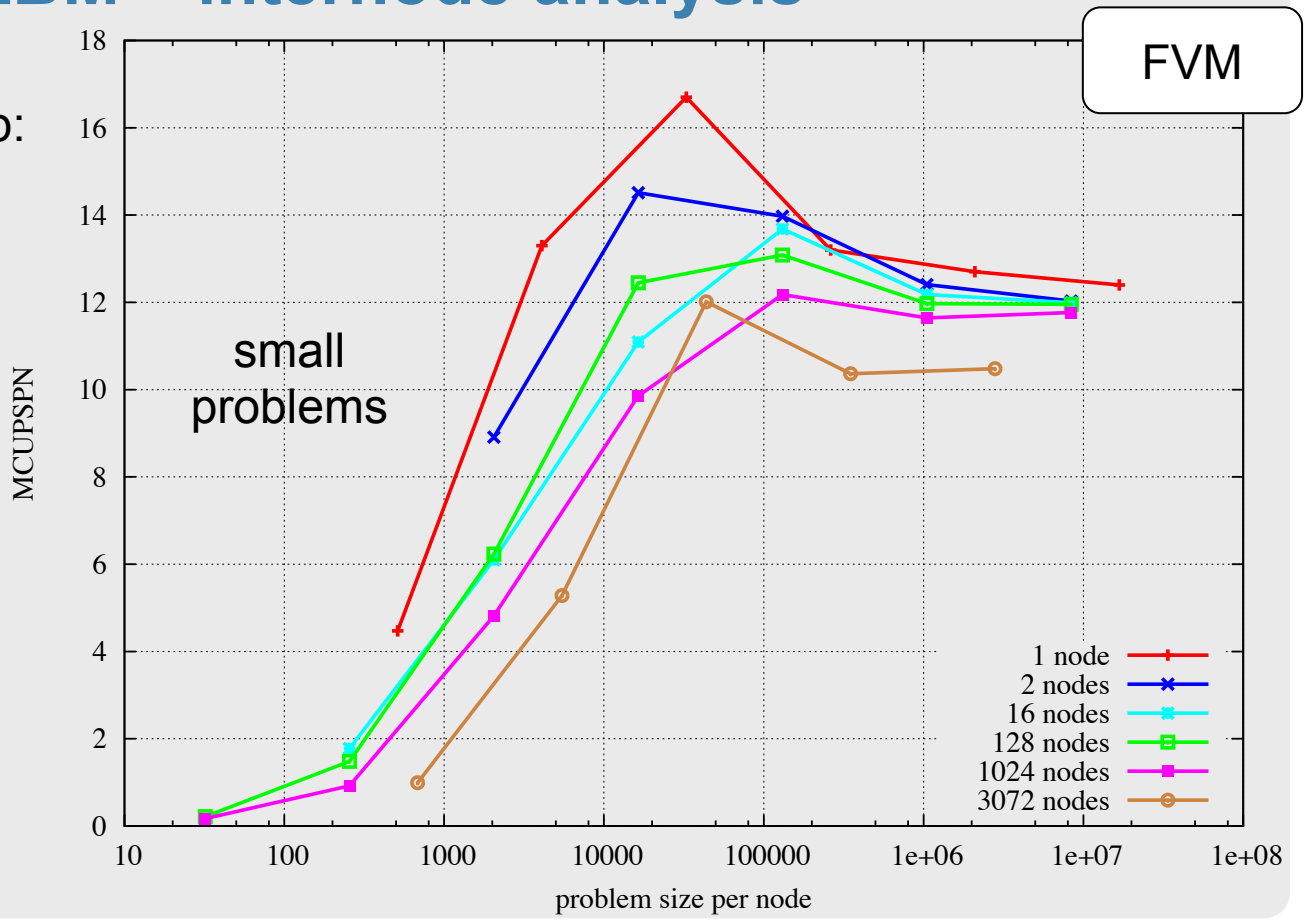
JÜLICH FORSCHUNGSZENTRUM

RWTH AACHEN UNIVERSITY

FVM and LBM – internode analysis

3 regions of the map:
- Communication dominated
- Caching effects
- Freq. memory access

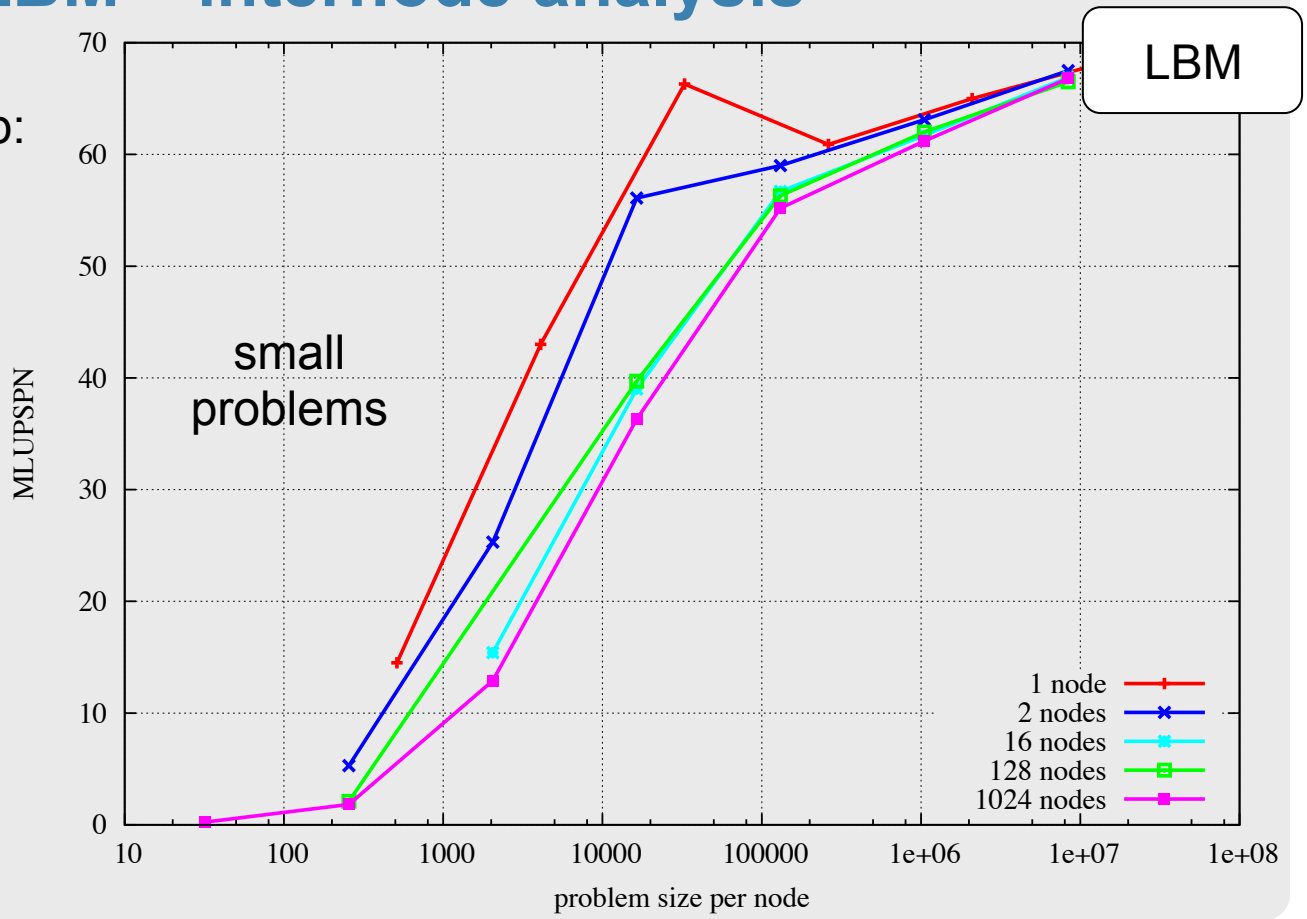
Further influences
- Method
- Optimization



FVM and LBM – internode analysis

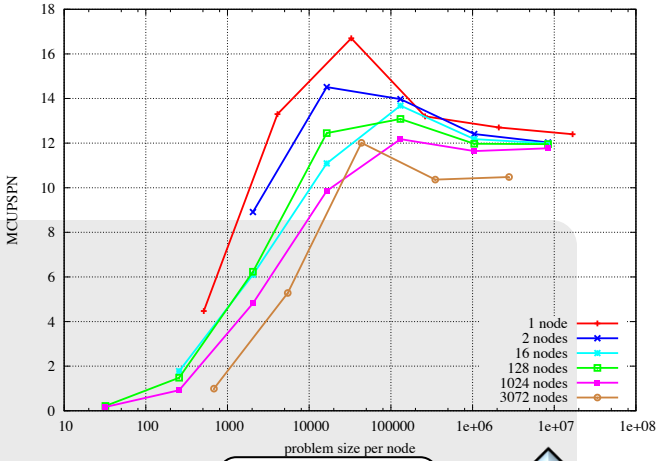
3 regions of the map:
- Communication dominated
- Caching effects
- Freq. memory access

Further influences
- Method
- Optimization



German Research School for Simulation Sciences

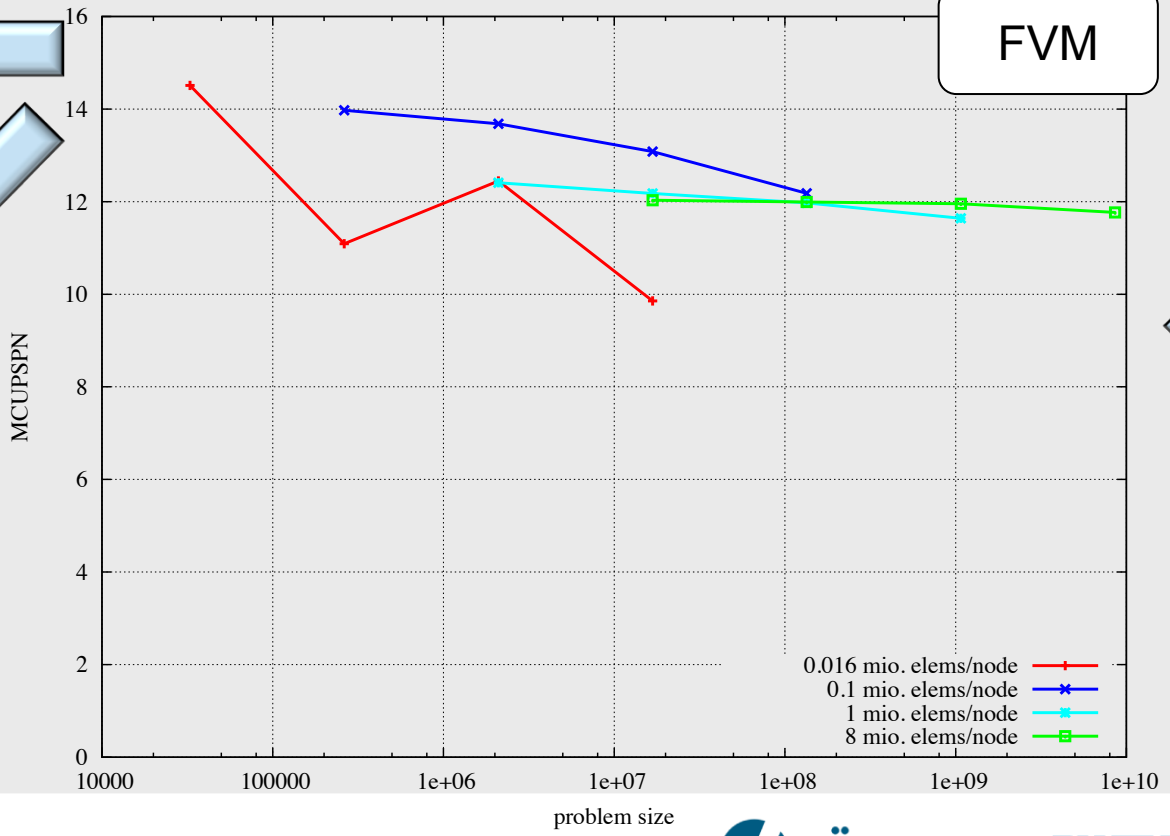
FVM (and LBM) – weak scaling



Small problem
- Cache effect
- Communication

Larger problem
- Freq. mem. access
- Nearly optimal

FVM

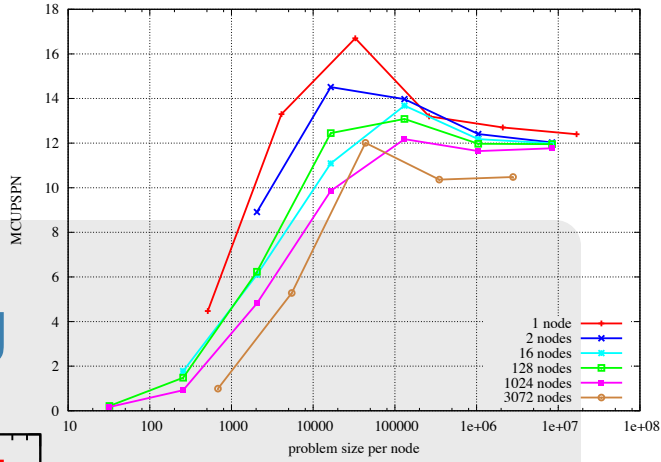


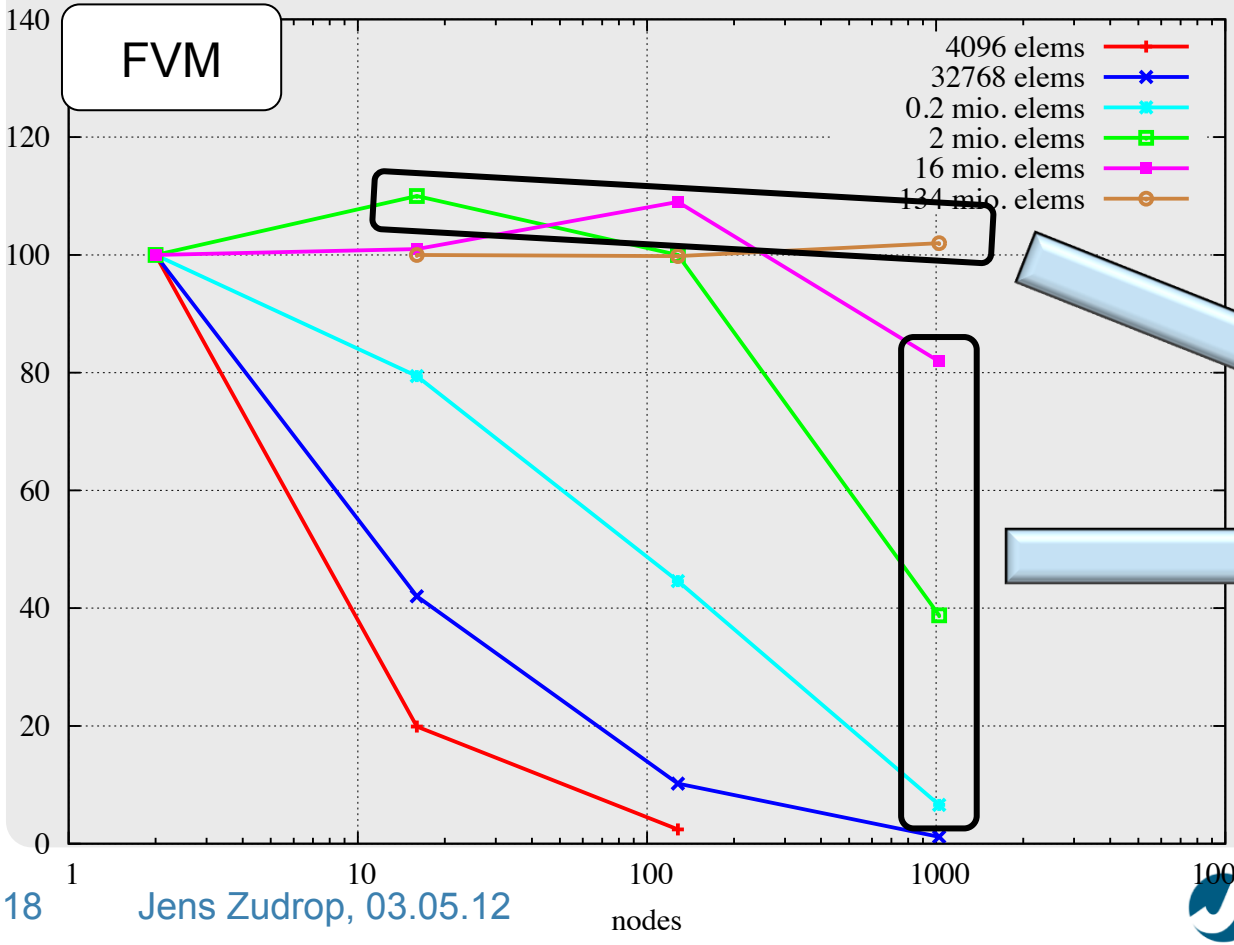
JÜLICH FORSCHUNGSZENTRUM

RWTH AACHEN UNIVERSITY

German Research School
for Simulation Sciences

FVM (and LBM) – strong scaling





Cache effect

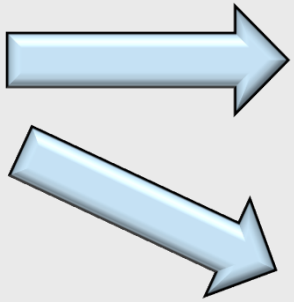
Communication

JÜLICH FORSCHUNGSZENTRUM

RWTH AACHEN UNIVERSITY

Conclusion

- Octree framework scales well
  - Promising end-to-end approach
- Scalabilty:
  - beyond factor of 500
  - up to cache size
- Works for complex geometries, too
- Relatively cheap algorithms, so far



Scales to more than 98304 cores / 3072 nodes

Up to 9% of peak performance, i.e. 76 TFLOPS

Outlook

- Higher order schemes (better strong scaling)
- Local refinement (load balancing)

German Research School
for Simulation Sciences

Thank you!

JÜLICH
FORSCHUNGSZENTRUM

RWTH AACHEN
UNIVERSITY