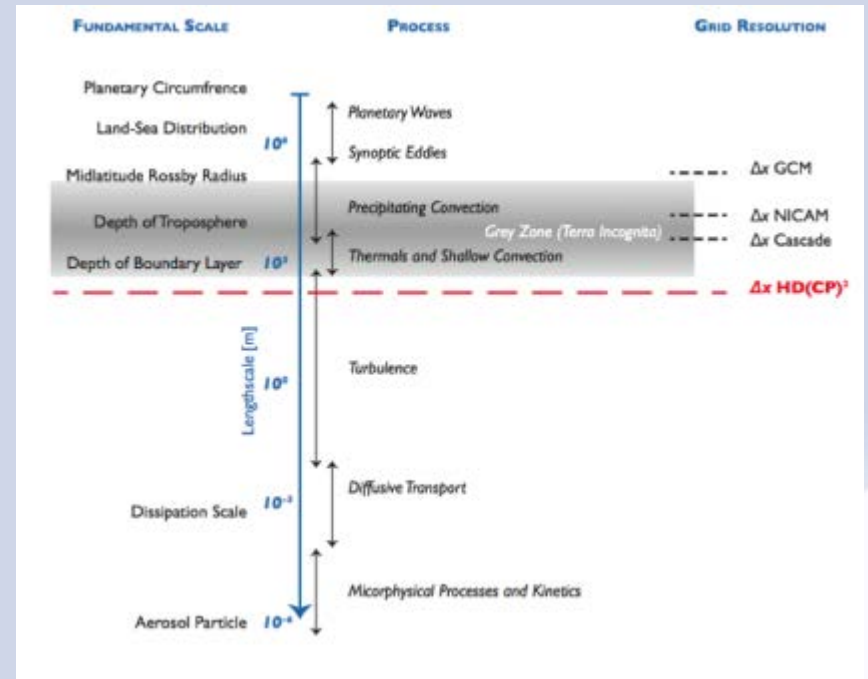


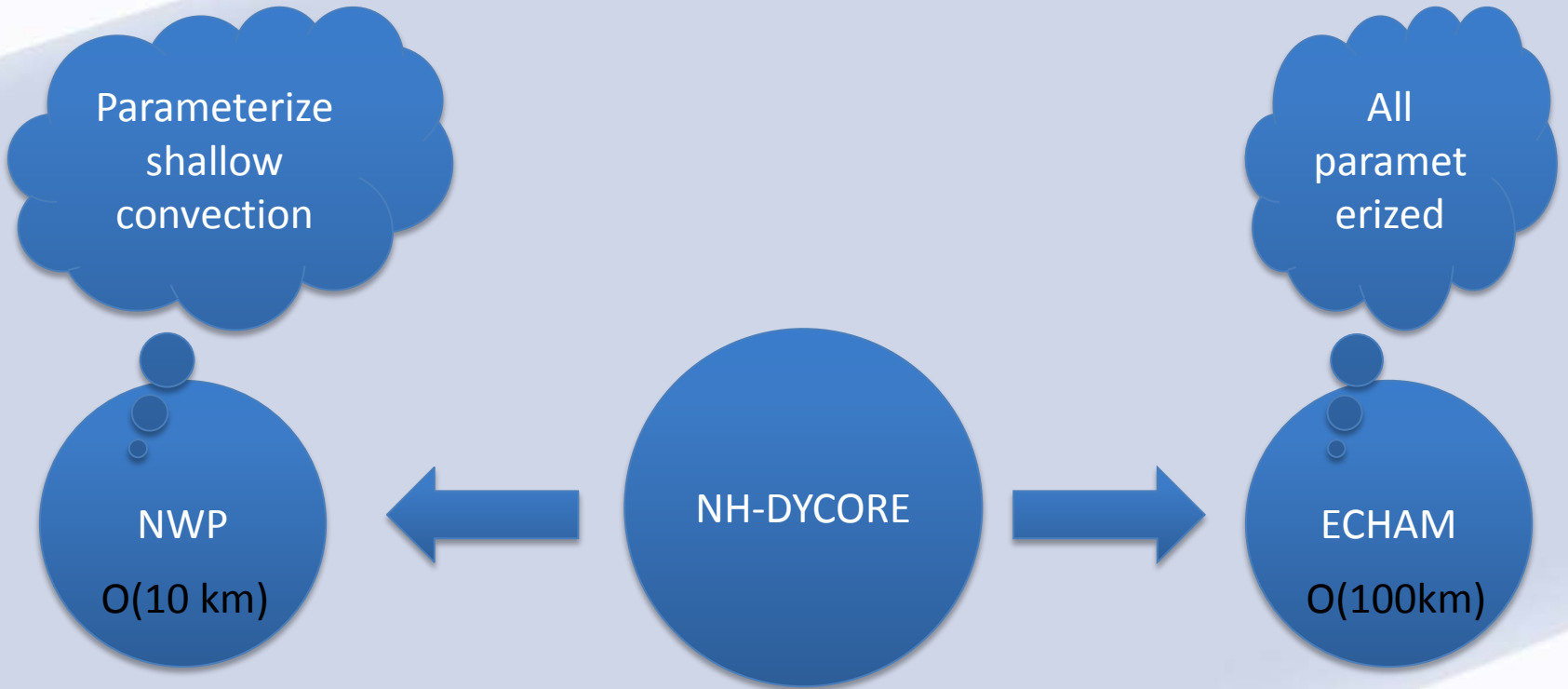
ICON for HD(CP)²

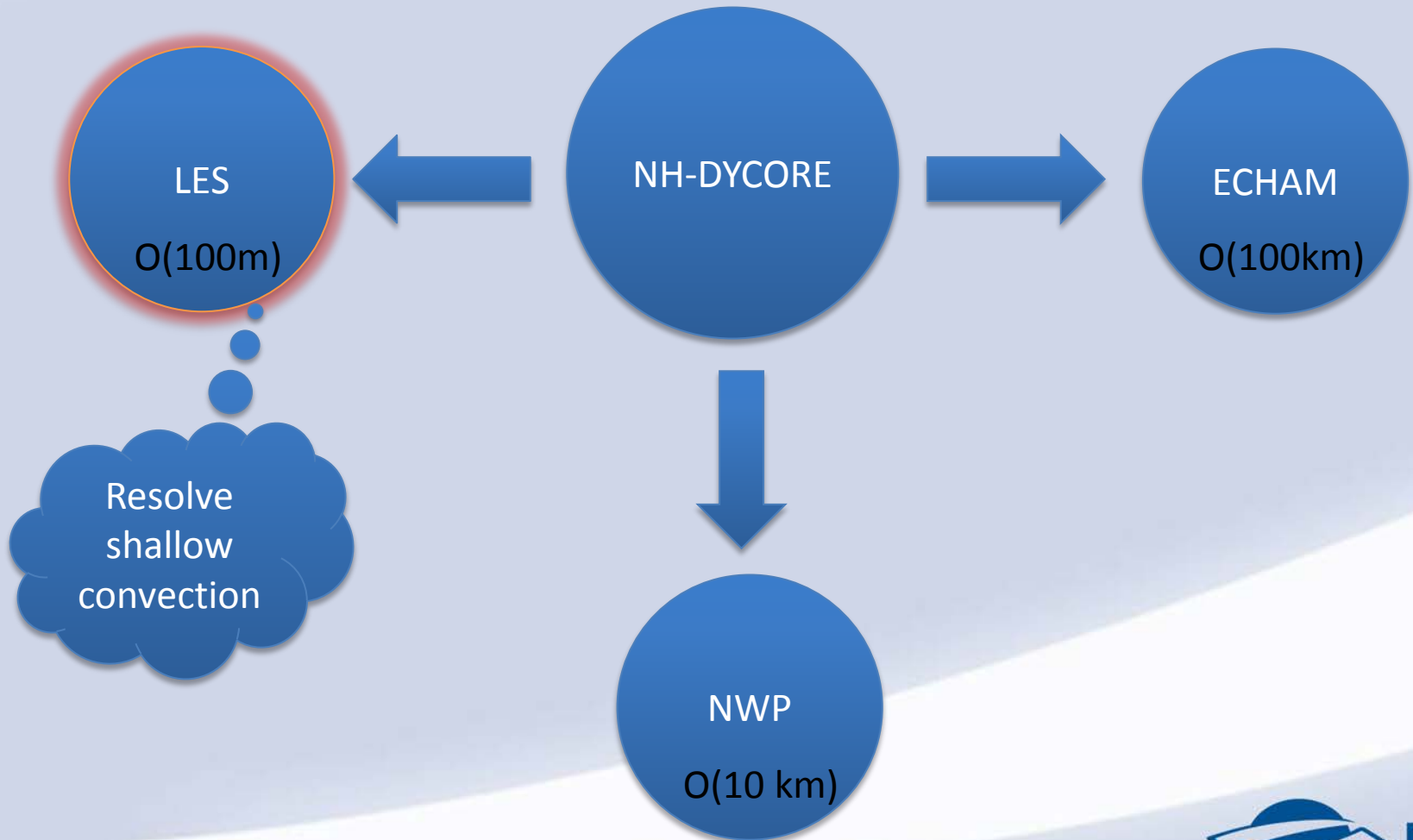
High Definition Clouds and
Precipitation for Advancing Climate
Prediction

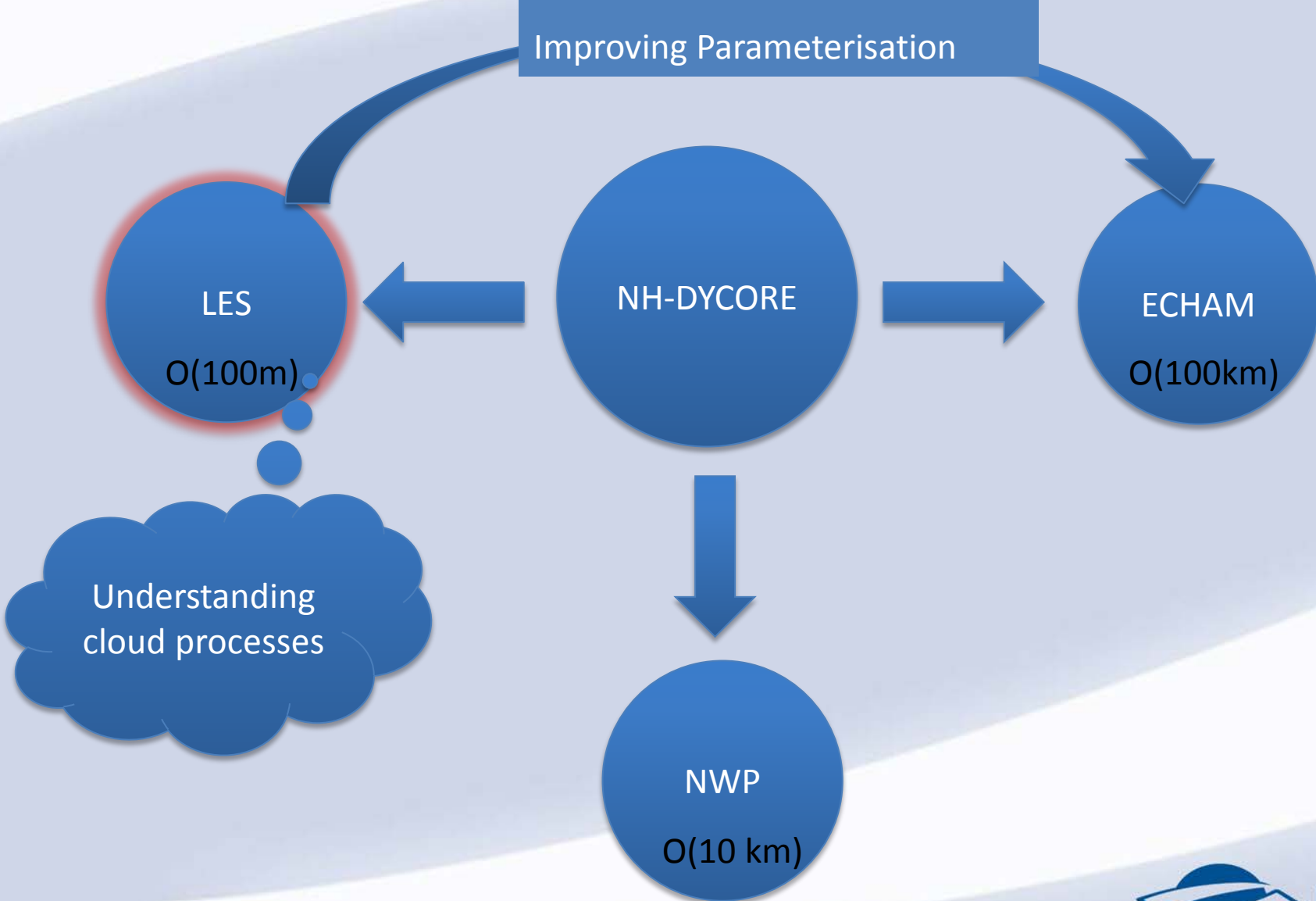
High Definition Clouds and Precipitation for Advancing Climate Prediction



ICON 2 years ago...







Computational Demands

Resolution	625m (50 level)	625m (50 level)	100m (200level)
Model time (days)	1	30	30
CPU time (hours)	6	180	180
Number of cores	448	448	70000

So we need ICON to scale to 100 thousands of core!!

Central Objective of HD(CP)²

- able to conduct production simulations on target grids as large as 10^{10} grid elements ($10^4 \times 10^4 \times 400$)
- grid spacing finer than 400 m (100 m being the target)
- capable of efficiently using a diversity of advanced high performance computing resources
- scaling of the model to many tens, possibly hundreds, of thousands of cores

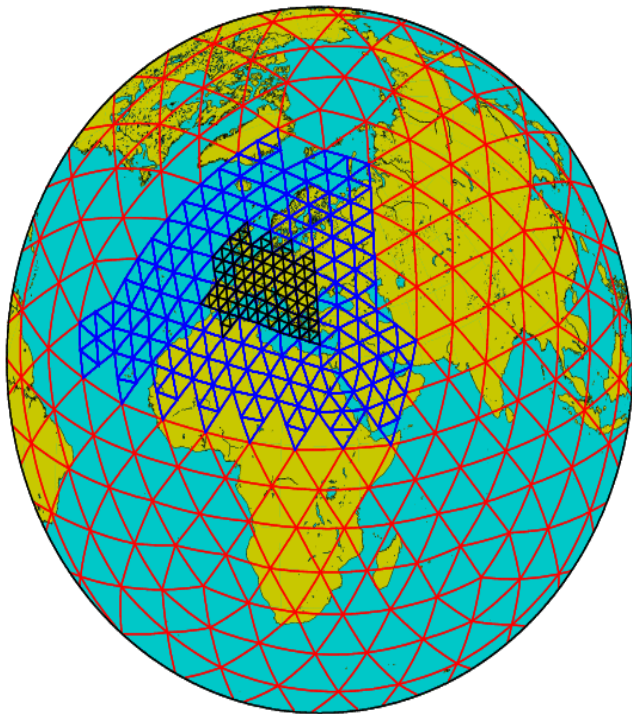
- Ensuring that the code runs efficiently on Peta- to Exa-Scale HPC systems
- This means the code must be
 - well optimized
 - scalable
 - portable
 - maintainable

Computational Challenges

- The complex architecture of Peta-ExaScale systems makes it hard to develop software which exploits the hardware in an optimal way
- **Computation is fast** (theoretical peak performance)
 - LINPACK
- **Moving data is the bottleneck**
 - between the levels of the memory hierarchy inside nodes/cores
 - communicating the data among processors
- **Need for new Algorithms with**
 - Optimal temporal/spatial locality

- **Minimizing („Avoiding“) Communication by**
 - Multilevel Parallelization
 - Locality and Topology aware Communication
 - Hybrid MPI/OpenMP Parallelization
- **Optimal mapping of the processes to the hierarchical architecture**
 - Find optimal load balancing
 - Minimize Communication
 - Graph partitioning algorithms using heuristics

Hybrid Parallelization on “blizzard”



- MPI (ST) 4 nodes
32 MPI-Processes/node
Wallclock : 779 sec
- MPI (SMT) 4 nodes
64 MPI-Processes/node
Wallclock : 549 sec
Gain : 29,5% wrt MPI (ST)
- MPI/OpenMP (SMT) 4 nodes
32 MPI-Processes x 2 OpenMP Threads
per node
Wallclock : 499 sec
Gain : 9% wrt MPI (SMT)
Gain : 35,9% wrt MPI (ST)

ICON on IBM Power6 “blizzard”

ST Mode

- MPI (ST) 1 node
Wallclock : 2981 sec
- MPI (ST) 2 nodes
Wallclock : 1516 sec
Strong Scaling=**1,96**
- MPI (ST) 4 nodes
Wallclock : 779 sec
Strong Scaling=**3,82**
- MPI (ST) 8 nodes
Wallclock : 410 sec
Strong Scaling=**7,27**

SMT Mode

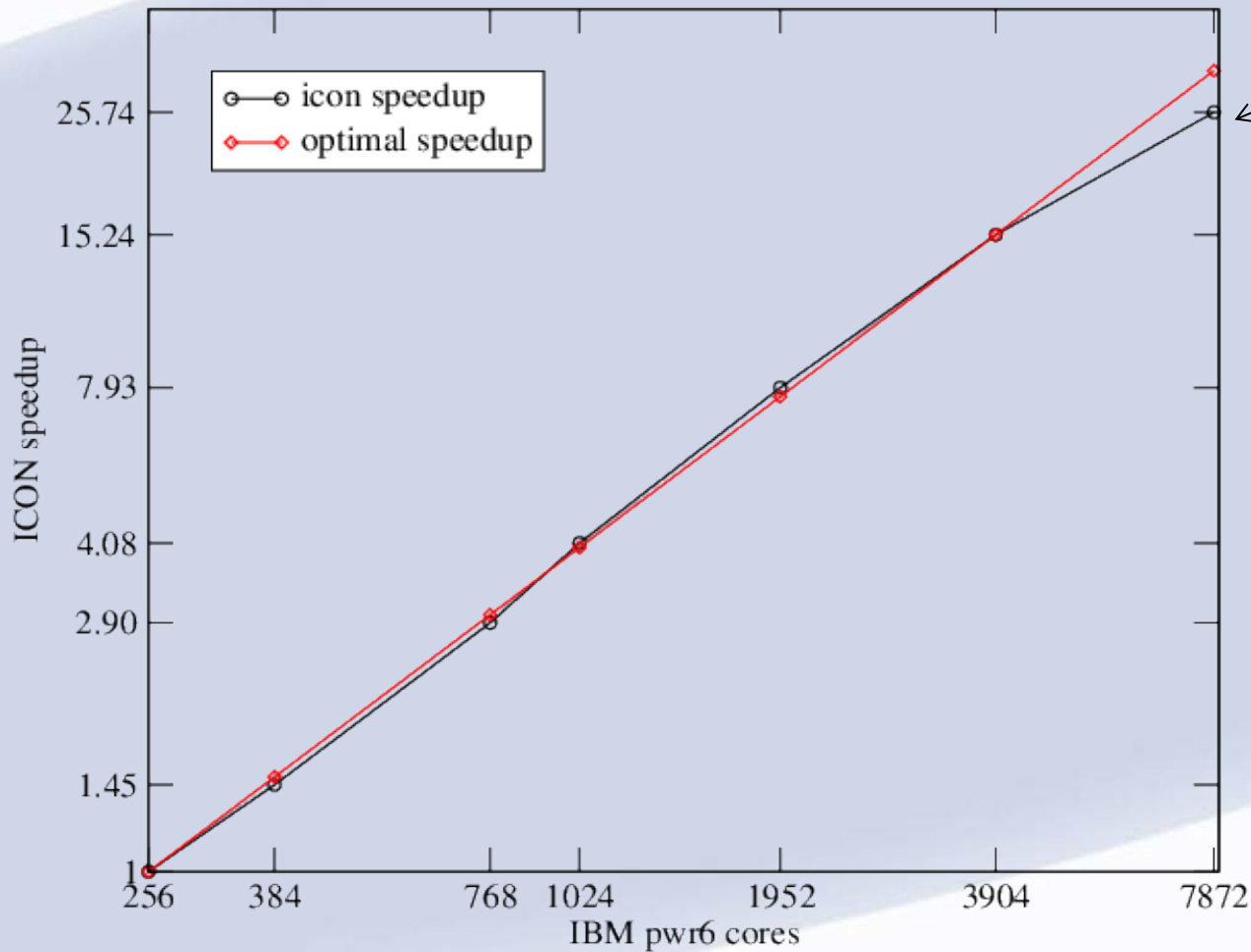
- MPI (SMT) 1 node
Wallclock : 2021 sec
Strong Scaling=1,47 -> **75% of the speedup on 2 nodes ST**
- MPI/OpenMP (SMT) 4 nodes
32 MPI-Processes x 2 OpenMP Threads
Wallclock : 499 sec
Strong Scaling = 5,97 -> **82% of the speedup on 8 nodes ST**

On the road to Petascale Systems

- R2B07, resolution 20km
- Limited Area Experiments:
 - resolution 416m with 3,514,000 cells
 - resolution 240m with 10,567,680 cells
- Supercomputing Systems
 - Blizzard : 8448 cores - 158 TeraFlop/s
 - SuperMUC : 147456 cores - 2.8 PetaFlop/s
 - JUQUEEN : 458752 cores - 5 PetaFlop/s

R2B07 GLOBAL 20km no I/O

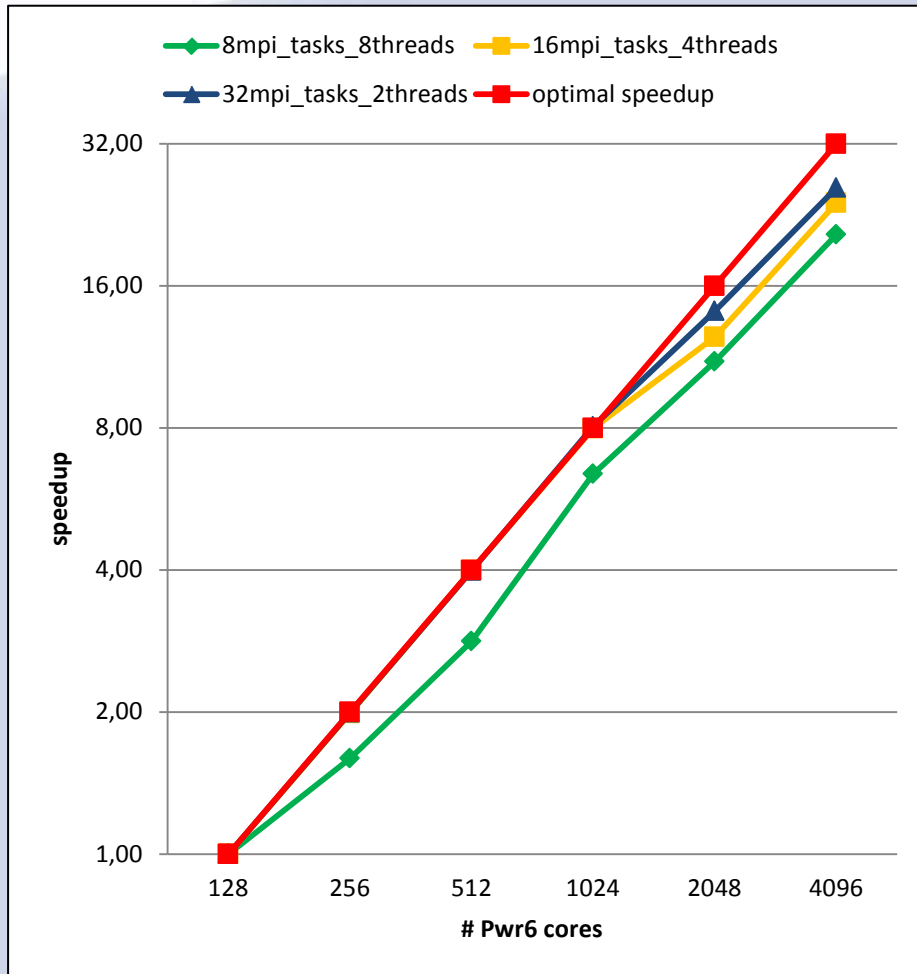
ICON strong scaling



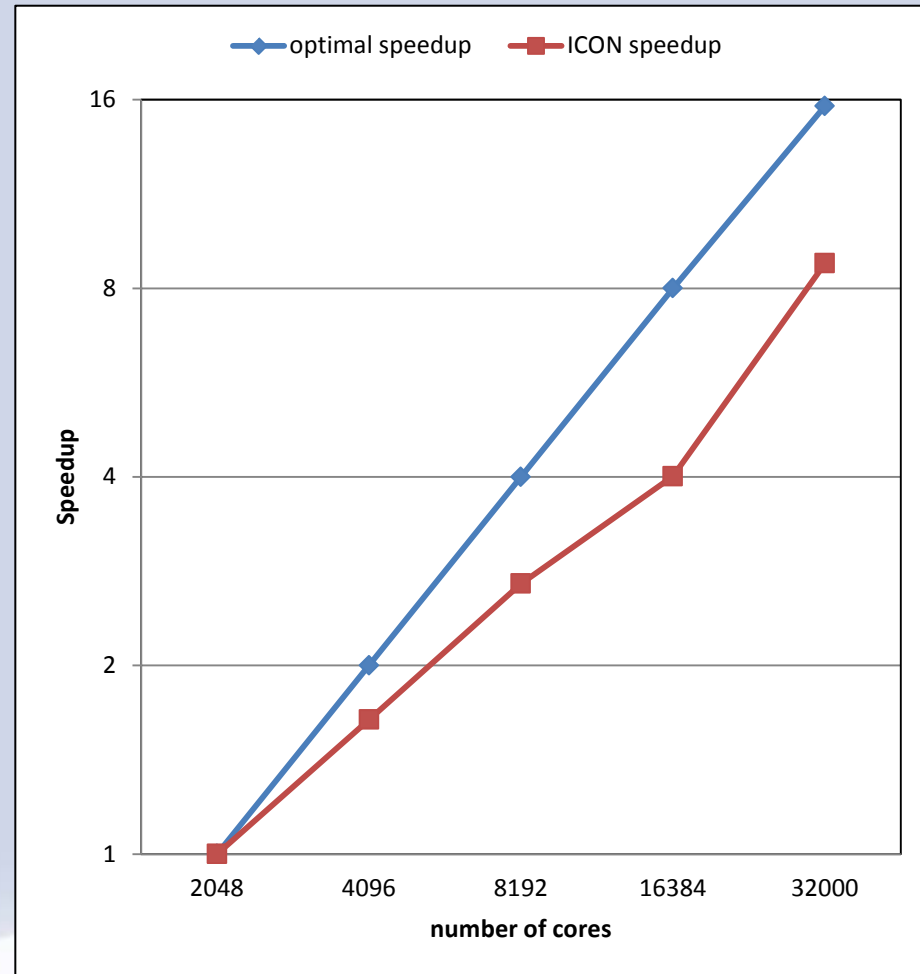
Efficiency 83%

Strong Scaling on Tera/Petascale Systems

LAM416m on BLIZZARD

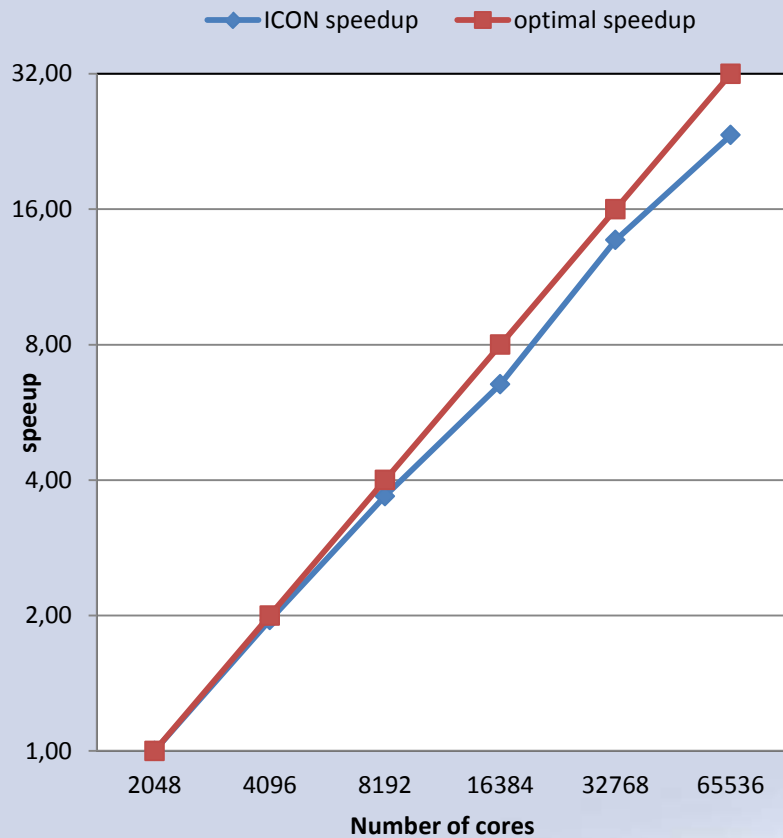


LAM416m on SuperMUC

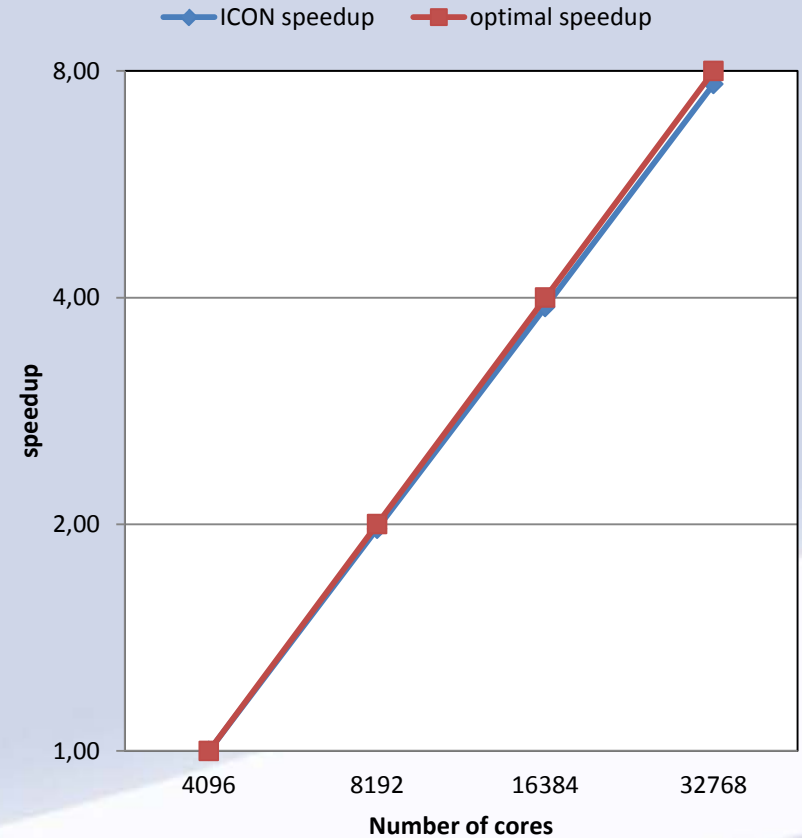


@Petascale with higher resolution

Strong Scaling on JUQUEEN LAM416m



Strong Scaling on JUQUEEN LAM240m

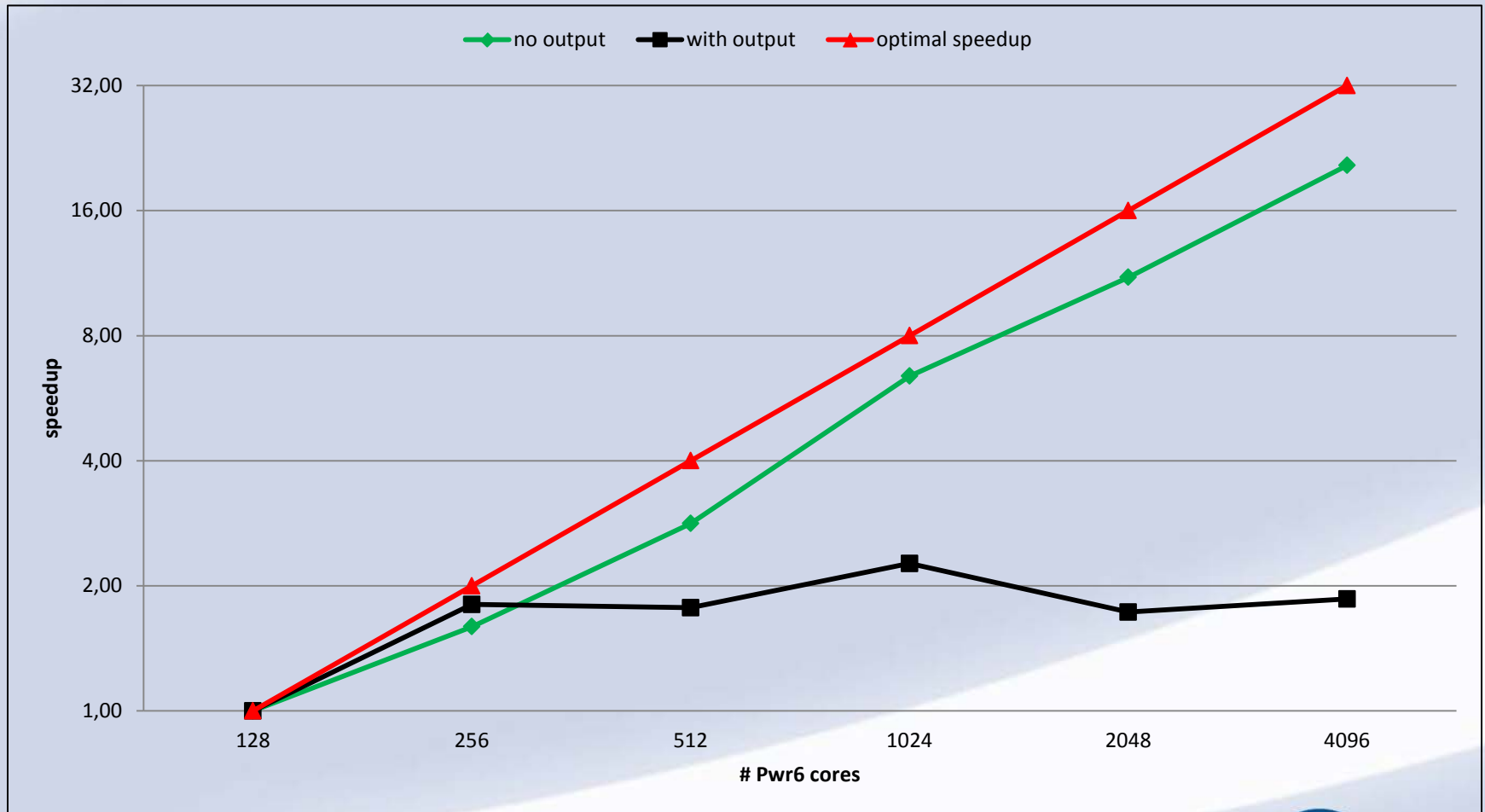


Current State with Output

- On Blizzard
- Experiment: exp.nh_hdcp2_lam_r0416m
- 64 nodes -> 2048 cores
- Simulated Time = 2 hours
- Wallclock= 1h 26min.
- Memory used = 5257 GB (82GB/pwr6 node)
- Size of output file = **289G**
- Aprox. 1.38 simulated days/day
- Serial I/O
 - ICON-output -> gathering of global data on output task

Challenges for Petascale Systems

Impact of Serial I/O for LAM416m on BLIZZARD



From Petascale to Exascale Systems

- Taking the challenges for Petascale HPC-Systems as chance to prepare for Exascale HPC-Systems
- Necessary changes of ICON are in progress
 - Parallelizing, Parallelizing,.....,Parallelizing
- Future Exascale architectures with accelerators ?
 - GPGPU's, Intel MIC (Xeon Phi)..

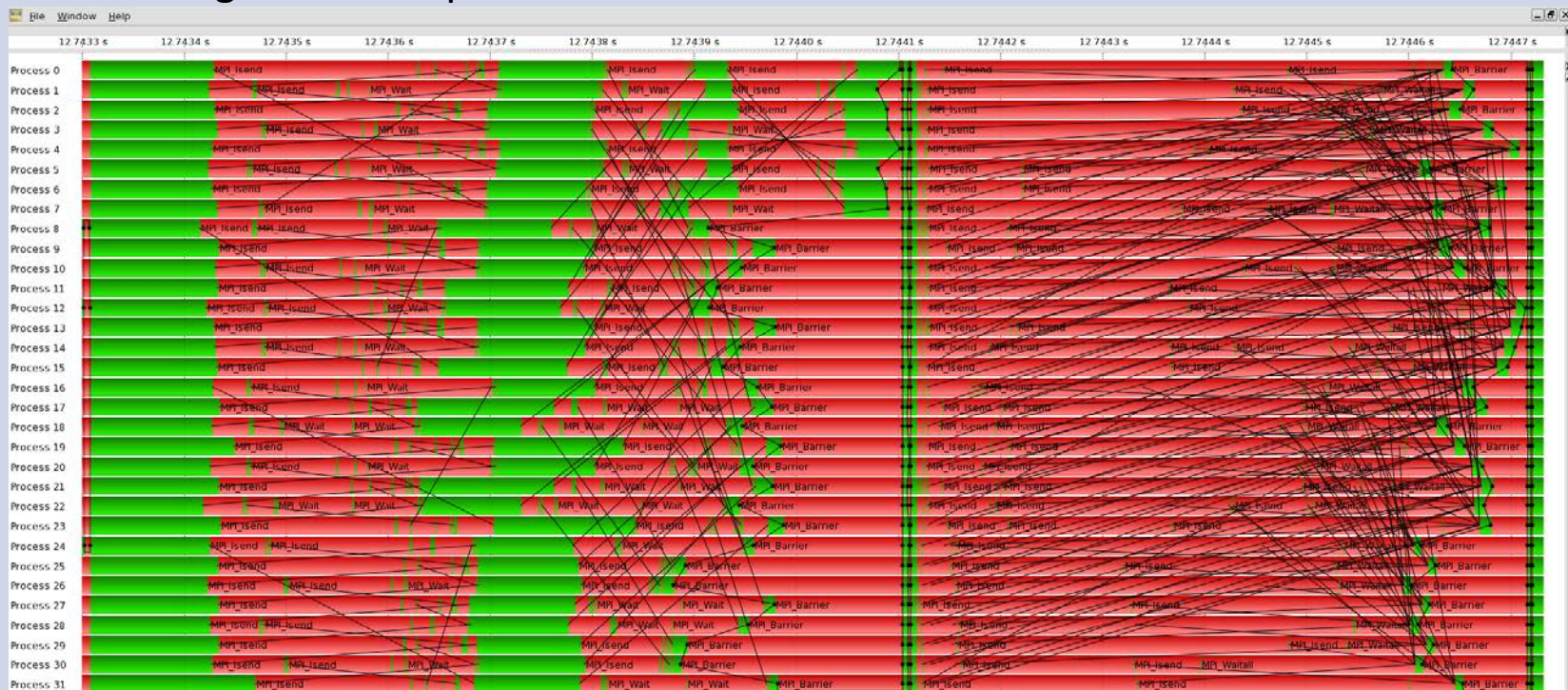
YAXT – Yet Another Communication Tool

- Library on top of MPI
- Simple to use:
 - Evaluates complete decomposition description
 - No explicit message passing visible
- Can reduce latency problem:
 - Climate models & high scaling: many small messages
 - Latency: constant overhead
 - Aggregation reduces communication overhead

MPI OM 3d-bounds-exchange measurement

Handwritten – requires symmetric
regular decomposition

YAXT general solution



green: data copy

no data copy: 25% faster