Zbigniew P. Piotrowski *,**

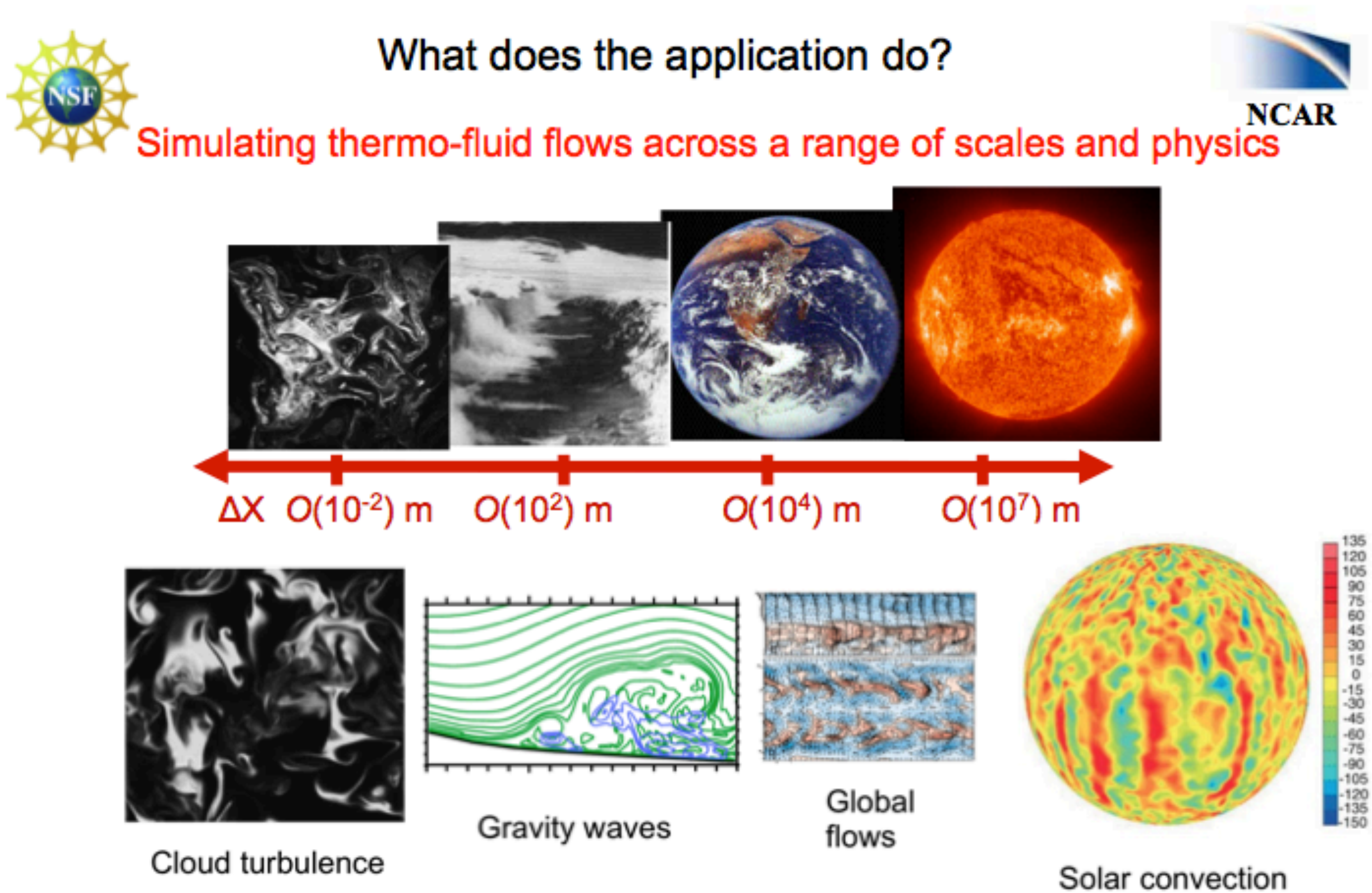
Collaborators: Andrzej A. Wyszogrodzki
Piotr K. Smolarkiewicz, NCAR

Towards petascale simulation of atmospheric circulations with soundproof equations

*Geophysical Turbulence Program,
National Center for Atmospheric Research, Boulder, Colorado, U.S.A.
**On the leave from Institute for Meteorology and Water Management, Warsaw, Poland

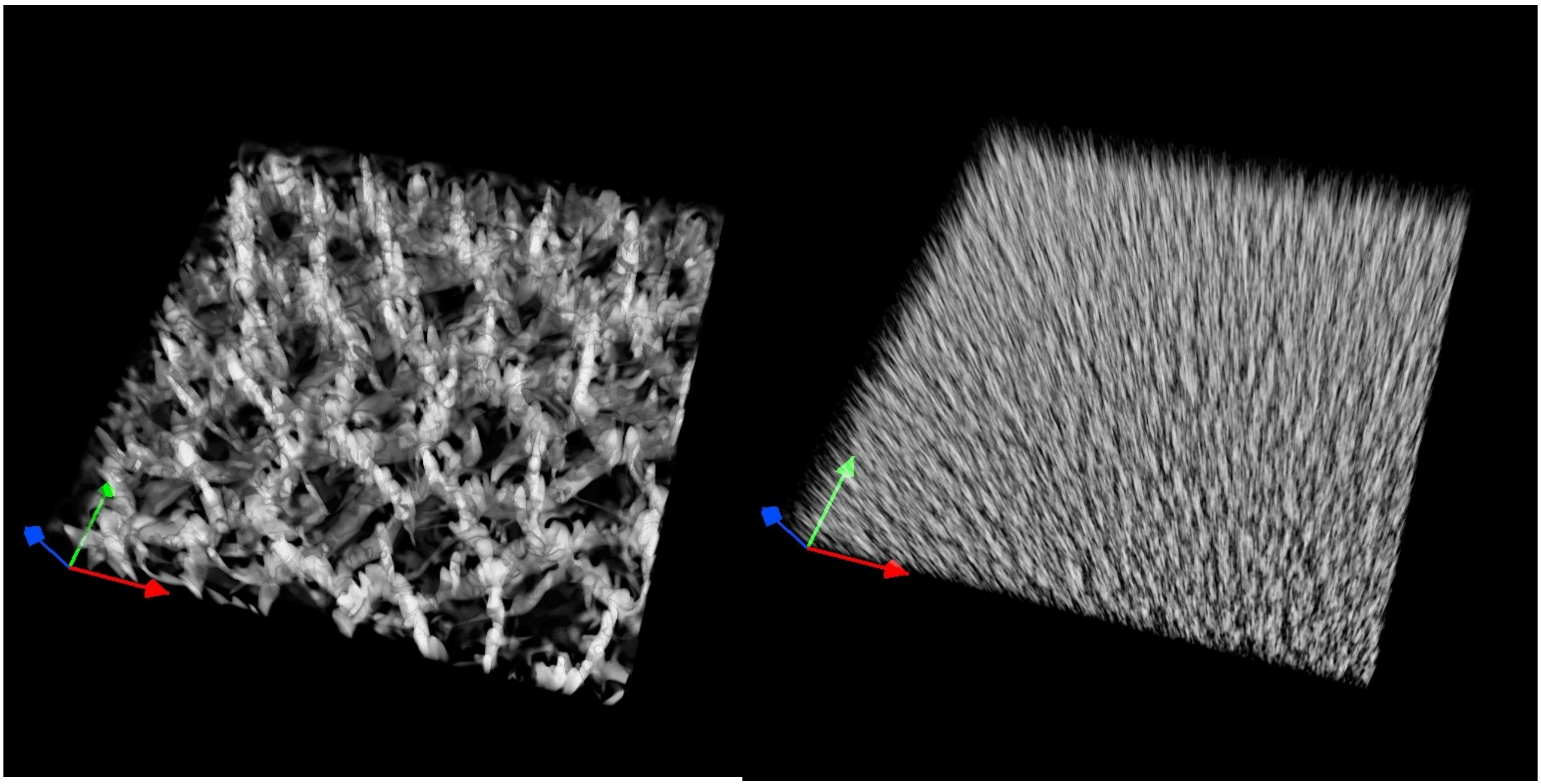
NCAR is sponsored by the National Science Foundation

EULAG model – powerful virtual laboratory



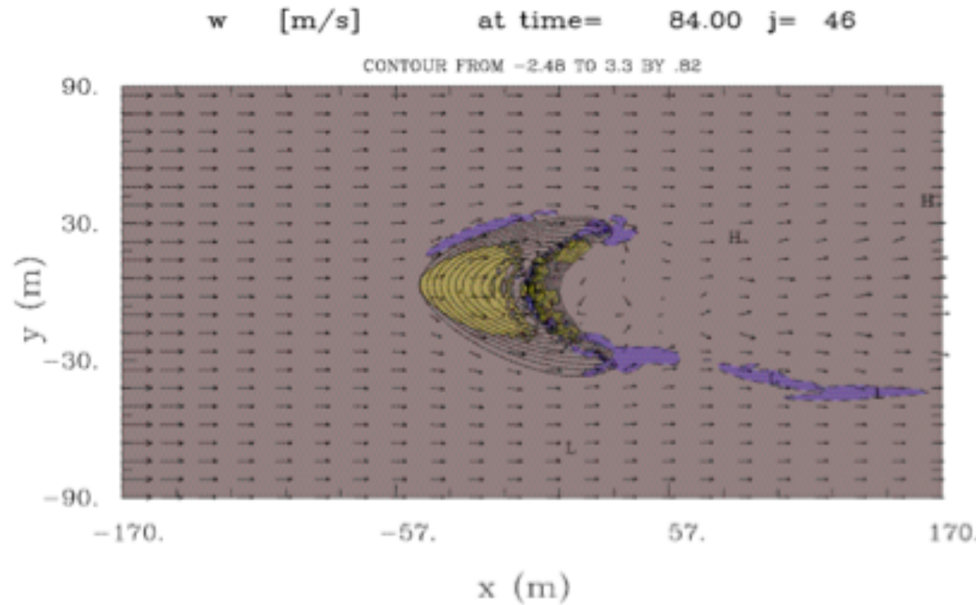
Examples of applications

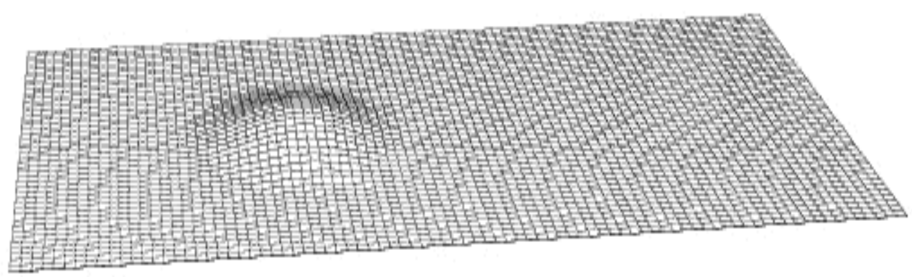
Investigation of numerical realizability of idealized thermal convection over heated plane (Piotrowski et al. 2009, JCP)



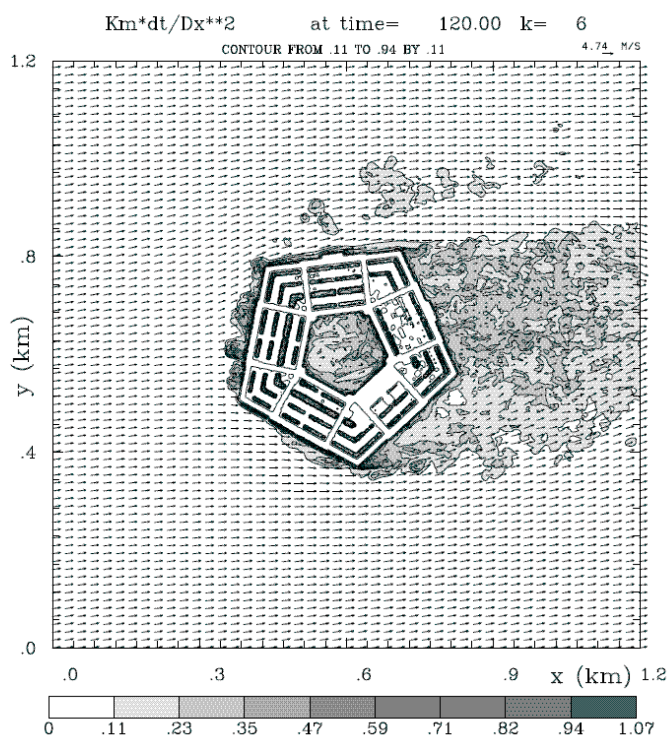
Imagery produced by VAPOR (www.vapor.ucar.edu), a product of NCAR

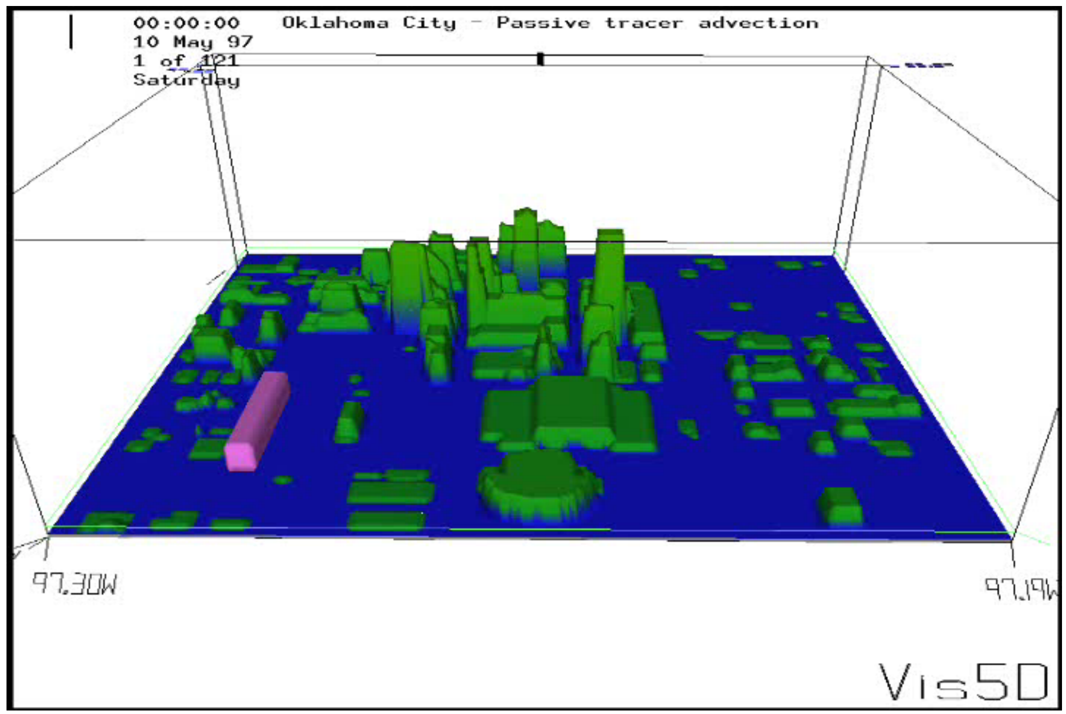
Simulations of boundary layer flows past rapidly evolving sand dunes
LES, with all relevant sub-grid scales parameterized
(Ortiz et al 2009, Phys. Rev.)





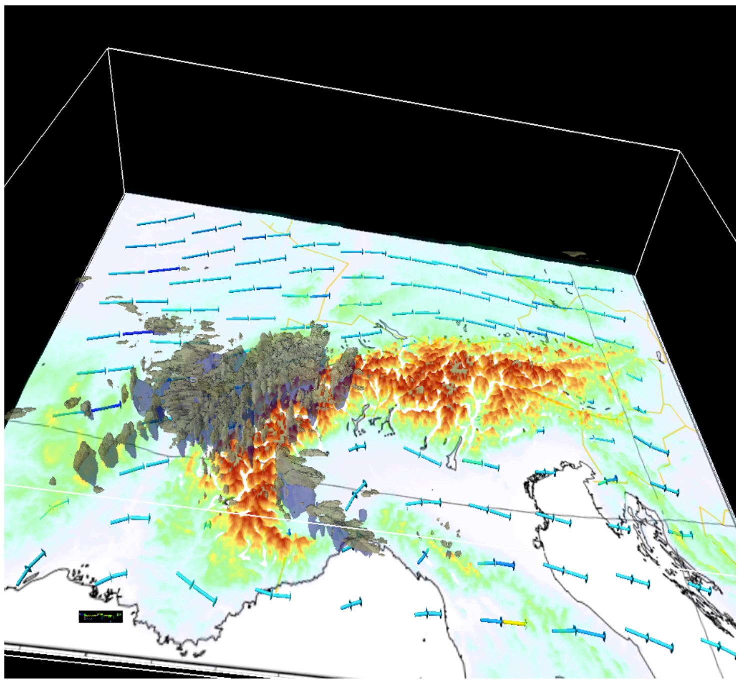
Urban planetary boundary layer

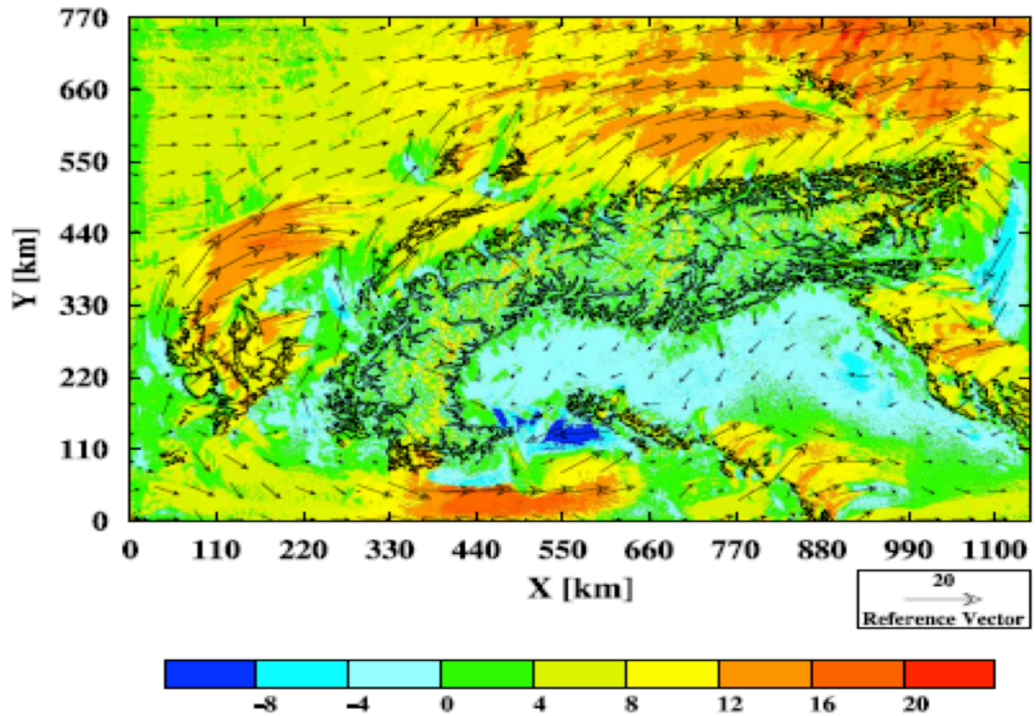


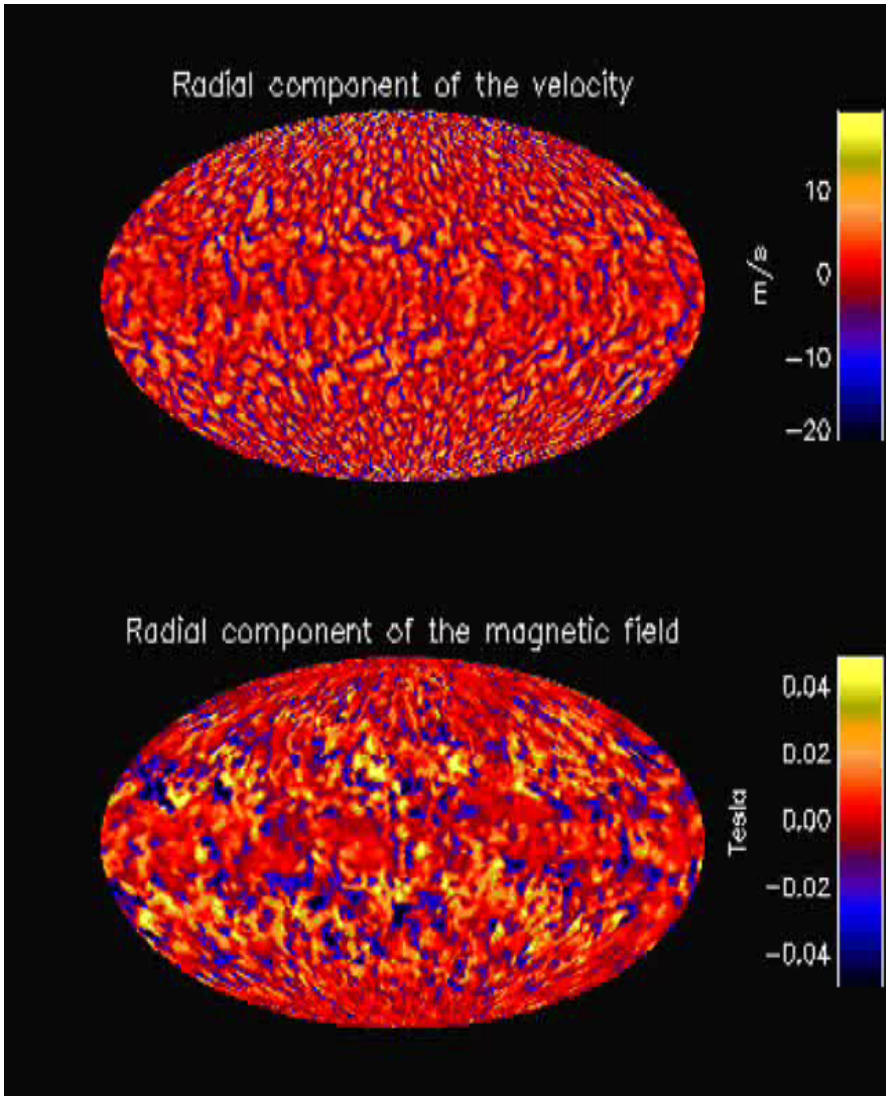


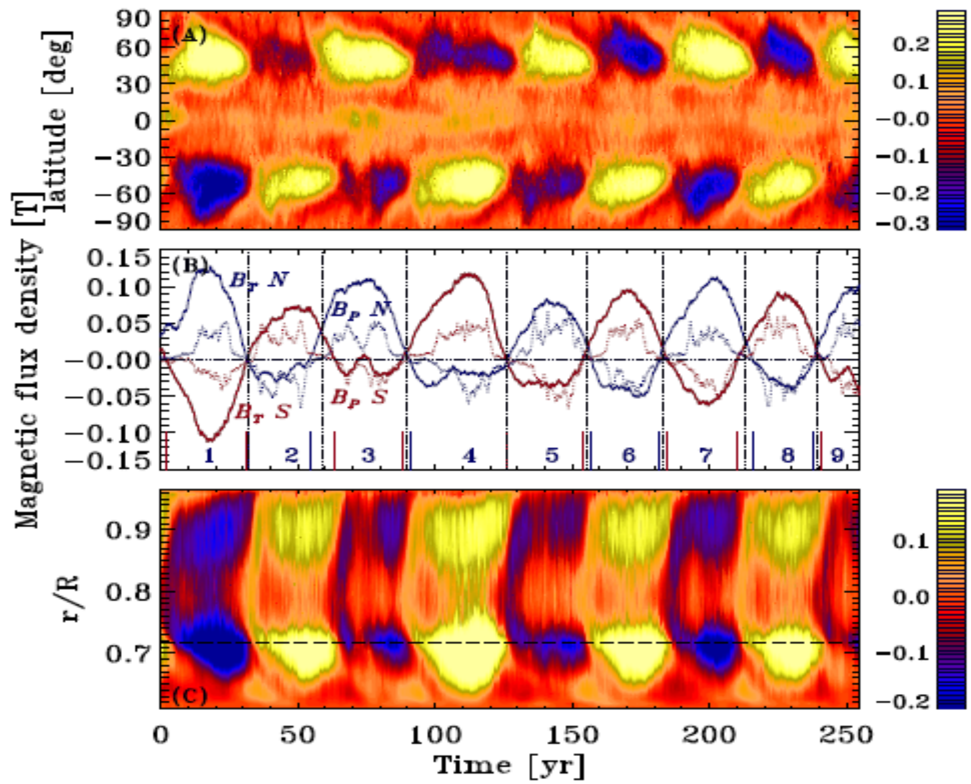
(Smolarkiewicz et al. 2007, *JCP*)

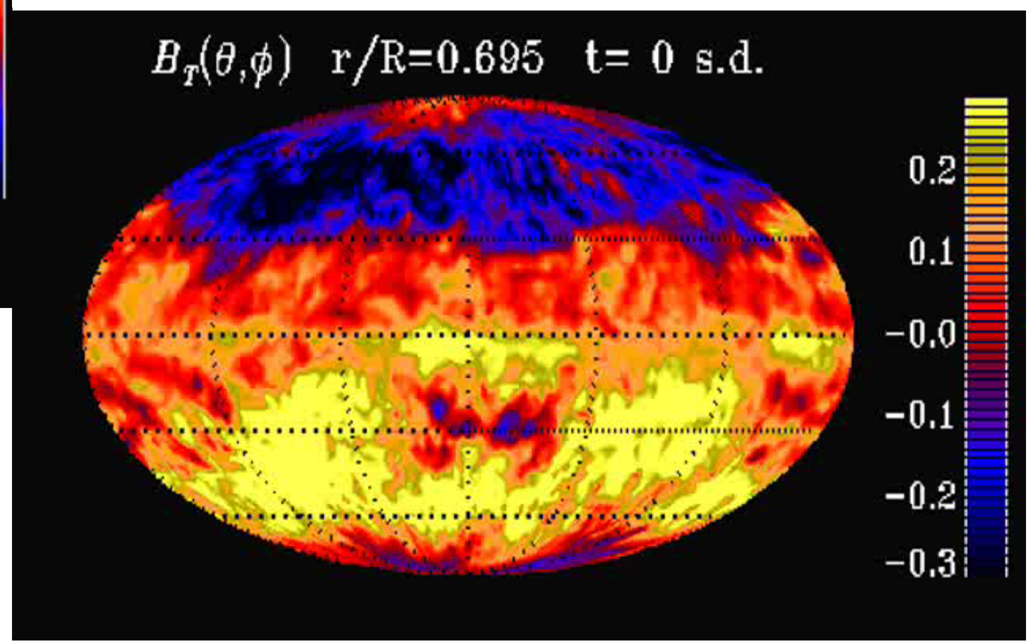
Prospective dynamical core for high resolution COSMO consortium regional NWP over Europe
(Ziemiański et al. 2011, Acta Geoph.)











Toroidal component of B in the uppermost portion of the stable layer underlying the convective envelope at r/R≈0 .7 ➔

Scientific approach

Two fundamental algorithms: MPDATA advection + GCRK pressure solver

Two optional modes for integrating fluid PDEs:
- Eulerian --- control-volume wise integral
- Lagrangian --- trajectory wise integral

Optional fluid equations (nonhydrostatic):
- Anelastic
- Compressible/incompressible Boussinesq
- Incompressible Euler/Navier-Stokes'
- Fully compressible for high-speed flows
- Anelastic MHD
- Anelastic for unstructured grid formulation

Available strategies for simulating turbulent dynamics:
- Direct numerical simulation (DNS)
- Large-eddy simulation, explicit and implicit (LES, ILES)

Multidimensional positive definite advection transport algorithm (MPDATA).

→ Starts with iteration of upwind scheme, then applies nonlinear corrective iterations of upwind with negative diffusion

$$\frac{\partial \phi}{\partial t} = -\nabla \bullet (\mathbf{V}\phi) \ , \qquad \phi_i^{n+1} = \phi_i^n - \frac{\delta t}{\mathcal{V}_i} \sum_{j=1}^{l(i)} F_j^{\perp} S_j$$

$$F_j^{\perp}(\phi_i, \phi_j, V_j^{\perp}) = [V_j^{\perp}]^{+} \phi_i + [V_j^{\perp}]^{-} \phi_j \ , \qquad [V]^{+} \equiv 0.5(V + |V|) \ \ , \ \ [V]^{-} \equiv 0.5(V - |V|) \ ,$$

$$\phi_i^{(k)} = \phi_i^{(k-1)} - \frac{\delta t}{\mathcal{V}_i} \sum_{j=1}^{l(i)} F_j^{\perp} \left(\phi_i^{(k-1)}, \phi_j^{(k-1)}, V_j^{\perp,(k)} \right) S_j$$

with $k = 1, .., IORD$ such that

$$\phi^{(0)} \equiv \phi^n \ \ ; \ \ \phi^{(IORD)} \equiv \phi^{n+1}$$

$$V^{\perp,(k+1)} = V^{\perp} \left(\mathbf{V}^{(k)}, \phi^{(k)}, \nabla \phi^{(k)} \right) \ \ ; \ \ V_j^{\perp,(1)} \equiv V^{\perp}|_j^{n+1/2}$$

$$V^{\perp}\Big|_{s_j}^{(k+1)} = \Bigg\{ 0.5|V^{\perp}| \left(\frac{1}{|\phi|} \frac{\partial |\phi|}{\partial r} \right) (r_j - r_i) \ - 0.5 V^{\perp} \left(\frac{1}{|\phi|} \frac{\partial |\phi|}{\partial r} \right) (r_i - 2 r_{s_j} + r_j)$$

$$- 0.5 \delta t V^{\perp} \left(\mathbf{V} \bullet \frac{1}{|\phi|} \nabla |\phi| \right) - 0.5 \delta t V^{\perp} (\nabla \bullet \mathbf{V}) \Bigg\} \Bigg|_{s_j}^{(k)}$$

Numerical design

All principal forcings are assumed to be unknown at *n*+1

$$\psi_{\mathbf{i}}^{n+1} = LE_{\mathbf{i}}(\psi^n + 0.5\Delta t R^n) + 0.5\Delta t R_{\mathbf{i}}^{n+1}$$

$\Rightarrow$ system implicit with respect to all dependent variables.

On grids co-located with respect to all prognostic variables, it can be inverted algebraically to produce an elliptic equation for pressure

$$\left\{ \frac{\Delta t}{\rho^*} \overline{\nabla} \cdot \rho^* \widetilde{\mathbf{G}}^T \left[\widehat{\widehat{\mathbf{v}}} - (\mathbf{I} - 0.5\Delta t \widehat{\mathbf{R}})^{-1} \widetilde{\mathbf{G}} (\overline{\nabla} \pi'') \right] \right\}_{\mathbf{i}} = 0$$

solenoidal velocity $\overline{\mathbf{v}}^s \equiv \overline{\mathbf{v}}^* - \dfrac{\partial \overline{\mathbf{x}}}{\partial t}$ *contravariant velocity* $\overline{\mathbf{v}}^* \equiv d\overline{\mathbf{x}}/d\overline{t} \equiv \dot{\overline{\mathbf{x}}}$

$$\widetilde{\mathbf{G}}^T [\widehat{\widehat{\mathbf{v}}} - (\mathbf{I} - 0.5\Delta t \widehat{\mathbf{R}})^{-1} \widetilde{\mathbf{G}} (\overline{\nabla} \pi'')] \equiv \overline{\mathbf{v}}^s$$

Boundary conditions on π'' Imposed on $\overline{\mathbf{v}}^s \bullet \mathbf{n}$ subject to the integrability condition

$$\int_{\partial\Omega} \rho^* \overline{\mathbf{v}}^s \bullet \mathbf{n} d\sigma = 0$$

Boundary value problem is solved using nonsymmetric Krylov subspace solver - a preconditioned generalized conjugate residual GCR(*k*) algorithm (Smolarkiewicz and Margolin, 1994; Smolarkiewicz et al., 2004)

Programming Model

- Fortran 77 (fixed form …)
- Model fits in one file
- C-shell preprocessor

Parallel features

- Two or three dimensional decomposition with MPI
- Libraries in parallel mode: serial and parallel Netcdf, Vis5d
- Currently run on Bluegene/L, POWER 6, Cray XT4, XT5, XE6, Linux clusters, PC workstations, etc.
- Performance testing with Tau, Scalasca, CrayPat

Eulag parallelization history

1996-1998: **compiler parallelization on NCAR's vector Crays J90**

1996-1997: **first MPP (PVM)/SMP (SHMEM) version at NCAR's Cray T3D based on 2D domain decomposition (Anderson)**

1997-1998: **extension to MPI, removal of PVM (Wyszogrodzki)**

2004: **attempt to use OpenMP (Andrejczuk)**

2009- : **development of GPU/OpenCL version (Rojek, Szustak, Kurowski)**

2010-2011: **extending 2D decomposition to 3D MPP (Piotrowski & Wyszogrodzki)**

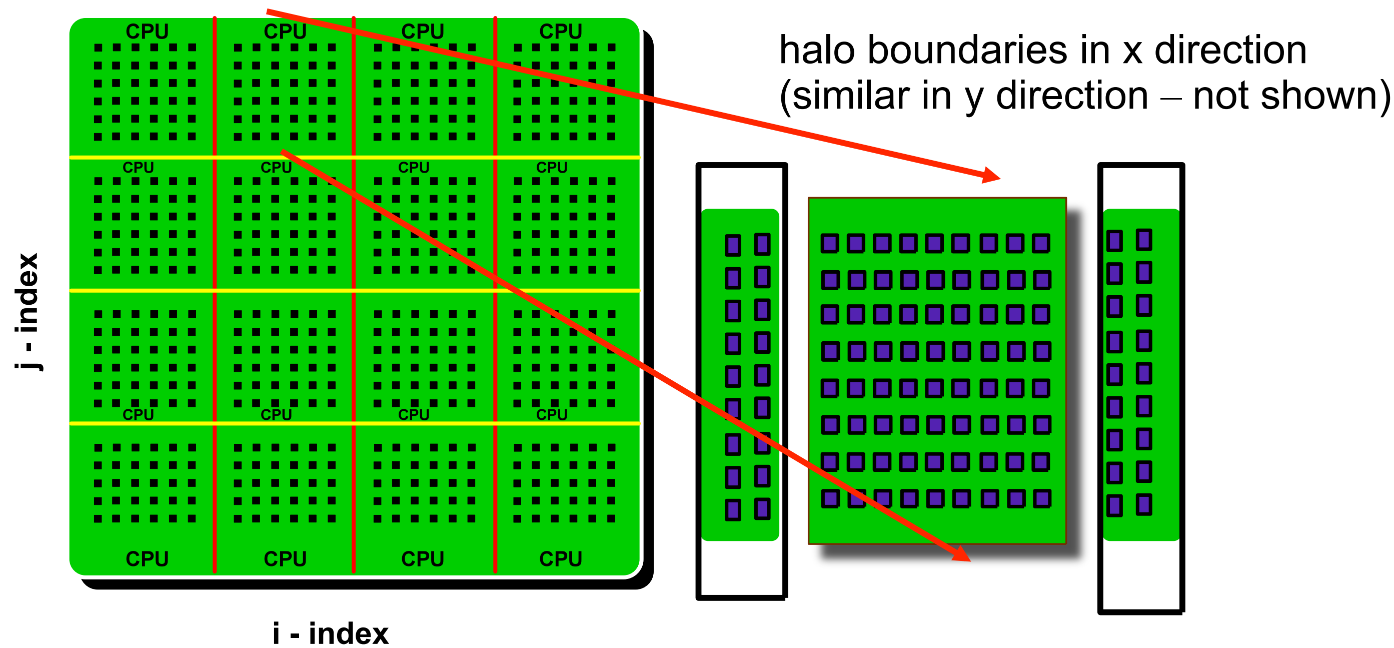
Motivation for 3D parallelization with MPI

- Improve scalability properties and enable efficient use of petascale-era supercomputers

- Increase maximum number of cores used in symmetric domains, like cloud turbulence studies

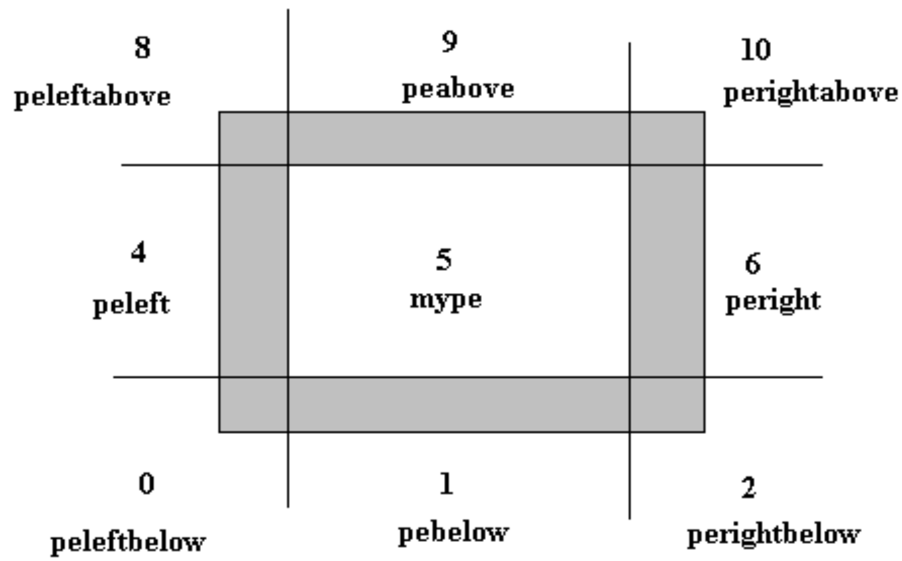
- Decreasing time-to-solution for problems demanding long integration in time

2D-MPI data decomposition in EULAG



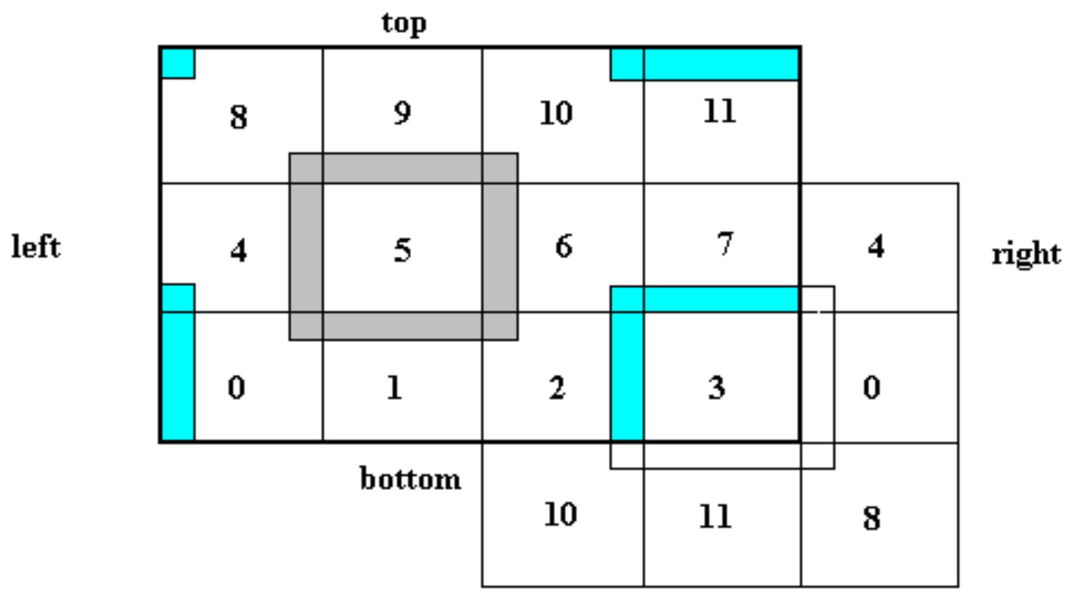
- 2D horizontal domain grid decomposition
- No decomposition in vertical Z-direction
- Hallo/ghost cells for collecting information from neighbors
- Predefined halo size for array memory allocation
- Selective halo size for update to decrease overhead

Typical processors configuration



- **Computational 2D grid is mapped onto an 1D grid of processors**
- **Neighboring processors exchange messages via MPI**
- **Each processor know its position in physical space (column, row, boundaries) and location of neighbor processors**

EULAG – Cartesian grid configuration



← **In the setup on the left**

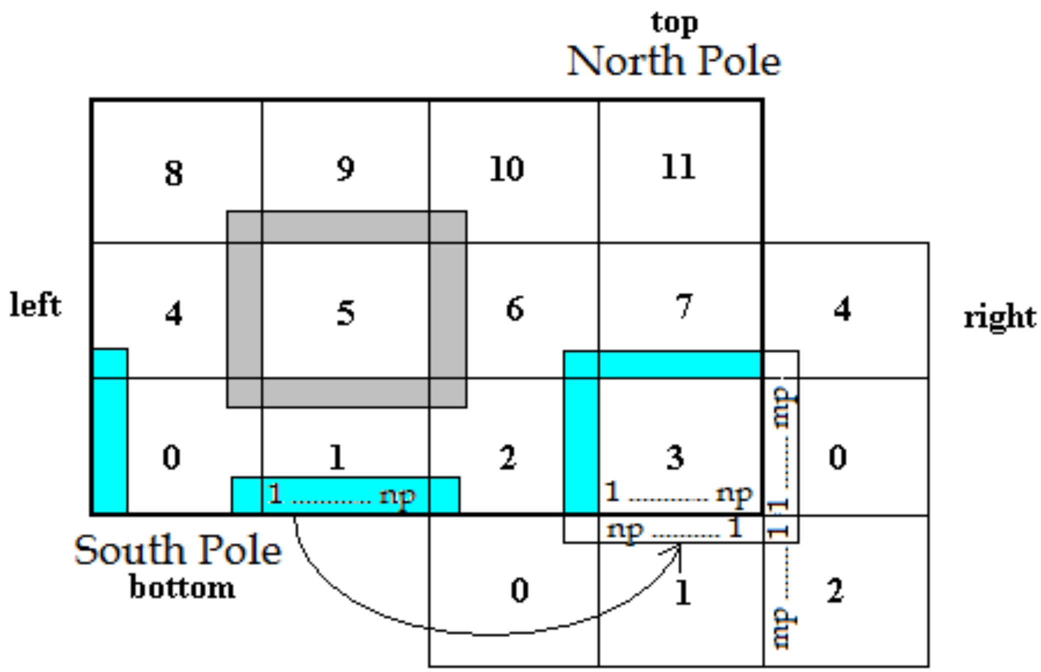
- **nprocs=12**
- **nprocx = 4, nprocy = 3**
- **if np=11, mp=11**

  then full domain size is

  N x M = 44 x 33 grid points

- **Parallel subdomians ALWAYS assume that grid has cyclic BC in both X and Y !!!**
- **In Cartesian mode, the grid indexes are in range: 1…N, only N-1 are independent !!!**
- **F(N)=F(1) –> periodicity enforcement**
- **N may be even or odd number but it must be divided by number of processors in X**
- **The same apply in Y direction.**

EULAG Spherical grid configuration with data exchange across the poles



← In the setup on the left

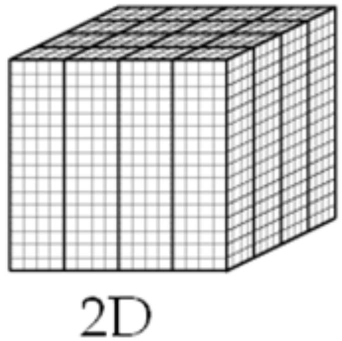
- nprocs=12
- nprocx = 4, nprocy = 3
- if np=16, mp=10

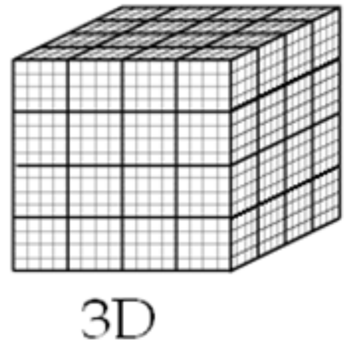
  then full domain size is

  N x M = 64 x 30 grid points

- Parallel subdomains in longitudinal direction ALWAYS assume grid in cyclic BC !!!
- At the poles processors must exchange data with appropriate across the pole processor.
- In Spherical mode, there is N independent grid cells F(N)≠ F(1) … required by load balancing and simplified exchange over the poles -> no periodicity enforcement
- At the South (and North) pole grid cells are placed at Δy/2 distance from the pole.

Development of EULAG 3D *domain decomposition*





Changes to model setup and algorithm design

- ***New processor geometry setup, option for MPI cartesian topology***
- ***Halo updates in vertical direction***
- ***Optimized halo updates at the cube corners (wider updates instead of many small messages)***
- ***Changes in vertical grid structure for all model variables***
- ***New loops structure due to differentiation and BC in vertical***

EULAG SCALABILITY TESTS

Weak Scaling

- Problem size/proc fixed
- Easier to see Good Performance
- Beloved of Benchmarkers, Vendors, Software Developers –Linpack, Stream, SPPM

Strong Scaling

- Total problem size fixed.
- Problem size/proc drops with P
- Beloved of Scientists who use computers to solve problems. Protein Folding, Weather Modeling, QCD, Seismic processing, CFD

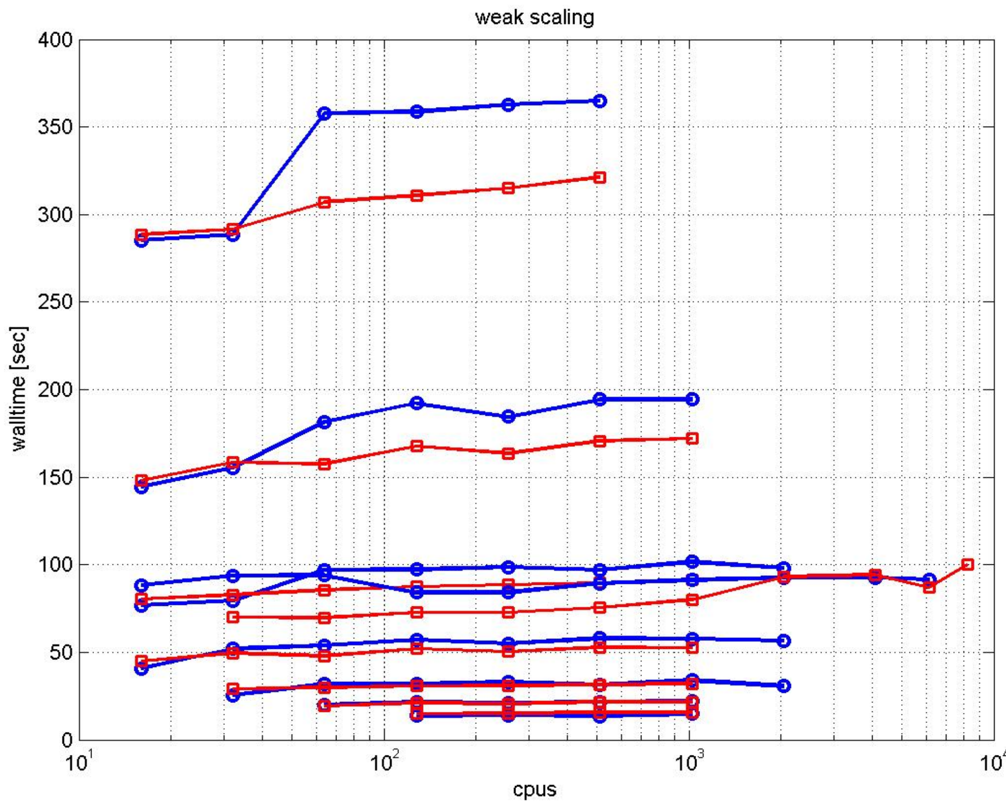
Scalability tests

- Idealized Held-Suarez climate benchmark
- Representative for global weather/climate studies in the thin atmospheric shell – ideal candidate for 2D MPI domain decomposition

Photo: Suomi NPP

EULAG SCALABILITY TESTS

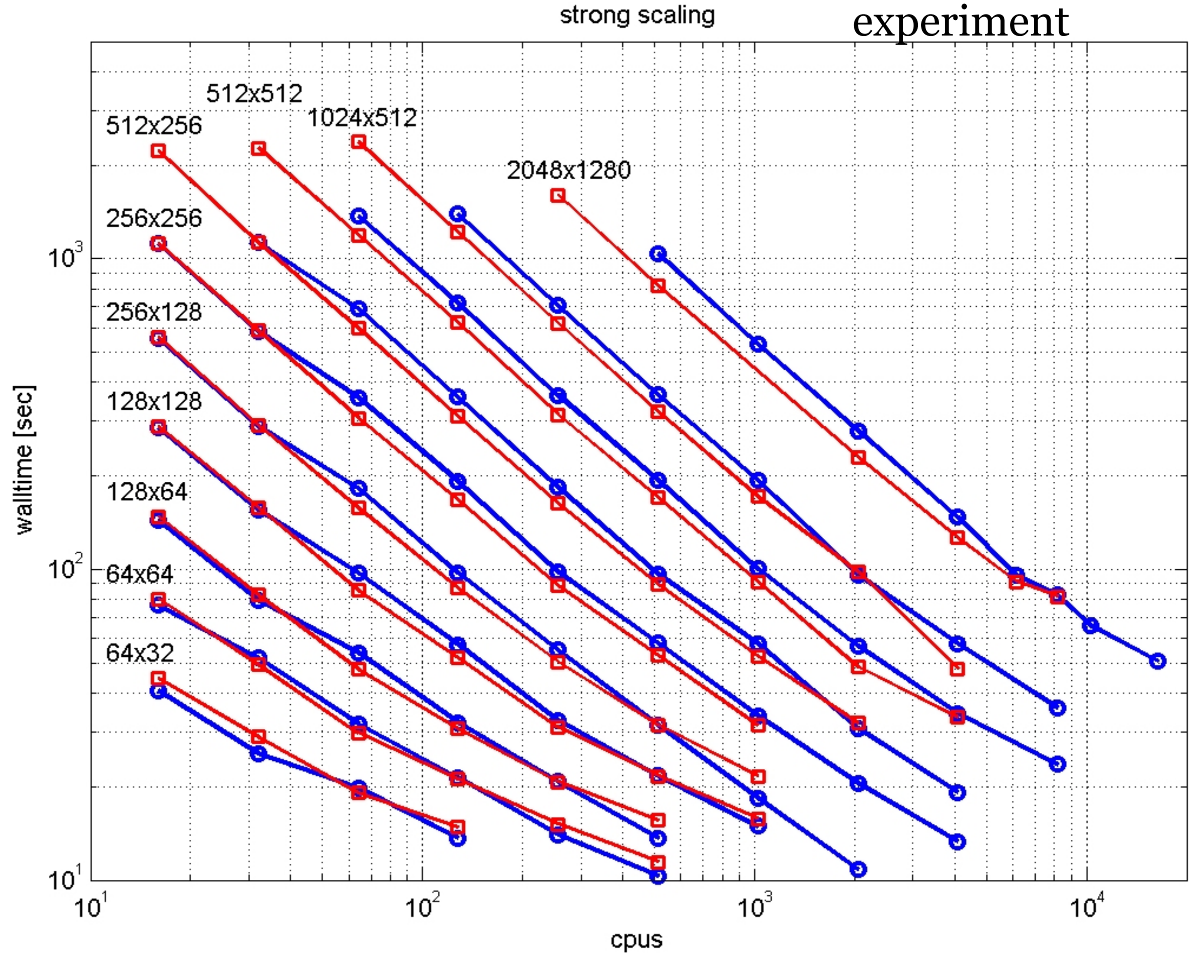
Benchmark results from the Eulag-HS experiments
NCAR/CU BG/L system 2048 processors (frost),
IBM/Watson Yorktown heights BG/L … up to 40 000 PE, only 16000 available during



Red lines – coprocessor mode, blue lines virtual mode

EULAG SCALABILITY

Benchmark results from the Eulag-HS experiments
NCAR/CU BG/L system 8384 processors (frost),
IBM/Watson Yorktown heights BG/W … up to 40 000 PE, only 16000 available during experiment



All curves except 2048x1280 are performed on BG/L system.

Numbers denote horizontal domain grid size, vertical grid is fixed l=41

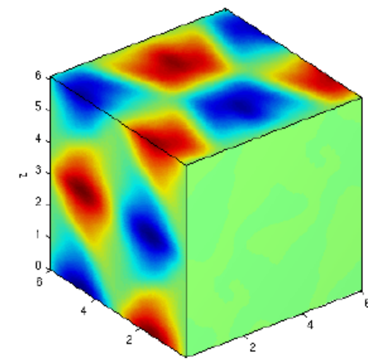
The Elliptic solver is limited to 3 iterations (iord=3)

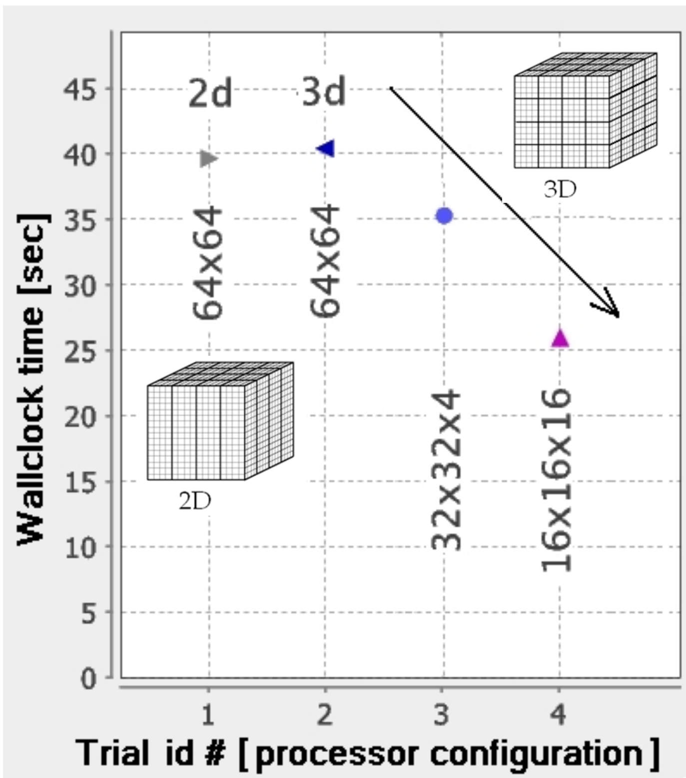
Red lines – coprocessor mode, blue lines virtual mode

Excellent scalability up to number of processors *NPE=sqrt* (*N*M*)

EULAG 3D *domain decomposition – turbulence in a box*

Taylor Green Vortex (TGV) turbulent decay.
Triple periodic cubic grid box - a perfect candidate for 3D decomposition





Only pressure solver and model initializations, no preconditioner

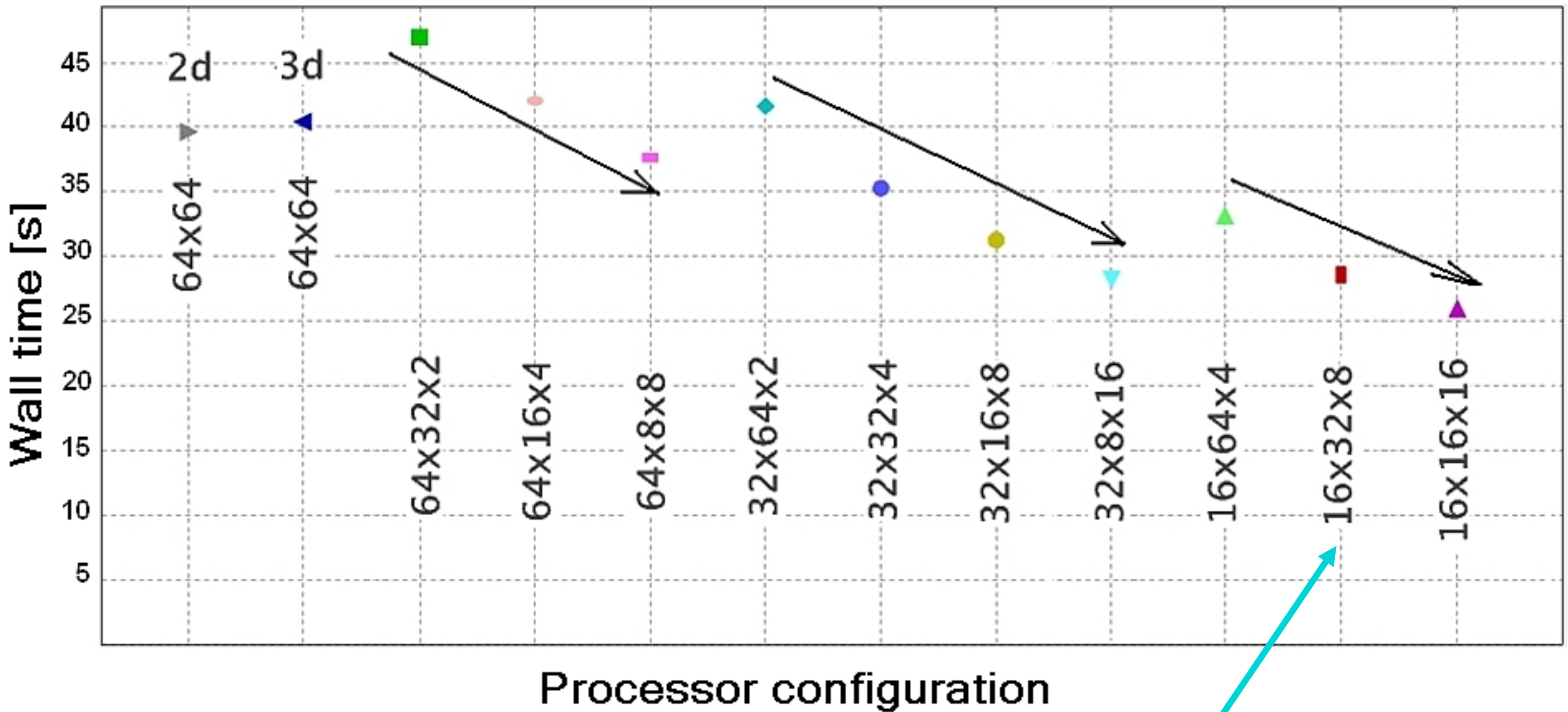
Fixed number of iterations

100 calls to solver

512^3 grid points

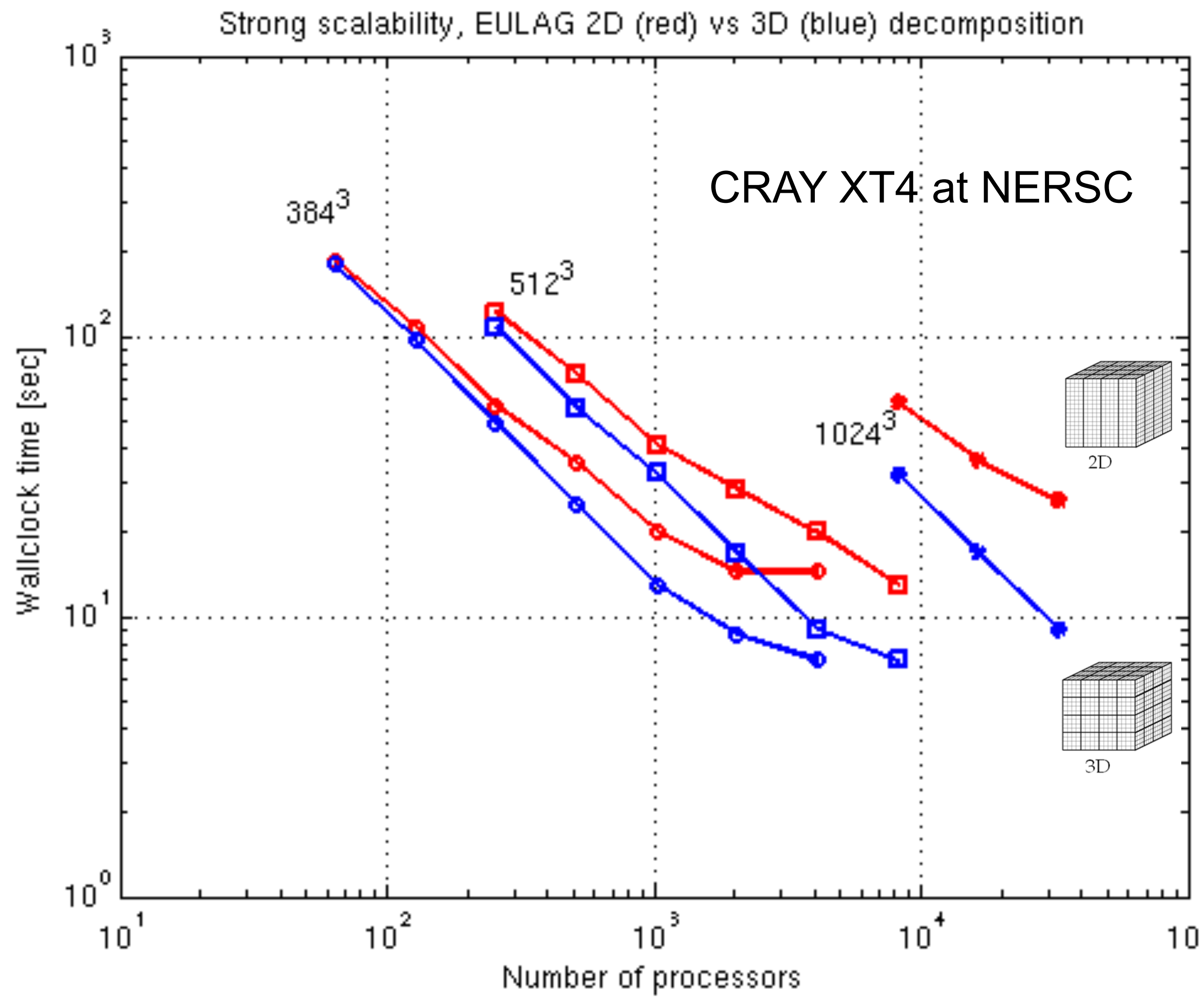
IBM BG/L system with 4096 PEs

512^3 gridpoints decaying turbulence - dependence of performance on the processor configuration on Bluegene/L



Longest innermost loop

Decaying turbulence scalability on CRAYs



Performance model for minimizing halo communication bandwidth

Examine $R = r_{3d}/r_{2d}$, where:

r3d = [(np3d + 2h) × (mp3d + 2h) × (lp3d + 2h) − V3d]/V3d
r2d = [(np2d + 2h) × (mp2d + 2h) × lp2d − V2d]/V2d .

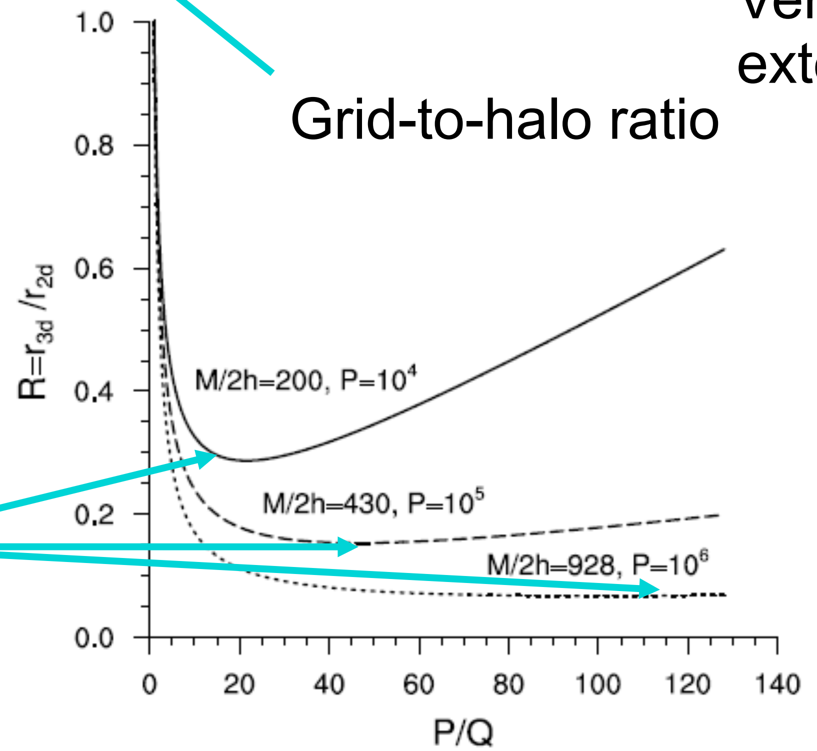
$$R\left(\frac{P}{bQ}; P, \widetilde{M}\right) = \frac{\left(1 + \frac{\sqrt{P/b}}{\widetilde{M}}\sqrt{\frac{bQ}{P}}\right)^2 \left(1 + \frac{a}{\widetilde{M}}\frac{P}{bQ}\right) - 1}{\left(1 + \frac{\sqrt{P/b}}{\widetilde{M}}\right)^2 - 1},$$

No. of cores in vertical

Total no. of cores

Grid-to-halo ratio

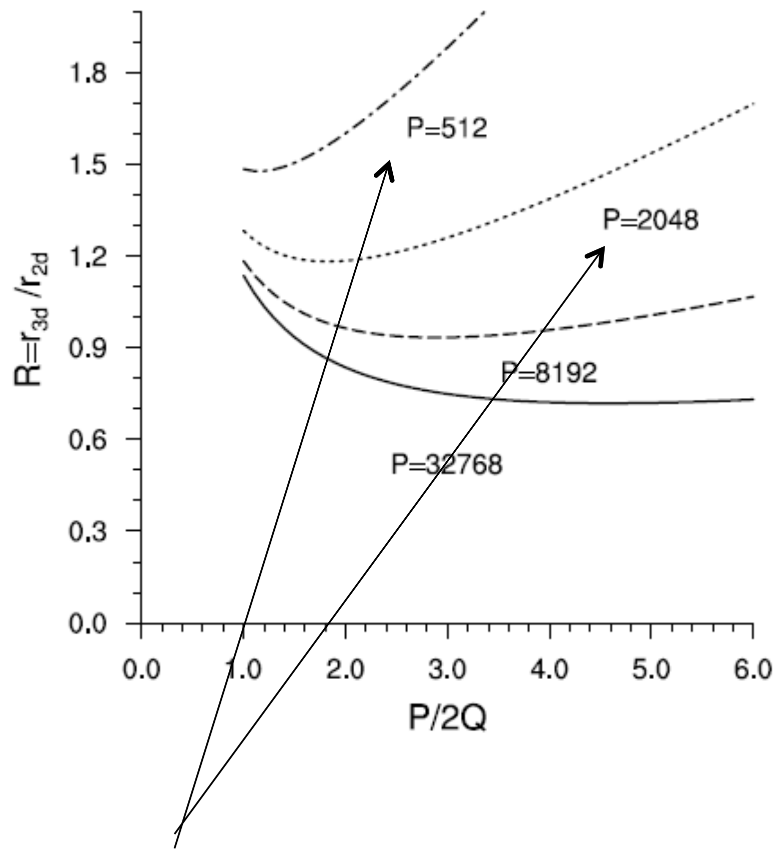
Vertical to horizontal extent ratio



Optimal no. of cores in the vertical

$$P/bQ = \sqrt[3]{P/ba^2}.$$

Performance model for 1024 × 512 × 41 grid of idealized climate simulation



Not always a performance gain from the 3D decomposition !

... but we can always use more cores to decrease time-to-solution !

Table 1: Strong scaling of idealized climate simulation on a $256 \times 128 \times 64$ grid, using 512 processors in the horizontal with increasing number of processors in the vertical.

Total processor number	512	1024	2048	4096
Processor configuration	$32 \times 16 \times 1$	$32 \times 16 \times 2$	$32 \times 16 \times 4$	$32 \times 16 \times 8$
Wallclock time [sec]	52	30	20	15

Remarks on vertical algorithms

- Strong domain anisotropy (thin shell) results in very bad conditioning hurting the performance of iterative solvers
- Effective preconditioning is a key to the iterative solver convergence …
- … but it demands direct inversion of the tridiagonal matrix in the vertical direction (same for radiation)

- Thomas algorithm is a embarrassingly serial recurrence → special treatment necessary
- Possible solution is the recurrence doubling approach

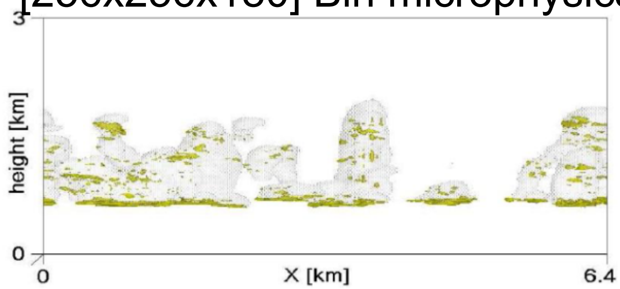
$a(n+1) = Ba(n)+C$ is rewritten as:

$a(n+1)=F(B,C)a(1)$ + parallel part

\+ pipelining or single GATHER/SCATTER in the vertical (depending on the machine and number of cores)

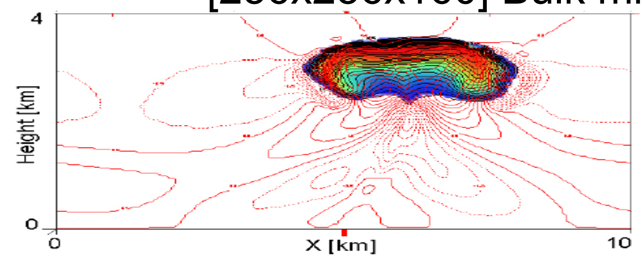
2D/3D *decomposition scalability (full model physics, Thomas preconditioner)*

[256x256x180] Bin microphysics

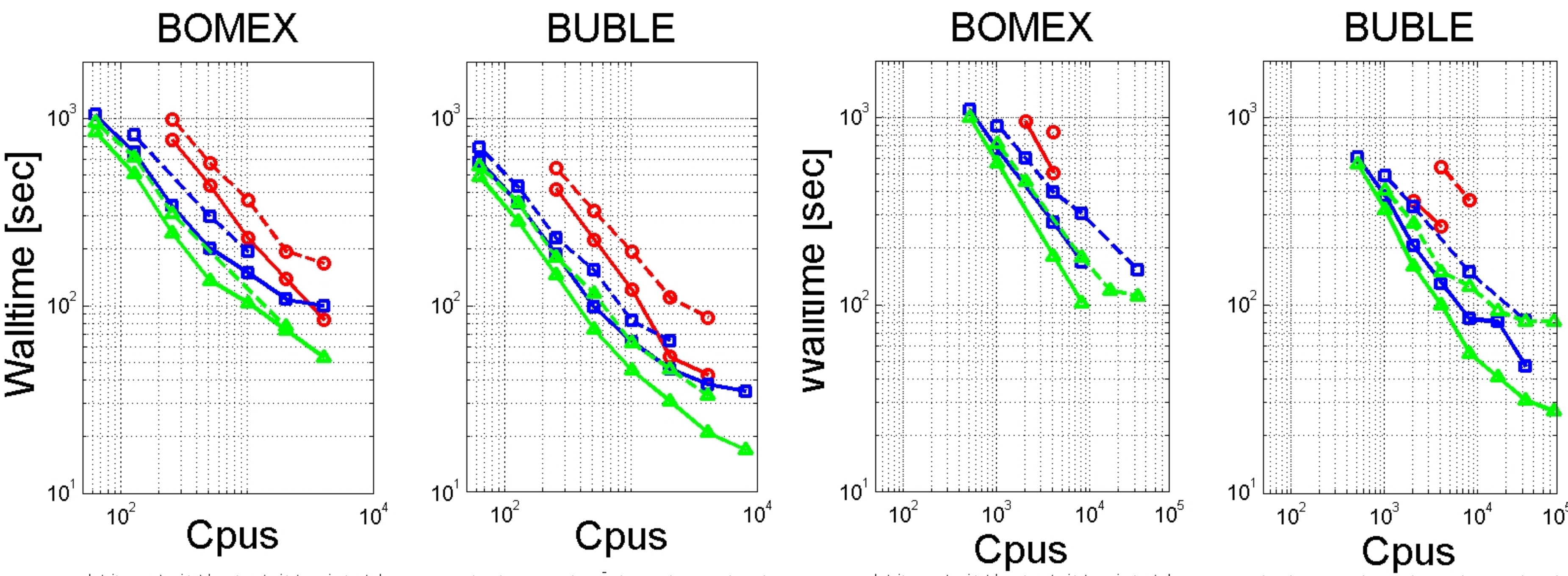


BOMEX

[256x256x160] Bulk microphysics



RISING BUBBLE



Strong scalability results with full model physics. The red, blue, and green lines shows results from IBM BG//L, CRAY XT4 and Cray XE6 respectively, the dashed lines represent 2D decomposition, the continuous lines 3D decomposition. Left and right panels show default and double resolution problems, respectively.

Additional benefits of 3D MPI parallelization

- Most part of the EULAG code is now symmetrical in x,y,z

- A number of long lasting bugs revealed and fixed

- For large part of experiments, time-to-solution significantly decreased for fixed number of cores

- Size of the innermost loop is more flexible – beneficial for vectorization

- Many optimizations introduced in process of coding and testing of the new code

More remarks …

With the new, 3D parallelization we can attempt to simulate much larger problems, BUT there is a memory wall ahead.

→ Need for improving memory locality and cache use efficiency

Also, we can decompose problem to use many more cores, BUT there is a communication wall ahead

→ Need for minimizing halo updates and, especially, reduce number of global MPI operations to minimum

Conclusions

- Three dimensional MPI parallel formulation, for symmetric (e.g. cubical turbulence) problems, can decrease time to solution for given number of cores used by factor of ~ 0.5.

- For thin-shell applications (weather and climate), it allows for decreasing time-to-solution by admitting much larger number of computing cores.