Large-eddy simulation of the atmospheric boundary layer: mastering model complexity and HPC performance

Mortikov E.V., Debolskiy A.V., Gashchuk E.M. , Glazunov A.V.

Research Computing Center, Lomonosov Moscow State University

Marchuk Institute of Numerical Mathematics, Russian Academy of Science

A.M. Obukhov Institute of Atmospheric Physics, Russian Academy of Science

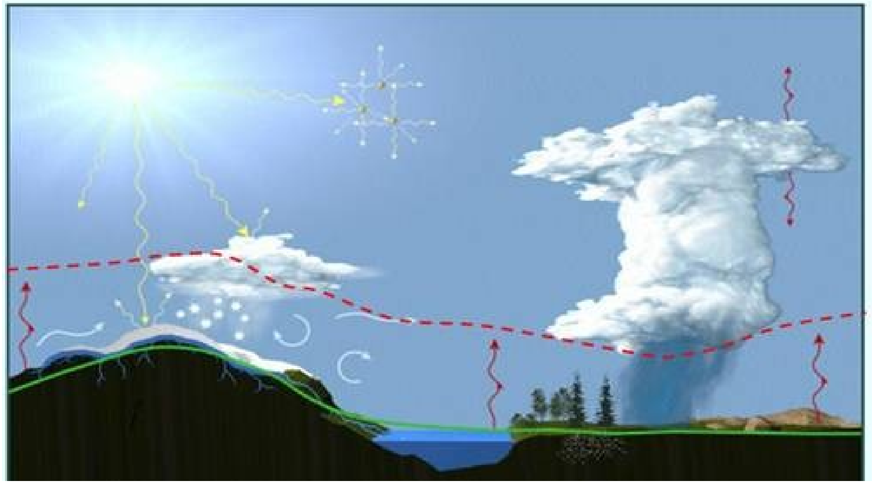
BRICS special session on the Digital Earth project

Russian Supercomputing Days 2024, Moscow, Russia

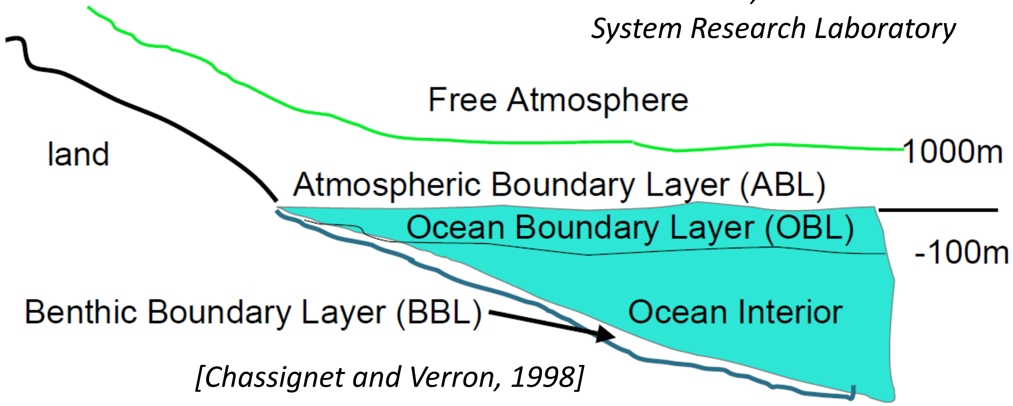
Atmospheric boundary layer

- **Atmospheric boundary layer**, $H_{ABL} \sim 10^2 - 10^3$ m
- **Diurnal transition between CBL & SBL**
- **Turbulence, stratification, solar radiation, clouds, aerosol transport, complex topography (e.g. urban environment), land surface interactions etc.**
- **Turbulence with very high Reynolds numbers**
  - Reynolds number up to $\mathbf{10^9}$
- **Numerical simulation of ABL**
  - Studies of ABL processes
  - Pollution and urban environment modelling
  - Development and improvement of parameterization in NWP and climate models

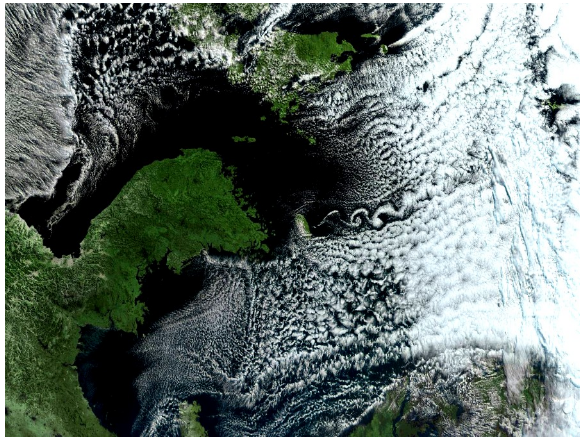
INMCM, Institute of Numerical Mathematics climate model
SL-AV, Vorticity-divergence semi-Lagrangian global atmospheric model



***ABL Scheme**, NOAA Earth System Research Laboratory*

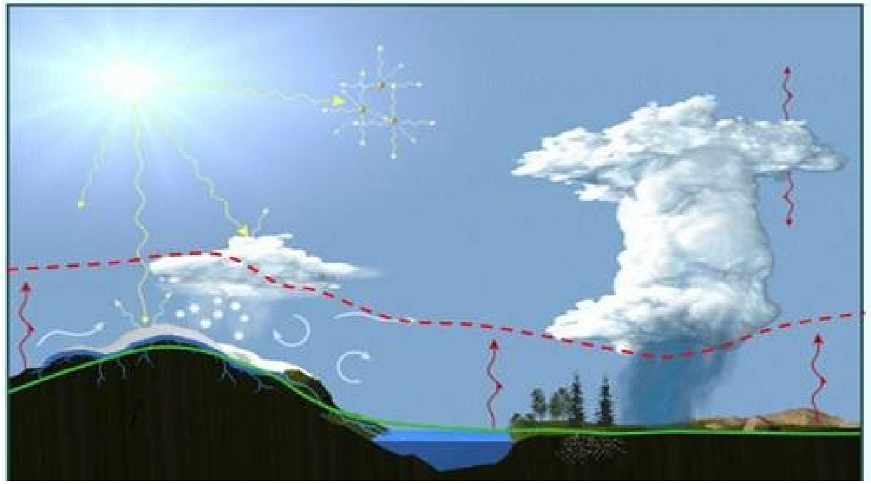


[Chassignet and Verron, 1998]



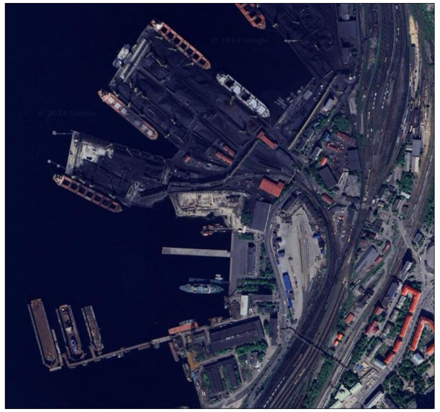
Unified DNS/LES/RANS model

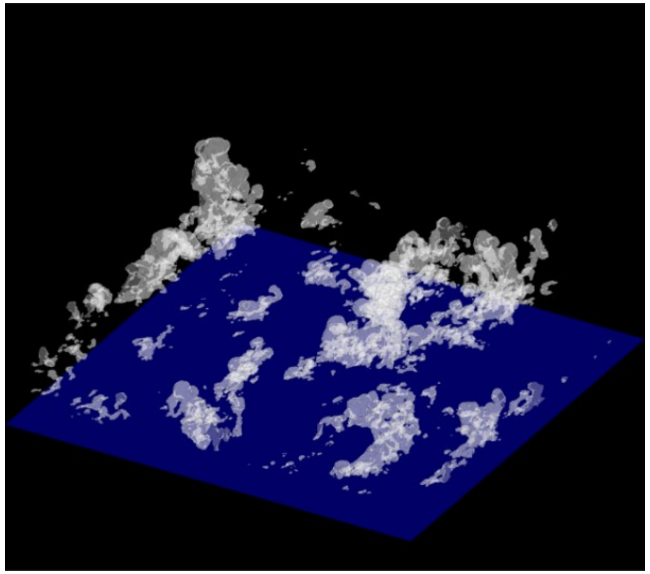
- **Code developed at RCC MSU & INM RAS (~1 million LOC)**
- **Boundary layer large-eddy simulation (LES) model**
    - **Mixed dynamic subgrid closures**
    - **Double-moment cloud microphysics**
    - **Radiation LW/SW module**
    - **Land surface coupling**
    - **Atmospheric chemistry & aerosol transport**



***ABL Scheme**, NOAA Earth System Research Laboratory*

Coal dust emissions, Murmansk port



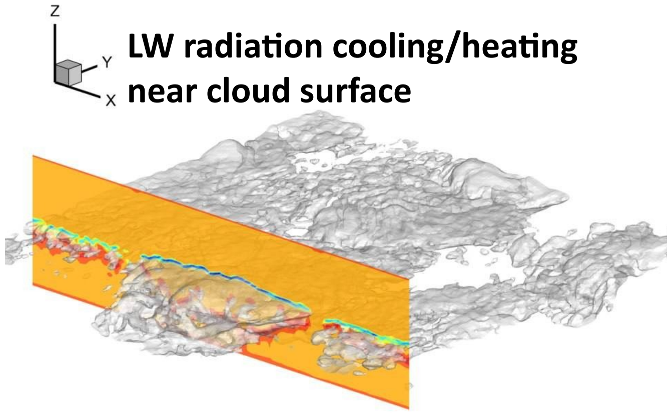


CBL entrainment

Shallow cumulus convection

MSU/INM LES model

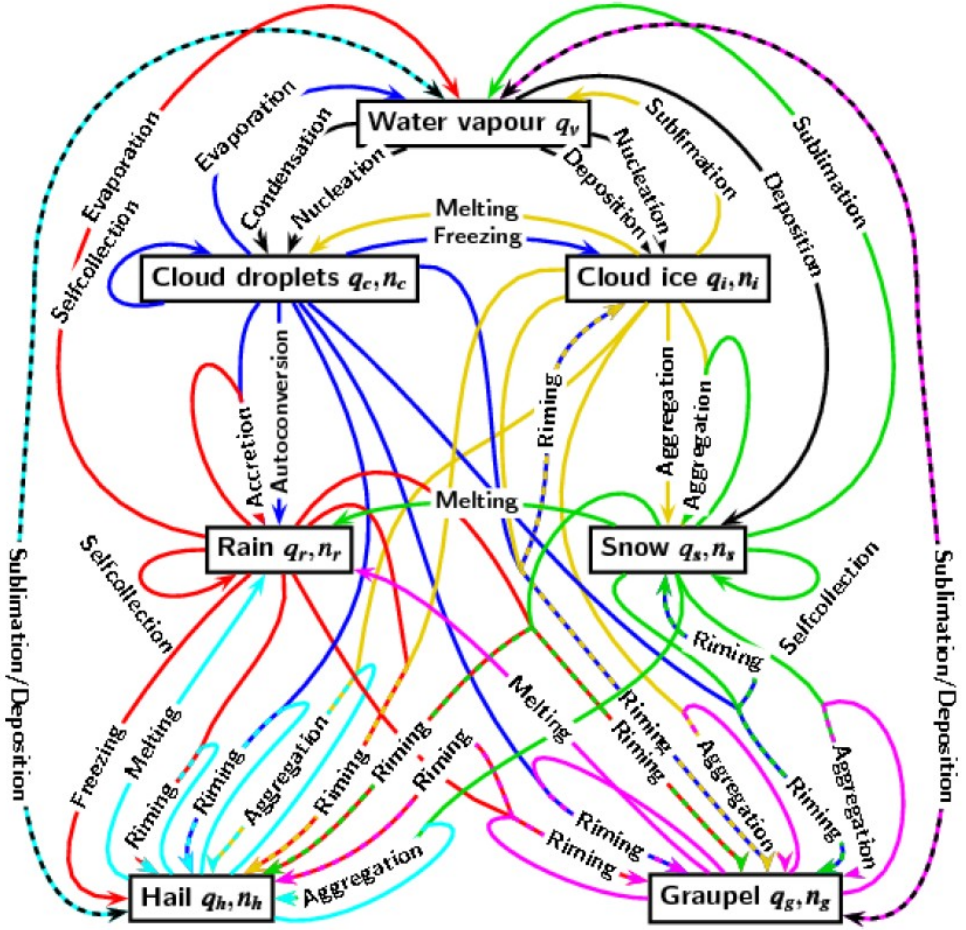
- **Hierarchy of microphysics schemes of different computational complexity**
  - Single-moment microphysics based on *Lin et al., 1983* model
  - Double-moment microphysics based on *Seifert & Beheng, 2006*
  - Includes cloud water, rain, ice, snow and graupel
- **Dynamic approach applied to subgrid momentum flux, sensible and latent heat fluxes**
- **RRTM radiation transport coupled model**



Isosurfaces of ice mixing ratio & color map of LW heating rate

COMBLE LES/SCM intercomparison:
Cold-Air Outbreaks in the Marine Boundary Layer Experiment

MSU/INM LES
DALES, MICROHH, UCLA-LES
PALM, DHARMA, MIMICA
SAM, WRF, ICON-LEM



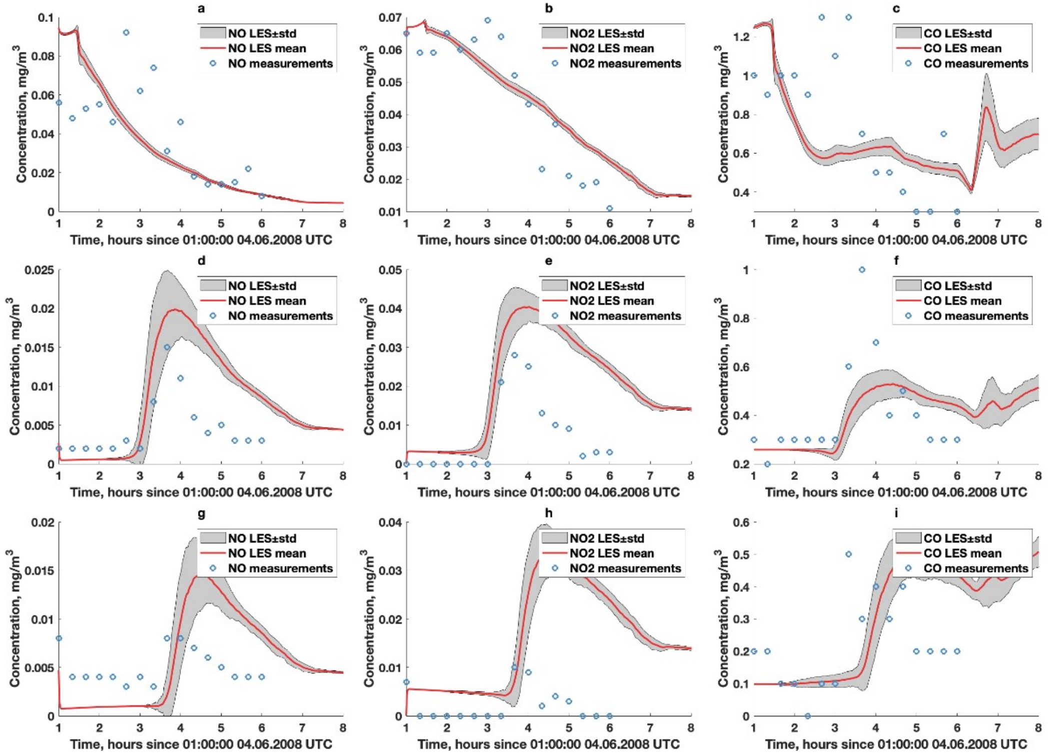
Seifert & Beheng, 2006 microphysics.
Current version of MSU-INM LES model doesn't include hail

Double-moment microphysics extremely computationally demanding – **4X LES dynamics**

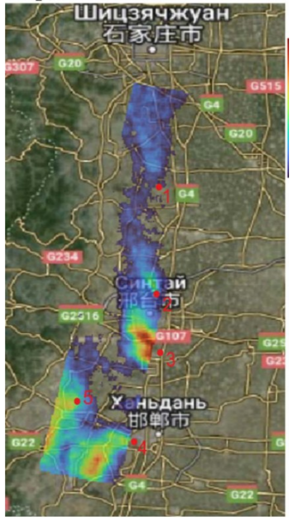
MSU/INM LES model

- Atmospheric chemistry model
  - **May include O(10-10[2]) species with O(10[2]-10[3]) reactions**
- **Reduced set of inorganic reactions intended for simulation of diurnal dynamics in urban environments coupled with LES**
  - Simplified **NO-NO_2-O_3** ‘fast’ cycle
  - RACM subset: **NO, NO_2, O_3, $O(^3P)$, $O(^1D)$, NO_3 ,N_2O_5, OH, HNO_2 , CO + ~40 chemical reactions**
  - Implicit in time numerical methods

LES model vs. measurements at Ostankino tower, Moscow



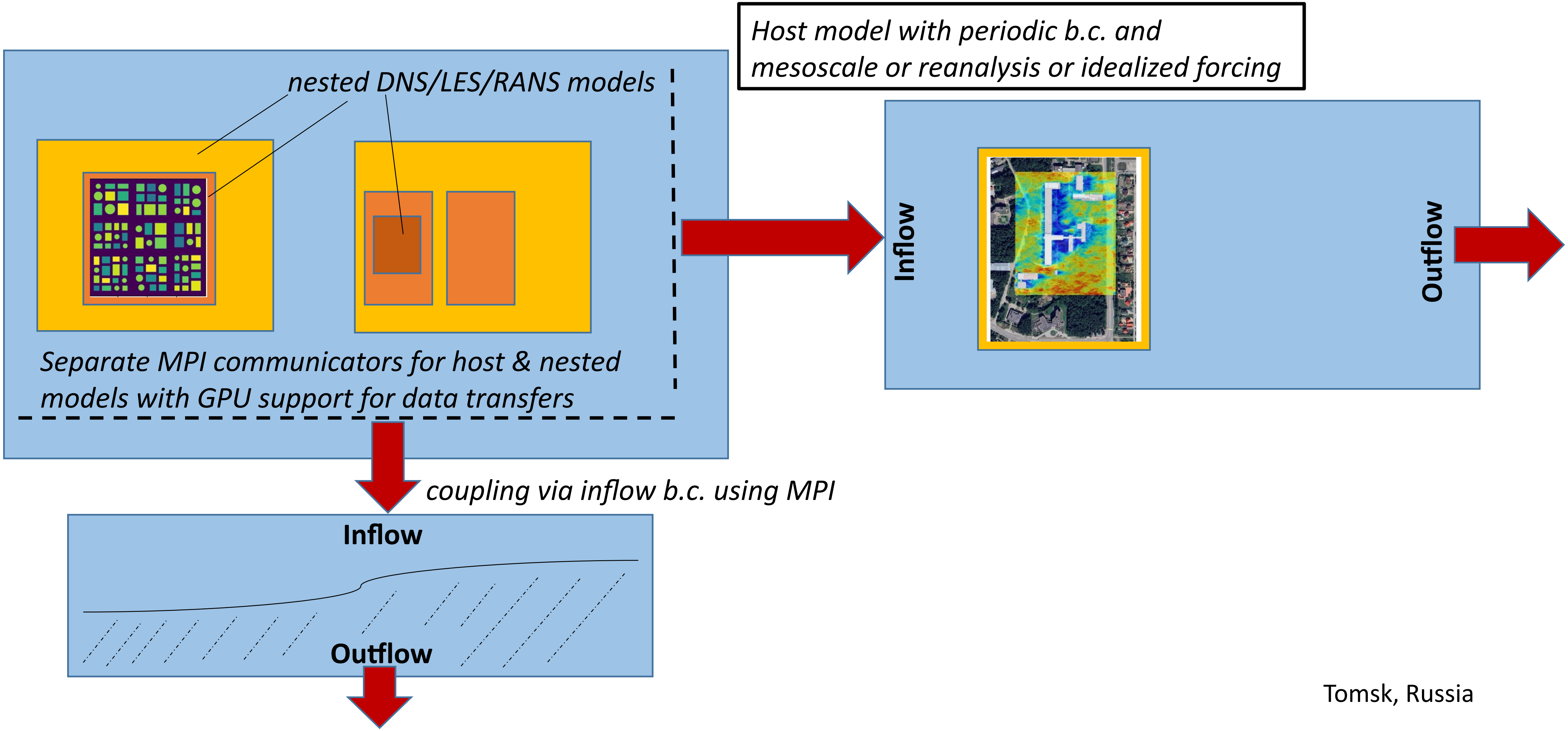
NO plume dynamics in ABL from ‘localized’ source – comparison with satellite images (China)



NO2 distribution with street level emissions

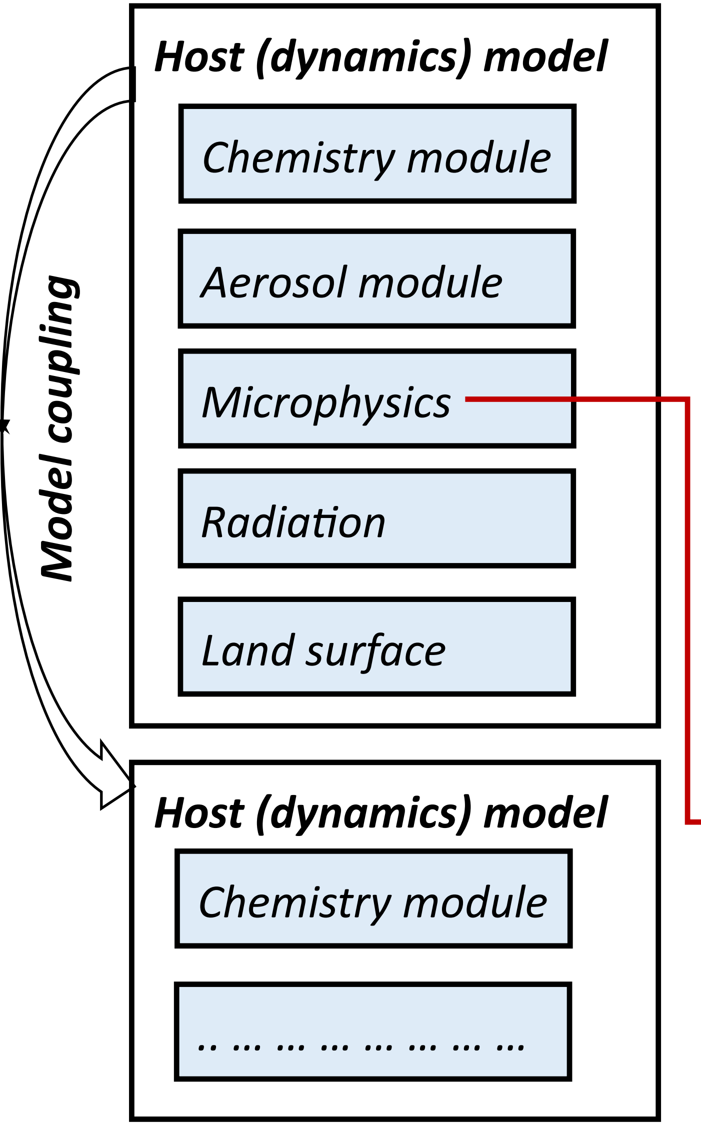
MSU/INM LES model

- **Urban environment modelling – high-resolution nested model approach**



MSU/INM LES model

- **Coupling ABL processes models**

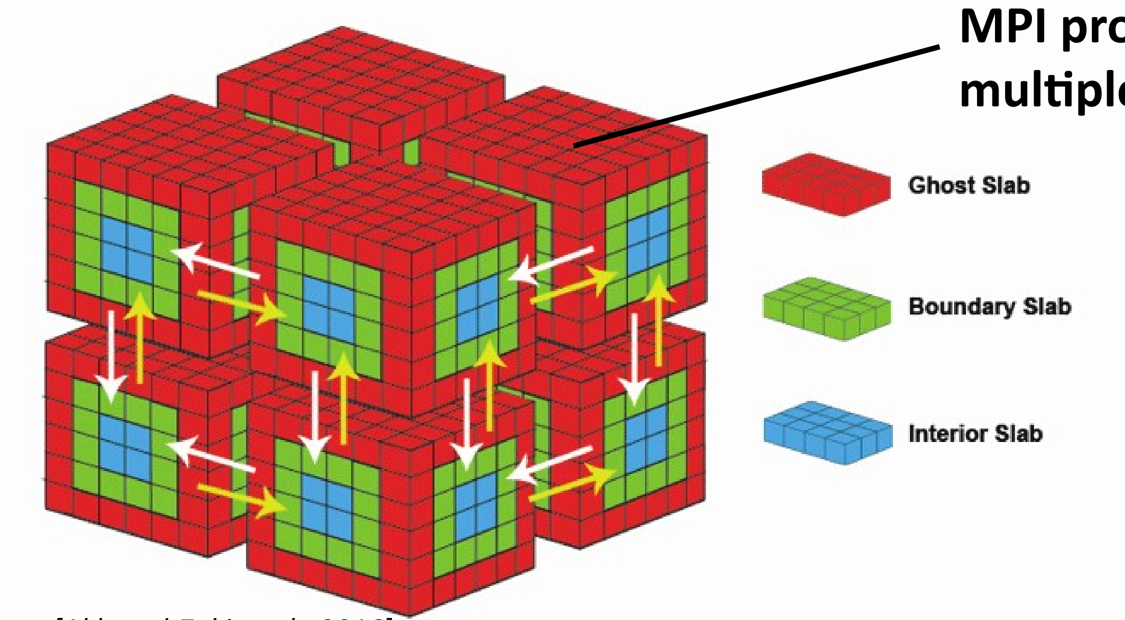


- **Host model controls the execution of nested models and describes the necessary interfaces between modules**
- **Each module has the same abstract structure as the host model:** initialization, time advancement, post-processing, output etc.
- **The modules support different concrete implementations:** e.g. built-in implementations or coupling with library or different code
- **Separates development of each component with little-to-none influence on the code of the host model or other components**

- Model is represented by **high-level code – independent of hardware implementation or optimizations**
- Only **low-level code specializes implementation:** CPU/GPU/ARM etc.

Parallel implementation

- **C/C++ code** with optional Fortran coupling (RRTM)
- **MPI domain decomposition**



[Akhmed-Zaki et al., 2016]

Lomonosov-2 supercomputer, ©MSU, T-platforms

Intel Xeon CPU
Nvidia GPU

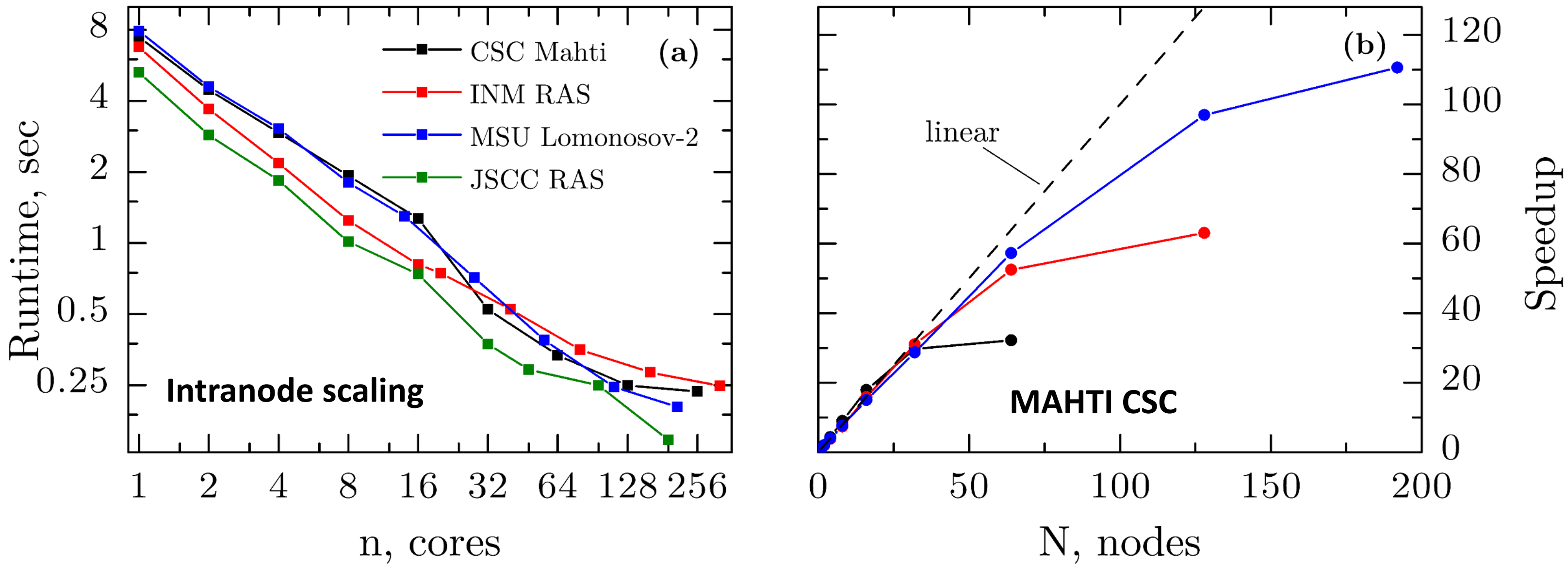
MVS-10P, -10Q supercomputers, ©RAS, RSC Group

Intel Xeon CPU
Intel Xeon Phi

- **Using OpenMP on multicore processors**
  - Overlap MPI communications with computations
  - Cache-aware algorithms/thread synchronization become more important
- **Ensuring use of CPU vector instruction sets (SSE, AVX etc.)**

Parallel implementation

- **MPI-OpenMP CPU scaling**



Scaling up to 25000 cores on CSC Mahti supercomputer (AMD EPYC)

- **Code ported on Intel Xeon Phi architecture**
- **Running on ARM-based CPUs (Kunpeng 920 processors)**

Argonne Theta Supercomputer (4096 Intel Xeon Phi cards)

Table 1. LR case run-time, in seconds per 1000 time steps

	AMD Rome 7H12	Intel Xeon Gold 6140	Kunpeng 920
single core, x2 and (x4)	39.94 (54.58)	48.53 (59.84)	123.70 (166.28)
max cores, x2 and (x4)	2.16 (2.89)	5.94 (7.94)	2.12 (2.90)

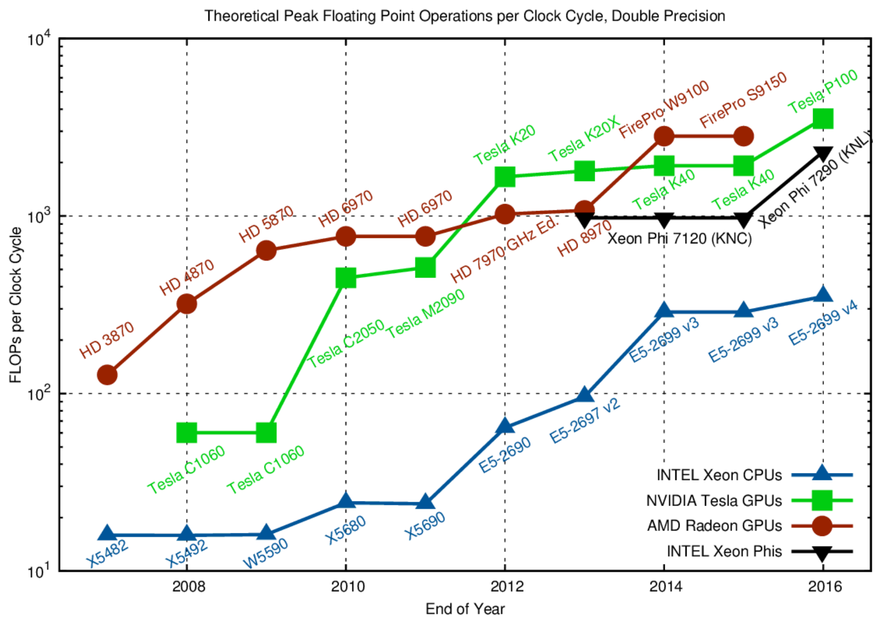
Table 2. HR case run-time, in seconds per 1000 time steps

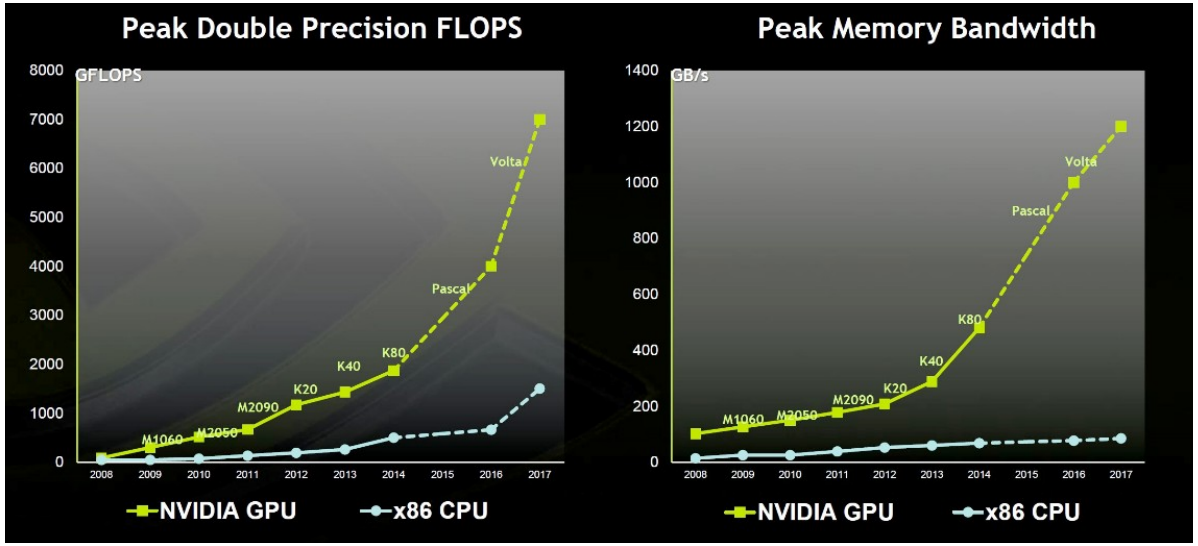
	AMD Rome 7H12	Intel Xeon Gold 6140	Kunpeng 920
single core, x2 and (x4)	285.23 (372.19)	391.11 (458.09)	1018.81 (1511.12)
max cores, x2 and (x4)	10.23 (13.49)	26.83 (32.81)	18.27 (25.02)

[Mortikov and Debolskiy, 2021]

Why GPUs?

- **Graphics Processing Units** (GPUs) – energy efficiency, cheap ($/FLOPs) & high performance ***for a number of problems***
- Increase in performance of supercomputers in the last 10 years **in large part due to the advent of coprocessors**: **GPUs** (*Lomonosov-2, Summit*) or **Intel Xeon Phi** (*Tianhe-2*)
- Speed-up of <u>hydrodynamic models</u> when ported to GPUs:
  - **x20-x40** compared with CPU core
  - **x2-x4** compared with CPU node
- Speed-up of <u>molecular dynamics</u> when ported to GPUs:
  - **x500-x1000** compared with CPU core
- Adapt models & algorithms to new *Frontiers*: ***exascale*** and ***post-exascale*** systems

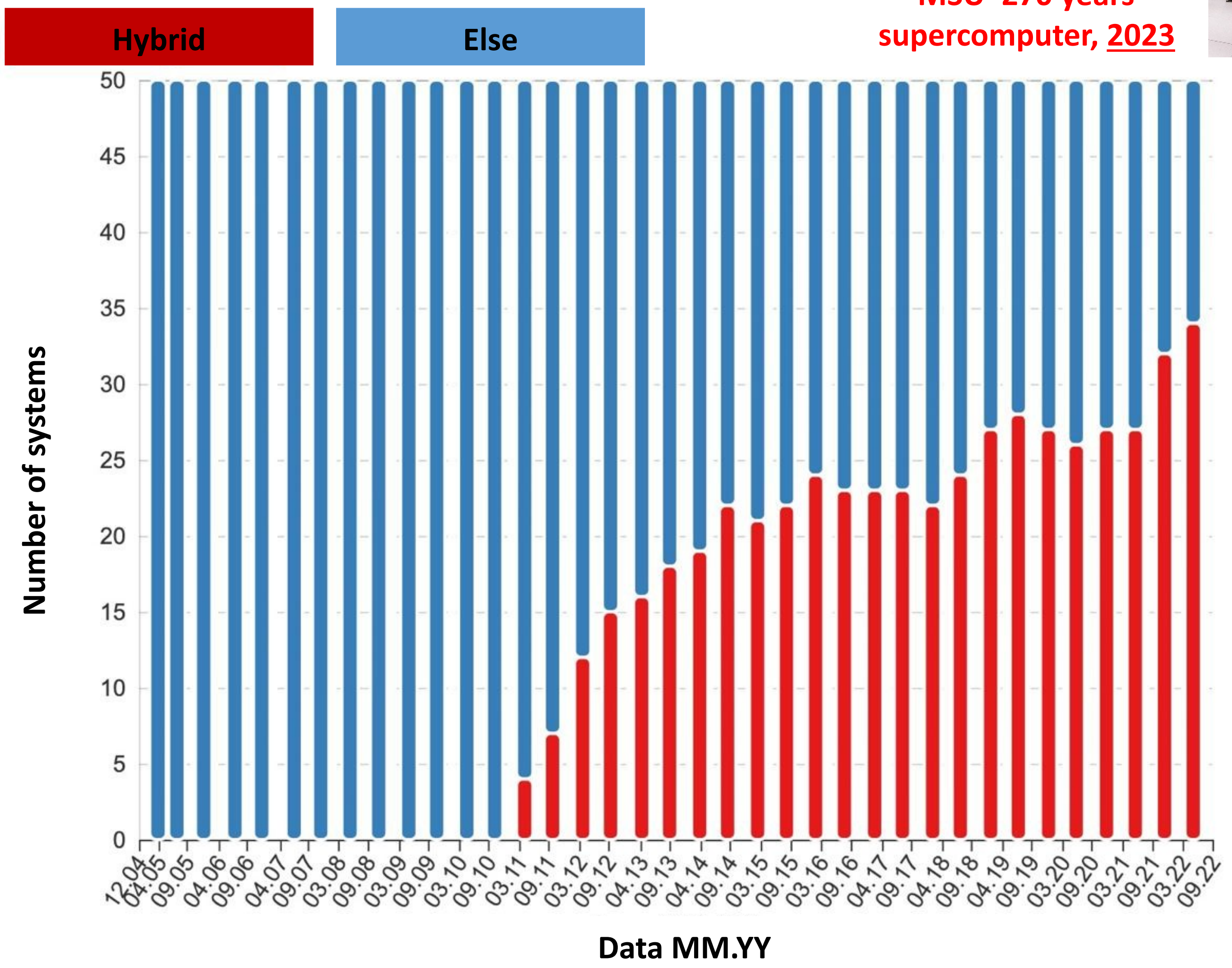




Why GPUs?

1	«Червоненкис» Яндекс, Москва	199 398 1592	199: CPU: 2x AMD EPYC 7702 , 1024 GB RAM Acc: 8x NVIDIA A100 HDR InfiniBand / нд / 100 Gigabit Ethernet
2	«Галушкин» Яндекс, Москва	136 272 1088	136: CPU: 2x AMD EPYC 7702 , 1024 GB RAM Acc: 8x NVIDIA A100 HDR InfiniBand / нд / 100 Gigabit Ethernet
3	«Ляпунов» Яндекс, Москва	137 274 1096	137: CPU: 2x AMD Epyc 7662, 512 GB RAM Acc: 8x NVIDIA A100 HDR InfiniBand / нд / 100 Gigabit Ethernet
4	«Кристофари Нео» SberCloud (ООО «Облачные технологии») , СберБанк, Москва	99 198 792	99: CPU: 2x AMD EPYC 7742, 2048 GB RAM Acc: 8x NVIDIA A100 HDR InfiniBand / 10 Gigabit Ethernet / 200 Gigabit Ethernet
5	«Кристофари» SberCloud (ООО «Облачные технологии») , СберБанк, Москва	75 150 1200	75: NVIDIA DGX-2 CPU: 2x Intel Xeon Platinum 8168 24C 2.7GHz, 1536 GB RAM Acc: 16x NVIDIA Tesla V100 EDR Infiniband / 100 Gigabit Ethernet / 10 Gigabit Ethernet
6	«Ломоносов-2» Московский государственный университет имени М.В.Ломоносова, Москва	1696 1696 1856	1536: CPU: 1x Intel Xeon E5-2697v3, 64 GB RAM Acc: 1x NVIDIA Tesla K40M 160: CPU: 1x Intel Xeon Gold 6126, 96 GB RAM Acc: 2x NVIDIA Tesla P100 FDR Infiniband / Gigabit Ethernet / FDR Infiniband

Number of hybrid systems in TOP50 Russian supercomputers



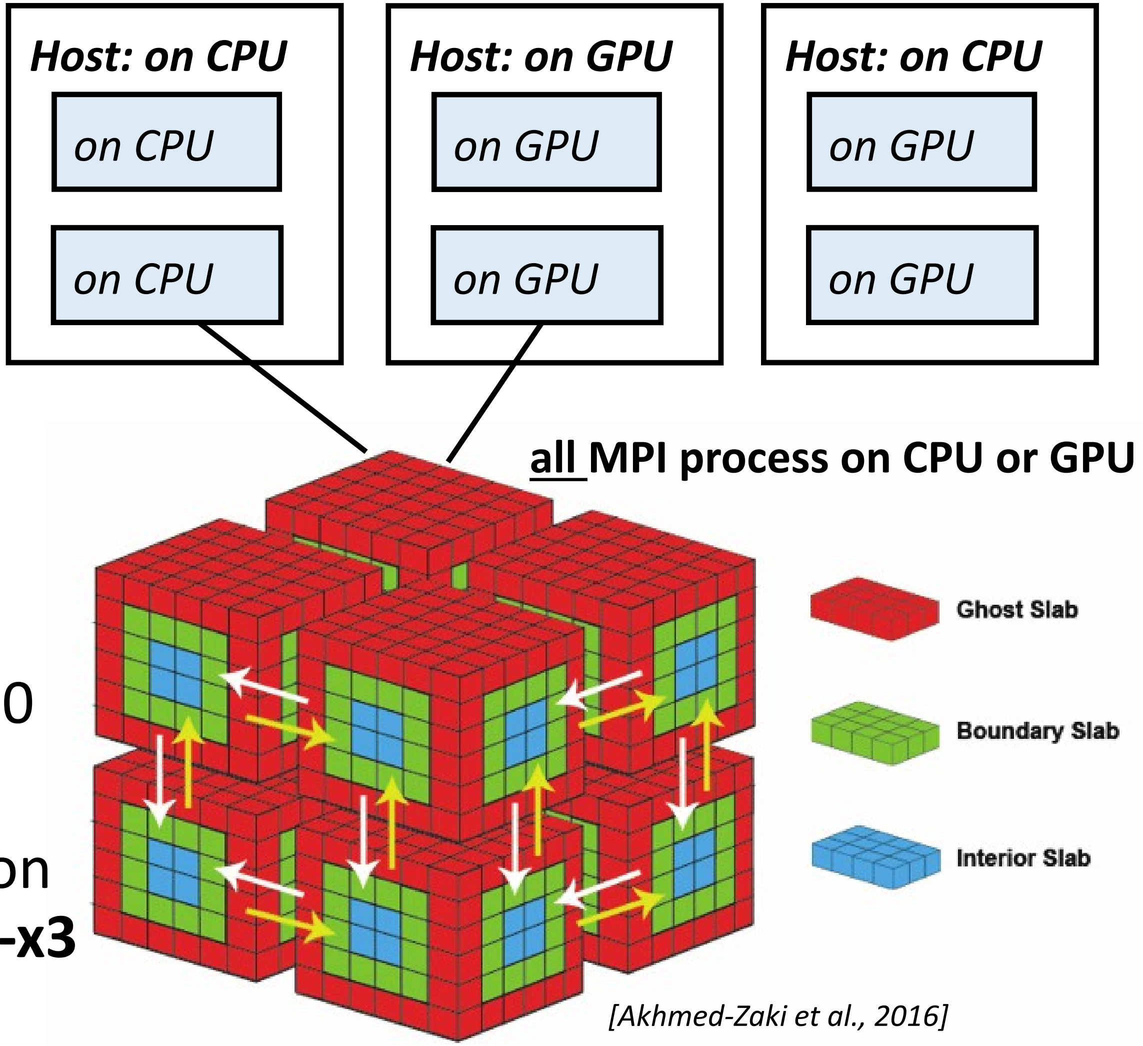
MSU '270 years' supercomputer, 2023

DNS/LES/RANS code on CPU/GPU systems

- **DNS/LES/RANS models fully ported to hybrid CPU/GPU systems**
  - Dynamics, chemistry, microphysics … & run-time flow processing support on GPUs
  - Using C/C++ & MPI/OpenMP/CUDA [only Nvidia GPUs]
- **Just compile & run** – single executable:

```
./exe -arch cpu   →   all on CPU
./exe -arch gpu   →   all on GPU
./exe -arch mix   →   mixed mode
```

- **Additional optimizations on GPU:**
  - optionally using half-precision (FP16) – **x2**
  - direct GPU-GPU memory transfers in MPI (NCL/IPC)
- **Microphysics** is highly efficient on GPUs (x100-x200 compared with Intel Xeon CPU core)
- **Atmospheric chemistry** model GPU implementation x2 more efficient than LES dynamics – **speedup x2-x3 compared with AMD EPYC 128 core node**



DNS/LES/RANS with offload on GPU

- **Offload modules or part of computations on GPU**
  - Running the model dynamics (+ other components) on CPU **except** the offloaded modules on GPU
  - **Atmospheric chemistry, aerosol transport and microphysics** are good candidates for offloading – more efficient (in terms of both performance and scaling) on GPUs compared with dynamics module

Lagrangian particles transport:

$$\frac{d\mathbf{x}_i(t)}{dt} = \mathbf{v}_i, m_i \frac{d\mathbf{v}_i(t)}{dt} = \mathbf{f}_B + \mathbf{f}_D + \ldots$$

Tracers transport:

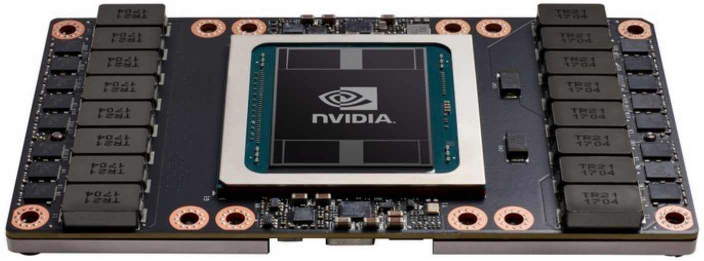
$$\frac{\partial C_k}{\partial t} + \frac{\partial u_j C_k}{\partial x_j} = \frac{1}{\mathrm{ScRe}} \frac{\partial^2 C_k}{\partial x_j \partial x_j} + F_k$$

Dynamics & data processing:

$$\frac{\partial u_i}{\partial t} = -\frac{\partial u_i u_j}{\partial x_j} - \frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 u_i}{\partial x_j \partial x_j} + F_i^e$$

$$\frac{\partial u_i}{\partial x_i} = 0$$

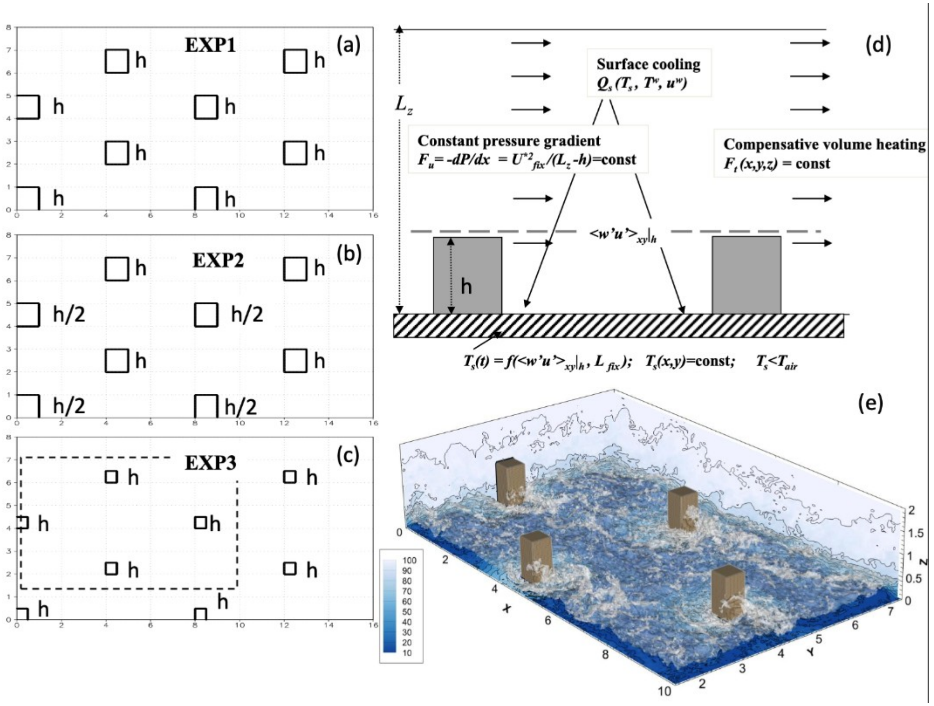
$$\frac{\partial T}{\partial t} + \frac{\partial u_j T}{\partial x_j} = \frac{1}{\mathrm{PrRe}} \frac{\partial^2 T}{\partial x_j \partial x_j}$$



CPU & GPU memory transfer each time step – may be overlapped with computations

Urban LES intercomparison

- **Bright future: microscale turbulence resolving models – *essential element of urban services and planning***
- ***Improving urban-canopy parameterizations* in meso-scale & global-scale atmospheric models**
- **How good is LES in reproducing urban boundary layer?**
- **What approaches (numerics + physics) work best?**



Moscow (Russia) central region / Institute of Atmospheric Physics

First stage: idealized urban configurations

MSU-INM LES (Russia)

Fluidity (China)

… … …

Contacts: mortikov@srcc.msu.ru

and.debol@srcc.msu.ru

Detailed information to appear on GitHub page

https://anddebol.github.io/ulescomp/

email: mortikov@srcc.msu.ru

GitLab: http://tesla.parallel.ru