



Urban Air Project

Castiel 2 – Code of the month presentation

Zoltán Horváth, László Környei, Mátyás Constans, Ákos Kovács, Csaba Tóth

10 July 2024



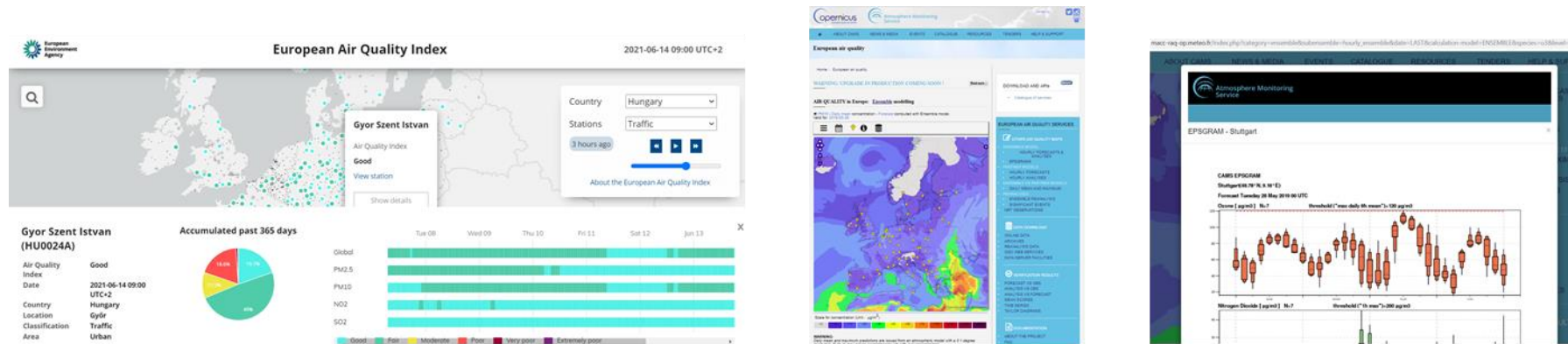
EuroHPC
Joint Undertaking

Grant number: 101093457

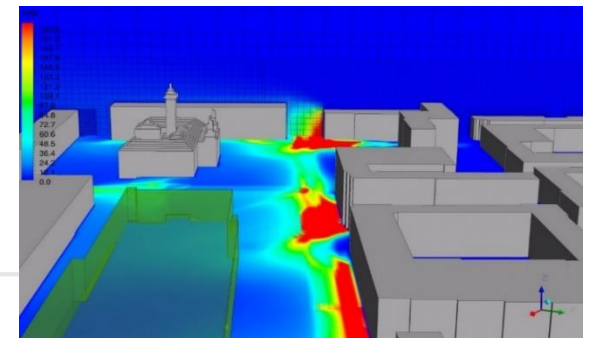
1. Motivation
 1. The addressed global and HPC challenges
 2. Business points
2. The UAP application and codes with demonstrations
 1. Workflow and portal
 1. MathSO-portal – configure, deploy, execute, monitor, post-process (Euro)HPC jobs from your web browser
 2. CFD codes
 1. OpenFOAM – scalability up to 200k CPU cores
 2. RedSim – scalable, native multi-GPU code with a flexible API
 3. CFDR – effective HPC-CFD visualizer on the web
3. Conclusions and further work

1. The global challenge: tackling urban air pollution

- EU Ambient Air Quality Directive [https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/734679/EPRS_BRI\(2022\)734679_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/734679/EPRS_BRI(2022)734679_EN.pdf)
 - Each EU country must report annual measures of air quality and punish violations.
 - Suitable proven digital solutions are eligible for reporting and policymaking
- **Current supporting services** of the EU: Monitoring, forecasts, and reanalysis for air quality of several cities on a coarse mesh (10 km x 10 km), see e.g.: <https://airindex.eea.europa.eu/Map/AQI/Viewer/>.



- **Pain point:** In many cities, local hot-spots (regions with excessive pollution) have been forming. Detection and evaluation of health effects of hot-spots are impossible with using simulations on coarse meshes. → **HPC, exascale**



1. Global challenges and business points

1. UAP.AQ: **Air quality simulation for cities at 1-2 meter resolution**

1. EU regulations have been proposed to be changed according to WHO's very strict thresholds ([link](#))
2. KPIs from EU regulations: threshold values, and monthly/annual mean concentrations and exceedance values for concentrations of gases (NO₂, O₃) and small particles (PPMs)
3. Extreme computing of the dispersion of the accidental/intentional gas release
4. Potential customers: National authorities, cities, defense

2. UAP.AC: **Computational design assessment for urban pedestrian air/wind comfort and safety**

1. Urban wind can cause discomfort for pedestrians and even critical safety situations, in particular near high buildings, see e.g. Chicago (the „Windy City”), Shinjuku District, Tokyo.
2. Potential customers: cities, designer companies

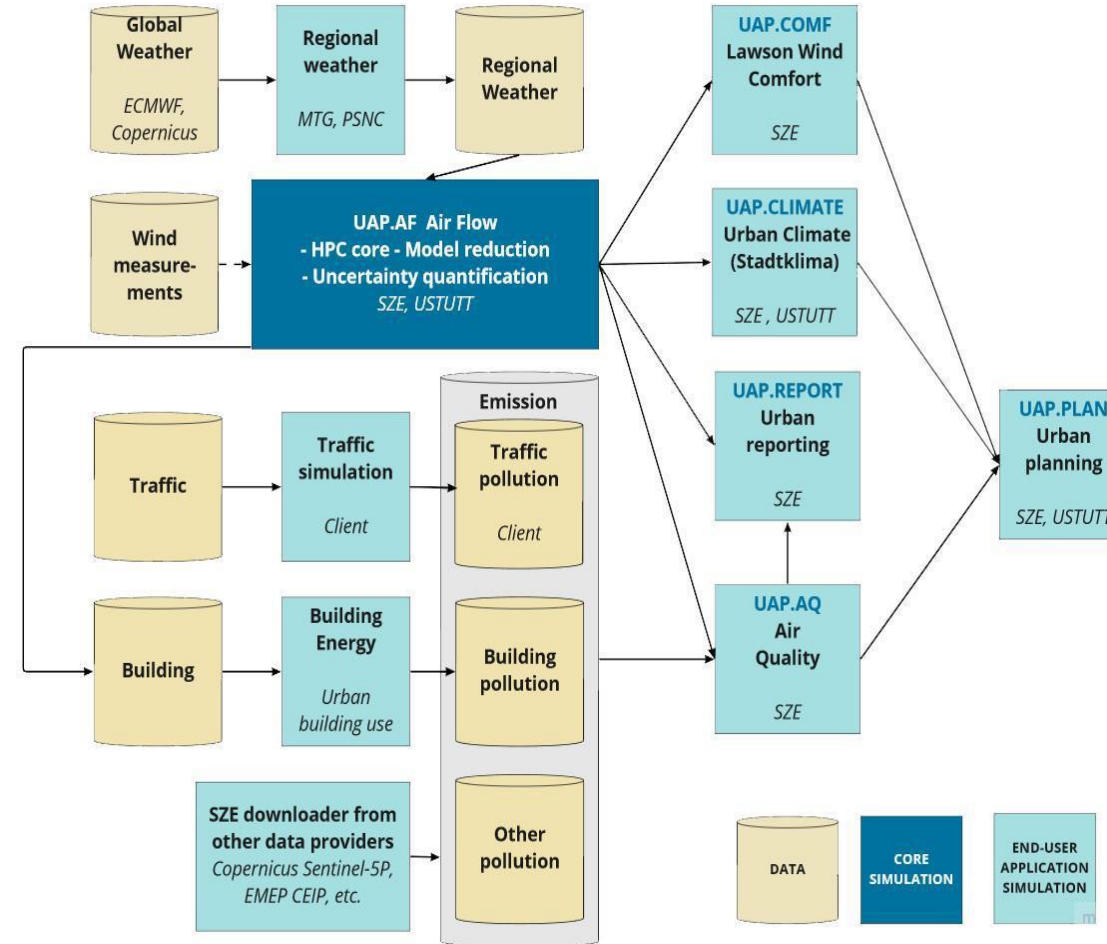
3. UAP.AP: **Urban planning**

1. Develop computational tools serving urban designers and policymakers to mitigate or cease the negative effects of urban challenges (Easy-to-use, tailored user interface, HPC in the background)
2. Potential customers: cities, architectural designer companies

4. CFD-GPU: **Native exascale CFD for compressible fluids on multi-GPUs**

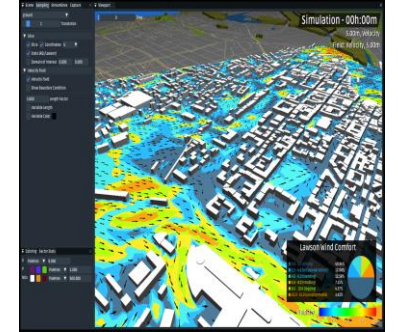
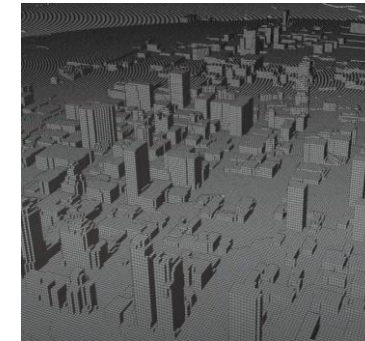
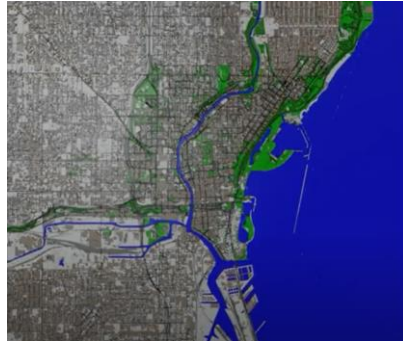
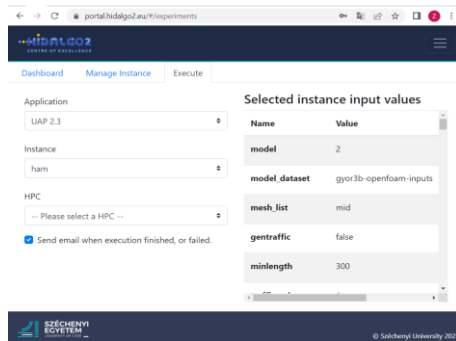
1. Develop solvers (e.g. RedSim) for MPI+multiGPU scaling on exascale problems
2. Potential customers: industry (aerospace, automotive, ...), interdisciplinary research

2.1 The UAP application and codes with demonstrations - Workflow



2.1 Overview of the UAP application from the portal

- **MathSO-portal: Automated deployment of containerized solutions to HPC platforms with**
 - Configuration (on the web application)
 - Preprocessing (geometry from OpenStreetMap, meshing),
 - Simulation (OpenFOAM, RedSim), and
 - Visualization (with CFDR).



Overview of the chapter on the HiDALGO2 Portal

- Features and Configuration
- UAP-FOAM deployment on EuroHPC machines
- Pilot blueprint, Workflow orchestration
- Integration and deployment workflow

Login page

Username / E-mail address

Password

[Forgot password](#)

[Submit](#)

Data Repository Configuration

MathSO CKAN [Update](#) [Delete](#)

Repository name

CKAN url

CKAN API key

HPC Configuration

LUMI-C [Update](#) [Delete](#) [Clear fields](#)

-- Add new HPC --
Solyom-C
MeluXina-C
LUMI-C

Host address

Port

Username

Password

Job type

Account ID (optional)

SLURM Partition (optional) [small](#) [standard](#) [Add new partition](#)

QoS (optional)

☐ SSH connection check

SSH key (only unencrypted supported)

— You have to set the password, or the SSH key

Portal - UAP-FOAM - Configure simulation

[Dashboard](#)
[Manage Instance](#)
[Execute](#)

Application

Instance

Submit

[Import from JSON](#)

New Instance name (max. char: 32) ✓

mesh

Input Dataset [i](#)

Source repository

MathSO CKAN

Dataset of the repository

antwerp-uap24-input

meshfile [i](#)

octree_antwerp1_with_vegetation_foam_9_4_3.3M.msh

traffic

Choose Emission [i](#)

-- Please select from the list --

wir

simT

-- Please select from the list --

No Emission (WindComfort)
 octree_antwerp1_with_vegetation_foam_9_4_3.3M_step1.nox.emi
 octree_antwerp1_with_vegetation_foam_9_4_3.3M_step2.nox.emi
 octree_antwerp1_with_vegetation_foam_9_4_3.3M_yearly_average.nox.emi
 Point Source (edit in advanced settings)

wind

☐ Generate Wind profile [i](#)

Generate Wind - Wind X coordinate [i](#)

5

Generate Wind - Wind Y coordinate [i](#)

0

Choose WIND file - No generate wind [i](#)

antwerp_ecmwf_2016-05-06

simTime

Simulation Start time [i](#)

06:00:00

Simulation End time [i](#)

07:00:00

HPC

OPENFOAM

writeIntervalSteady [i](#)

600

pointSampleInterval [i](#)

600

sampleInterval [i](#)

600

writeInterval [i](#)


3600

sliceInterval [i](#)

3600

maxIterationsTransient [i](#)

20


[Home](#)
[Applications](#)
[Data Catalogue](#)
[Configurations](#)
[Experiments](#)
[User Guide](#)
[leslie](#)

[Dashboard](#)
[Manage Instance](#)
[Execute](#)


Application
Instance
HPC
partition

☐ Send email when execution finished, or failed.

Execute instance

Selected instance input values

Name	Value
3dpostproc	true
datavis	http://datavis.mathso.sze.hu:8000
datavisapikey	
emifile	octree_antwerp1_with_vegetation_
genwind	false
maxCo	80
maxDeltaT	30
maxIterationsTransient	20
maxtime	08:00:00
meshfile	octree_antwerp1_with_vegetation_


[Home](#)
[Applications](#)
[Data Catalogue](#)
[Configurations](#)
[Experiments](#)
[User Guide](#)
[leslie](#)

[Dashboard](#)
[Manage Instance](#)
[Execute](#)

3 executed deployment has been removed! (0 delete failed)

Select

Cancel selected

Delete selected

Last update: 23:07:37

Selected	ID	Name	HPC	Status	Created	Finished	Actions
<input type="checkbox"/>	1098	uap-2.5L2-ant_test_128c-2024-7-8_23_6_49	MeluXina-C	running	2024-07-08 21:06:51	-	
<input type="checkbox"/>	1097	uap-2.5L2-ant_test_128c-2024-7-8_15_33_43	MeluXina-C	completed	2024-07-08 13:33:45	2024-07-08 14:03:49	
<input type="checkbox"/>	1096	uap-2.5L2-ant_test_128c-2024-7-8_15_33_37	LUMI-C	failed	2024-07-08 13:33:38	-	
<input type="checkbox"/>	1093	uap-2.5L1-asdf-2024-7-7_23_33_32	MeluXina-C	completed	2024-07-07 21:33:34	2024-07-08 01:59:48	
<input type="checkbox"/>	1087	uap-2.5L1-asdf32-2024-7-7_18_35_33	Solyom-C	completed	2024-07-07 16:35:32	2024-07-07 17:13:44	
<input type="checkbox"/>	1075	uap foam v2.5a-antwerp_1h_32c-2024-7-6_22_46_38	Solyom-C	completed	2024-07-06 20:46:39	2024-07-06 21:25:26	

Application Manager

[Upload](#)[My Applications](#)

Application
name

File







 [Browse](#)

☐ Available for demo users?

[Submit](#)

UAP-FOAM Blueprint input fields

char: 32)

mesh	
traffic	
wind	
simTime	
Simulation Start time 	<input type="text" value="06:00:00"/>
Simulation End time 	<input type="text" value="07:00:00"/>
HPC	
Number of Nodes 	<input type="text" value="1"/>
ntasks 	<input type="text" value="128"/>
ntaskspernode 	<input type="text" value="128"/>
ncorespernode 	<input type="text" value="128"/>

```

148 simstarttime:
149   name: Simulation Start time
150   description: "Simulation Start time"
151   default: "00:00:00"
152   type: text
153   group: simTime
154   order: 9
155   optional: false
156
157 simendtime:
158   name: Simulation End time
159   description: "Simulation End time"
160   default: "23:59:00"
161   type: text
162   group: simTime
163   order: 10
164   optional: false
165
166
167 jobmanager_list:
168   name: Jobmanager
169   description: "Choose the used JobManager"
170   type: hidden
171   default: "SLURM"
172   group: HPC
173   order: 11
174   optional: false
175
176 nnodes:
177   name: Number of Nodes
178   description: "Number of Nodes"
179   default: "2"
180   type: text
181   group: HPC
182   order: 40
183   optional: false
184

```

UAP-FOAM Workflow in TOSCA

```

487 copying:
488     type: hpc.nodes.Job
489     properties:
490         job_options:
491             type: 'HEAD'
492             command: { concat: [ 'wget ', { get_input: pilot_url }
493         deployment:
494             #bootstrap: 'scripts/bootstrap_traffic.sh'
495             #revert: 'scripts/revert_traffic.sh'
496             skip_cleanup: true
497
498 sim_conf:
499     type: hpc.nodes.Job
500     properties:
501         job_options:
502             type: 'HEAD'
503             command: { concat: [ 'sleep 15 && ./mathso-gencfg.sh '
504     relationships:
505         - type: job_depends_on
506           target: copying
507
508 preproc:
509     type: hpc.nodes.Job
510     properties:
511         job_options:
512             type: 'SRUN'
513             command: './run.sh'
514             nodes: '1'
515             tasks: '1'
516             cpus_per_task: '12'
517             max_time: 01:00:00
518         deployment:
519             #bootstrap: 'scripts/bootstrap_traffic.sh'
520             #revert: 'scripts/revert_traffic.sh'
521             skip_cleanup: true
522         relationships:
523             - type: job_depends_on
524               target: sim_conf

```

```

525 simu:
526     type: hpc.nodes.Job
527     properties:
528         job_options:
529             type: 'SBATCH'
530             command: './cf/simulate.job'
531             max_time: '48:00:00'
532             std_out: 'slurm_out'
533             err_out: 'slurm_err'
534         deployment:
535             #bootstrap: 'scripts/bootstrap_traffic.sh'
536             #revert: 'scripts/revert_traffic.sh'
537             skip_cleanup: true
538         relationships:
539             - type: job_depends_on
540               target: preproc
541
542 upload:
543     type: hpc.nodes.Job
544     properties:
545         job_options:
546             type: 'HEAD'
547             command: './upload.sh'
548             std_out: 'upload_out'
549             err_out: 'upload_err'
550         deployment:
551             skip_cleanup: true
552         relationships:
553             - type: job_depends_on
554               target: simu
555
556 expected_results:
557     - type: 2D
558     - type: 3D

```


Home Applications Data Catalogue Configurations Experiments User Guide lesli

New Instance name (max. 32 char: 32)

image

Application image *i*

Source repository: MathSO CKAN

Dataset of the repository: uap-foam_images

File of the dataset: -- Please select from the list --

mesh

Source repository: MathSO CKAN

Dataset of the repository: antwerp_uap25_mesher

File of the dataset: octree_antwerp1_with_vegetation_

emission

Choose Emission type *i*: EMI file based (select from next list)

EMI file *i*

Source repository: MathSO CKAN

Dataset of the repository: antwerp_uap25_emission

File of the dataset: antwerp_step_3_month_3.nox.zip

```

app_image:
  name: Application image
  description: Singularity image for UAP-FOAM applicat
  default: ""
  type: ckan_file
  ckan_tag: uap25 image
  group: image
  order: 2
  optional: false
  dataset_optional: false

model_mesh:
  name: Mesh
  description: Mesh dataset and mesh.
  default: ""
  type: ckan_file
  ckan_tag: uap25 mesh
  group: mesh
  order: 3
  optional: false
  dataset_optional: false

emitype:
  name: Choose Emission type
  description: "Choose Emission Type Source"
  type: list
  choices:
    - first:
      text: No Emission (WindComfort)
      value: noemission
    - second:
      text: EMI file based (select from next list)
      value: emiemiission
    - fifth:
      text: Point Source (edit in advanced settings)
      value: pointsource
  group: emission
  order: 11
  optional: true

```

MathSO

Datasets Organizations Gr

/ Datasets

Filter by location Clear

+ -

30 datasets found for "uap25"

Map data © OpenStreetMap contributors
Tiles by Stamen Design (CC BY 3.0)

Organizations

SZE 29

SZE-only 1

Groups

There are no Groups that match this search

Tags

Inputs 9

2.0 7

OPENFOAM 7

PRIVATE antwerp_uap25_mesher
This dataset has no description
msh

PRIVATE antwerp_uap25_emission
This dataset has no description
ZIP

PRIVATE antwerp_uap25_wind
This dataset has no description
ZIP

PRIVATE uap-foam_images
This dataset has no description

Application: wf_sing_integration_v1

Instance: uap_foam_integration

[Update](#)
[Submit](#)
[Delete](#)
[Export into JSON](#)

New Instance name (max. char: 32):

basic

Image repository settings

```
1 # repo type
2 WF_REPO_TYPE="ckan"
3
4 # access url and key
5 WF_REPO_URL="https://datarepo.mathso.sze.hu"
6 WF_REPO_KEY=""
7 WF_REPO_DATASET="sing_images"
8
```

Repository for storing singularity images.

Configuration

```
1 # base image for container
2 WF_CONT_BASE="ubuntu:22.04"
3 #
4 # OpenMPI versions to compile
5 WF_CONT_OPENMPI_VERSIONS="3.1.6 4.1.6"
6 # MPICH versions to compile
7 WF_CONT_MPICH_VERSIONS="3.1.4"
8 #
9 # Framework and versions to compile (optional)
10 WF_CONT_FRAMEWORK_NAME=""
11 # WF_CONT_FRAMEWORK_VERSION=""
12 WF_CONT_FRAMEWORK_NAME="openfoam"
13 WF_CONT_FRAMEWORK_VERSION="v2212 v2312"
14 #
15 # Application to compile
16 WF_CONT_APP_NAME="uap_foam"
17 WF_CONT_APP_VERSION="main"
18
```

advanced

OpenMPI recipe

```
1 # Docker file
2 bootstrap: docker
3 from: DIST
4 #
5 # Recipe is processed with m4. Singularity does not support
6 # Variables:
7 # DIST: WF_CONT_BASE
8 # VER: OpenMPI version
9 #
10 %post
11 # Update ubuntu
12 apt-get update
13 apt-get install -y file g++ gcc gfortran make gdb
14 mkdir -p /opt/mpi
15
16 # Install OpenMPI
17 major=$(echo VER | cut -d. -f-2)
18 wget -q https://download.openmpi.org/release/open-
19 tar xf openmpi-VER.tar.bz2
20 cd openmpi-VER
21
```

Docker recipe for OpenMPI integration.

MPICH recipe

```
1 bootstrap: docker
2 from: DIST
3 #
4 # Recipe is processed with m4. Singularity does not support
5 # Variables:
6 # DIST: WF_CONT_BASE
7 # VER: MPICH version
8 #
9 %post
10 # Install software
11 apt-get update
12 apt-get install -y file g++ gcc gfortran make gdb
13
14 # Install MPICH
15 wget -q http://www.mpich.org/static/downloads/VER/
16 tar xf mpich-VER.tar.gz
17 cd mpich-VER
18 ./configure --disable-fortran --enable-fast=all,03
19 make -j$(nproc)
20 make install
21
```

Docker recipe for MPICH integration.

Framework recipe

```
9 %post
10 apt-get install -y libz-dev libfl-dev
11
12 # Install OpenFOAM
13 mkdir -p /opt/OpenFOAM
14 cd /opt/OpenFOAM
15 wget -q https://dl.openfoam.com/source/VER/OpenFOAM
16 wget -q https://dl.openfoam.com/source/VER/ThirdParty
17 tar xf OpenFOAM-VER.tgz
18 tar xf ThirdParty-VER.tgz
19 cd OpenFOAM-VER
20 cd OpenFOAM-VER
21 sed -i 's/PROJECT_DIR/$(pwd)/' ./etc/defaults/
22 sed -i 's/PROJECT_DIR/$(pwd)/' ./etc/defaults/

```

© Széchenyi University 2024

2.2 CFD codes in UAP

1. OpenFOAM

1. 3D incompressible Navier-Stokes,, FVM on unstructured meshes, RANS and URANS, (pimpleFOAM, simpleFOAM), passive scalar (for pollutant dispersion)
2. Optimized code for CPU (OpenMPI)
 1. **Parallel efficiency: >80% (#cells=10M)**
3. 1 paper published, joint with the other (non-HPC) FAIRMODE-teams

2. RedSim

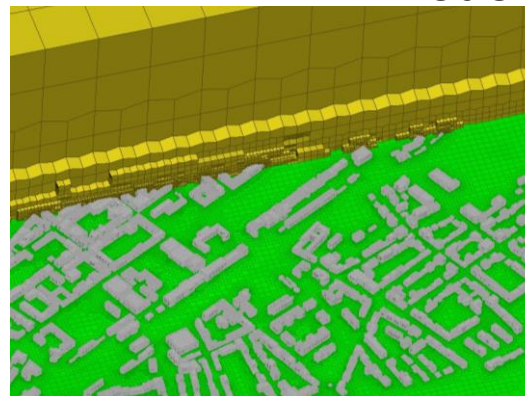
1. C/C++/CUDA code for 2D/3D Euler and Navier-Stokes, compressible, 1st & 2nd order FVM, unstructured polyhedral meshes, reduced order mode (POD, POD-DEIM)
2. Optimized code for CPU (OpenMPI, cray-mpich on LUMI), Multi-GPU (CUDA, NVLINK), randomized SVD
 1. **Parallel efficiency > 85% with 8 GPUs (KAROLINA) and #cells = 30M**

3. Testing: Xyst CFD code

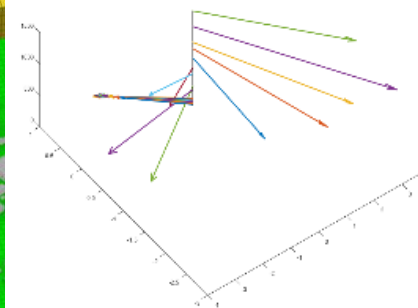
1. 3D Navier-Stokes, compressible, FEM on unstructured tetrahedral mesh, adaptive load distribution, asynchronous parallel programming with the Charm++ runtime system, Low-Mach solver under development
2. Optimized code for CPU (OpenMPI), started as a fork of the Quinoa-code, <https://quinoacomputing.github.io>

- Input
 - 3D Mesh City Model
 - Weather-based boundary, ECMWF coupled
 - Traffic or point-based pollution source
- Equations and solution
 - Incompressible URANS solved with k-e turbulence model
 - NO_x Pollution s calculated with scalar transport with volumetric source S_s .
 - Transient simulation with PIMPLE method, initialized with SIMPLE method

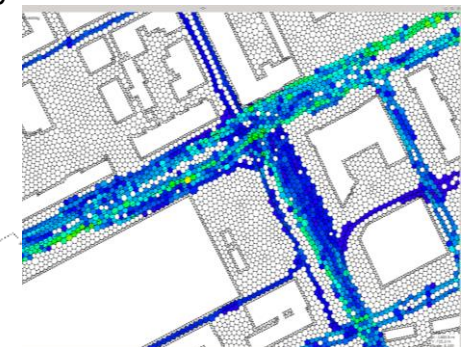
Model



Boundary



Traffic

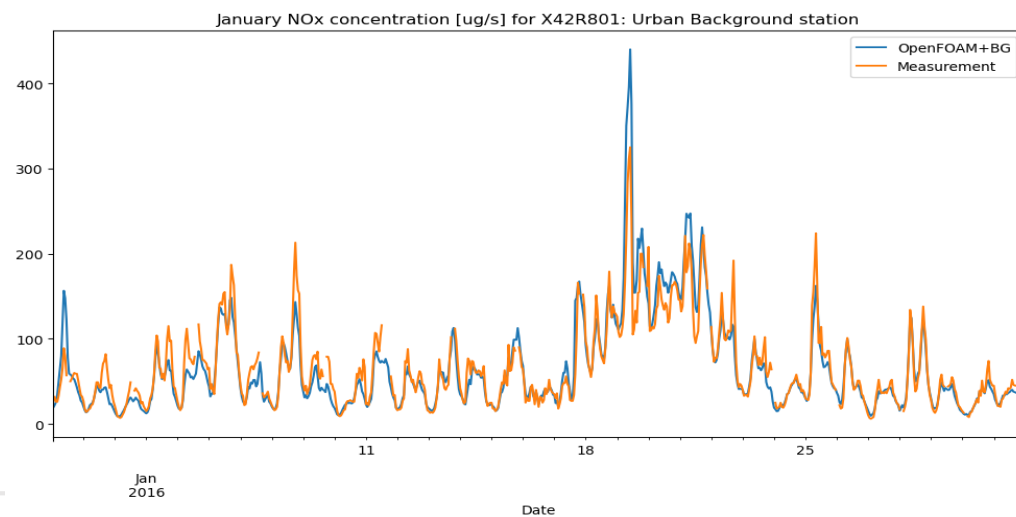
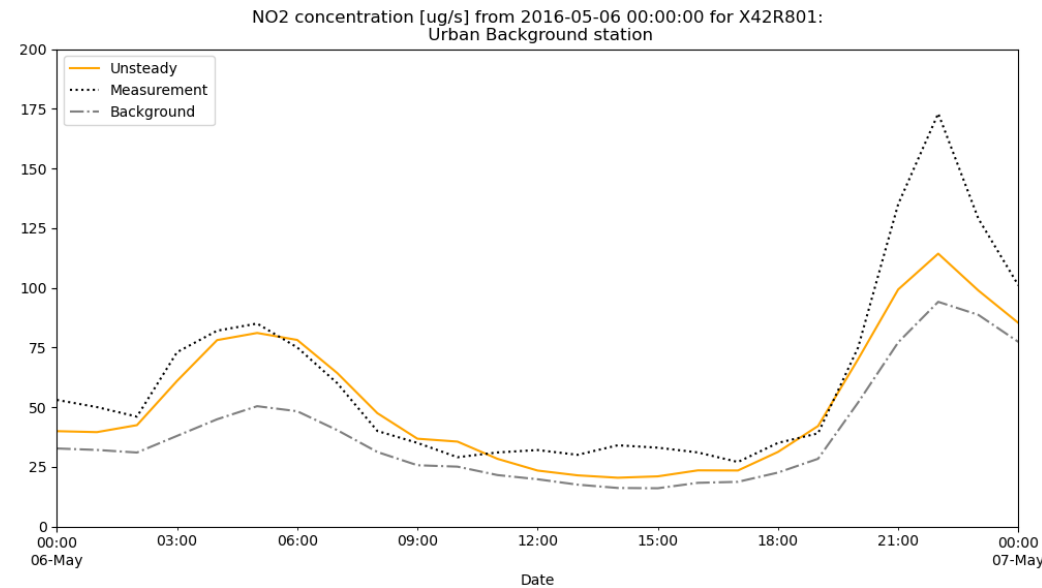


3D Result



UAP-FOAM Validation

- Antwerp, 2016, full year simulation, measured wind data, yearly average based traffic modelling
- 2 EU measurement stations, several passive samplers
- Comparison with several modelling groups in FAIRMODE: Martin et. al., STOTEN, Volume 925, May 2024, 171761

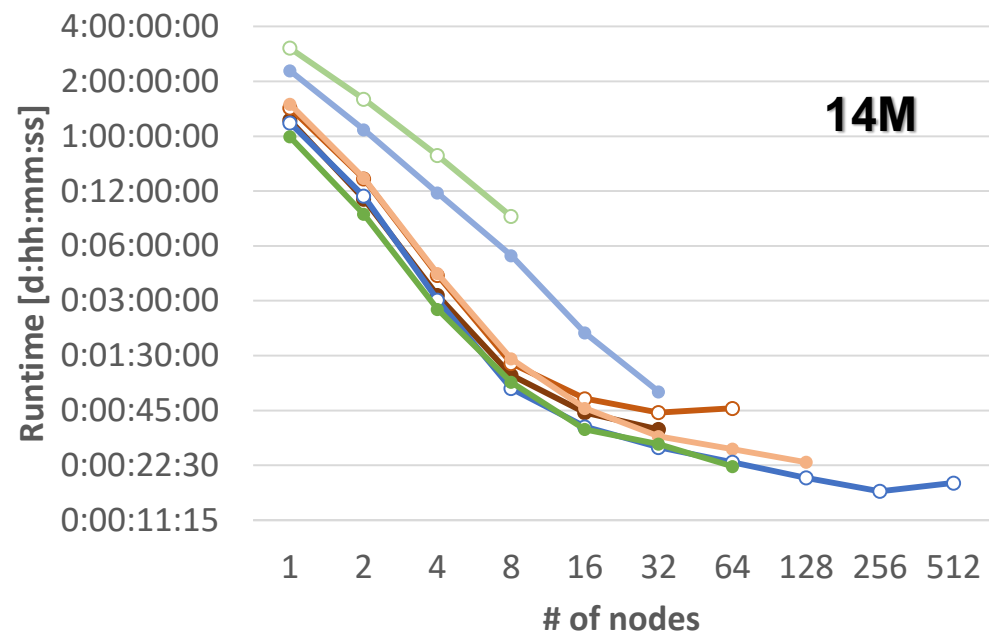
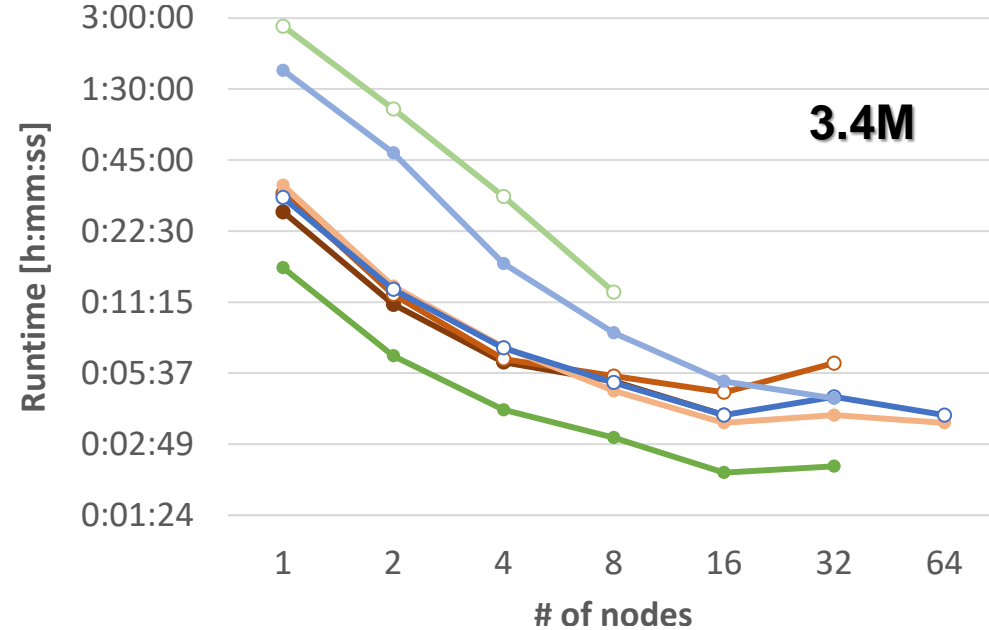
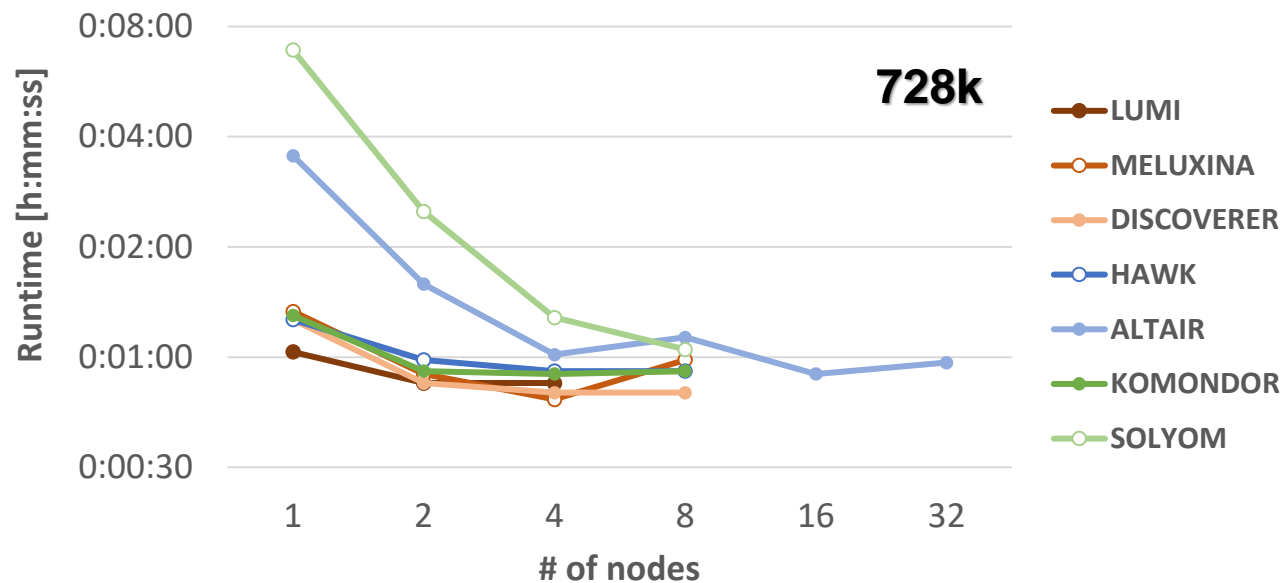


2.2 UAP-FOAM Benchmark - Hardware Overview

		LUMI	MELUXINA	DISCOVERER	HAWK	ALTAIR	KOMONDOR	SOLYOM
Location		Finland	Luxembourg	Bulgaria	Germany	Poland	Hungary	Hungary
<i># of CPU nodes available</i>		1022	573	1110	5632	1321	184	12
<i># of sockets per node</i>		2	2	2	2	2	2	2
<i># of cores per socket</i>		64	64	64	64	24	64	16
<i># of cores total</i>		130816	73344	142080	720896	63408	23552	384
CPU	<i>vendor</i>	AMD	AMD	AMD	AMD	INTEL	AMD	INTEL
	<i>type</i>	EPYC 7763	EPYC 7H12	EPYC 7H12	EPYC 7742	XEON 8268	EPYC 7763	XEON 6226R
RAM	<i>per node</i>	256 GB	512 GB	256 GB	256 GB	192 GB	256 GB	192 GB
Interconnect	<i>type</i>	Slingshot	Infiniband	Infiniband	Infiniband	Infiniband	Slingshot	Infiniband
	<i>card</i>	SS11	MT4123	MT4123	HDR	MT4119	SS11	MT4123
	<i>BW</i>	200 Gbit/s	200 Gbit/s	200 Gbit/s	200 Gbit/s	100 Gbit/s	200 Gbit/s	100 Gbit/s
Compiler		cray clang	gnu gcc	gnu gcc	gnu gcc	gnu gcc	gnu gcc	gnu gcc
	<i>version</i>	14.0.2	11.3.0	11.3.0	9.2	10.2	12.2.0	9.4.0
MPI		cray-mpich	openmpi	openmpi	hpe mpt	openmpi	cray-mpich	openmpi
	<i>version</i>	8.1.18	4.1.4	4.1.4	2.23	4.1.0	8.1.24	4.1.2
OpenFOAM	<i>version</i>	v2112	v2206	v2206	v2012	v2012	v2112	v2112

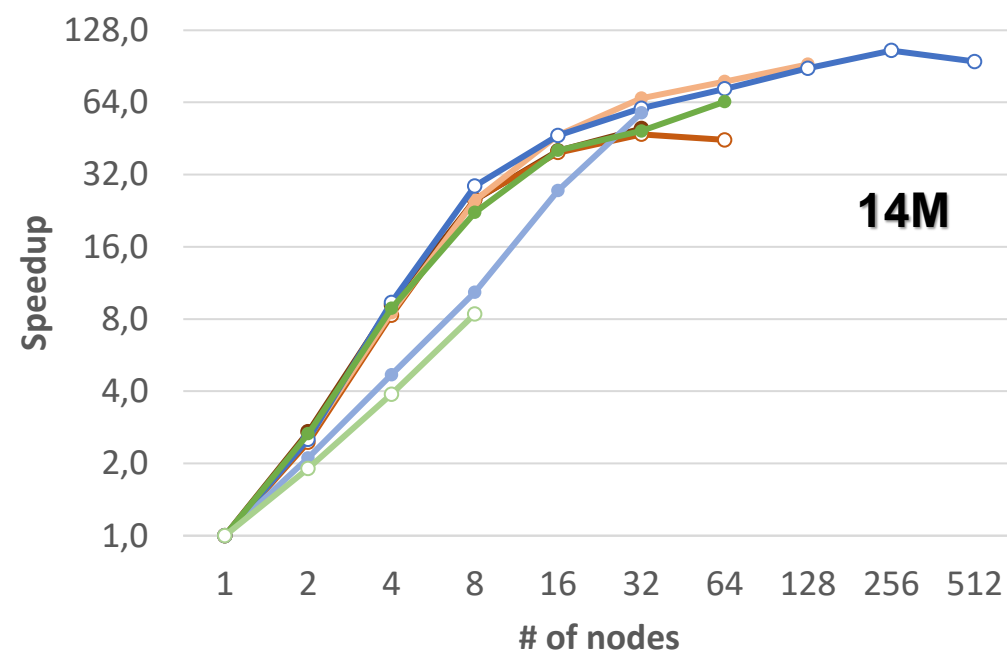
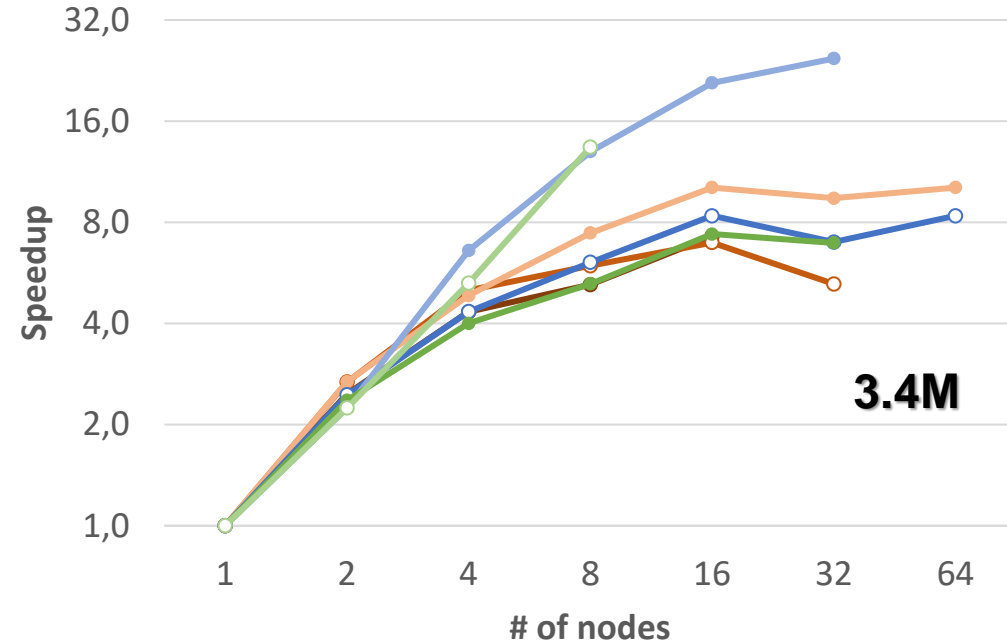
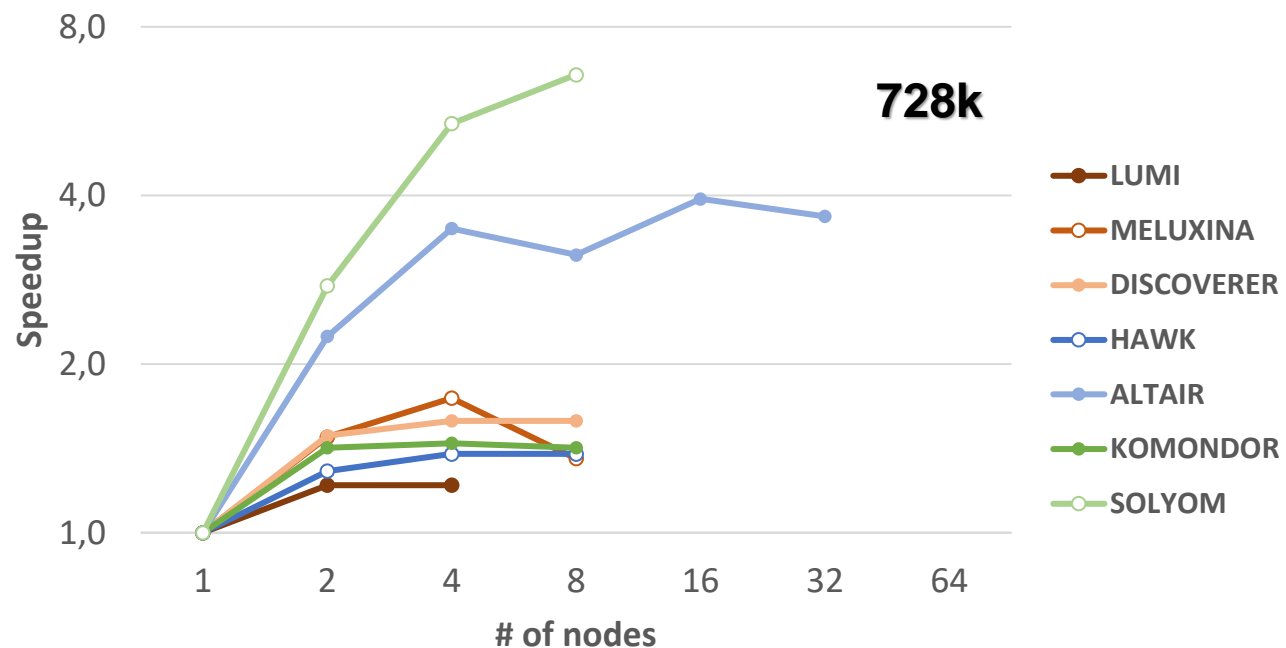
 LUMI
  MELUXINA
  DISCOVERER
  HAWK
  ALTAIR
  KOMONDOR
  SOLYOM

2.2 Benchmarking: UAP-FOAM Runtime



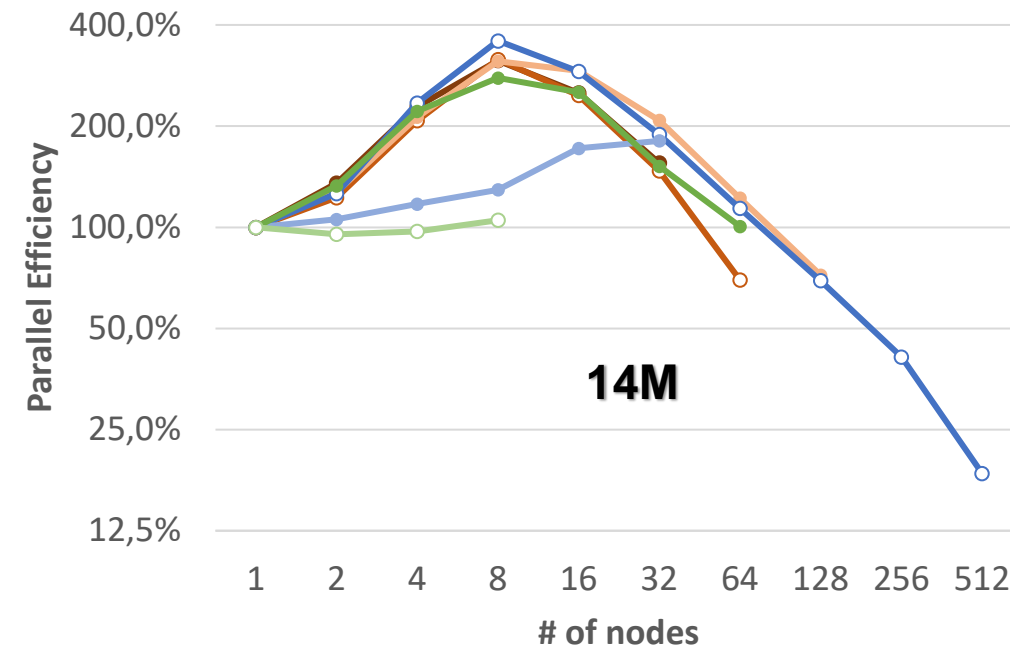
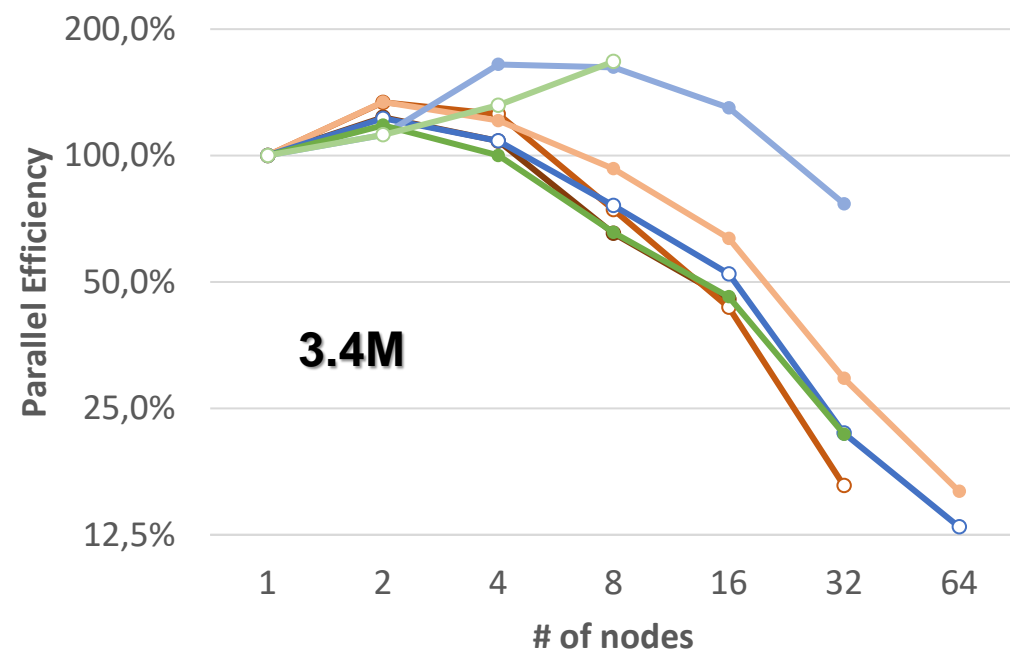
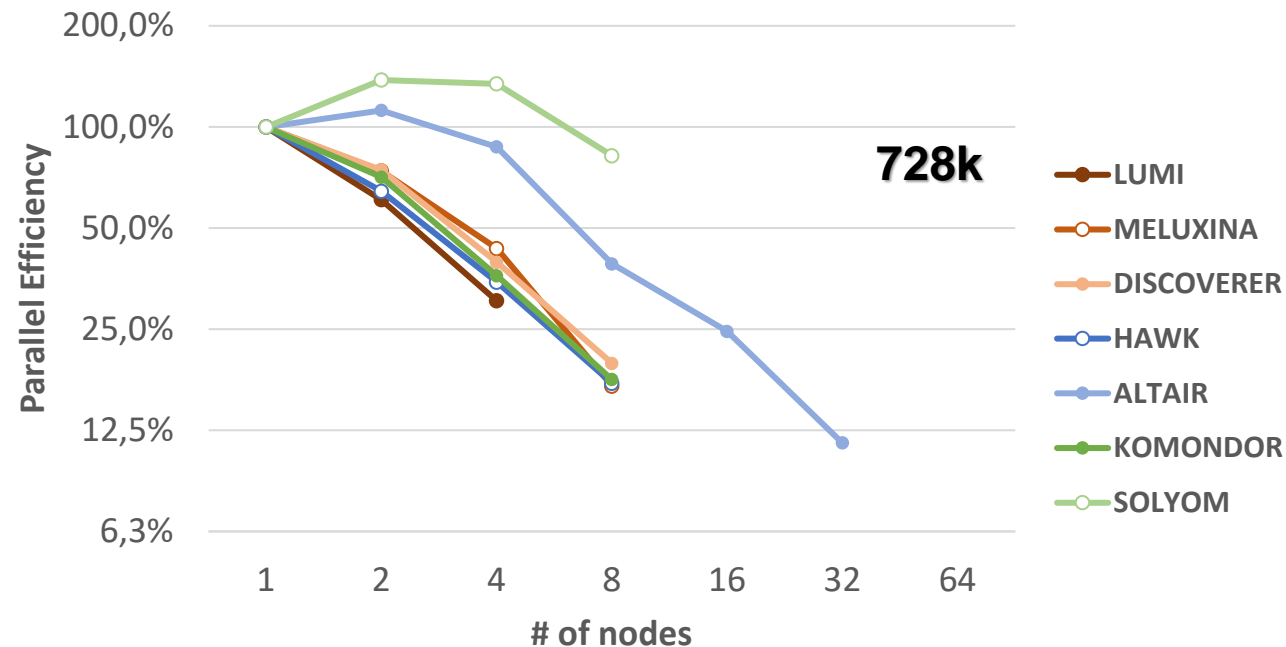
- Győr geometry with mesh sizes 728k, 3.4M, 14M
- Testruns on 7 architectures
- Runtime estimation [hh:mm:ss] for 1h simulated time

2.2 Benchmarking: UAP-FOAM Speedup



- Győr geometry with mesh sizes 728k, 3.4M, 14M
- Testruns on 7 architectures
- Speedups with regard to 1 node

2.2 Benchmarking: Parallel Efficiency



- Győr geometry with mesh sizes 728k, 3.4M, 14M
- Testruns on 7 architectures
- Parallel efficiency with regard to 1 node

2.3 The RedSIM and CFDR software

<https://redsim.mathso.sze.hu/>

RedSim:

- native multi-GPU CFD-solver to simulate compressible fluids on unstructured, polyhedral meshes.

CFDR:

- visualization software for CFD running on HPC and to visualize data on your web browser live.

API:

- expressive, compact API written in Lua, helps the users to write their applications with RedSim and CFDR.

2.3 The RedSIM software: The FOM algorithm

Algorithms and data structures

1. Solves the compressible Euler and Navier-Stokes equations with the ideal equation of states (EOS)
2. Uses finite volume method for the spatial (semi)discretization
 1. Control volumes = unstructured polyhedra
 2. Upwinding with the Vijayasundaram flux-vector splitting
 3. 2nd order scheme with averaging to vertices and linear reconstruction on faces
3. (Yet) Explicit time-stepping with the Euler and optimal TVD schemes
4. One single algorithm for 2D/3D computations
5. Input/Output data formats: industrial standards, e.g. Enight Gold, Nastran.

Algorithms

- Basic method: the POD – proper orthogonal decomposition method
 - Snapshot collection from the representative states and the SVD to define the projections
- Several variants of DEIM (discrete empirical interpolation methods) are implemented for numerical performance purposes
- For the snapshot collection the RedSim FOM multi-GPU version was applied
- For the SVD computations
 - From the Eigen library, and
 - the RedSVD randomized numerical solver, see <https://github.com/cequencer/redsvd>

2.3 The RedSIM software: Code features

Coding style

- Developed from scratch by the authors (algorithms: ZH, programming: MC)
- Written in C99-style, compiled as C++, avoids C++ features (RAII, reflections, exceptions, ...)
- Custom memory allocators, such as Linear Arena, Heaps, or Pools when appropriate; no use of new
- "Data Oriented Approach" (cf. Mike Acton), SOA paradigm
- One single algorithm and one source file, runs easily on each platform
- Parallel:
 - CPU (OpenMPI + OpenMP)
 - Multi-GPU (with CUDA)
- One single algorithm and code for handling 2D/3D
- Integrated in-house real-time 3D visualizer
- Reconfigurable during running, due to suitable lua-scripts
- Highly optimized code for data size, data copy, and computations

Benchmark architectures

1. Solyom local cluster
 1. 1x FAT node, #CPU-cores = 80, RAM = 3TB
 2. 1x V100s NVIDIA GPU
2. Komondor HPC-machine of the EuroHPC Hungarian National Competence Center KIFÜ
 1. 1x GPU-node with 8x A100 NVIDIA

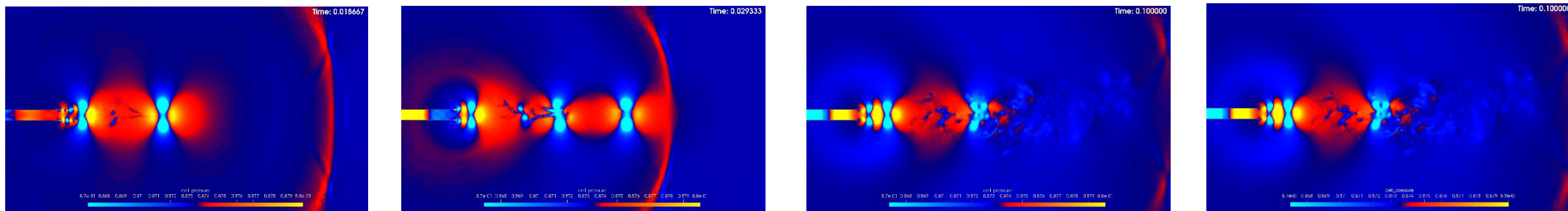
Benchmark problems

1. Exhaust pipe acoustics problem – for FOM
 1. Origin: industrial problem from automotive industry
 2. #cells = 70M, d.o.f. = 350M
 3. Simulated time (physical time): 1.0 sec
2. Urban air flow computation for the city of Gyor – for FOM and ROM
 1. Small mesh: #cells = 1.4M (d.o.f. = 7M) (spatial resolution: 5 m) (Remark: 1 state vector = 100 MB)
 2. Medium mesh: #cells = 18M (d.o.f. = 90M) (spatial resolution: 2 m) (Remark: 1 state vector = 1 GB)

2.3 RedSim benchmarks

Benchmark problem 1: Exhaust pipe acoustics problem

- Runtime: 6 hours with the multi-GPU version, [video: https://www.youtube.com/watch?v=rAKsChm9-b8](https://www.youtube.com/watch?v=rAKsChm9-b8)



2. Validation to measurements

Comparison: AUDI measurements / SZE results.

Evaluated Time: [0.1s, 0.4s]

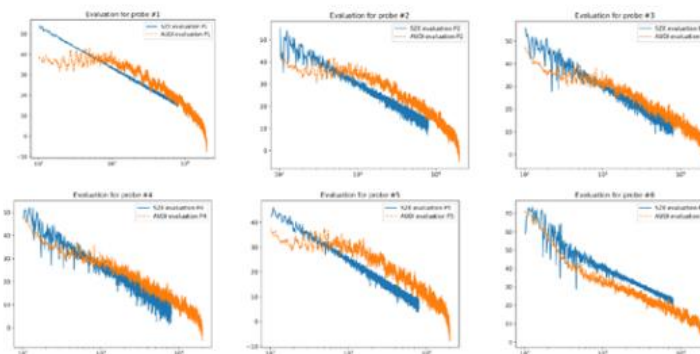


Figure 5: Comparison between the measured SPL values for the straight-pipe provided by the industrial partner (see [3]), and the simulated SPL values by SZE.

2.3 RedSim benchmarks

Table 1. Parallel efficiency for the 2D SOD Shock Tube Problem with $N = 16,000,000$ squares as cells. P denotes the number of GPUs.

EFFICIENCY	KOMONDOR 8x NVIDIA A100-SXM4-40GB	VEGA 4x NVIDIA A100-SXM4-40GB
P=1	1	1
P=2	0.968994657	0.9772293065
P=4	0.9142748011	0.9205859105
P=8	0.7212876395	NO DATA

Table 2. Parallel efficiency for the 2D SOD Shock Tube Problem with $N = 64,000,000$ squares as cells. P denotes the number of GPUs.

EFFICIENCY	KOMONDOR 8x NVIDIA A100-SXM4-40GB	VEGA 4x NVIDIA A100-SXM4-40GB
P=1	1	1
P=2	0.9897779737	0.9930057839
P=4	0.9706092646	0.9753116722
P=8	0.9105837686	NO DATA

Table 3. Parallel efficiency for the 3D Gyor airflow problem with $N = 18,342,623$ tetrahedra as cells. P denotes the number of GPUs.

EFFICIENCY	KOMONDOR 8x NVIDIA A100-SXM4-40GB	VEGA 4x NVIDIA A100-SXM4-40GB
P=1	1	1
P=2	0.9225121646	0.9742910708
P=4	0.8208693259	0.9753116722

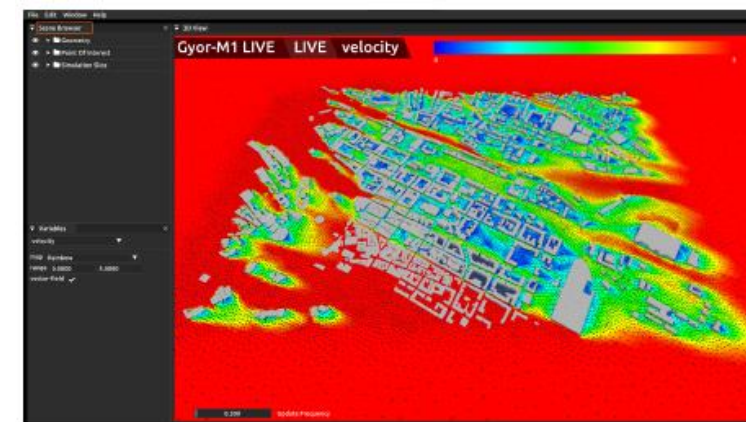


Fig. 1. Live visualization of the HPC-computations.

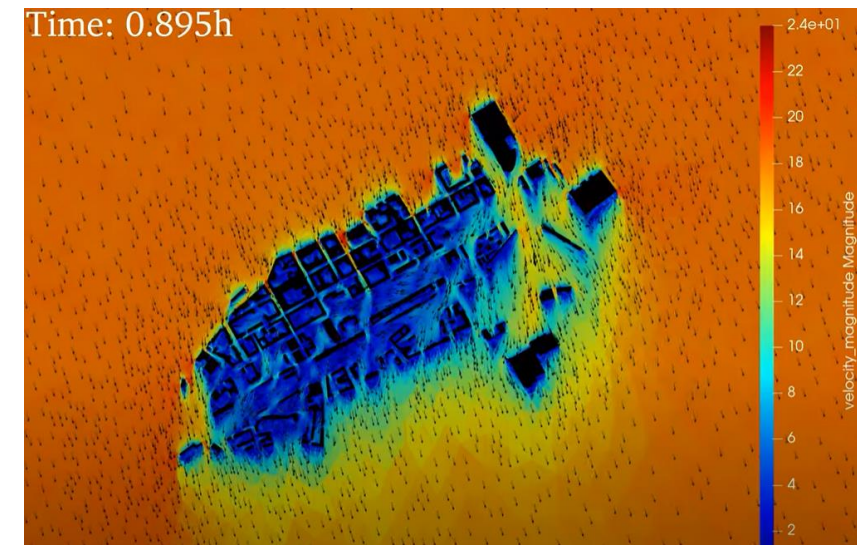
Video: <https://www.youtube.com/watch?v=-pJAISsKJHQ>

2.3 RedSim benchmarks

Benchmark problem 2.2: Urban air flow, #cells=18M

1. Video: <https://www.youtube.com/watch?v=-pJAISsKJHQ>

2. Runtimes, and parallel scalability: 82% on p=4 GPUs
(i.e. more than 3 times faster on 4 GPU than on 1 GPU)



Wall-Clock Time: Urban Airflow, Gyor. N = 18342623			
wall-clock (s)	ONDOR, 4x NVIDIA A100-SXM4-4	4x NVIDIA A100-SXM4	SOLYOM, 1x NVIDIA Tesla V100S-PCIE-32GB
P = 1	2781.590725	2783.904732	2127.01869
P = 2	1507.617369	1428.682257	NO DATA
P = 4	847.147846	1475.724026	NO DATA
Wall-Clock Time: Urban Airflow, Gyor. N = 3154126			
wall-clock (s)	KOMONDOR	VEGA	SOLYOM, 1x NVIDIA Tesla V100S-PCIE-32GB
P = 1	432.583269	433.15866	276.966294
P = 2	271.443758	234.848385	NO DATA
P = 4	182.695222	269.181903	NO DATA

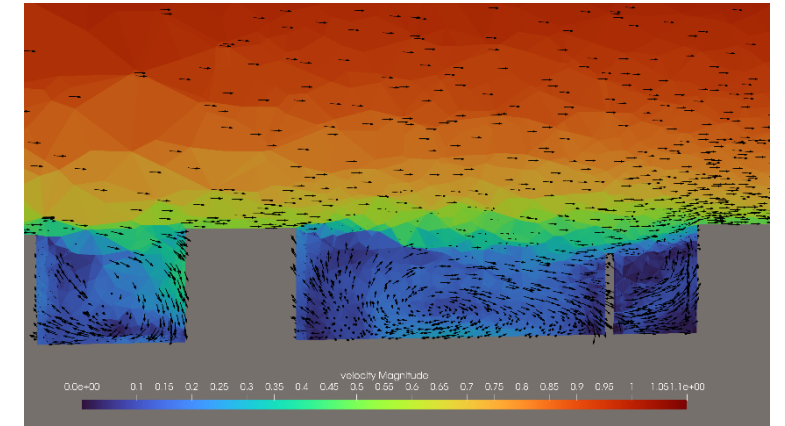
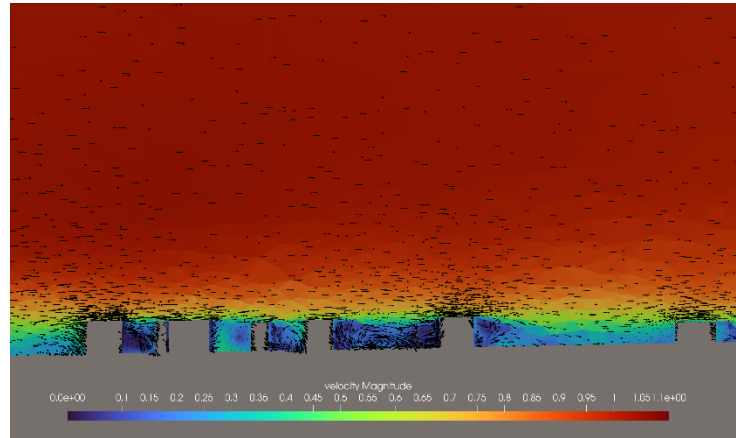
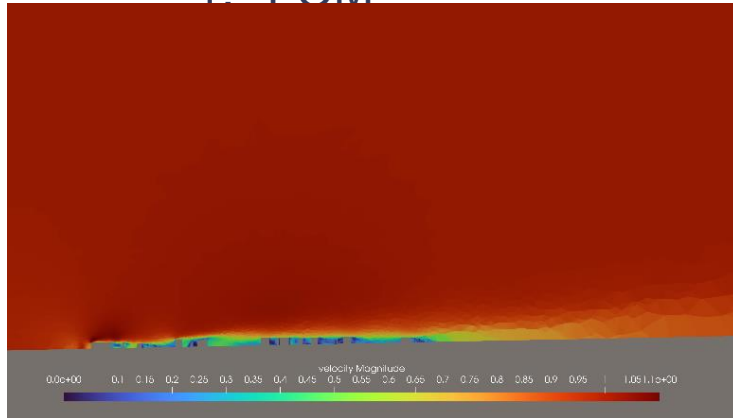
Efficiency: Urban Airflow, Gyor. N = 30,291,099		
efficiency	KOMONDOR, 4x NVIDIA A100-SXM4-40GB	VEGA, 4x NVIDIA A100-SXM4-40GB
P = 1	1	-
P = 2	0.9449242544	-
P = 4	0.8752212861	-
Efficiency: Urban Airflow, Gyor. N = 18,342,623		
efficiency	KOMONDOR, 4x NVIDIA A100-SXM4-40GB	VEGA, 4x NVIDIA A100-SXM4-40GB
P = 1	1	1
P = 2	0.9225121646	0.9742910708
P = 4	0.8208693259	0.4716167595
Efficiency: Urban Airflow, Gyor. N = 3,154,126		
efficiency	KOMONDOR, 4x NVIDIA A100-SXM4-40GB	VEGA, 4x NVIDIA A100-SXM4-40GB
P = 1	1	1
P = 2	0.7968193341	0.9222091521
P = 4	0.5919466096	0.4022917729

2.3 Redsim benchmarks: FOM vs ROM

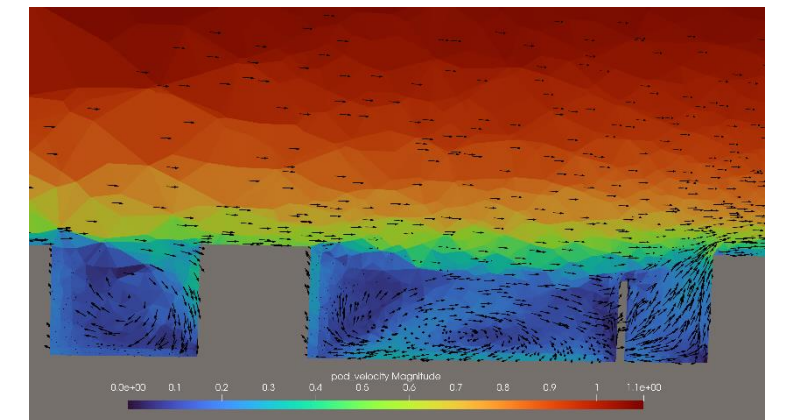
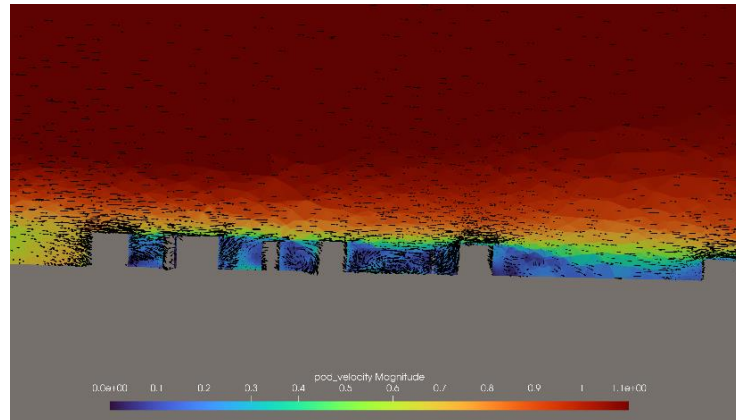
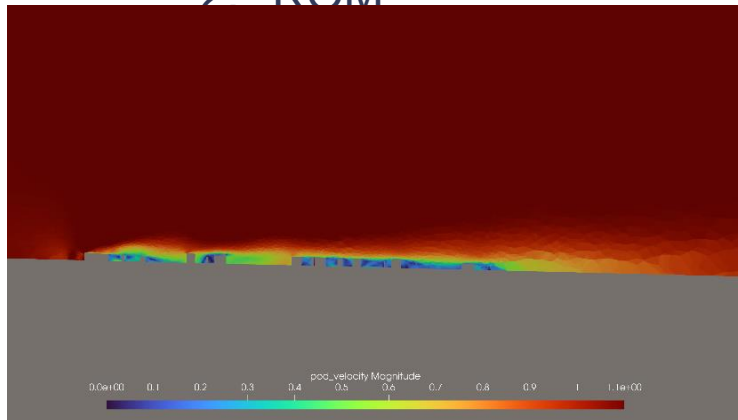
Benchmark problem 2.2: Urban air flow, #cells=18M

3. Visual comparison of FOM and ROM with $\eta = 85\%$; [video: https://youtu.be/wTGuyMImFcE](https://youtu.be/wTGuyMImFcE) (#cells=1.4M)

1. FOM



2. ROM



2.3 RedSim benchmarks: FOM vs. ROM

Benchmark problem 2.2: Urban air flow

ETA + (1, 2, 2, 5, 1)	r	CFL	rho	rho v1	rho v2	rho v3	e	V100 GPU runtime (seconds)
100%	(100, 100, 100, 100, 100)	25	0.00232	0.05780	0.07466	0.20311	0.00313	95.83356
99%	(2, 63, 63, 84, 2)	50	0.00021	0.06854	0.07245	0.18365	0.00011	31.20451
98%	(2, 48, 48, 72, 2)	60	0.00036	0.06847	0.07175	0.18855	0.00021	27.59252
95%	(2, 26, 27, 51, 2)	60	0.00031	0.07146	0.07409	0.19842	0.00017	21.91905
90%	(2, 13, 13, 35, 2)	60	0.00150	0.07298	0.06699	0.19363	0.00265	18.16461
80%	(2, 6, 6, 20, 2)	60	0.01023	0.11446	0.15723	0.18984	0.01157	15.54122
70%	(2, 4, 4, 13, 2)	100	0.02326	0.20657	0.32415	0.28957	0.02581	7.50004
60%	(2, 3, 3, 10, 2)	100	0.02158	0.21504	0.32845	0.31873	0.01073	7.31519
50%	(2, 3, 3, 8, 2)	100	0.02190	0.21563	0.33013	0.31821	0.01151	7.20283
40%	(2, 3, 3, 7, 2)	100	0.02252	0.21503	0.33032	0.31542	0.01251	7.17410
20%	(2, 3, 3, 6, 2)	100	0.02206	0.21370	0.32785	0.31684	0.01176	7.12716

N = 18342623, t = 3600s, ||v||=1m/s Static Wind (80 deg), REDUCED DOMAIN, L1 ERROR MEDIAN FROM 11 DIFF TIME POINTS

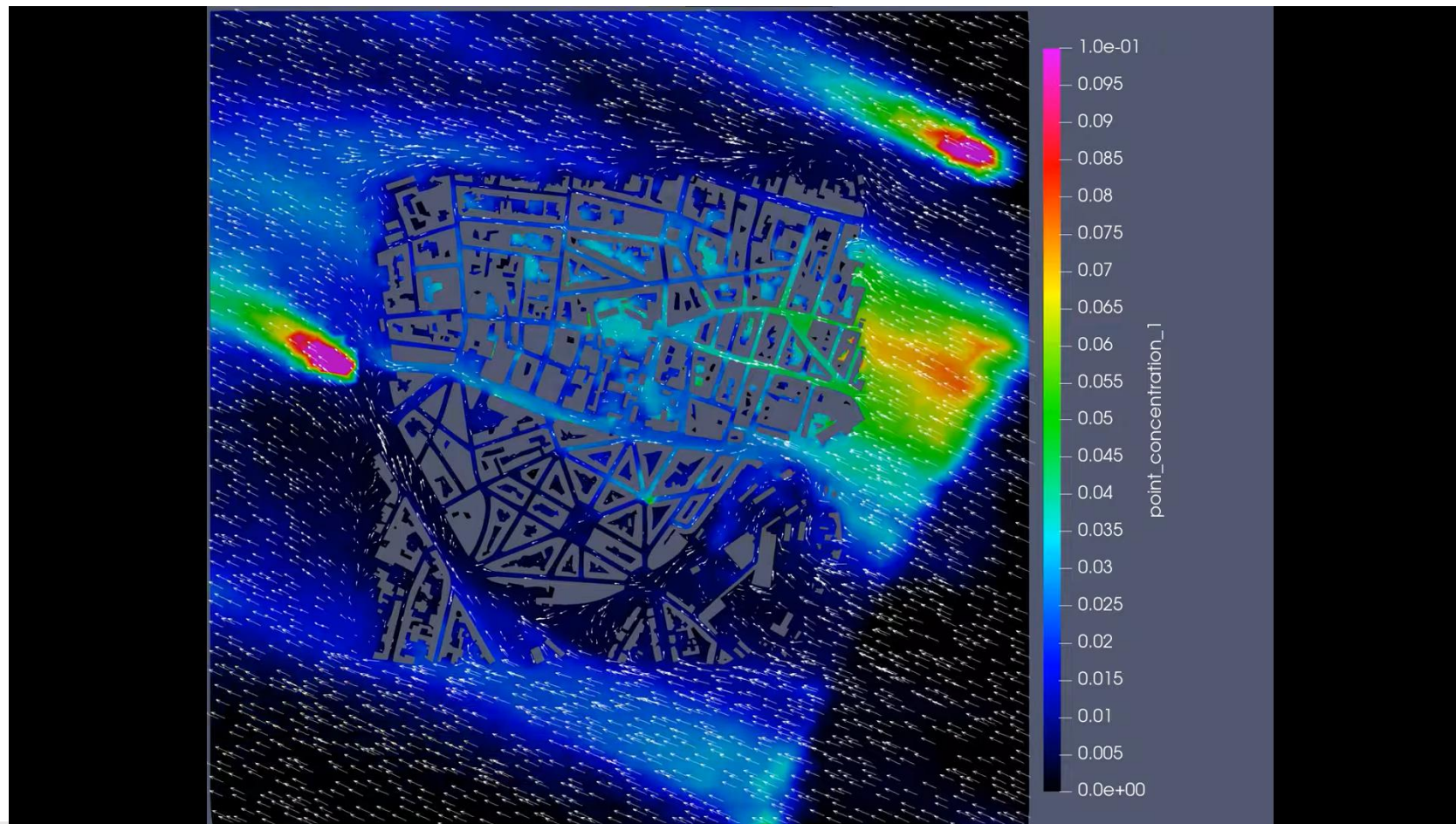
ETA + (1, 2, 2, 5, 1)	r	CFL	rho	rho v1	rho v2	rho v3	e	V100 GPU runtime (seconds)
85%	(2, 39, 39, 56, 2)	25	0.013936	0.175821	0.179983	0.439115	0.017311	979.648389
85%	(2, 39, 39, 56, 2)	50	0.01398	0.17586	0.18003	0.43916	0.01733	494.295947
80%	(2, 28, 28, 45, 2)	50	0.00922	0.17070	0.17400	0.43124	0.01314	440.468834
70%	(2, 14, 14, 29, 2)	100	0.00594	0.16619	0.15655	0.40173	0.01383	188.047495
60%	(2, 8, 8, 19, 2)	100	0.01509	0.19095	0.18687	0.51170	0.00099	168.482461
20%	(2, 3, 3, 7, 2)	100	0.08093	0.25193	0.28858	0.43526	0.01737	154.073113

Physical time: 3,600s	Runtime for #cells=1.4M:		Runtime for #cells=18M:	
	FOM	712.85s	FOM	13,723.79s
	POD eta=90%	18.16s	POD eta=85%	494.30s

3. Conclusions 1

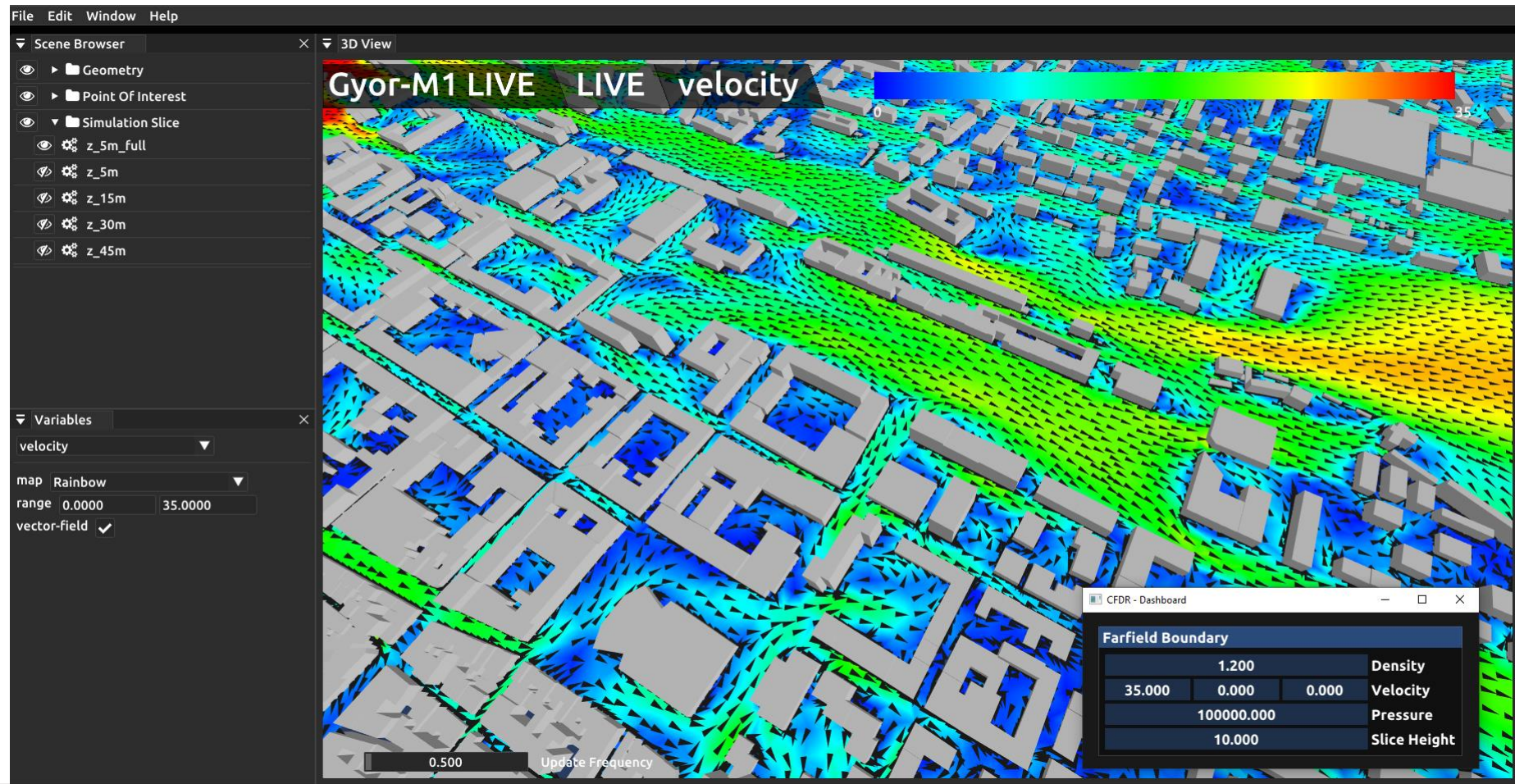
NOx concentration propagation in Antwerp for **1 full year**, FAIRMODE intercomparison exercise.

Link: <https://www.youtube.com/watch?v=sikD2pbpS9A>



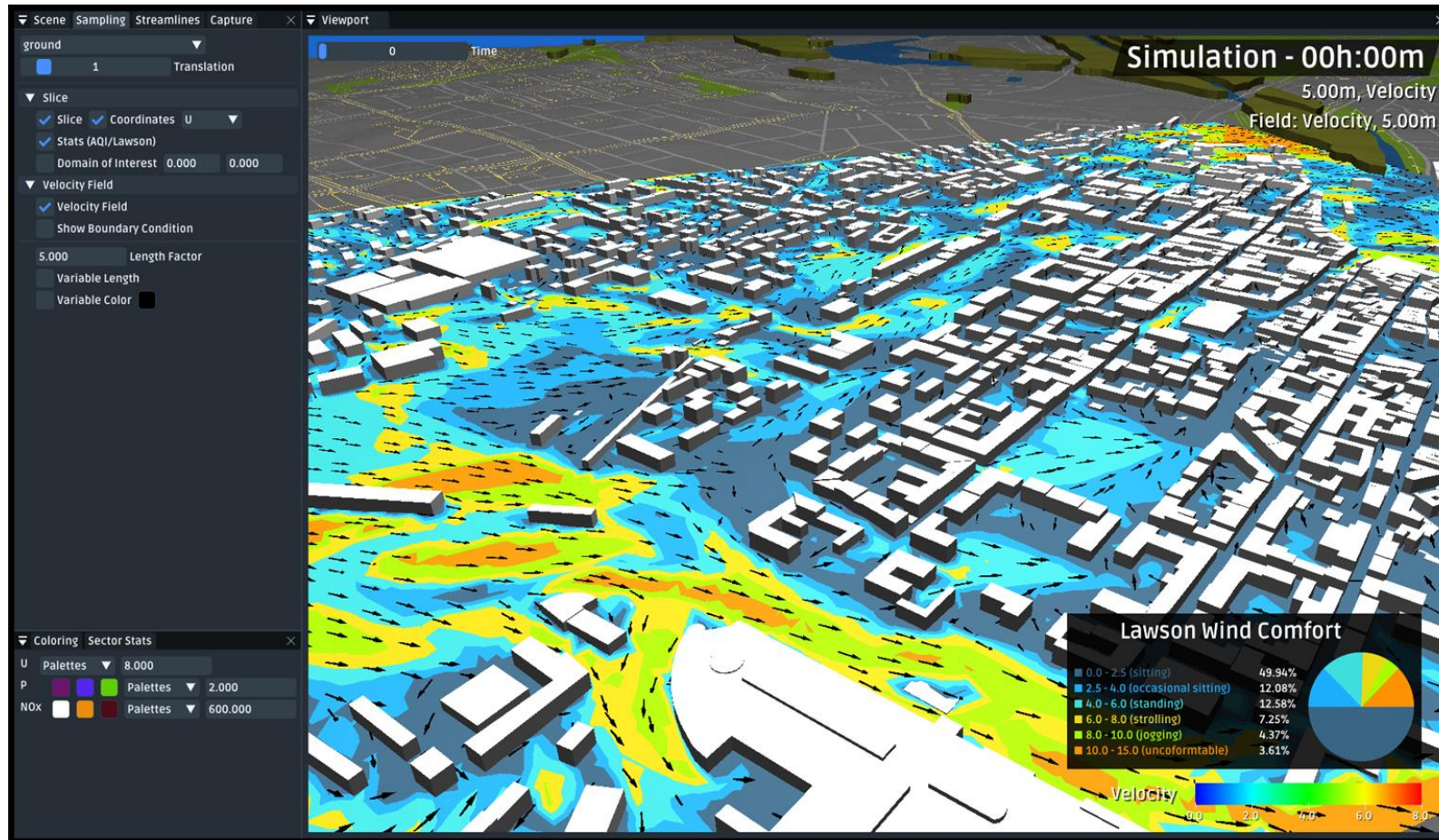
3. Conclusions 2

Real-time digital twin prototype for the airflow for the city of Győr



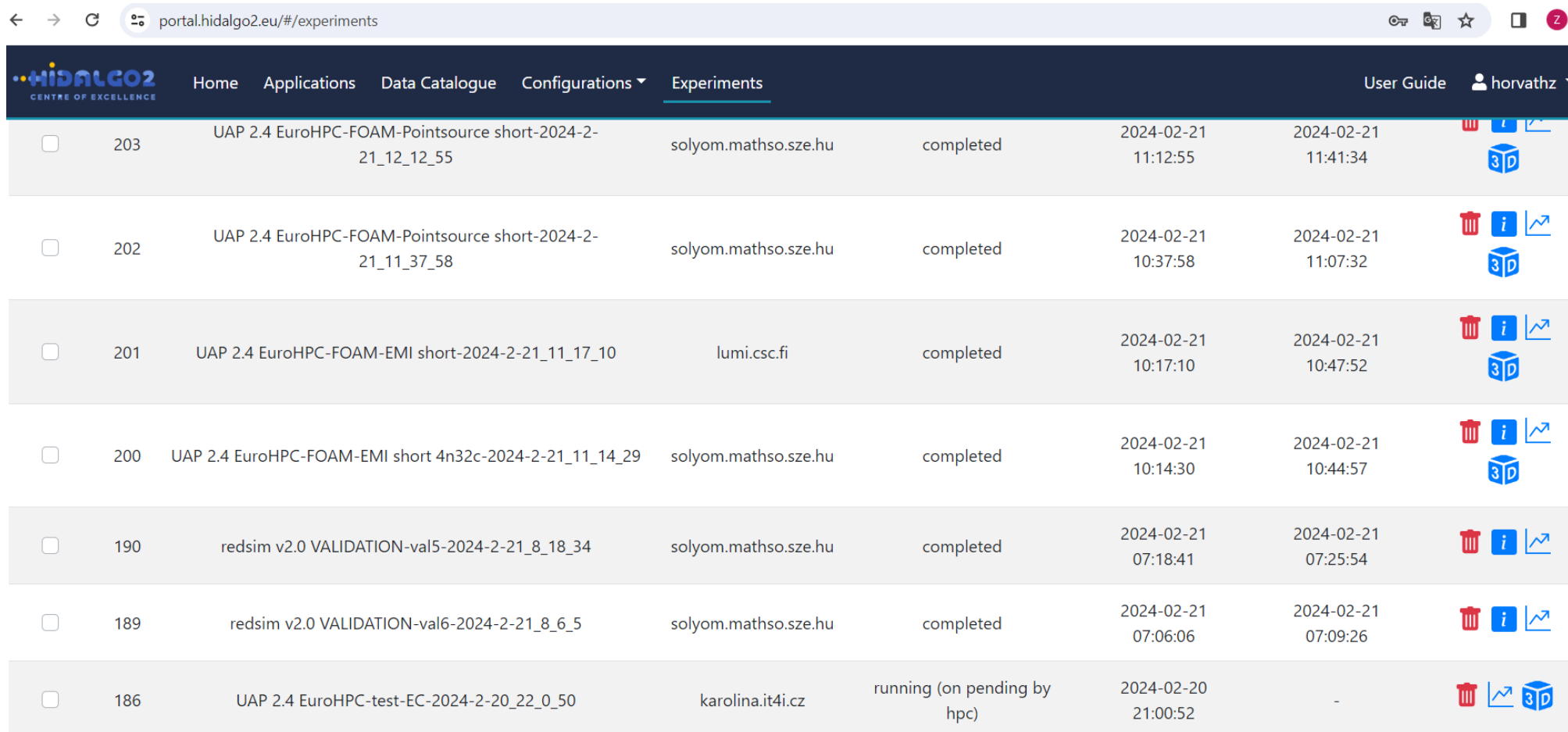
3. Conclusions 3

Real-time HPC + HPDA (Lawson wind comfort)



3. Conclusions 4: UAP from the portal

<https://portal.hidalgo2.eu/>



The screenshot shows the 'Experiments' page of the HIDALGO2 portal. The page has a dark blue header with the HIDALGO2 logo and navigation links: Home, Applications, Data Catalogue, Configurations, and Experiments (which is highlighted). On the right of the header, there is a 'User Guide' link and a user profile 'horvathz'. Below the header is a table of experiments. Each row represents an experiment with columns for a checkbox, ID, Name, User, Status, Start Time, End Time, and Action icons (trash, info, line graph, and 3D model). The experiments listed are numbered 203 down to 186. Experiments 203-189 are in a 'completed' state, while experiment 186 is 'running (on pending by hpc)'. The table is styled with alternating light and dark grey rows.

	ID	Name	User	Status	Start Time	End Time	Action
<input type="checkbox"/>	203	UAP 2.4 EuroHPC-FOAM-Pointsource short-2024-2-21_12_12_55	solyom.mathso.sze.hu	completed	2024-02-21 11:12:55	2024-02-21 11:41:34	Trash, Info, Line Graph, 3D Model
<input type="checkbox"/>	202	UAP 2.4 EuroHPC-FOAM-Pointsource short-2024-2-21_11_37_58	solyom.mathso.sze.hu	completed	2024-02-21 10:37:58	2024-02-21 11:07:32	Trash, Info, Line Graph, 3D Model
<input type="checkbox"/>	201	UAP 2.4 EuroHPC-FOAM-EMI short-2024-2-21_11_17_10	lumi.csc.fi	completed	2024-02-21 10:17:10	2024-02-21 10:47:52	Trash, Info, Line Graph, 3D Model
<input type="checkbox"/>	200	UAP 2.4 EuroHPC-FOAM-EMI short 4n32c-2024-2-21_11_14_29	solyom.mathso.sze.hu	completed	2024-02-21 10:14:30	2024-02-21 10:44:57	Trash, Info, Line Graph, 3D Model
<input type="checkbox"/>	190	redsim v2.0 VALIDATION-val5-2024-2-21_8_18_34	solyom.mathso.sze.hu	completed	2024-02-21 07:18:41	2024-02-21 07:25:54	Trash, Info, Line Graph, 3D Model
<input type="checkbox"/>	189	redsim v2.0 VALIDATION-val6-2024-2-21_8_6_5	solyom.mathso.sze.hu	completed	2024-02-21 07:06:06	2024-02-21 07:09:26	Trash, Info, Line Graph, 3D Model
<input type="checkbox"/>	186	UAP 2.4 EuroHPC-test-EC-2024-2-20_22_0_50	karolina.it4i.cz	running (on pending by hpc)	2024-02-20 21:00:52	-	Trash, Line Graph, 3D Model

3. Further work for the next year

1. RedSim

1. Co-design: Optimization of RedSim for special hardware (e.g. support tetrahedral mesh only) – we expect significantly (maybe 100x) faster code than the current multi-GPU code for polyhedral meshes
2. MPI + Multi-GPU with CUDA and then with OpenCL
3. Develop implicit time-stepping and an operational real-time digital twin for urban airflow

2. More physics

1. A more detailed atmosphere model, coupled with WRF
2. Couple with other HiDALGO2 use-cases
 1. Urban Building Model
 2. waLBerla for small particle propagation
 3. WildFIRES

3. Couple with the Destination Earth platform

4. Collaborations with UAP in EuroHPC: GPU-porting,

5. Services with UAP for the environmental sector and the general industry

Acknowledgments

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Poland, Germany, Spain, Hungary, France, Cyprus under grant agreement number: 101093457.

This publication expresses the opinions of the authors and not necessarily those of the EuroHPC JU and Associated Countries which are not responsible for any use of the information contained in this publication.



**Co-funded by
the European Union**



Disclaimer

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European High Performance Computing Joint Undertaking (JU) and Poland, Germany, Spain, Hungary, France, Cyprus. Neither the European Union nor the granting authority can be held responsible for them.

Further acknowledgments

SZE acknowledges KIFÜ, the Hungarian NCC for providing HPC resources for the GPU code development and benchmarking.

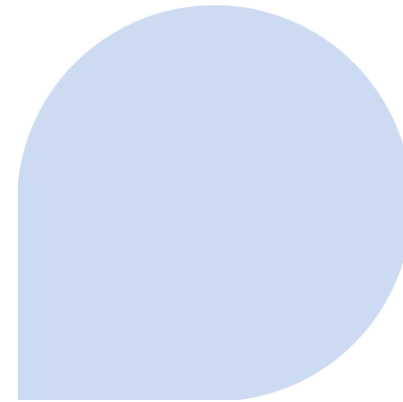




**Thank you for your
attention**

www.hidalgo2.eu

e-mail: office@hidalgo2.eu



Zoltán Horváth, László Környei, Mátyás Constans

Széchenyi István University (SZE)

Egyetem tér 1.
9026 Győr, Hungary

phone: (+36 96) 613-657
e-mail: {horvathz, laszlo.kornyei,
matyas.constans}@math.sze.hu