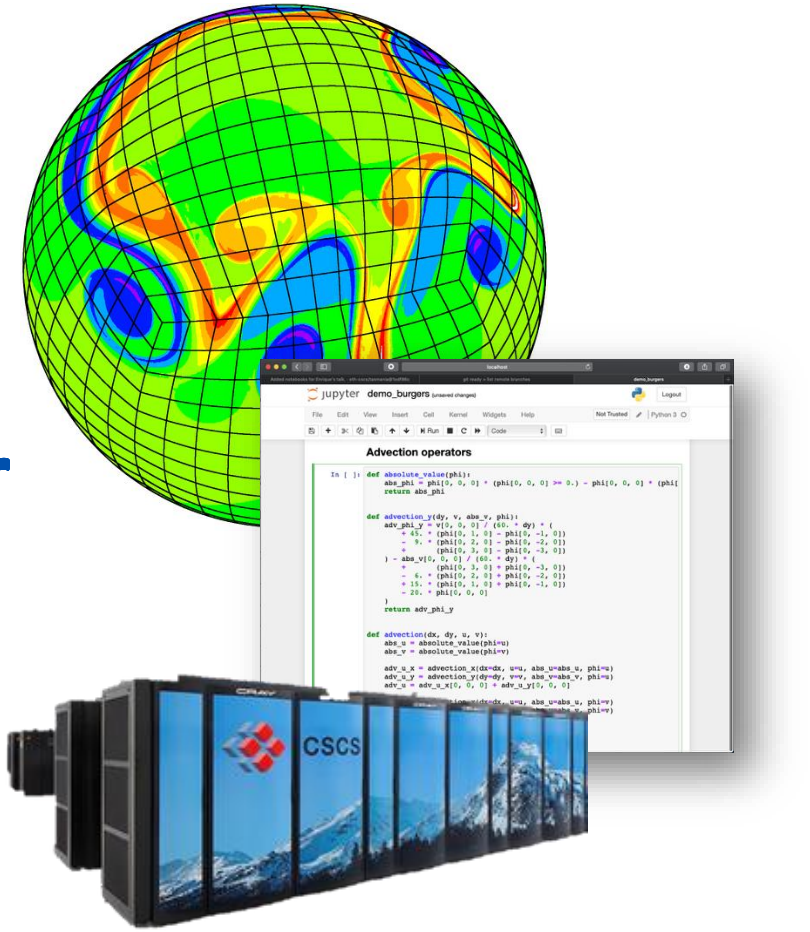
High Performance Computing for Weather and Climate (HPC4WC)

Content: Wrapup

Lecturers: Oliver Fuhrer, Christophe Müller

Block course 701-1270-00L

Summer 2025

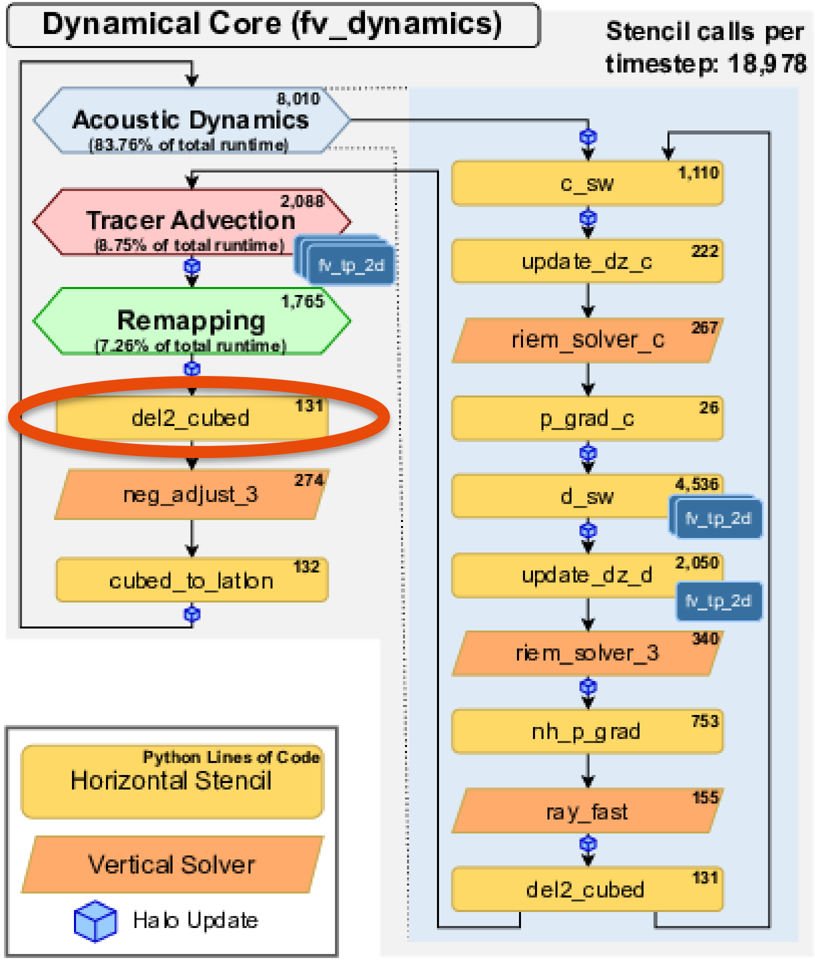


Pace: GT4Py based FV3

GFDL Finite-Volume Cubed-Sphere Dynamical Core (FV3)

Finite volume transport on a cubed sphere grid

- Integrated into several models, including
  - Operational weather models (Global Forecast System)
  - Next Generation Global Prediction System



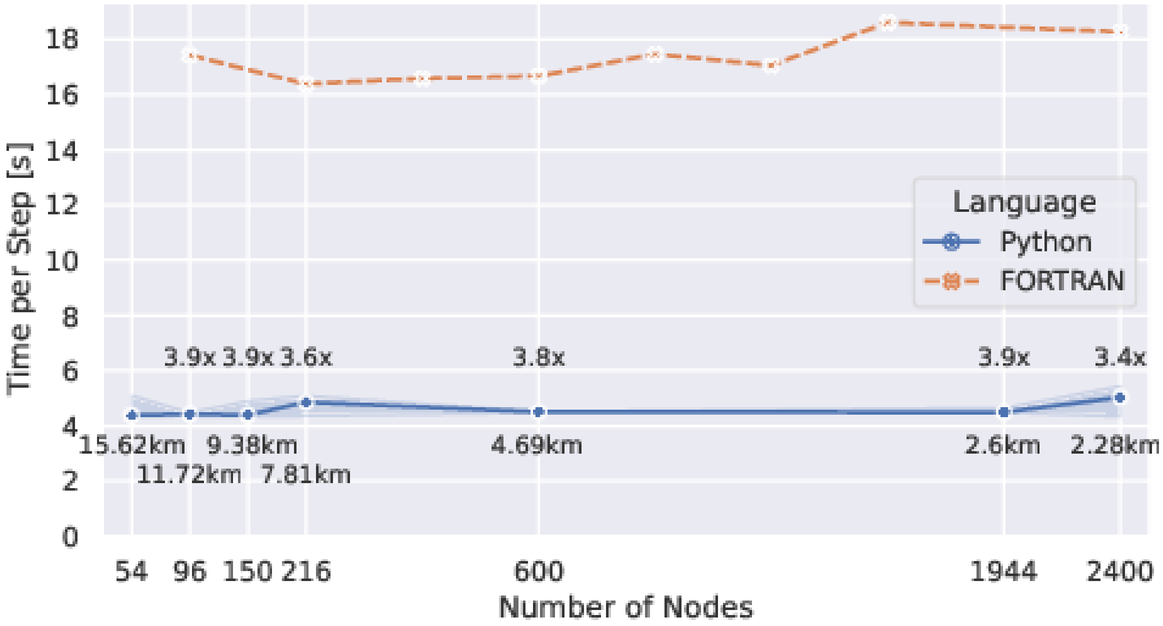
The Pace Model

Full program optimization

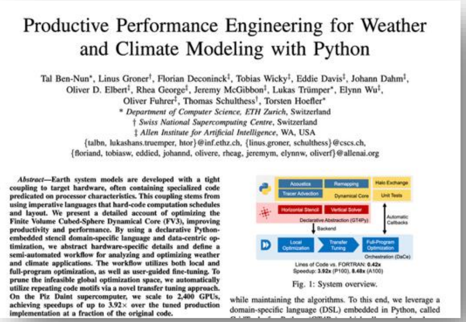
DSL coverage of all main numerical computation

Custom code for halo updates

New DSL concepts for FV3-specific motifs



Separation of Concerns



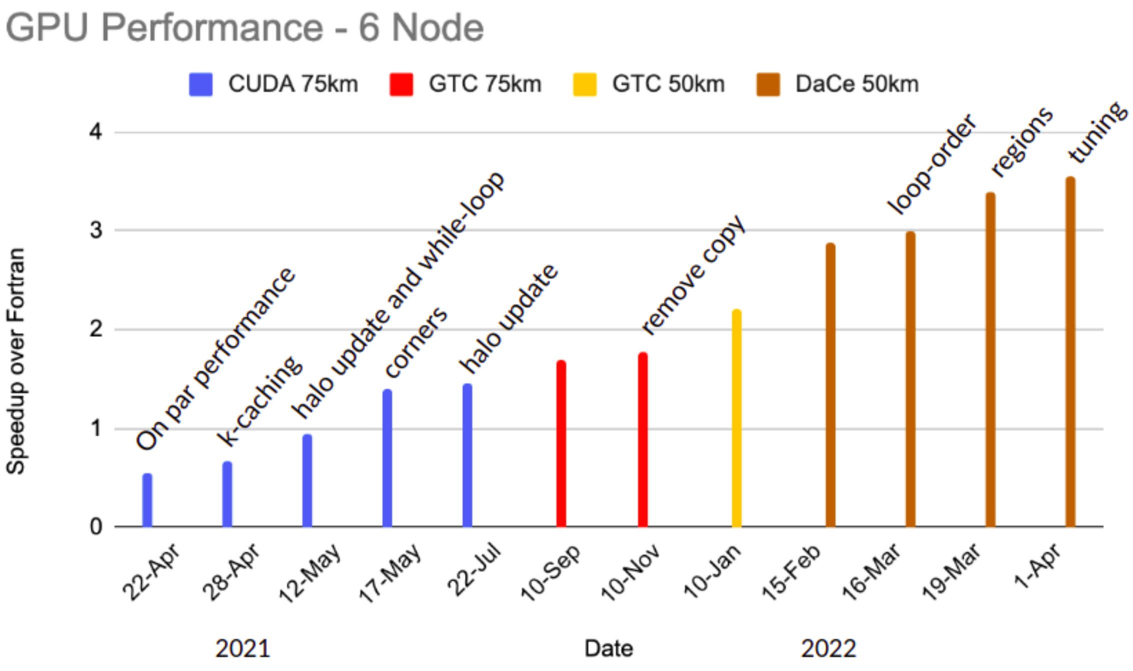
Productive Performance Engineering for Weather and Climate Modeling with Python

Tal Ben-Nun*, Linus Groner†, Florian Deconinck‡, Tobias Wicky‡, Eddie Davis‡, Johann Dahm‡, Oliver D. Elbert‡, Rhea George‡, Jeremy McGibbon‡, Lukas Trümper*, Elynn Wu‡, Oliver Fuhrer‡, Thomas Schulthess†, Torsten Hoefler*
* *Department of Computer Science, ETH Zurich*, Switzerland
† *Swiss National Supercomputing Centre*, Switzerland
‡ *Allen Institute for Artificial Intelligence*, WA, USA
{talbn, lukashans.truemper, htor}@inf.ethz.ch, {linus.groner, schulthess}@cscs.ch,
{floriand, tobiasw, eddied, johannd, olivere, rheag, jeremym, elynnw, oliverf}@allenai.org

***Abstract*—Earth system models are developed with a tight coupling to target hardware, often containing specialized code predicated on processor characteristics. This coupling stems from using imperative languages that hard-code computation schedules and layout. We present a detailed account of optimizing the Finite Volume Cubed-Sphere Dynamical Core (FV3), improving productivity and performance. By using a declarative Python-embedded stencil domain-specific language and data-centric optimization, we abstract hardware-specific details and define a semi-automated workflow for analyzing and optimizing weather and climate applications. The workflow utilizes both local and full-program optimization, as well as user-guided fine-tuning. To prune the infeasible global optimization space, we automatically utilize repeating code motifs via a novel transfer tuning approach. On the Piz Daint supercomputer, we scale to 2,400 GPUs, achieving speedups of up to 3.92× over the tuned production implementation at a fraction of the original code.**

Fig. 1: System overview.

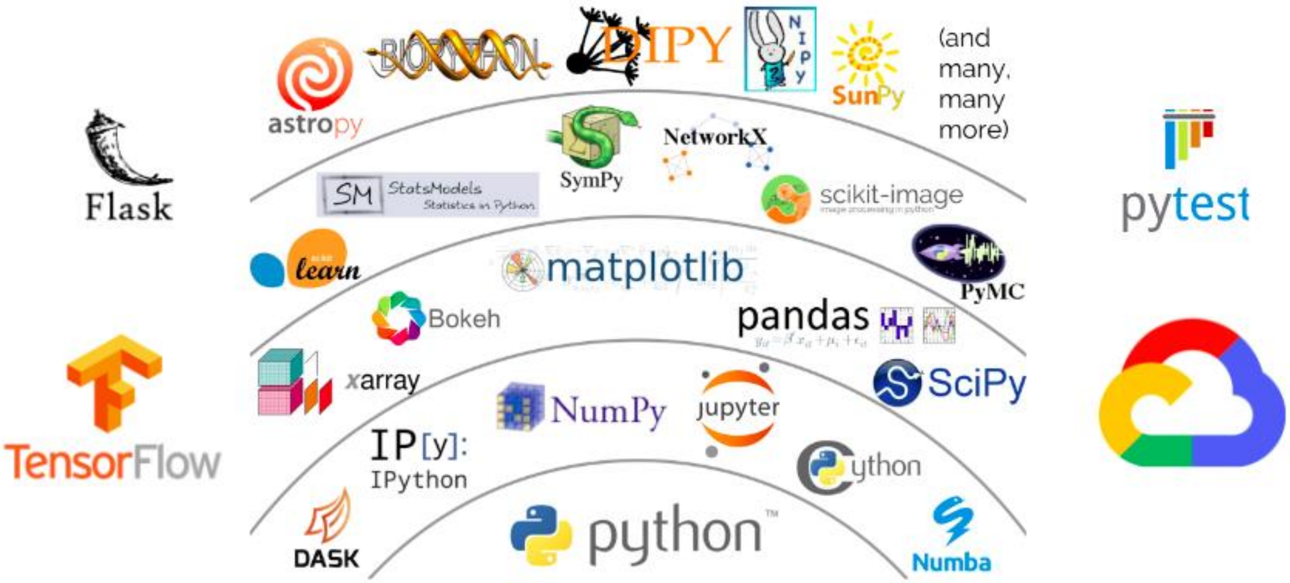
while maintaining the algorithms. To this end, we leverage a domain-specific language (DSL) embedded in Python, called



7x improvement

Leveraging the Python ecosystem

The rich python ecosystem is valuable – new options for development



Credit: Jake VanderPlas, "The Unexpected Effectiveness of Python in Science", PyCon 2017

Example: Integration of ML emulator

```
class Physics:
...
prepare_microphysics(physics_state)
microph_state = physics_state.microphysics
microphysics(microph_state)
```

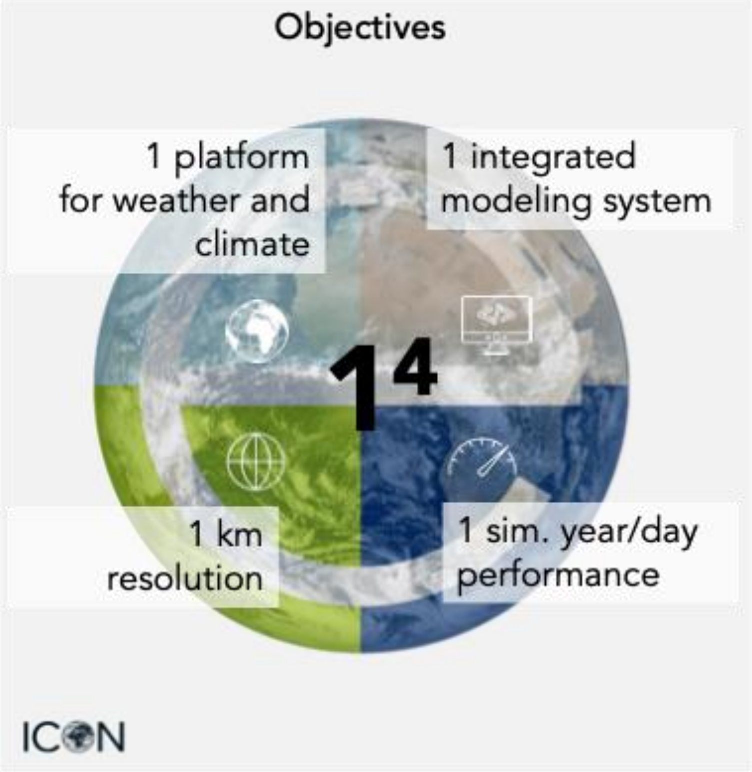
GT4Py stencil-based

```
emulation_model = tf.keras.models.load_model("model.tf")
```

ML-based microphysics

```
emulation_dict = prepare_emulation_data(physics_state.microphysics)
predictions = emulation_model(emulation_dict)
model_outputs = unpack_predictions(predictions, emulation_model.output_names, …)
```

EXCLAIM Project

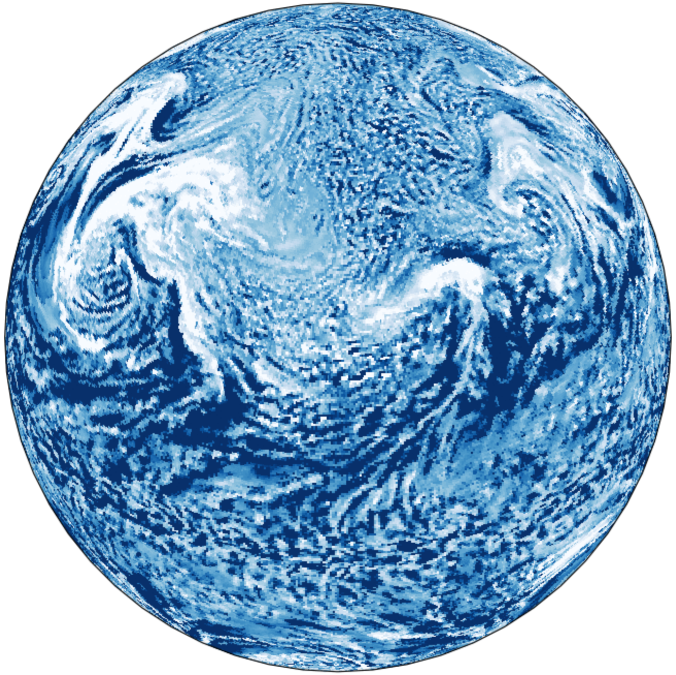


Six year (2021-27) open ETH project aiming to develop an ICON infrastructure capable of running km-scale climate simulations

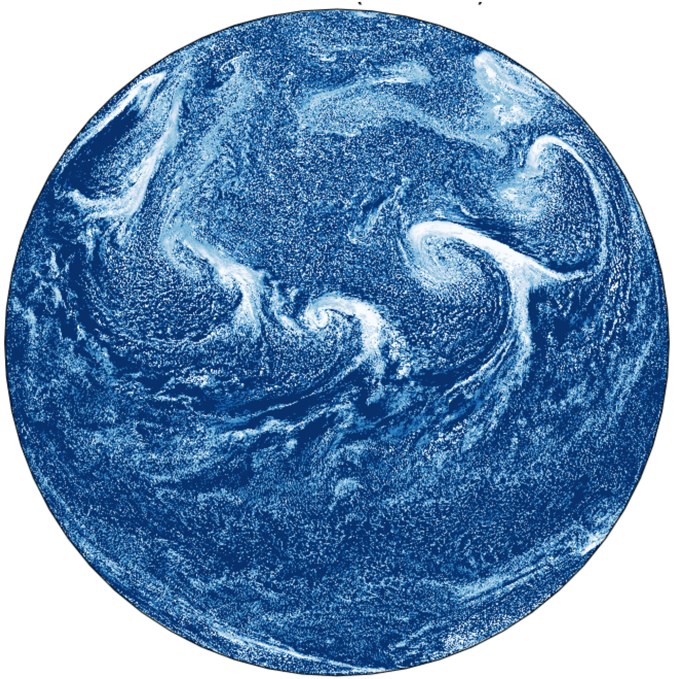
EXCLAIM: Rewriting ICON in GT4Py

Proof of concept: Aquaplanet with GT4Py

Visualization of low level clouds

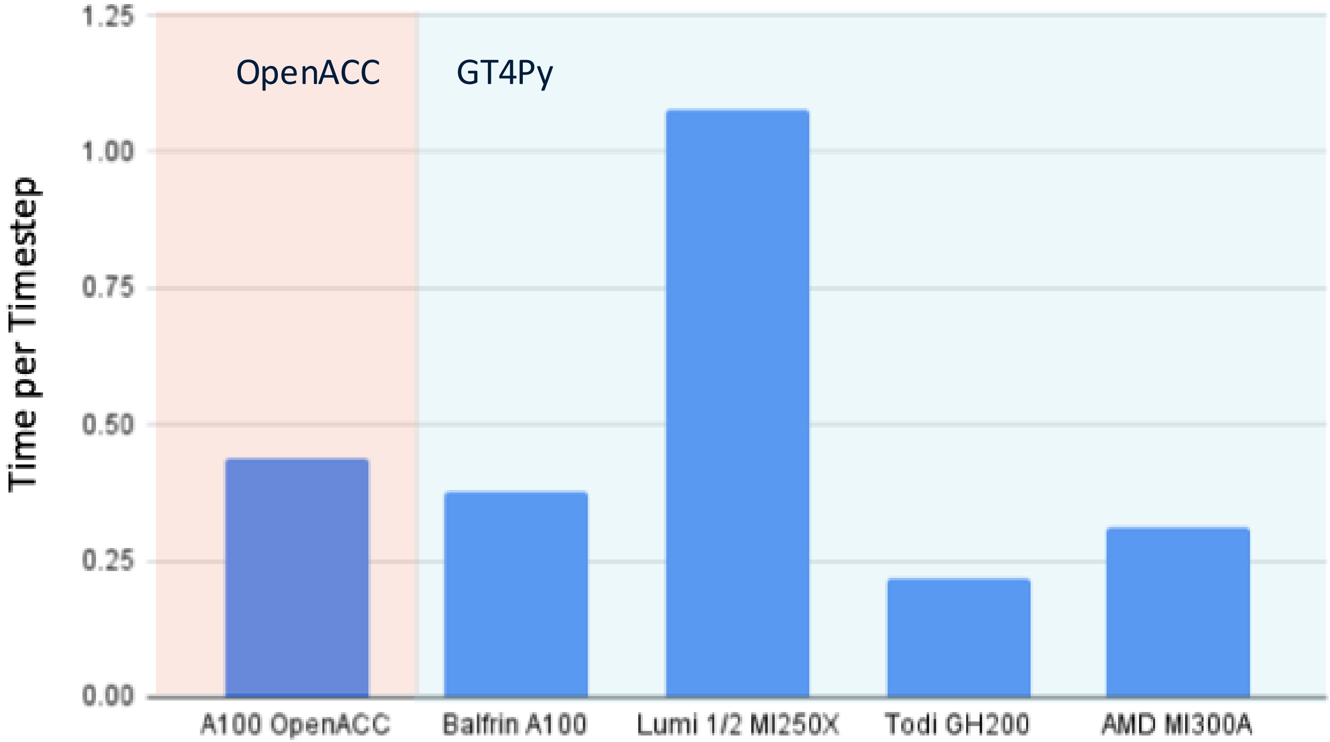


ICON OpenACC R2B6 (40 km)



ICON GT4Py R2B9 (5 km)

Performance & Portability



Supercomputer Architecture

(Numbers are for Alps and vary from system to system)

Day 1

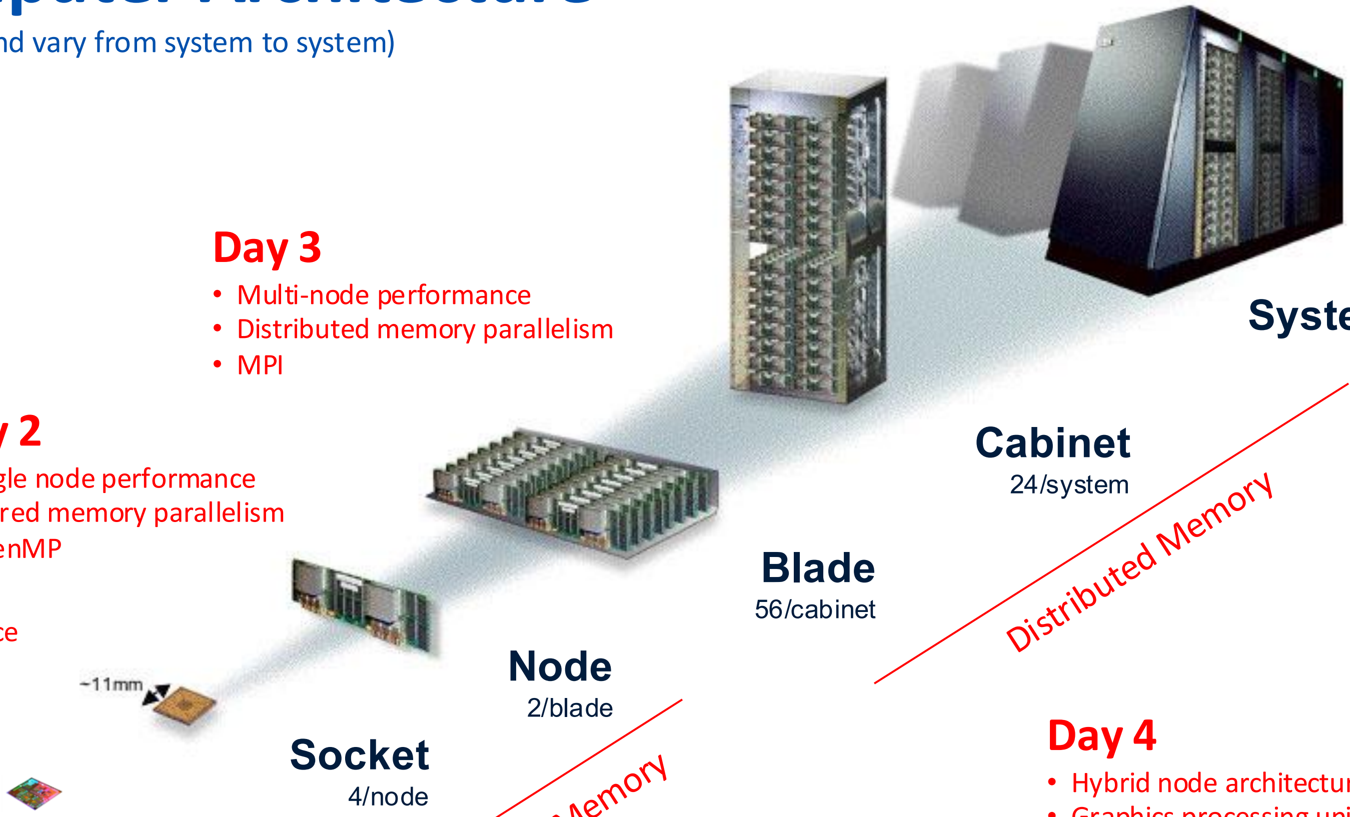
- Single core performance
- Caches

Day 2

- Single node performance
- Shared memory parallelism
- OpenMP

Day 3

- Multi-node performance
- Distributed memory parallelism
- MPI



Day 4

- Hybrid node architectures
- Graphics processing units (GPUs)
- CuPy

Weather and Climate Models

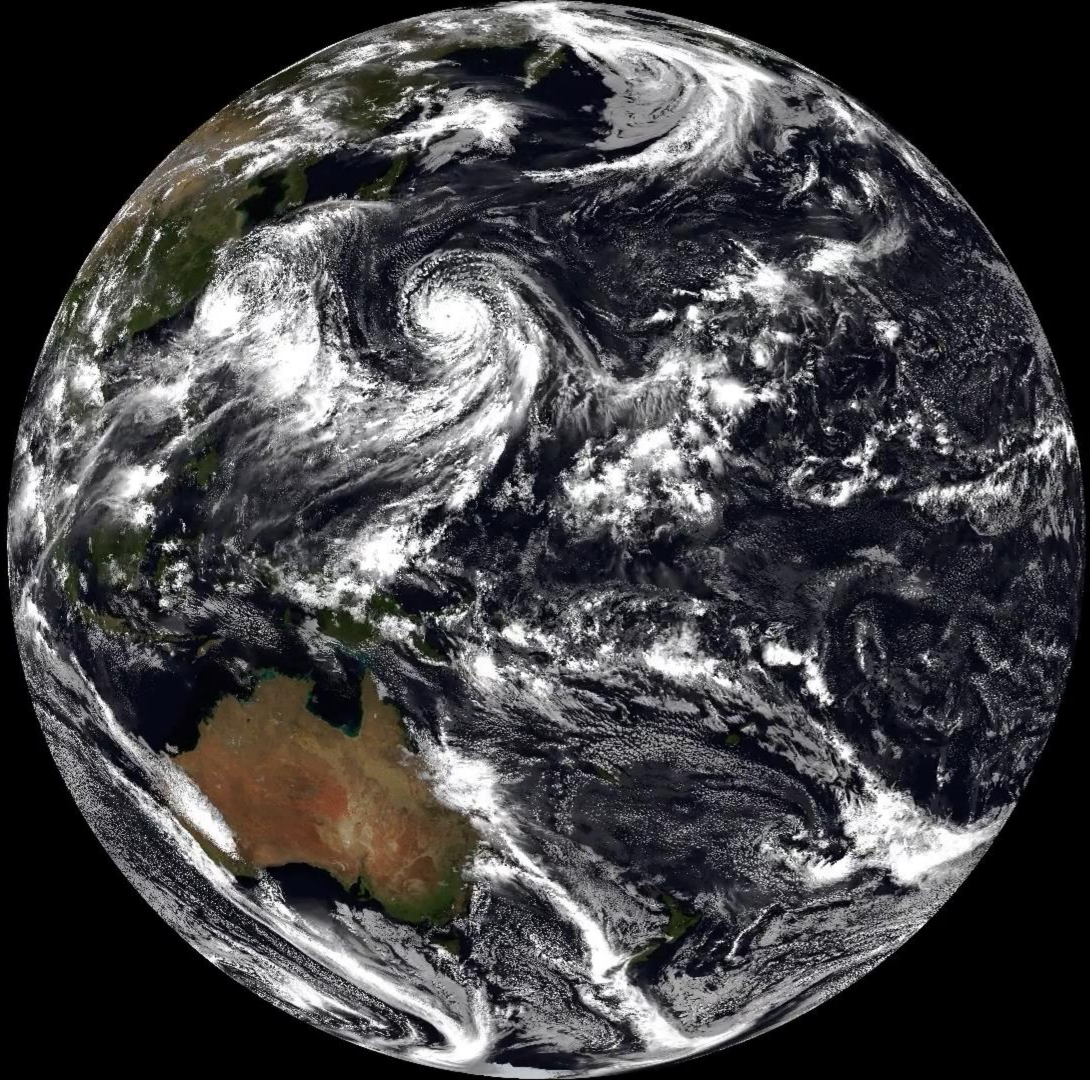
ICON, COSMO, WRF, LFRic, SHiELD, GEOS, … all are

- large, Fortran-dominated code-bases
- stencil-dominated and memory bandwidth bound
- optimizing cache efficiency using blocking, tiling, loop fusion, …
- using MPI to decompose the computational domain along xy
- using OpenMP to parallelize using threads
- using OpenACC compiler directives to leverage GPUs
- run on a range of different supercomputing architectures
- exploring or already using domain-specific languages

You can apply everything you've learnt with stencil2d!

Goals of Course

- Understand high performance computing concepts relevant for weather and climate simulations
- Able to work with weather and climate simulation codes that run on large supercomputers


2016-08-11 18:00Z
258 Forecast Hours
FV3 3km
Visualization
Xi Chen@FV3 team