Challenges of NICAM toward the exascale era

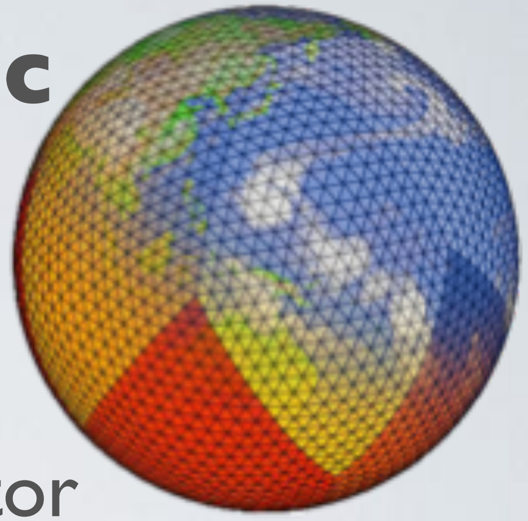
Hisashi YASHIRO
RIKEN Center for Computational Science
Kobe, Japan

and

NICAM team

NICAM

Non-hydrostatic **I**cosahedral **A**tmospheric **M**odel (NICAM)



- Development was started since 2000
  Tomita and Satoh (2005), Satoh et al. (2008, 2014)
- First global dx=3.5km run in 2004 using the Earth Simulator
  Tomita et al. (2005), Miura et al. (2007, Science)
- First global dx=0.87km run in 2012 using the K computer
  Miyamoto et al. (2013, 2015), Kajikawa et al. (2016)

Prof. Satoh (AORI, Tokyo univ.)

Dr. Tomita (RIKEN)

- Main target : high-resolution simulation without convection parameterization, without lateral boundary
- Compressive, non-hydrostatic equations are solved using finite volume method on the icosahedral grid
  - Most part is written by Fortran90
  - ~50 users, ~10 active developers

NICAM and Supercomputers

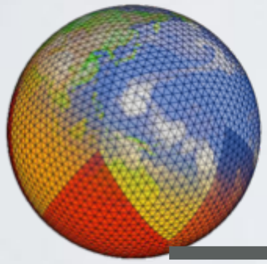
JAMSTEC
- Earth Simulator 36TFLOPS (2002)
- Earth Simulator 2 122TFLOPS
- Earth Simulator 3 ~1PFLOPS

RIKEN
- The K computer 10PFLOPS
- Post-K

The U. of Tokyo & Tsukuba U.
- T2K-Tsukuba 100TFLOPS
- Oakleaf-FX 1PFLOPS
- Oakforest PACS 14PFLOPS

JAXA
- SORA-MA 3PFLOPS

Tokyo Tech.
- TSUBAME2.5 3PFLOPS
- TSUBAME3 8PFLOPS



2000 2004 2011 2018

3.5km MJO sim.

870m run
14km 30yr AMIP run

International collaborations

- CMIP6/HighResMIP
- RCEMIP
- DYAMOND project
- SPPEXA/AIMES & GridTools-NICAM

Current development target **from the viewpoint of HPC**

Cloud-permitting (14km~3.5km) simulation + X

+ Eddy resolving ocean
+ Ensemble data assimilation system
+ Aerosol & chemistry

Keywords

- Approximate computing
- Data centric design
- Performance portability

Today's talk

Our efforts on the petascale computing era

- Practical cases of the extreme scale global simulation

Our efforts toward the exascale computing era

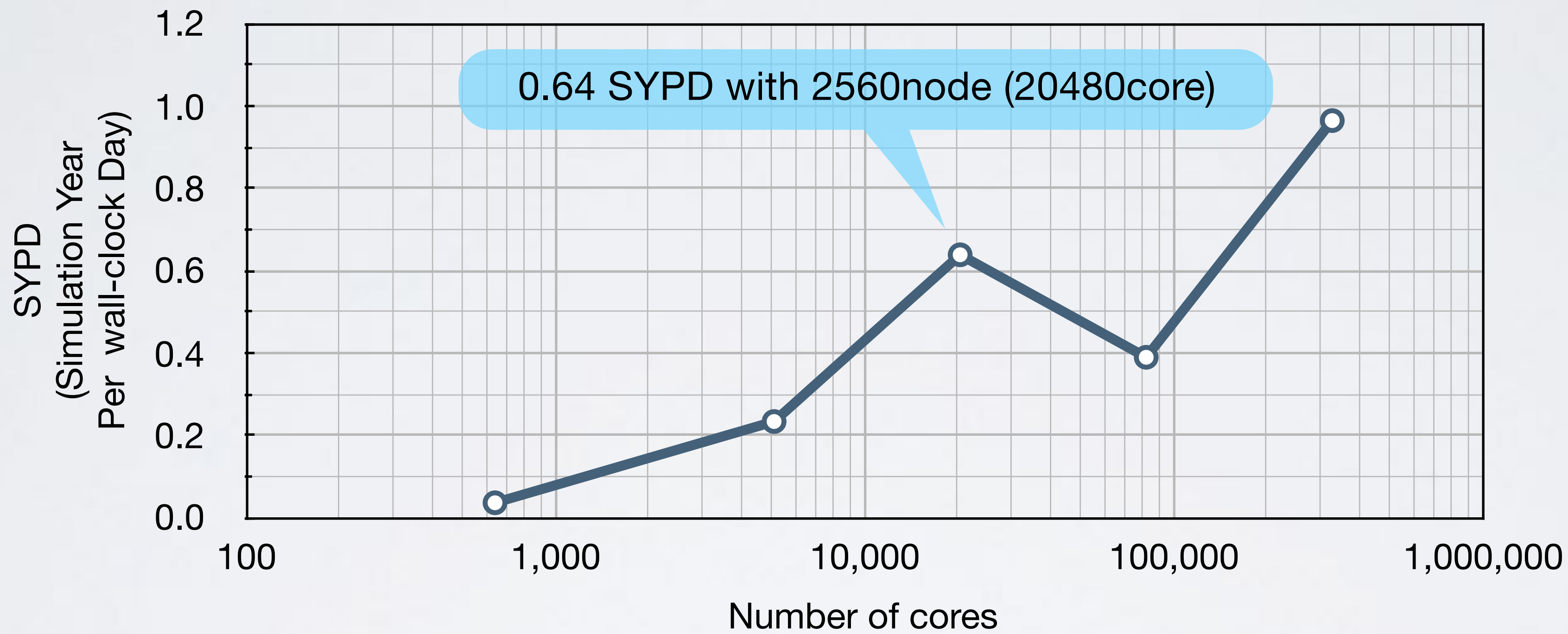
- Post-K project
- SPPEXA/AIMES & GridTools-NICAM

Efforts on the petascale computing era

NICAM on the K computer

Performance optimization (Yashiro et al., 2017, PASC'16)

- Time-consuming parts are removed: zero-filling, copying, lots of intermediate arrays, "if" in the loop
- Good weak scalability up to 81,920 nodes x 8 threads with 0.9PFLOPS
- Good strong scalability: with 14km horizontal, 38layers:



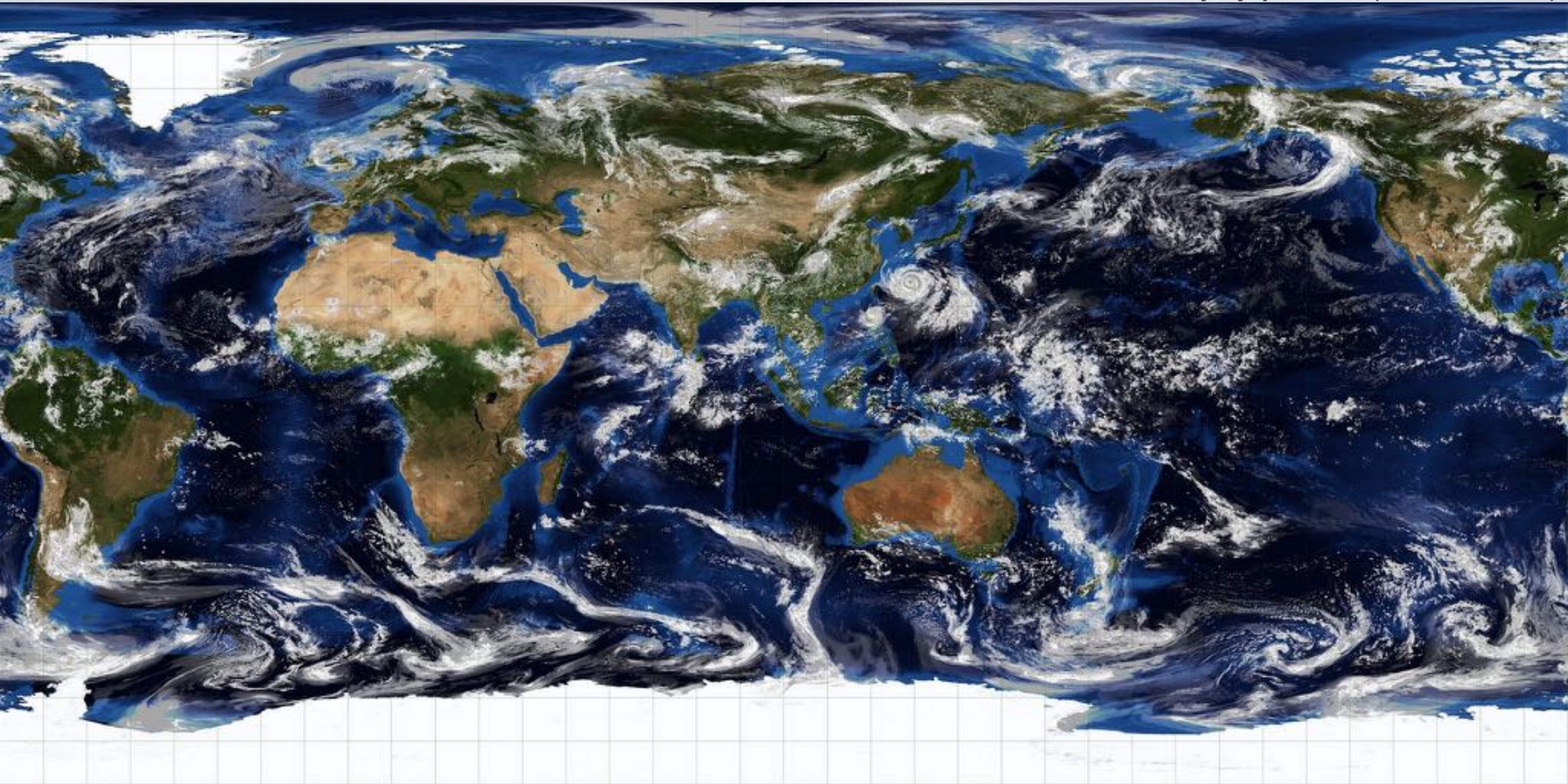
The cost ranking cannot find the time-consuming part

- Tiny "time eaters" are hiding everywhere in the code
- It is necessary to collect the information about the elapsed time, the memory throughput, and the number of floating operations for each loop nests

The first global sub-km weather simulation on the K computer

Miyamoto et al., 2013, GRL

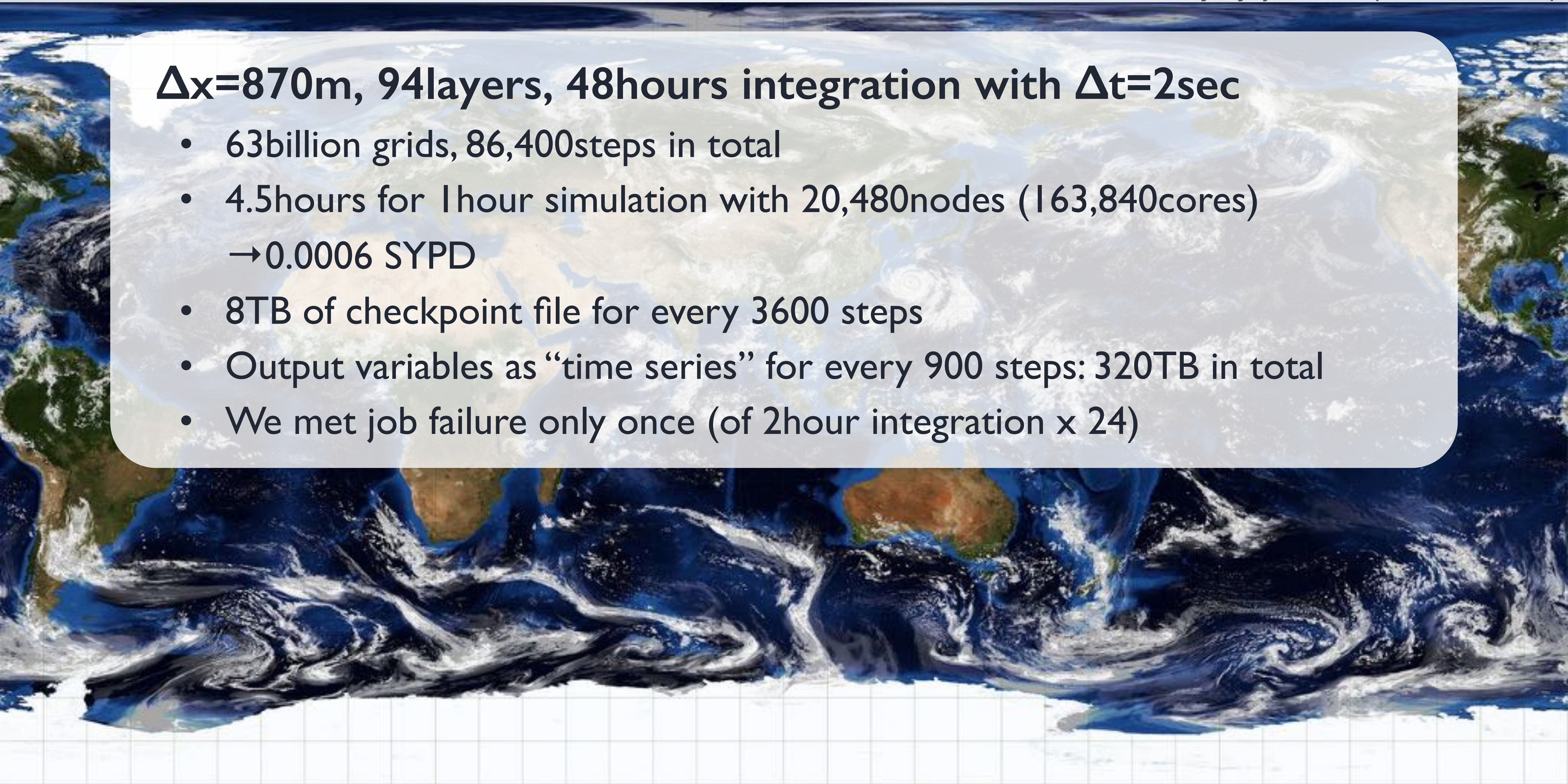
Visualized by Ryuji Yoshida(RIKEN,Kobe U.)



The first global sub-km weather simulation on the K computer

Miyamoto et al., 2013, GRL

Visualized by Ryuji Yoshida(RIKEN,Kobe U.)



Δx=870m, 94layers, 48hours integration with Δt=2sec

- 63billion grids, 86,400steps in total
- 4.5hours for 1hour simulation with 20,480nodes (163,840cores)
  →0.0006 SYPD
- 8TB of checkpoint file for every 3600 steps
- Output variables as "time series" for every 900 steps: 320TB in total
- We met job failure only once (of 2hour integration x 24)

The first global sub-km weather simulation on the K computer

Miyamoto et al., 2013, GRL

Visualized by Ryuji Yoshida(RIKEN,Kobe U.)

Δx=870m, 94layers, 48hours integration with Δt=2sec

- 63billion grids, 86,400steps in total
- 4.5hours for 1hour simulation with 20,480nodes (163,840cores)
  →0.0006 SYPD
- 8TB of checkpoint file for every 3600 steps
- Output variables as "time series" for every 900 steps: 320TB in total
- We met job failure only once (of 2hour integration x 24)

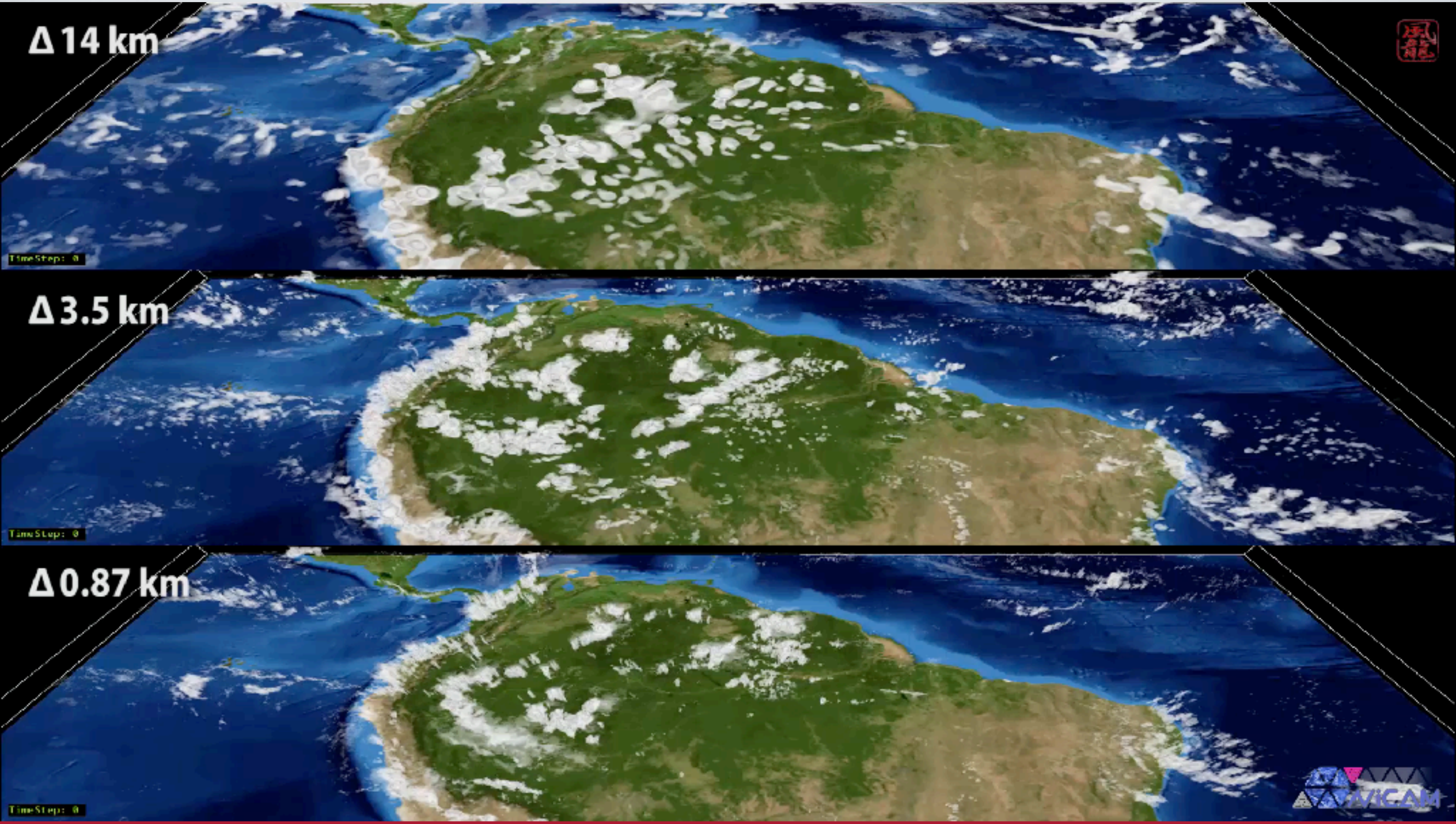
Our simulation didn't have any problems in I/O: Why?

- File staging : isolated from the crowded global file system
  - A different storage disk is assigned to each MPI rank
- I/O node : we don't have to wait writing due to the large buffer
- Distributed file I/O : each MPI rank writes the files

The diurnal cycle of precipitation over land in the tropics

Yashiro et al., 2016, SOLA

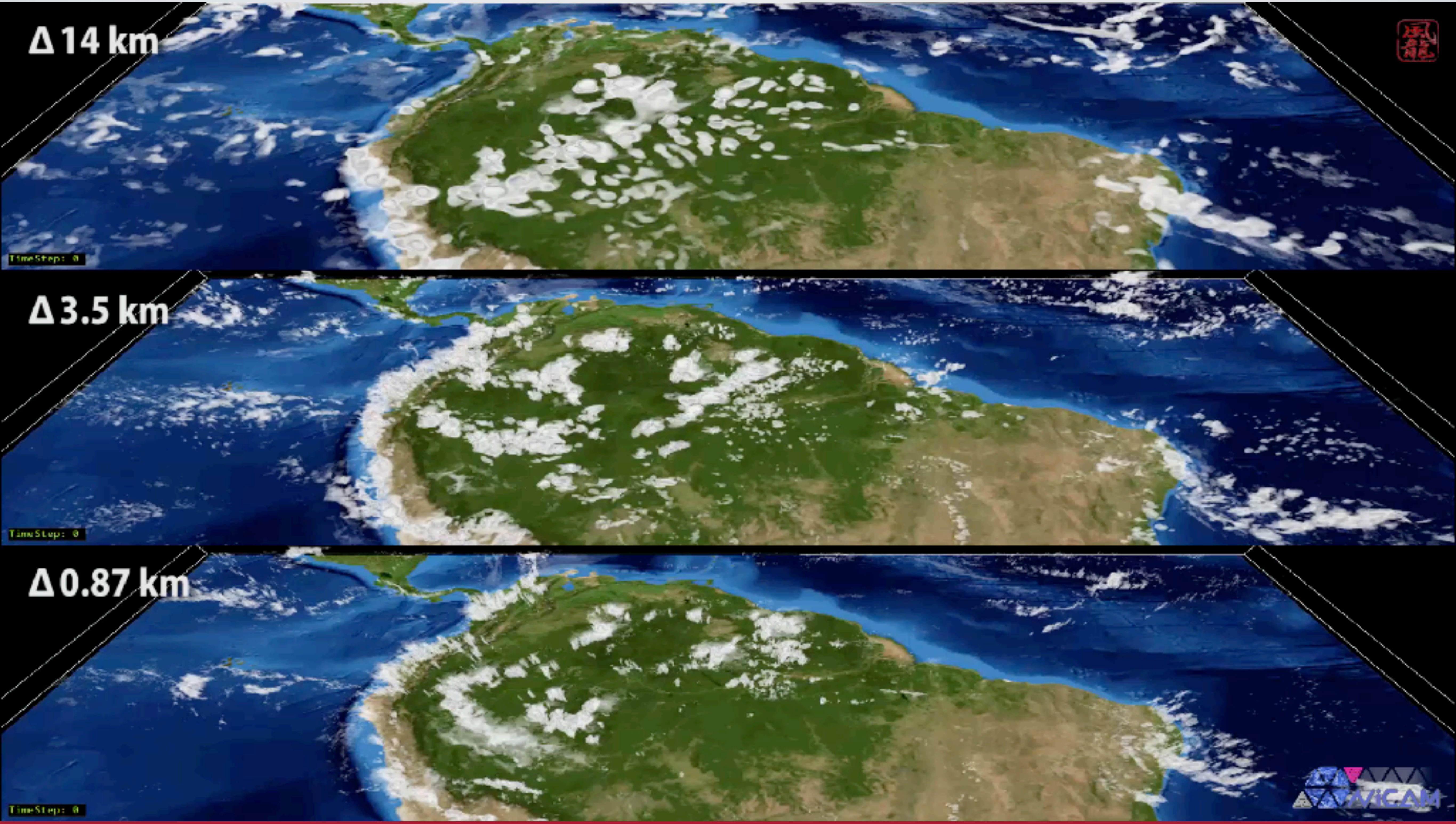
Visualized by Ryuji Yoshida(RIKEN,Kobe U.)



The diurnal cycle of precipitation over land in the tropics

Yashiro et al., 2016, SOLA

Visualized by Ryuji Yoshida(RIKEN,Kobe U.)



'Big' data analysis in the weather/climate study

Every 30min Snapshot for 0.87km run: ~3TB
x 48 steps (for last 1day output) = 160TB

↓

Grid remapping
from icosahedral
to latitude-longitude

2 months on the post-process cluster

↓

Analysis on
latitude-longitude grid

2 months on the post-process cluster

'Big' data analysis in the weather/climate study

Every 30min Snapshot for 0.87km run: ~3TB
x 48 steps (for last 1day output) = 160TB

Grid remapping
from icosahedral
to latitude-longitude

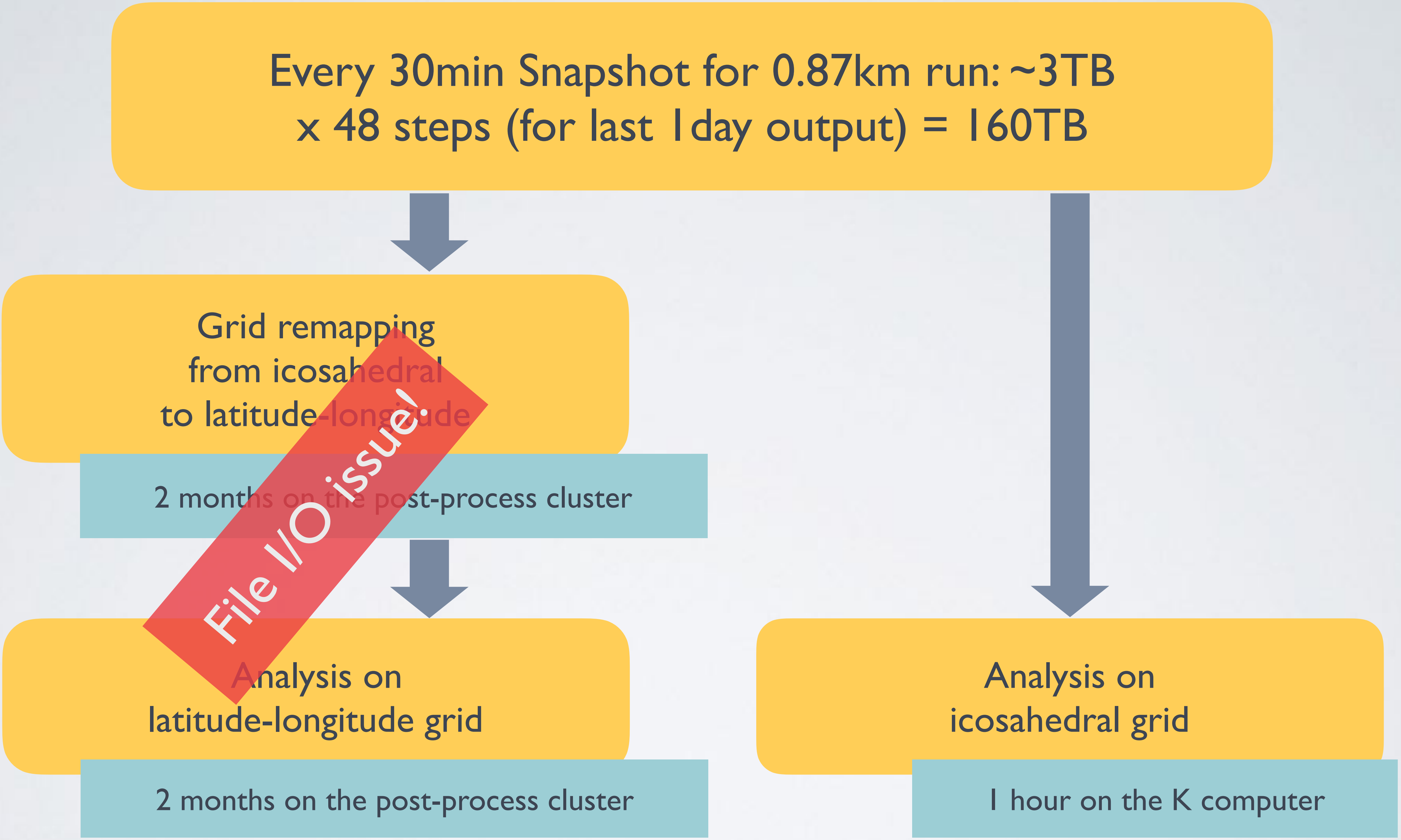
2 months on the post-process cluster

File I/O issue!

Analysis on
latitude-longitude grid

2 months on the post-process cluster

'Big' data analysis in the weather/climate study



NICAM on KNL cluster

Oakforest-PACS: the largest KNL cluster in Japan

- 8208 nodes, Intel Xeon Phi 7250

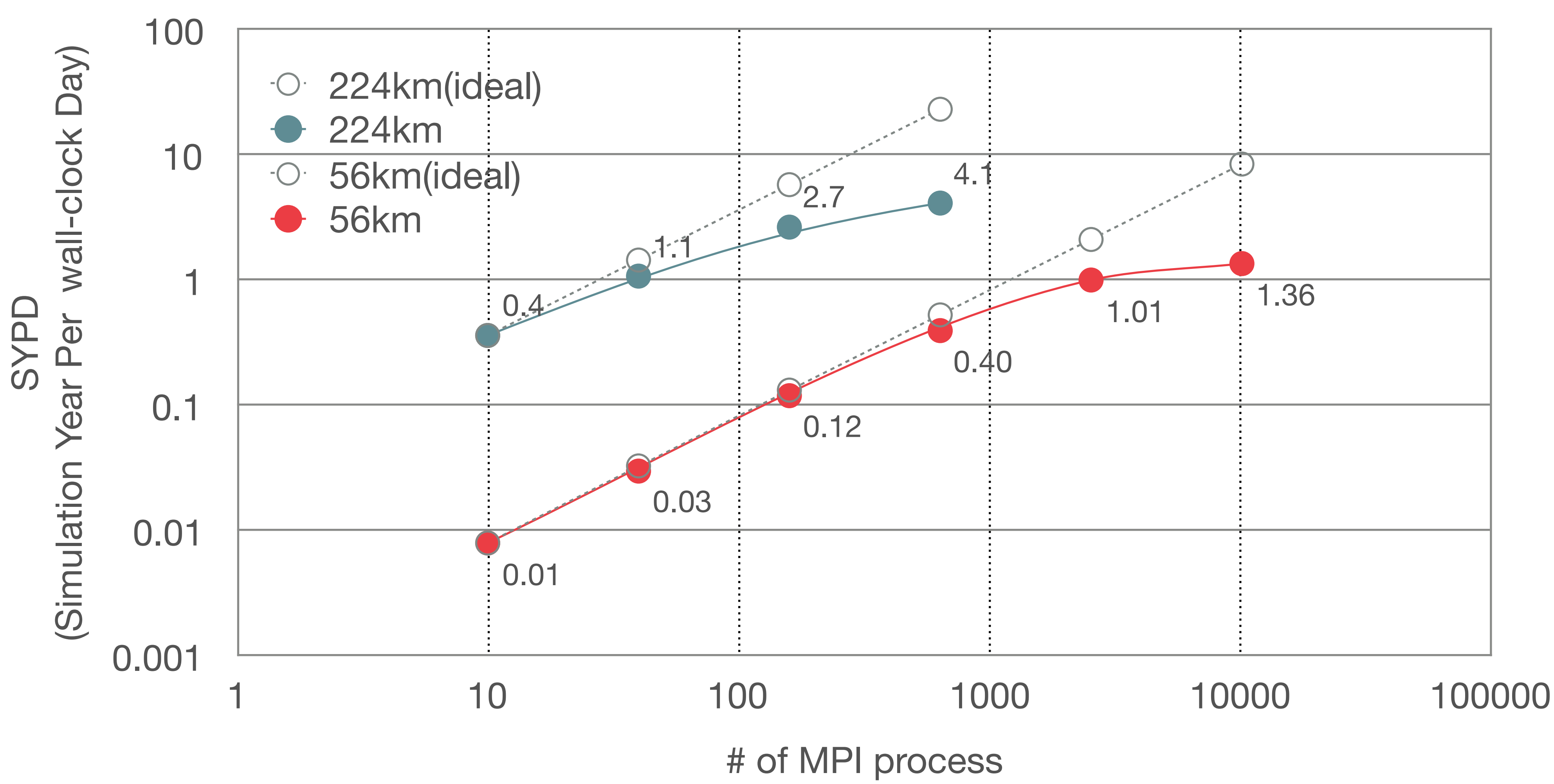
Optimization on Oakforest-PACS

- This was the first time that we inserted OpenMP directives in the NICAM code
- Fine-grained thread parallelization → **worse** thread scalability
  :The cost of fork/join is large on the KNL
- The flat-MPI execution shows fairly good performance
  - The Omni-Path is good at handling many small p2p communications?
  - I/O did not become critical issue

NICAM on KNL cluster

Weak scaling w/ low resolution run on Oakforest-PACS

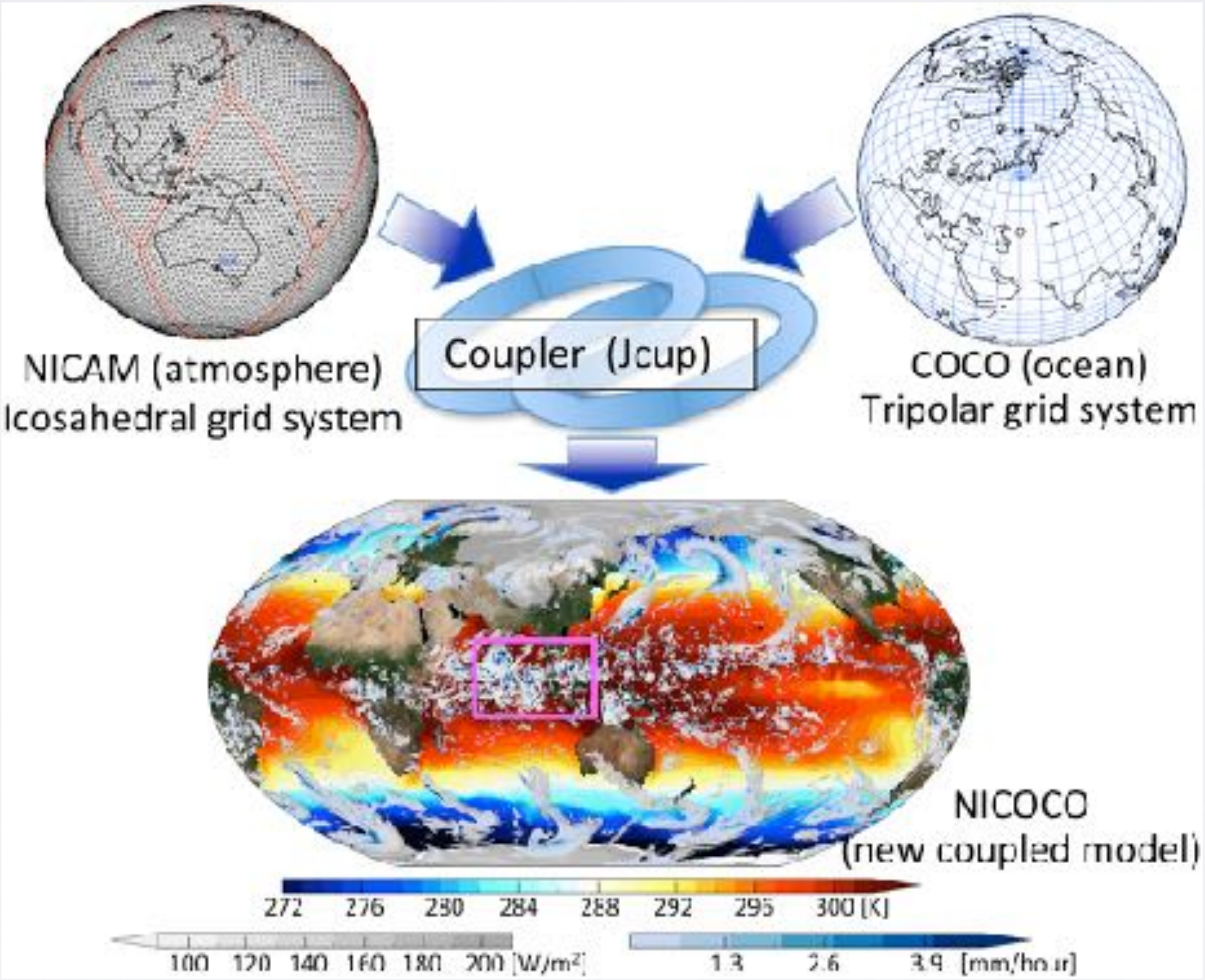
- Result shows good scalability when the grid point size per process is enough
- More process → less grid per proc. → lack of parallelism



NICAM on KNL cluster

Atmosphere-Ocean coupling studies on Oakforest-PACS

- A whole-year simulation with 14km atmosphere (NICAM) + 0.1deg ocean (COCO)
- Atmosphere: 0.3 billion grid points, 0.8 million steps, 10240 MPI processes
- Ocean: 0.6 billion grid points, 0.2 million steps, 1600 MPI processes



Picture from Miyakawa et al., 2017, GRL

Efforts toward the exascale computing era

From "K" to "post K"

Post K is…

- Next Japanese flagship supercomputer
- Manycore architecture
  - ARM v8 + Scalable Vector Extension
  - No accelerators
- 6-D mesh/torus network

- Designed for general purpose
  - The proxy applications are selected from nine priority research fields
  - ➡ System-Application co-design

Co-design in post-K project

Estimation of computational performance

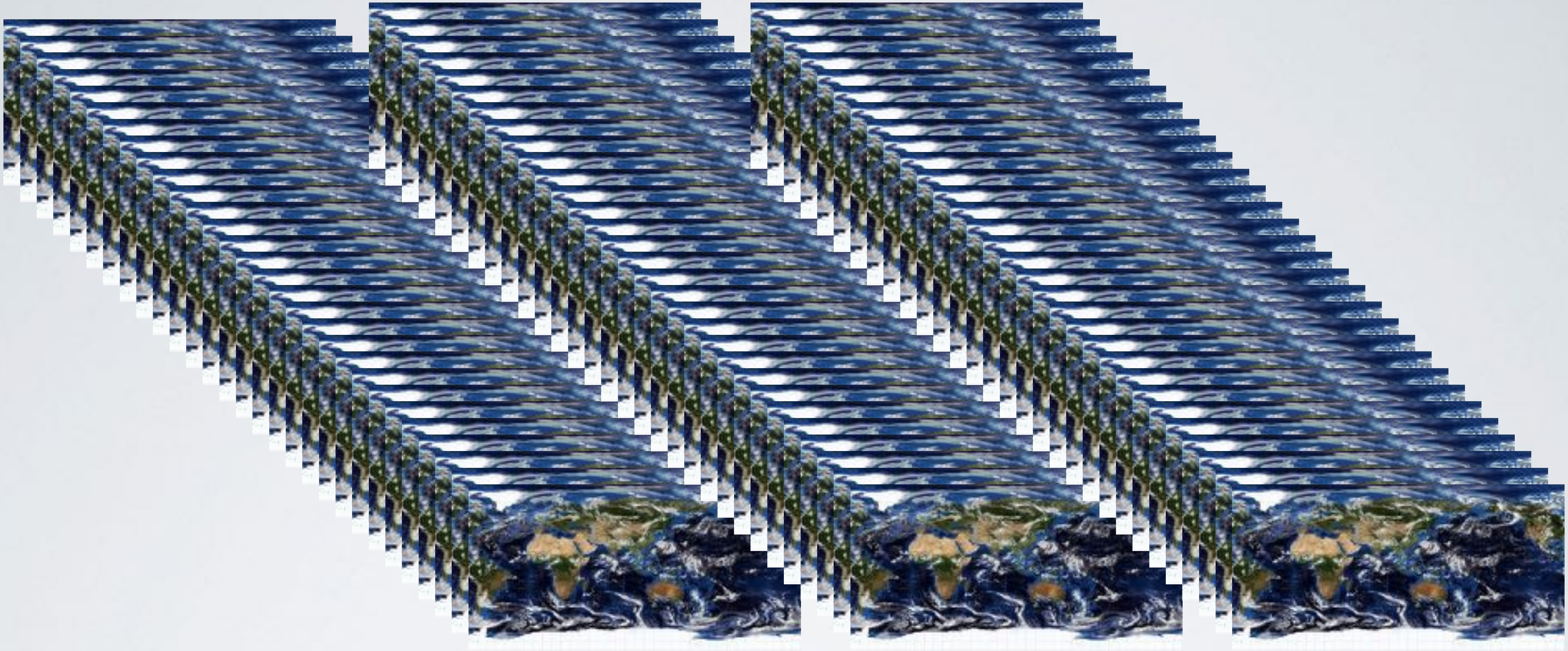
- ~10 kernels are extracted from dycore & physics
- Estimation using a performance profiler of FX100 and parameter of post-K
- Evaluation using post-K software simulator

- Basic design phase (~2015): Contribute to the decision of machine parameter
- Detailed design phase (2015~): Contribute to the compiler development

Change of application side

- All of things beyond the subroutine-level optimization
  : refactoring, data layout, algorithm, framework, etc.
- Optimization with the risk of deterioration of simulation results: precision

Grand challenge on the post-K computer



- 3.5km-mesh, 100 layers x 1000 members
  - It takes **2 weeks** for 1 DA cycle using full node of the K computer
  - **3 PByte** of the data will be exchanged between NICAM and LETKF for 1 DA cycle
- 1 million observation including satellite data

New design of the DA system

Data centric design

Problem:
Huge amount of data exchange/transpose occurs between the weather model and the ensemble DA system

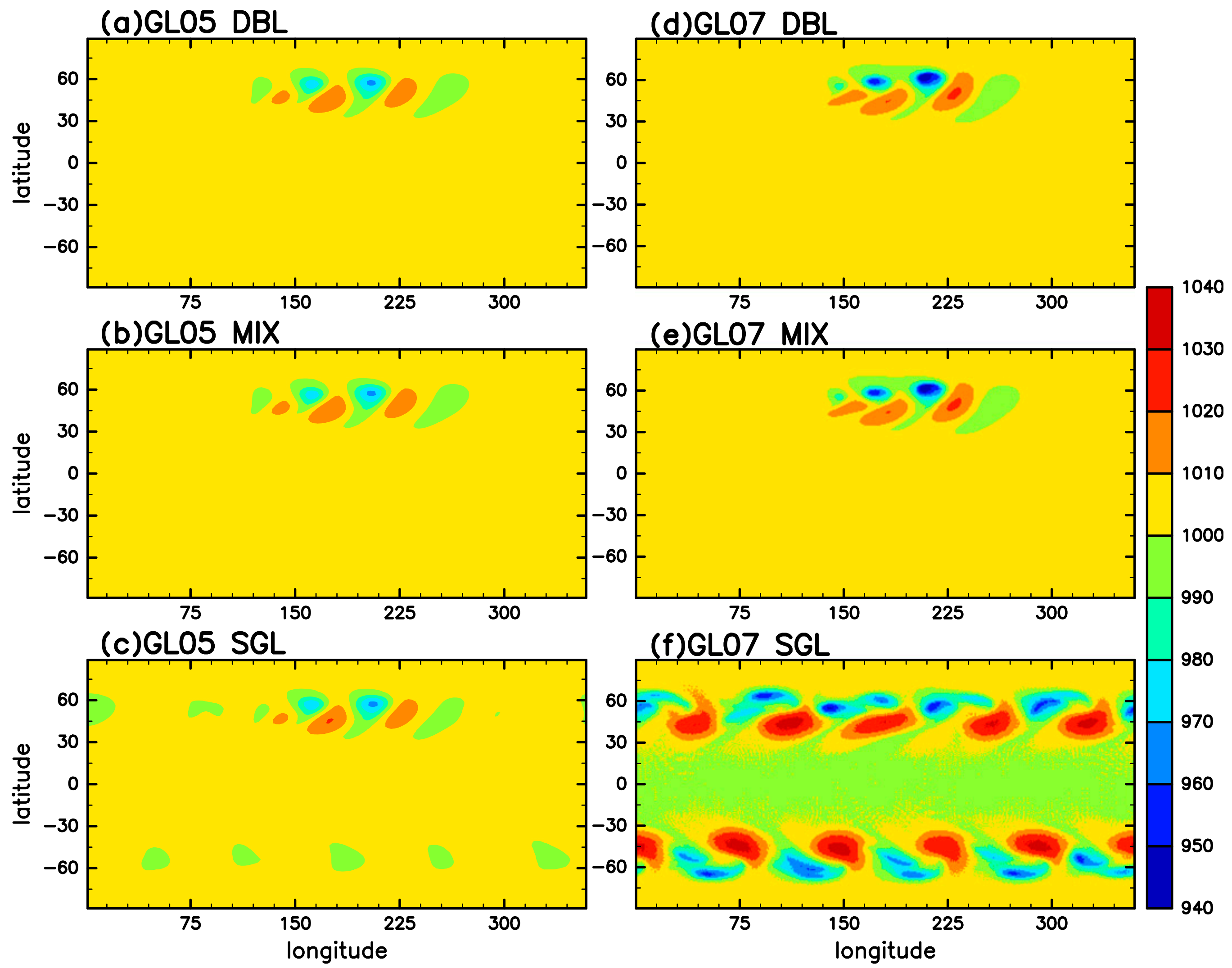
"Throughput-aware" design of the DA system
(Yashiro et al., 2016, GMD)

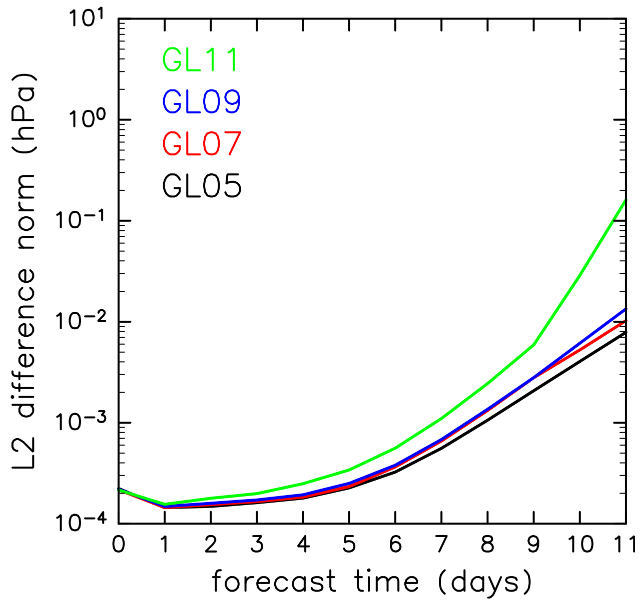
- reduce data movement
- use local storage
- avoid global communication

Evaluation of mixed precision

We had better utilize the single/half precision more

- More evaluation of physical performance is necessary
  : Ideal/real case, deterministic/statistic case, unit/total tests, etc.

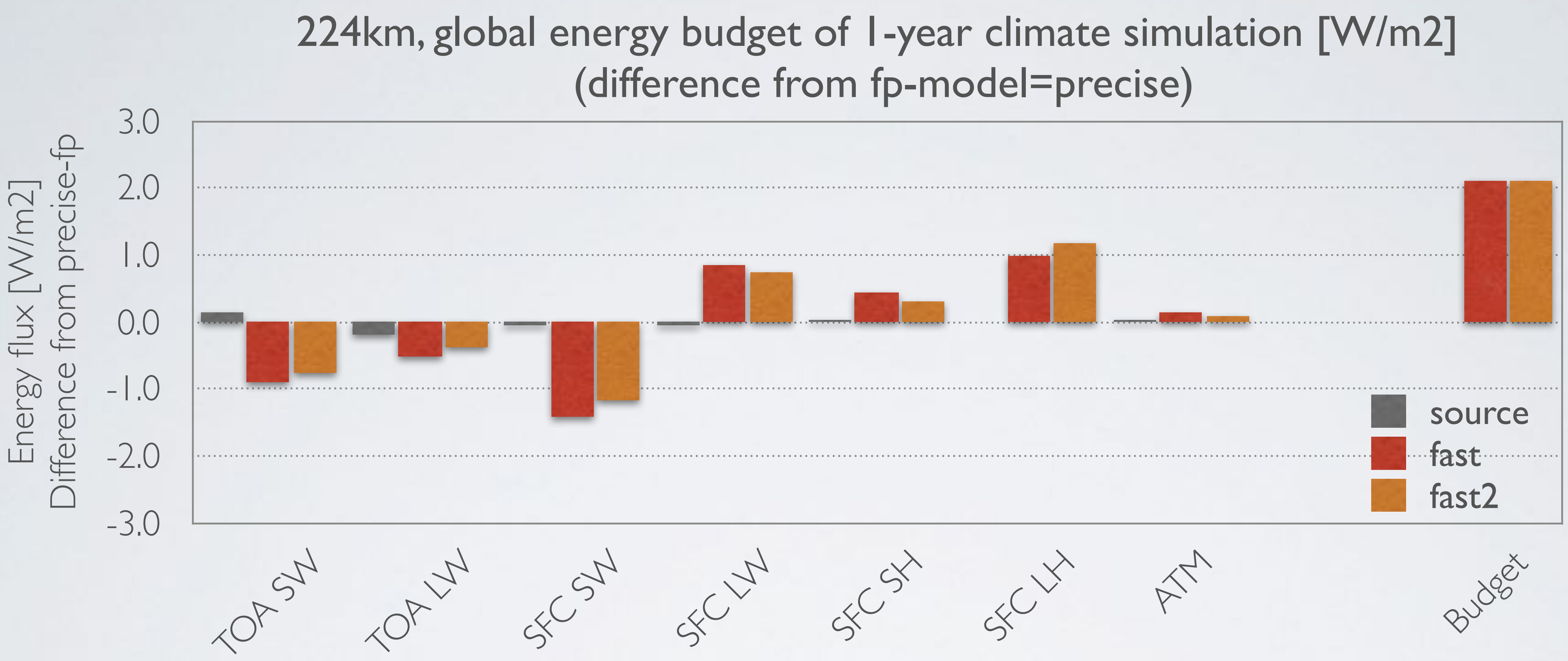




Baloclinic wave test with single precision
(Nakano et al., 2018, MWR)

Approximate Computing

Scientific performance evaluation is important

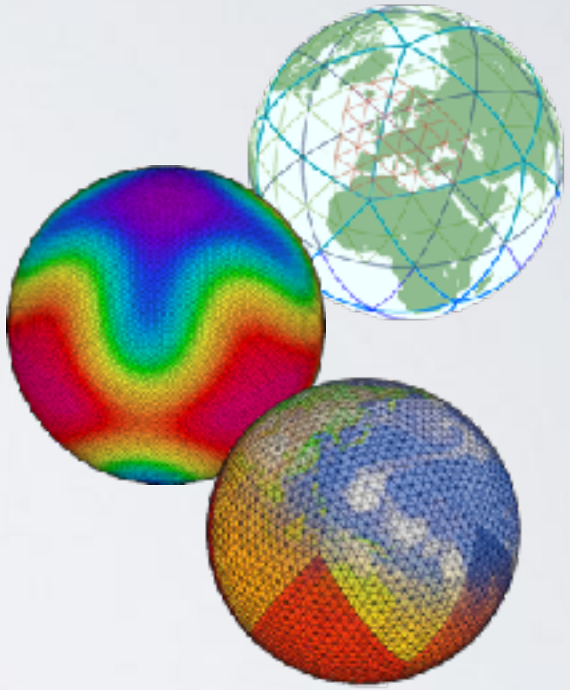


- For intel Fortran compiler, fp-model=fast2 is x1.6 faster than precise mode
- However, the budget imbalance occurs in the cases of fp-model=fast/fast2
  - We know there are few lines to keep precision in radiation scheme

SPPEXA/AIMES project (1)

AIMES **(Advanced Computation and I/O Methods for Earth-System Simulations)**

- Tri-lateral collaborative project funding
- Collaboration of icosahedral atmosphere model
  - U. Humburg, DWD, DKRZ (German) : ICON
  - IPSL (France) : DYNAMICO
  - RIKEN, Tokyo Tech., U. Tokyo (Japan) : NICAM



Targets

- DSL benefit for icosahedral atmospheric models
- Massive I/O
- Kernel suites and mini-apps from three state-of-art climate models

SPPEXA/AIMES project (2): Benchmarking

IcoAtmosBenchmark

- https://aimes-project.github.io/IcoAtmosBenchmark_v1/
- Ver.1: A kernel package from icosahedral models
  - For the performance evaluation of stencil calculation
  - For the development of domain specific languages (DSLs)

SPPEXA/AIMES project (3): I/O

Approximate Computing

Data centric design

SCIL: Scientific Compression Interface Library

- User can control precision of output data for each model variable
- Library selects compression algorithm (lossy/lossless)
- HDF5/NetCDF4 integration

Integration into NICAM

- History output: evaluation of compression efficiency is ongoing
- Checkpoint input/output: planned

Performance Portability

GridTools-NICAM(1)

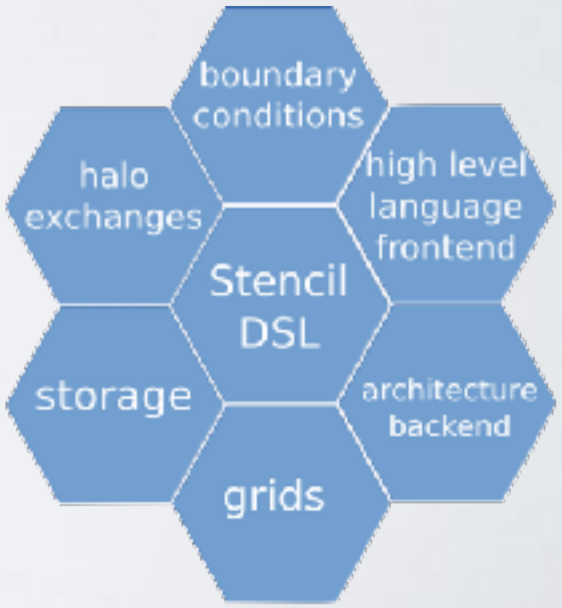
Can we become released from the curse of directives?

- Maintenance of directives is costful

```
!OCL XFILL
!$acc kernels pcopy(PROGq00) pcopyin(PROGq) async(0)
!$omp parallel do default(none),private(g,k,l,nq), &
!$omp shared(gall,kall,lall,nall,PROGq00,PROGq), &
!$omp collapse(3)
do nq = 1, nall
do l  = 1, lall
do k  = 1, kall
do g  = 1, gall
   PROGq00(g,k,l,nq) = PROGq(g,k,l,nq)
enddo
enddo
enddo
enddo
!$omp end parallel do
!$acc end kernels
```

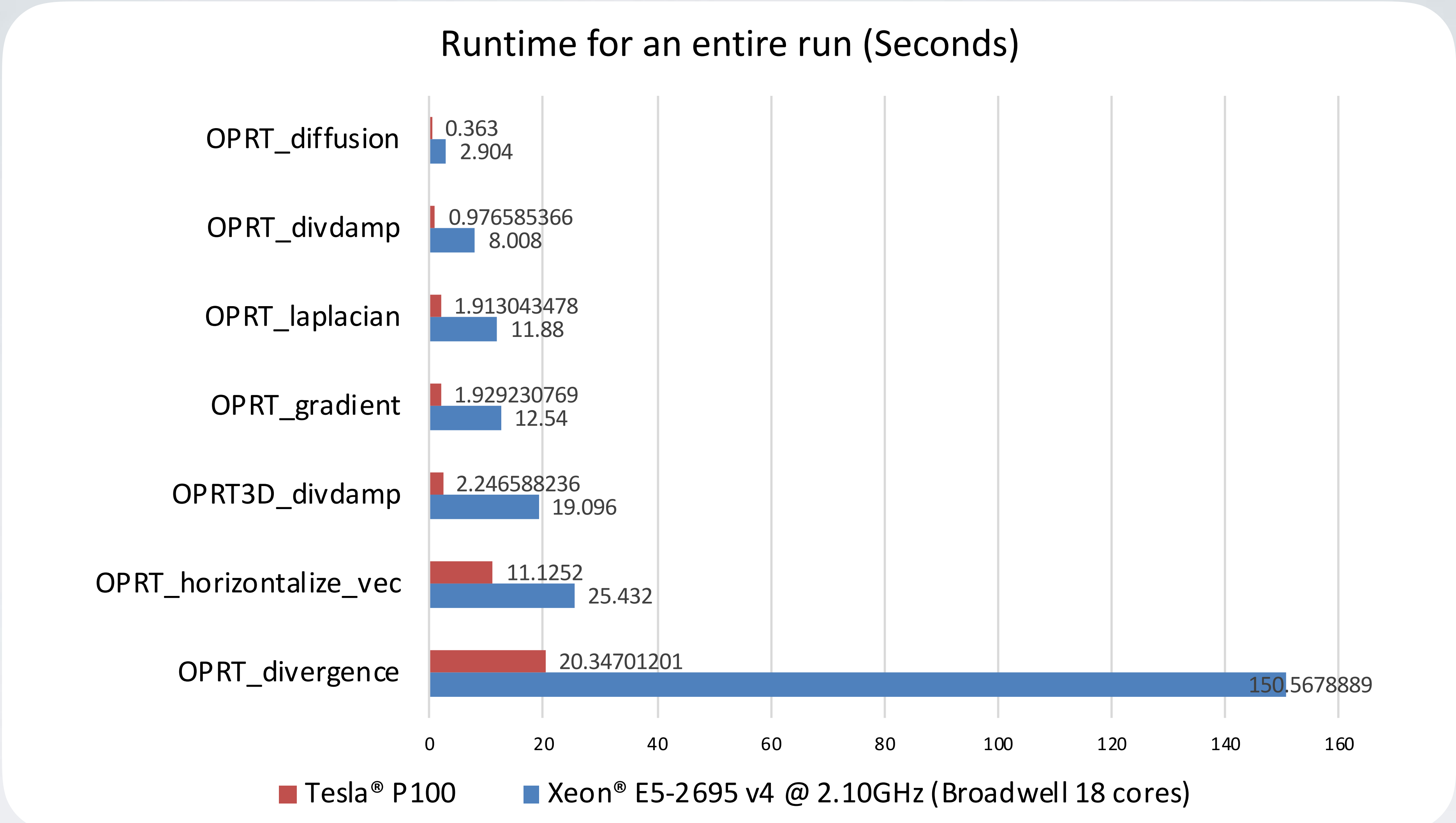
GridTools-NICAM project

- Collaboration started in the AIMES project
- NICAM is favorable for GridTools
  - Structured grid in tile: cartsian-like data layout



GridTools-NICAM(2)

Performance Portability



The way to full implementation with GridTools

- The full-dycore is almost finished: very good results on GPU
- Communication part is more difficult: node topology is complex
- Physics library? : exploring solutions using Python

Summary

- **In the petascale era, the efforts on the performance have meant the utilization of more cores and accelerators**
  - Labor intensive works were effective: code refactoring and directives

- **In the exascale era, the efforts will mean how the application developers accept trade-offs.**
  - ~~Sub-grid parameterization vs super high resolution~~
  - The floating-point precision vs simulation result
  - DSL affects everything of the ecosystem of weather/climate studies: from education to operation.
    - Can we change the mind of community people?

"Rebuild myself while running at full speed"