Data Storage Concepts for pre-exascale Earth System Model Output handling

Karsten Peters-von Gehlen
Deutsches Klimarechenzentrum (DKRZ), Data Management, peters@dkrz.de

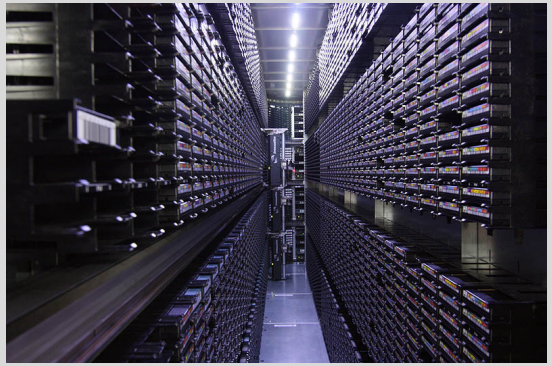
22 September 23, NHR Storage Workshop, virtual

DKRZ – very short intro

DKRZ is a **topical IT infrastructure provider** for the Earth System Science (ESS) community (in Germany)

Levante Supercomputer

(130PB disk storage, DDN)



Tape archive

(300PB capacity, StrongLink)

+

MinIO Cloud (~20PB) in testing

A suite of services specifically tailored towards the needs of ESS researchers

+

- Model improvement
- Analysis support
- **Data management**

Earth System Science until about 2015

Relativel coarse resolution models

Grid spacing ~100km

Output volumes manageable by “common approaches”

The traditional way of working

Common approaches?

An example (the “individual scientist” case)

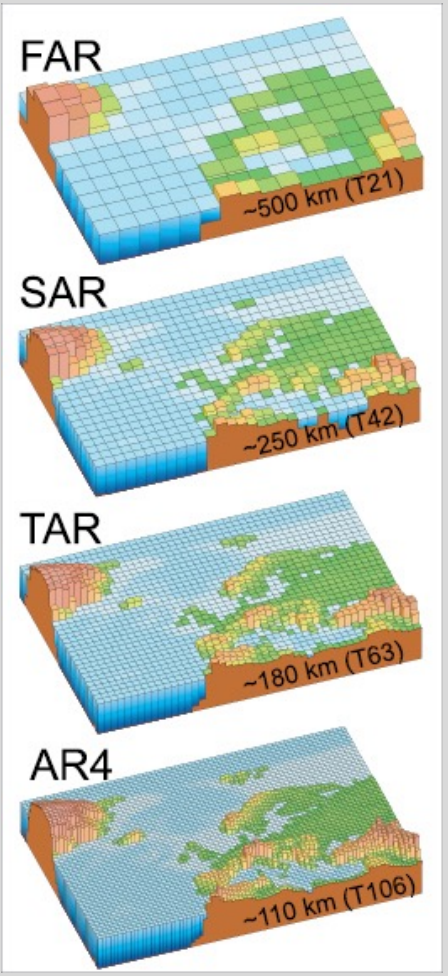
- simulation(s) configured and performed by individual scientist (on the HPC)
- output written to specified location on disk in private user/group space
    - Usually in relatively large individual files
- Postprocessing to fit analysis demands (possibly involving duplication)
- Data analysis
- Paper publication
- Transfer to tape archive at the end of a project to free up space
- Documentation of data handling did often not take place, data therefore “got lost in the archive”
- Data volume on the order of several 10s of TBs for multiple simulations

The traditional way of working

Common approaches?

Another example (the "reuse" case, e.g. CMIP, IPCC)

- simulation(s) configured according to common approaches and performed by a group of scientists
- output postprocessed and standardised according to agreed protocols
  - Usually in relatively large (global) files
- Data publication via ESGF (a global federation of data nodes)
- Transferred to long-term archive for preservation and global availability
  - World Data Center for Climate @DKRZ
- Total data volume on the order of several 10s of PBs for multiple simulations (still all available via ESGF)



https://archive.ipcc.ch/publications_and_data/ar4/wg1/en/ch1s1-5.html

The traditional way of working

Short summary

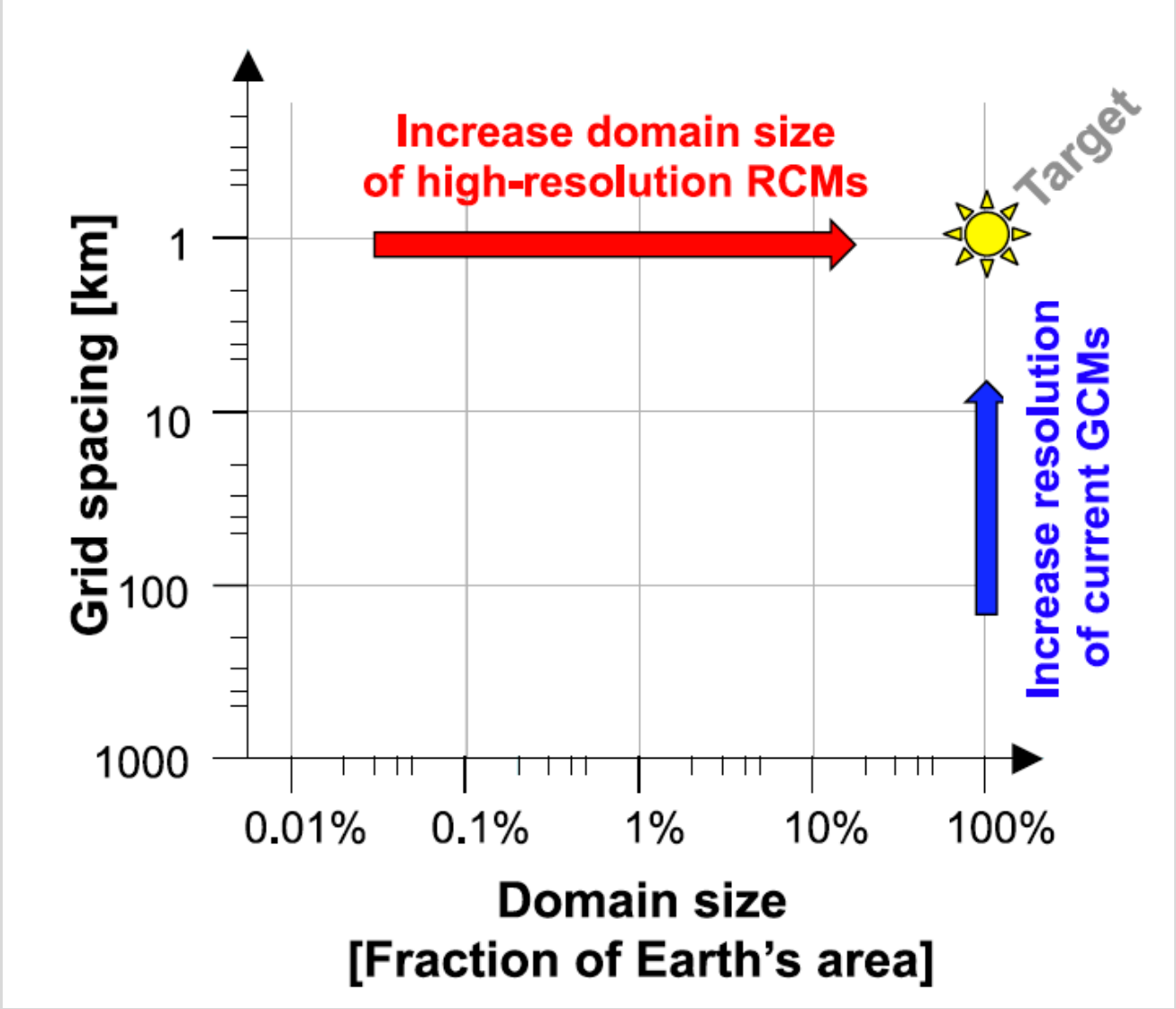
Characteristics

- Model output often still manageable at the individual scientist level
- Often used only by one scientist
- Analysis methods often not optimized (but that did not matter)
- Climate change projections with a large ensemble of models possible and still manageable

www.digitalbevaring.dk

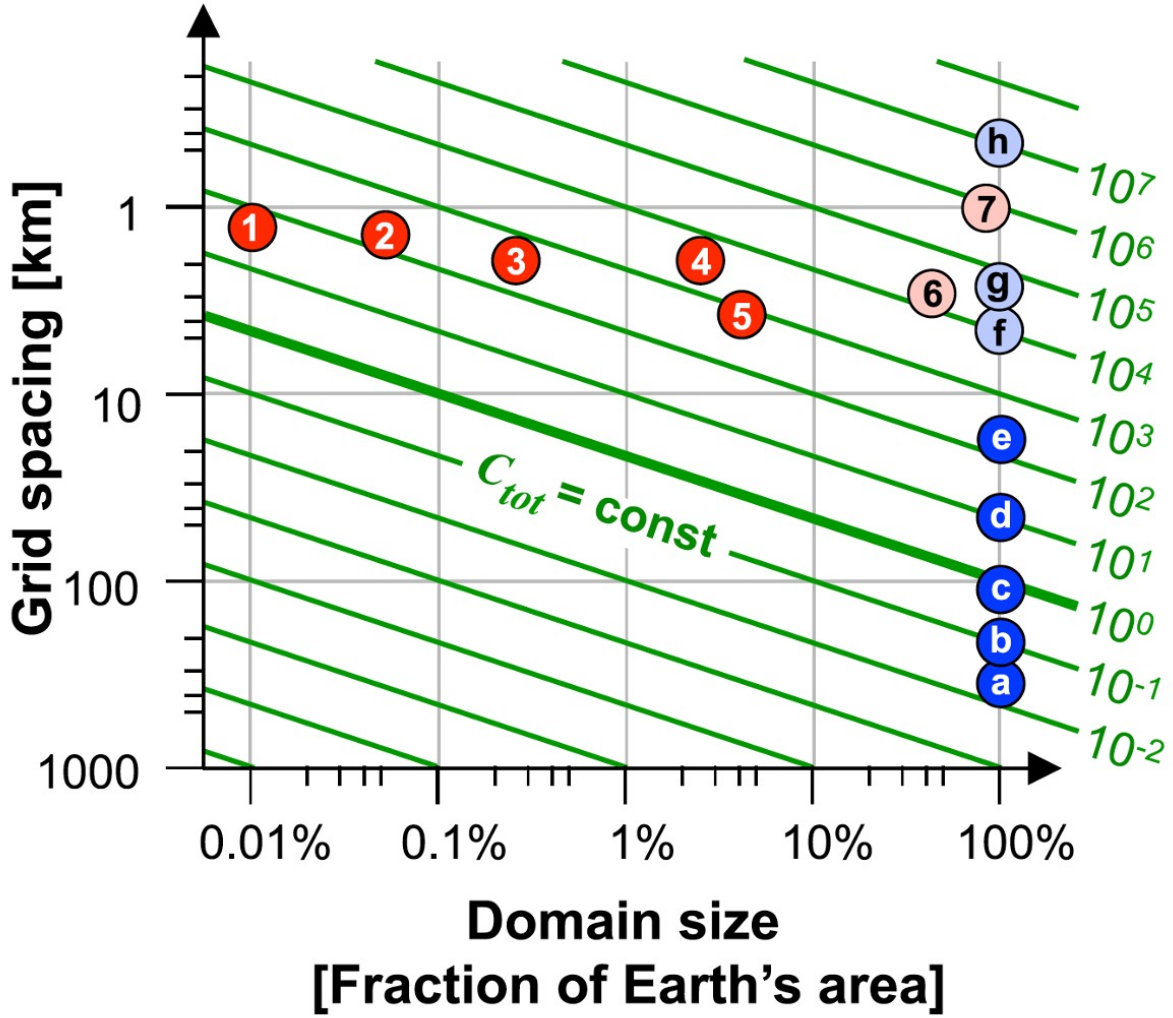
But what changed?

- Increasing push towards higher resolution
  - Hopes for more "physical" models
- In recent years, simulations have approached the limits of what is computationally feasible



But what changed?

Approaching



1. Rheinland-Pfalz,1.3km grid spacing (Knote et al, 2010)
2. Southern UK, 1.5km grid spacing (Kendon et al, 2014)
3. Alps, 2.2km, 2.2km grid spacing (Ban et al 2014)
4. Europe, 2.2km grid spacing (Leutwyler et al, 2017)
5. North America, 4km spacing (Liu et al 2017)
6. Aquaplanet, 4km spacing (Bretherton and Khairoutdinov, 2015)
7. Near-global, 1km spacing (Fuhrer et al, 2018)

---

a. CMIP1 (1995)
b. CMIP3 (2001)
c. CMIP5 (2013)
d. MIROC4h (Sakamoto et al, 2012)
e. CMIP6 HighRes MIP (Haarsma et al., 2016)
f. ICON and IFS Scaling Experiments (Neumann et al, 2019)
g. DYAMOND (Stevens et al., 2019)
h. NICAM (Miyamoto et al., 2013)

Implications for data handling

- First simulations attempted to follow established approaches
  - Multiple GB sized files
  - Labour-intensive postprocessing for reuse by many local scientists (regridding, renaming, reorganization)
  - Very slow anlaysis/visualisation
  - Global reuse/sharing practically impossible

  - Output volumes of ~1PB per ONE 5yr simulation

https://www.istockphoto.com/de/grafiken/frustrated-man

Implications for data handling

- First simulations attempte… …w established approaches
  - Multiple GB sized fi…
  - Labour-intensive postp… …for reuse by many local scientists (regrid… …aming, reorganization)
  - Very slow anlays… …lisation
  - Global reuse/… …ically impossible

- Output volum… …of ~1PB per ONE 5yr simulation

https://www.istockphoto.com/de/grafiken/frustrated-man

Ways to go forward

“efficient access to and discovery and processing of (very) large in-house and remote datasets”

Comprehensive metadata cataloguing and metadata-driven access across storage tiers

Data formats allowing for fast access and processing

Possibility of very specific database queries

Efficient use of existing hardware

Ways to go forward

Semantic data access

DKRZ

Intake catalogs

INTAKE

- a simple programmatic data access layer allowing for **semantic data access**
  - available for Python
- DKRZ provides these catalogs for PBs worth of data
  - CMIP5/6
  - CORDEX
  - ERA5
  - **DYAMOND**
  - **nextGEMS**

```
import intake
col = intake.open_esm_datastore("/work/ik1017/Catalogs/dkrz_cmip6_disk.json")
col.df.head()
```

Features

- display catalogs as clearly structured tables inside jupyter notebooks for easy investigation
- browse through the catalog and select your data without being on the pool file system
- open climate data in an analysis ready dictionary of `xarray` datasets

https://cmip-data-pool.dkrz.de/intake-catalog-service.html

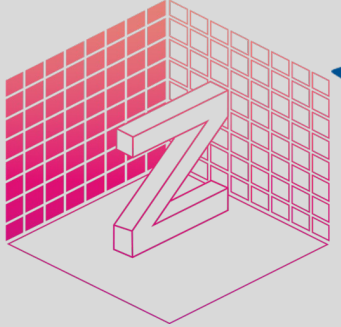
Benefit:

Large dataset collections can be stored anywhere on the system and users do not need to know where it acutally is.

BUT: catalogs have to be maintained

Ways to go forward

Data format



DKRZ

Zarr - a file storage format for chunked, compressed, N-dimensional arrays based on an open-source specification

For climate data:

- "feels like netCDF"
- full datasets can be larger than memory
- I/O and analysis can be easily parallelized (w/o MPI)
- optimized for cloud storage
- possibility to just access the data you need (through semantics)

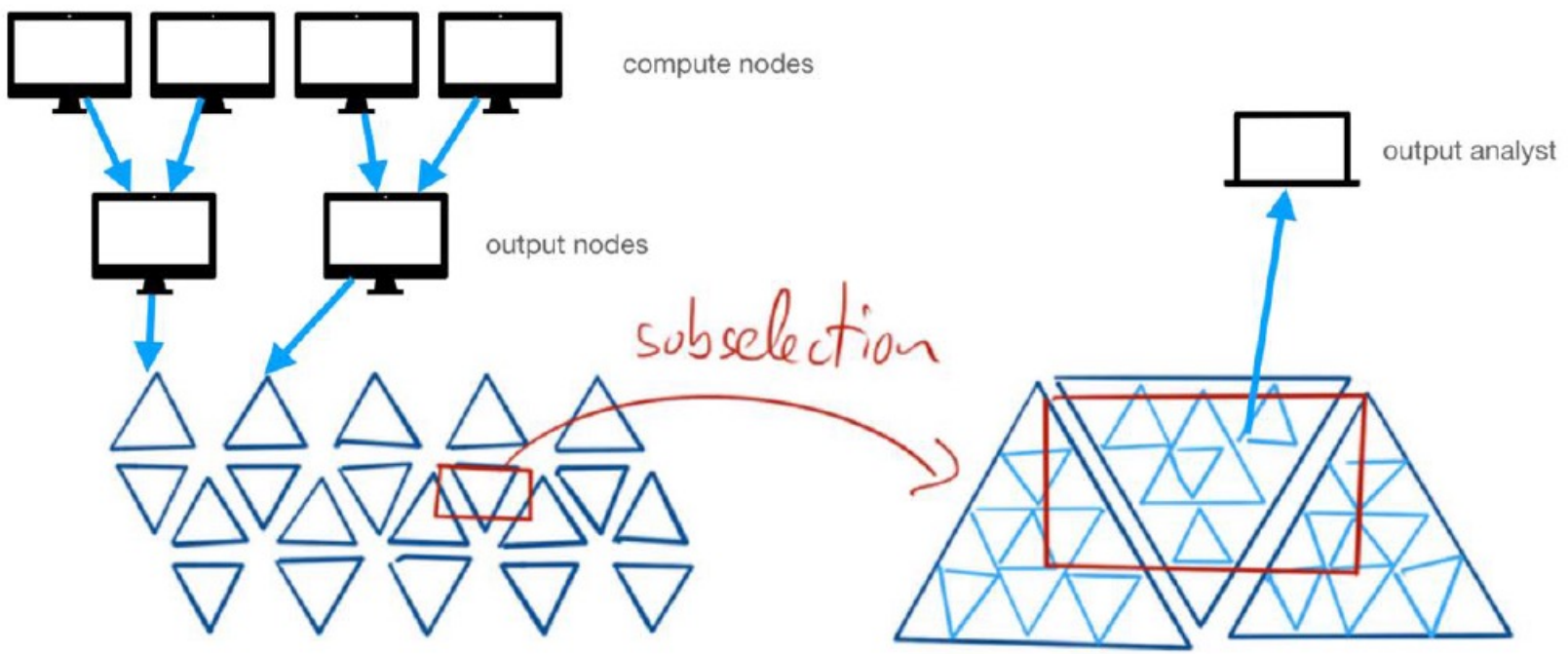
https://zarr.readthedocs.io/en/stable/

Ways to go forward

Data format

Some performance checks:

Store data in horizontal chunks, load only a few of them

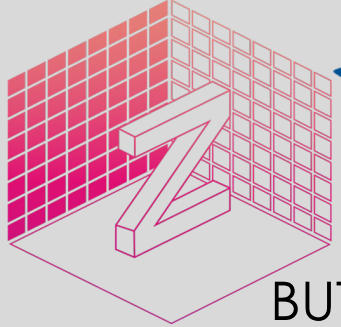


T. Koelling

Ways to go forward

Data format

Some performance checks:



DKRZ

It works

Plot a time series averaged over a certain area (0.75 % of globe)

Chunk size	$20*4^{10}$	$6*4^{9}$
format	grib/aec	blosc/lz4
1 thread	20 min	6.8 s
48 threads	3 min 29 s	4.8 s

T. Koelling

BUT:

- individual "files" are fairly small (about 10-100 MB)
- 800 TB of data result in millions of files
- performance issues on a parallel file system and in a tape archive

- further research needed for optimal setup

Ways to go forward

Monitoring data use

/fastdata at DKRZ (by DDN)

Configuration

We have 200 TB SSD via nvme (OST 0-15) and 3 PiB HDD (OST16-19).

They are organized in pools `ddn_ssd` and `ddn_hdd`.

Default write would spread data randomly across the SSD and HDD parts.

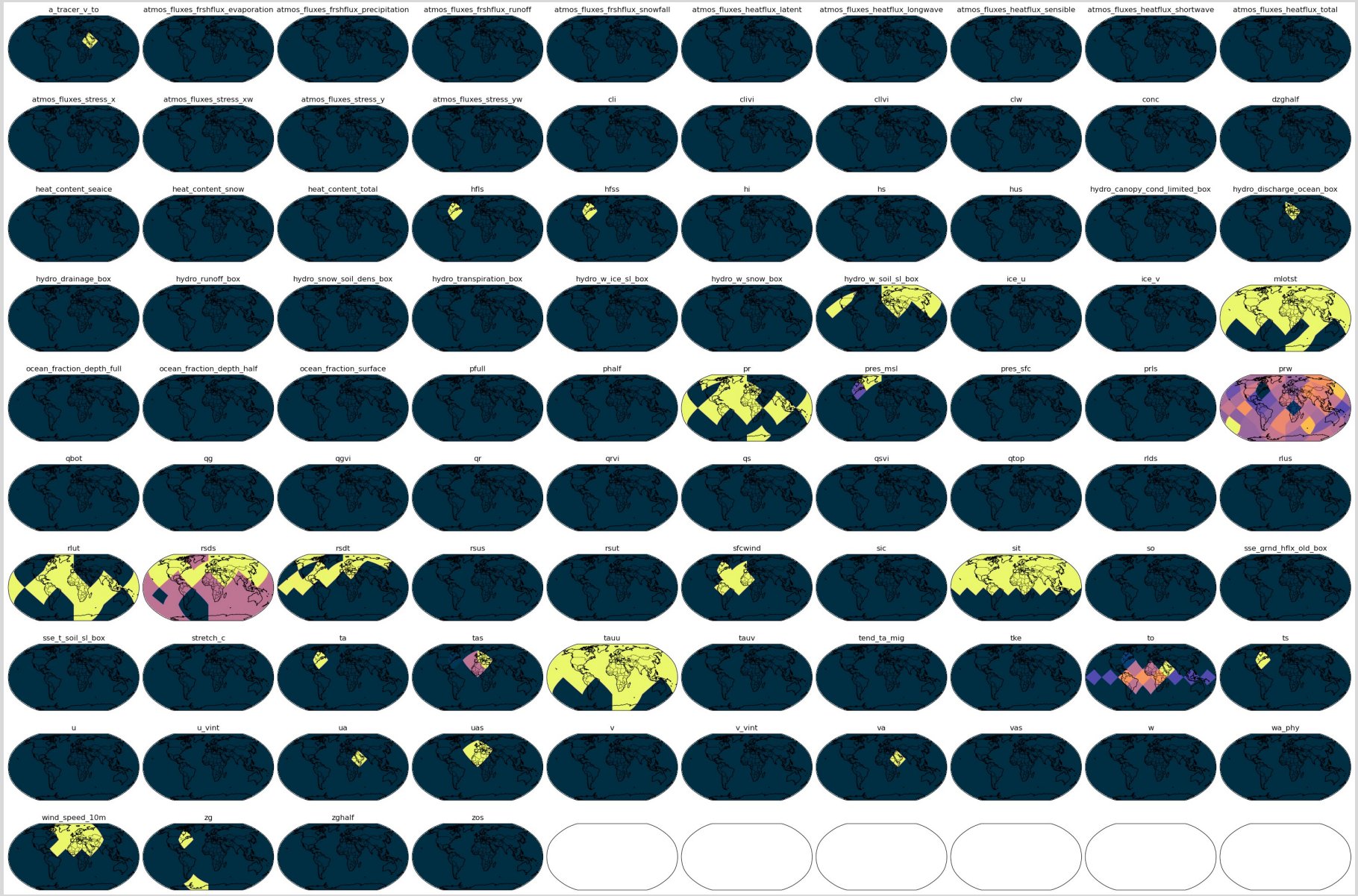
The fancy magic can move/distribute data between SSD and HDD.

Writing to this part of the file system by request, e.g. for large projects, data access is logged ☺

Test case: use it for a Hackathon in the beginning of June 2023 (about 100+ scientists working on several PBs of data at once)

Ways to go forward

Monitoring data use



What data is actually used at the Hackathon?

- 94 parameter fields in total
- only a fraction was touched
- for many, only selected regions were used

Of course only a snapshot, but gives valuable information for system optimization

Figure: Tobias Kölling, MPI-M

Ways to go forward

Connection the tape achive

How to deal with all these small files?

Goal: enable catalog-based access to chunked data using the DKRZ tape archive system (HSM)

Home-built solution: “outtake”

- archive tar’ed zarr datasets along with index files
- request a chunk
- outtake finds the tarball needed for this chunk
- downloads this tarball to a ”disk-cache”
- extracts the chunk and provides it to the user

It works, but heavily depends on the performance of the HSM system.

More stable application featuring these capabilities will be built

https://easy.gems.dkrz.de/Processing/Intake/tape_archive.html

The end…

Thank you for your attention!

peters@dkrz.de