# DKRZ's km-scale cloud

## formerly known as eerie.cloud

## 3/3: Coming of age

Fabian Wachsmann (DKRZ)

**In ESSI2.3 Pangeo EGU26-9485**

https://km-scale-cloud.dkrz.de

The cloudify codebase

# Available Km-scale climate simulations

We host >10PB of prominent High-Resolution Earth System Model output stored at the German Climate Computing Center (DKRZ) on the open platform https://km-scale-cloud.dkrz.de

CORDEX Coordinated Regional Climate Downscaling Experiment

ERA5

epoc explaining and predicting the ocean conveyor

EERIE

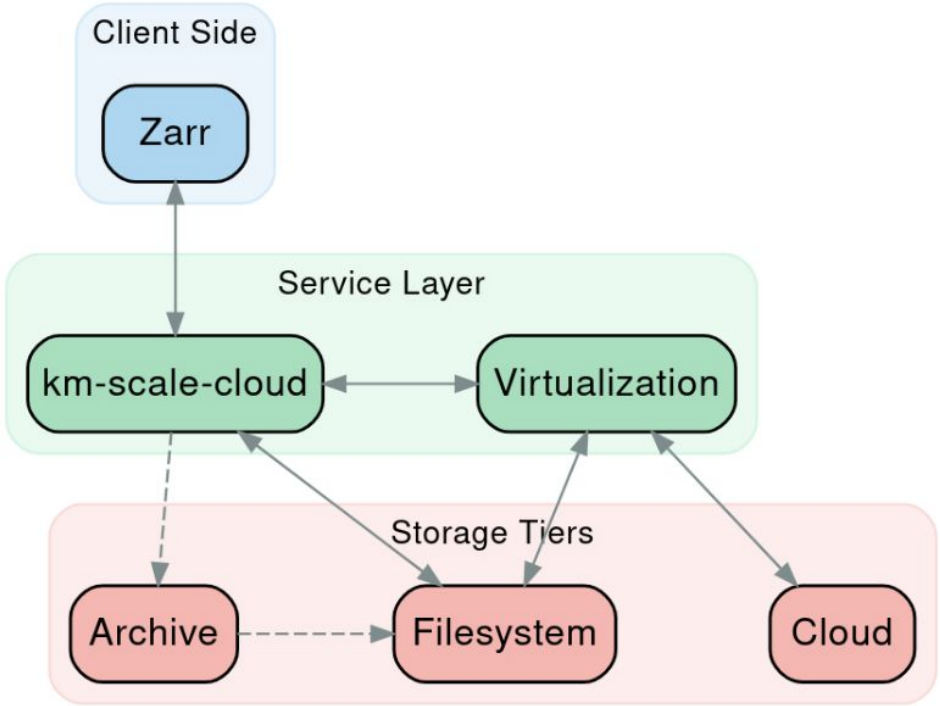
nextGEMS

ORCESTRA

CONSORTIUM FOR SMALL SCALE MODELING COSMO


https://km-scale-cloud.dkrz.de

# What's the magic? 🪄

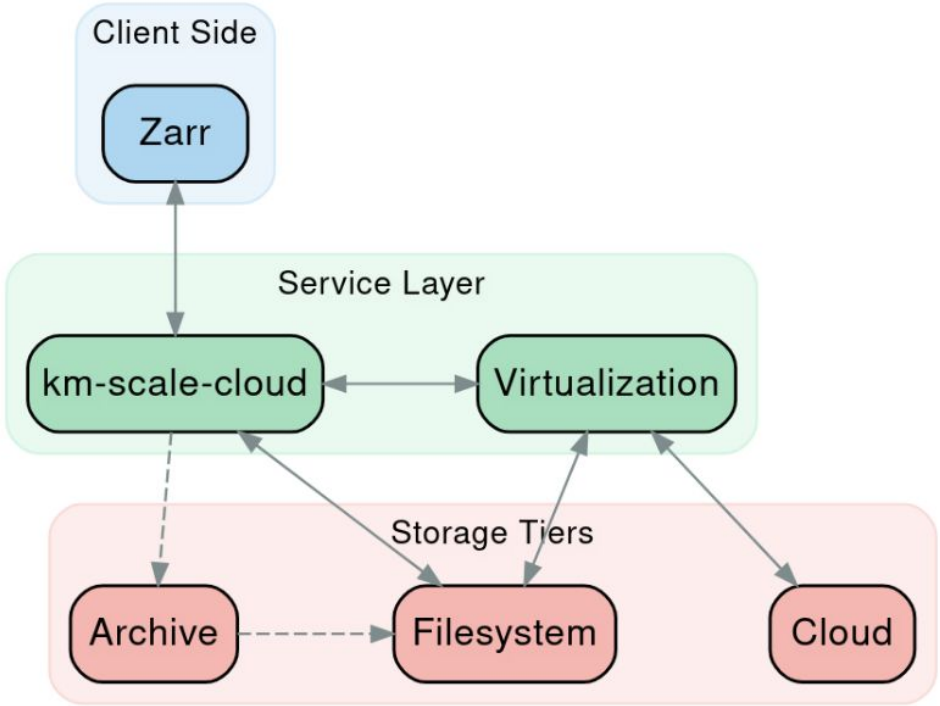



**The km-scale-cloud**

**is an open data server that takes over virtualization tasks from clients and provides uniform Zarr access.**

https://km-scale-cloud.dkrz.de

# What's the magic? 🪄




**The km-scale-cloud**

**is an open data server that takes over virtualization tasks from clients and provides uniform Zarr access.**

- **+ Smaller Client environments**
  **→ Broader user community**
- **+ COARD from any storage and format**
  **→ Cloud migration teaser**

https://km-scale-cloud.dkrz.de

# Give credit where credit is due 💳

The km-scale-cloud is based on xpublish plus

+ a kerchunk proxy (bypassing array loading)
+ a dataset to STAC item
+ statistics endpoints

and runs on 16CPUs and 64GB memory

https://github.com/xpublish-community/xpublish

https://km-scale-cloud.dkrz.de

# Available Km-scale climate simulations

We host >10PB of prominent High-Resolution Earth System Model output stored at the German Climate Computing Center (DKRZ) on the open platform https://km-scale-cloud.dkrz.de

ERA5

EERIE

nextGEMS

ORCESTRA

CONSORTIUM FOR SMALL SCALE MODELING COSMO


https://km-scale-cloud.dkrz.de

# Available Km-scale climate simulations

We host >10PB of prominent High-Resolution Earth System Model output stored at the German Climate Computing Center (DKRZ) on the open platform https://km-scale-cloud.dkrz.de

CORDEX Coordinated Regional Climate Downscaling Experiment

ERA5

epoc explaining and predicting the ocean conveyor

EERIE

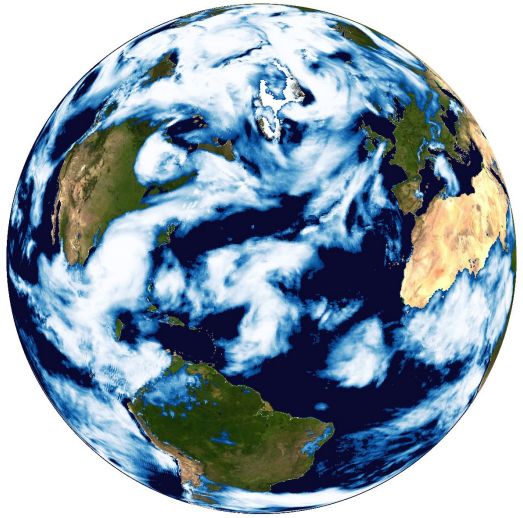
nextGEMS

ORCESTRA

CONSORTIUM FOR SMALL SCALE MODELING COSMO


https://km-scale-cloud.dkrz.de

# Celebrate success 🎊: ERA5



Virtually standardized following a Data Lakehouse Approach.

Renamed variables and added attributes

O(PB) original GRIB covering 1940-2026

ERA5-Land added

← Total Cloud Cover

xarray.Dataset

▸ Dimensions: (**time**: 31412, cell: 542080)

▾ Coordinates:

| | | | |
|---|---|---|---|
| **time** | (time) | datetime64[ns] | 1940-01-01T11:30:00 ..... |

```
array(['1940-01-01T11:30:00.000000000', '1940-01-02T11:30:00.000000000',
       '1940-01-03T11:30:00.000000000', ..., '2025-12-29T11:30:00.000000000',
       '2025-12-30T11:30:00.000000000', '2025-12-31T11:30:00.000000000'],
      shape=(31412,), dtype='datetime64[ns]')
```

| | | | |
|---|---|---|---|
| lat | (cell) | float64 | dask.array<chunksize=(... |
| lon | (cell) | float64 | dask.array<chunksize=(... |

▾ Data variables:

| | | | |
|---|---|---|---|
| bld | (time, cell) | float32 | dask.array<chunksize=(... |
| cape | (time, cell) | float32 | dask.array<chunksize=(... |

| | |
|---|---|
| typeOfLev... | surface |
| stepType : | avg |
| gridType : | reduced_gg |
| units : | J kg-1 |
| gridDefini... | Gaussian Latitude/Longitude Grid |
| original_p... | /pool/data/ERA5/E5/sf/fc/1D/059 |
| original_n... | 059 |
| long_name : | Convective Available Potential Energy |
| standard_... | atmosphere_convective_available_potential_energy |
| original_c... | gribscan |

https://km-scale-cloud.dkrz.de

# Celebrate success 🎊: ORCESTRA

EERIE

- Zarr V3 support

2m Temperature in a
ICON Limited Area 1.25km simulation (credits: Romain Fievet)

https://km-scale-cloud.dkrz.de

# Celebrate success 🎊:

- Full 1950-2050 km-scale simulations available
  Enabled async access to GRIB data in S3 Cloud Storage

Sea Surface Temperature in the
EERIE IFS-FESOM2 ssp245 scenario on ~5km resolution

https://km-scale-cloud.dkrz.de

# km-scale Metrics 🏃‍♀️

## Can you give us KPIs?

https://km-scale-cloud.dkrz.de

# km-scale Metrics 🏃‍♀️

Can you give us KPIs?

- Maximum 2.5GB/s http throughput reached for 2 processes API and datasets stored on Lustre on a single HPC node
- Throughput of 100MB/s (typical ethernet card limit) sustained in real world applications as an average over all datasets

Large chunks → large memory usage (client side caching recommended)

https://km-scale-cloud.dkrz.de

# New features: The Km-scale Tree

The whole 10PB
from the km-scale cloud

- programmatically accessible
- initialized in ms
- with a memory footprint of <100MB

as a DataTree object (replacing
intake catalog)

```
dt = xr.open_datatree(
    "https://km-scale-cloud.dkrz.de/datasets",
    engine="zarr",
    zarr_format=2,
    chunks=None,
    create_default_indexes=False,
    decode_cf=False
)
```

```
path_filter="s2024-08-10"
filtered_tree=dt.filter(lambda ds: ds if path_filter in ds.path else None)
```

https://km-scale-cloud.dkrz.de

# New features: Catalogs and Statistics

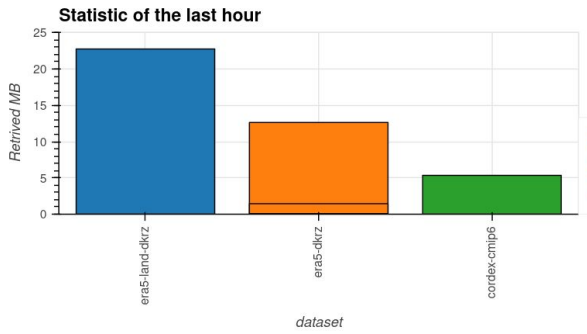
**The whole 10PB**

**from the km-scale cloud**

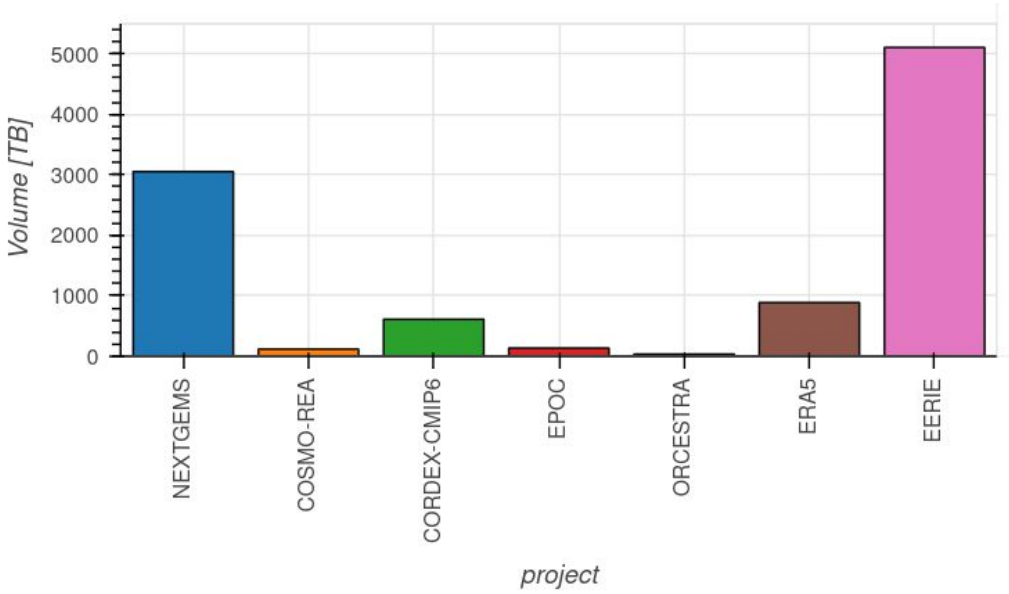
**interoperable through STAC**

**Live monitoring**

```
df=pd.read_json(
    "https://km-scale-cloud.dkrz.de/stats_nginx.json"
)
```




```
import pystac
import pandas as pd
import hvplot.pandas
cat=pystac.Catalog.from_file(
    "https://km-scale-cloud.dkrz.de/stac-catalog-all.json"
)
stat_assets=[]
for col in cat.get_children():
    df = pd.read_csv(col.assets["stat"].href.replace(".html",".csv"))
    df["project"]=col.title.split(' ')[0]
    stat_assets.append(df)
concat_df = pd.concat(stat_assets)[['total_nbytes [TB]', 'project']]
```




https://km-scale-cloud.dkrz.de

DKRZ's km-scale cloud makes HR-ESM output usable and tangible!

- data access as simple and effective as possible with zarr endpoints
- STAC catalog items for each dataset

Meet the team at the DKRZ Booth 23

## Worth a look:

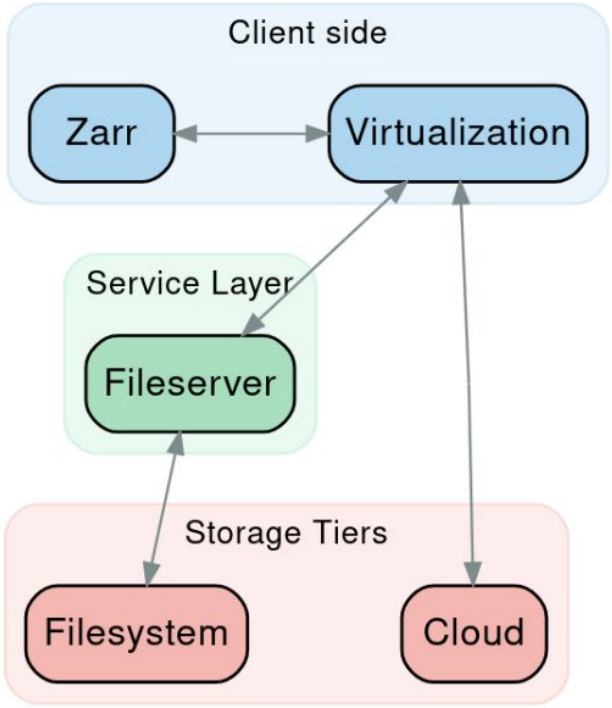
**Today 14-16**
**Details on our workflow**
**X4.117**

**Afterwards,16-18:**
**km-scale ESMs**
**Room 0.31/32**


**Tomorrow, 8-12:**
**Zarr Visualization**
**X4.89**

**Wednesday 16-17:**
**Fantastic climate model output + gridlock at the DKRZ booth**

# What's the magic? 🪄

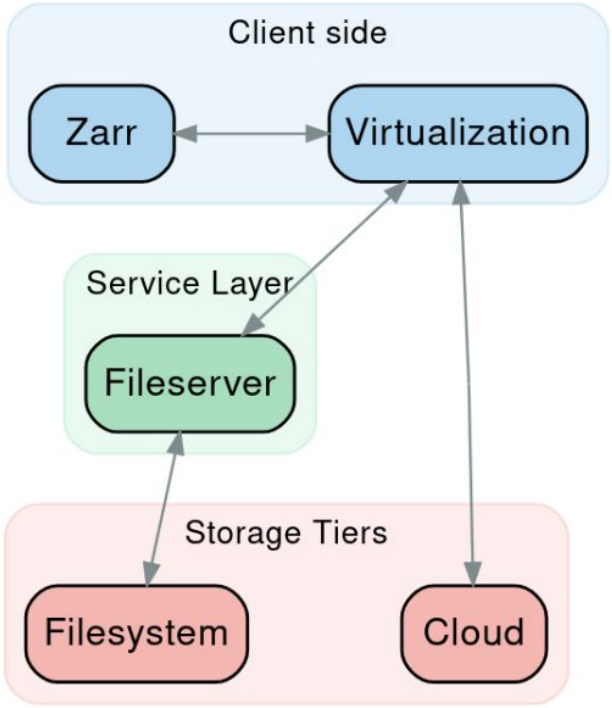



## Virtual Zarr Datasets

- turn differently formatted source data into COARD
- promotes cloud migration

https://km-scale-cloud.dkrz.de

# What's the magic? 🪄




## Virtual Zarr Datasets

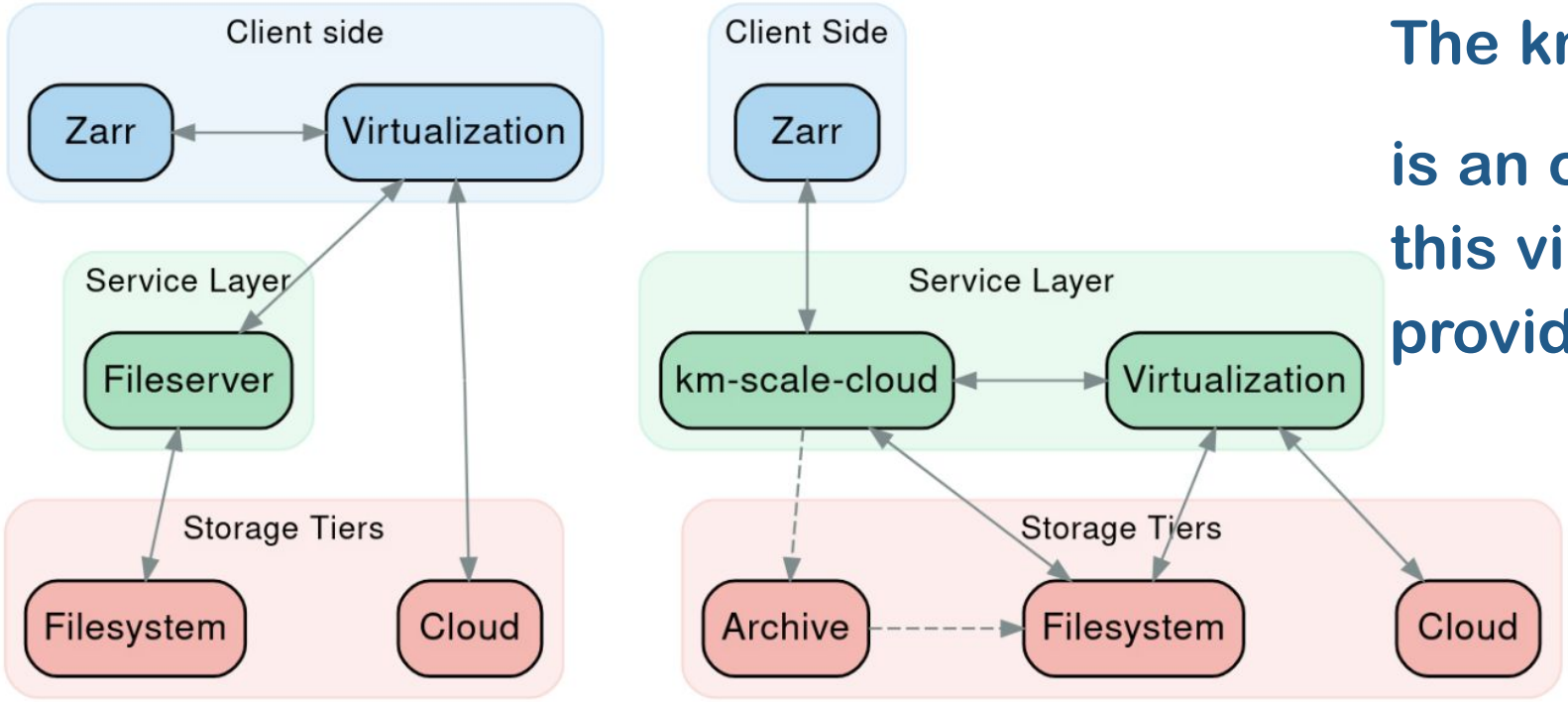
- turn differently formatted source data into COARD
- simplify cloud migration

… but may require heavy client resources for

- speaking protocols
- decoding formats
- loading chunks

https://km-scale-cloud.dkrz.de

# What's the magic? 🪄




**The km-scale-cloud**

**is an open data server that does this virtualization for you and provides uniform Zarr access.**

https://km-scale-cloud.dkrz.de

# Future

Upcoming mission: Unify access for AI Climate model output

Changes required:

- Support rectangular chunk shapes
- Support icechunk

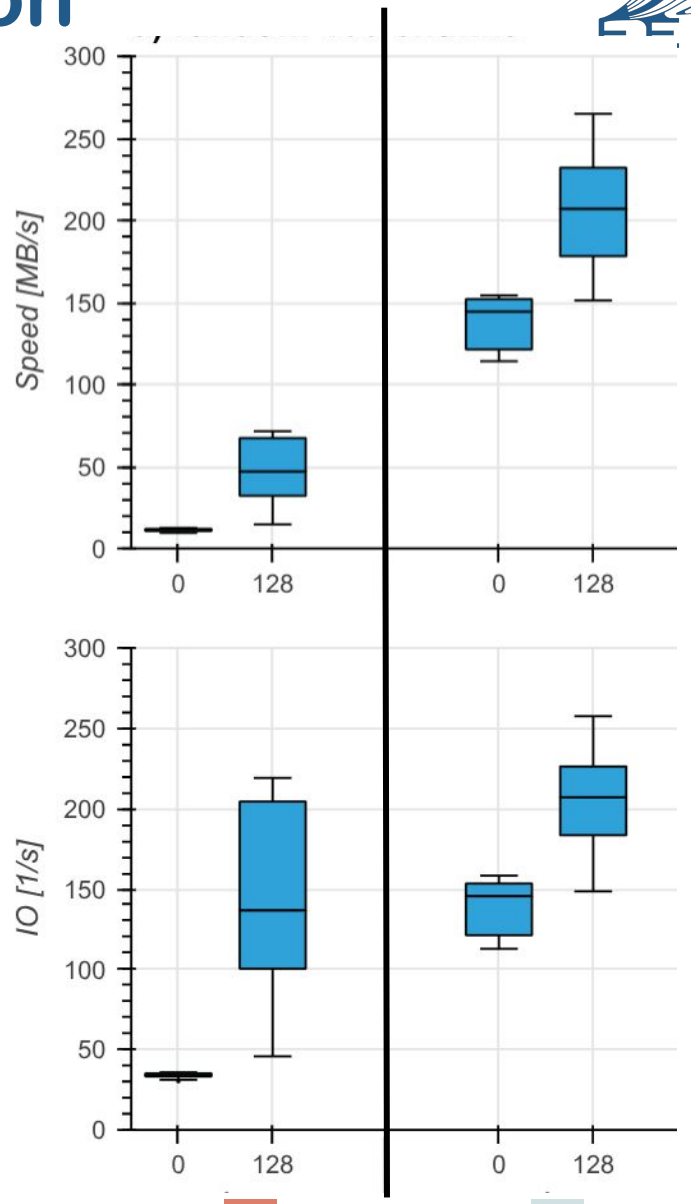
Hypothesis:
Repacked NetCDF (which is still not cloud-optimized) glued to an index can be read by Zarr V3 without references

https://km-scale-cloud.dkrz.de

# Good Virtualization, bad virtualization

Benchmark for single process App

Speed and IOPS in 10x100 parallel random accesses to chunks with cache enabled (128) and disabled (0) for datasets A and B with same avg chunk sizes (~1MB). Why is A bad?




left: A. right B.

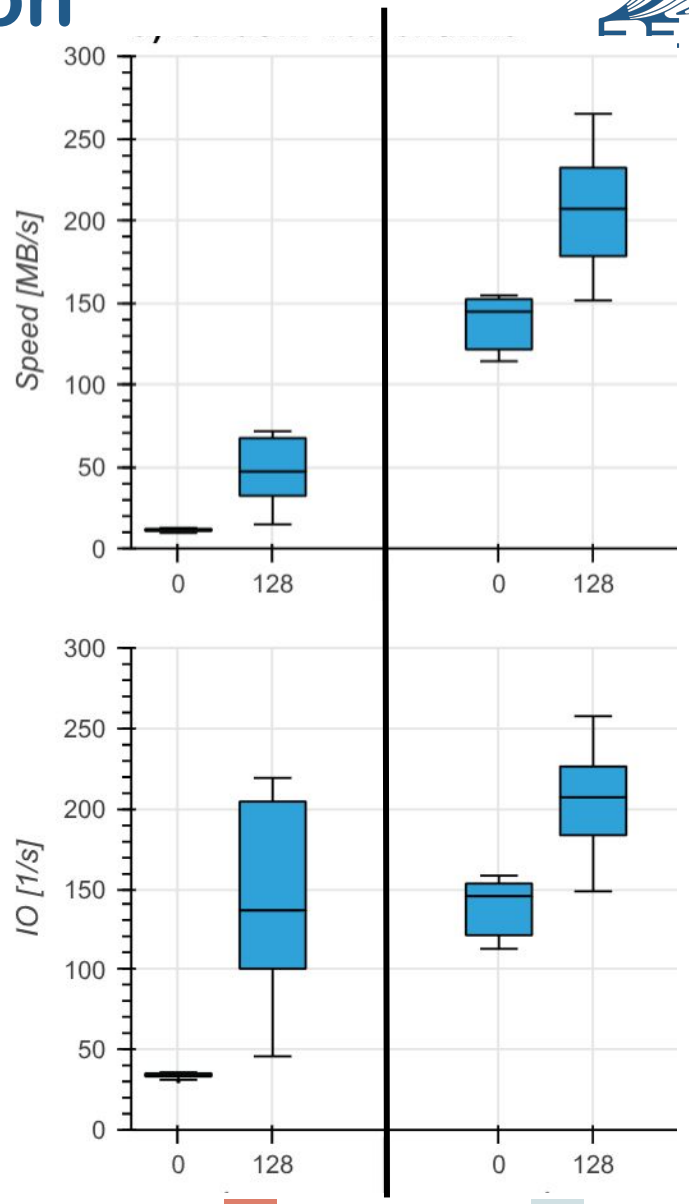
https://km-scale-cloud.dkrz.de

# Good Virtualization, bad virtualization

Benchmark for single process App

Speed and IOPS in 10x100 parallel random accesses to chunks with cache enabled (128) and disabled (0) for datasets A and B with same avg chunk sizes (~1MB).

A: Too large single reference table size (25MB) leads to overhead.




left: A. right B.

https://km-scale-cloud.dkrz.de

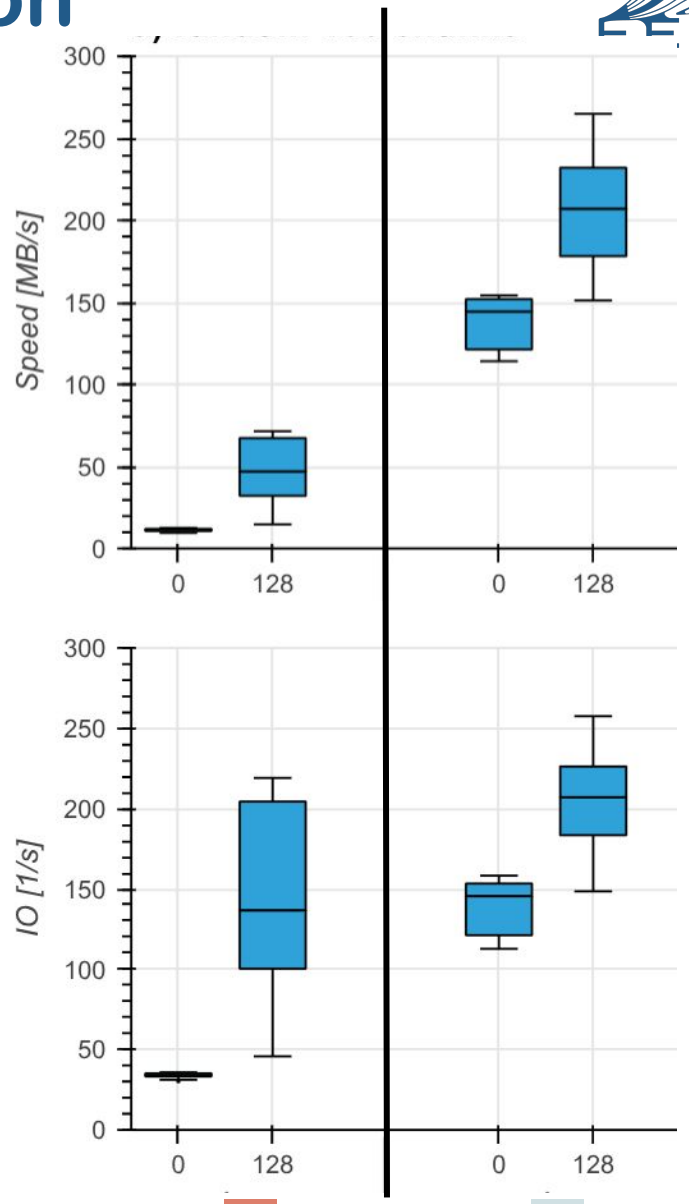
# Good Virtualization, bad virtualization

Benchmark for single process App

Speed and IOPS in 10x100 parallel random accesses to chunks with cache enabled (128) and disabled (0) for datasets A and B with same avg chunk sizes (~1MB).

A: Too large single reference table size (25MB) leads to overhead.

→ *Pay attention to reference table configuration*

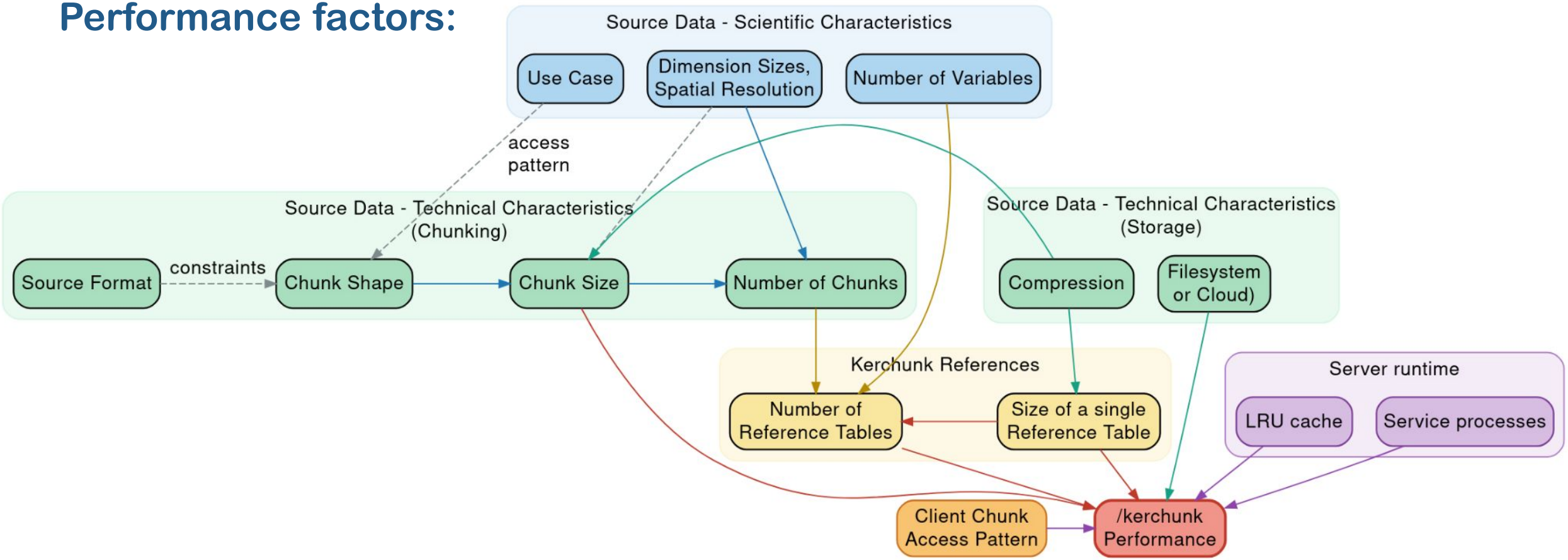



left: A. right B.

https://km-scale-cloud.dkrz.de

# Good Virtualization, bad virtualization

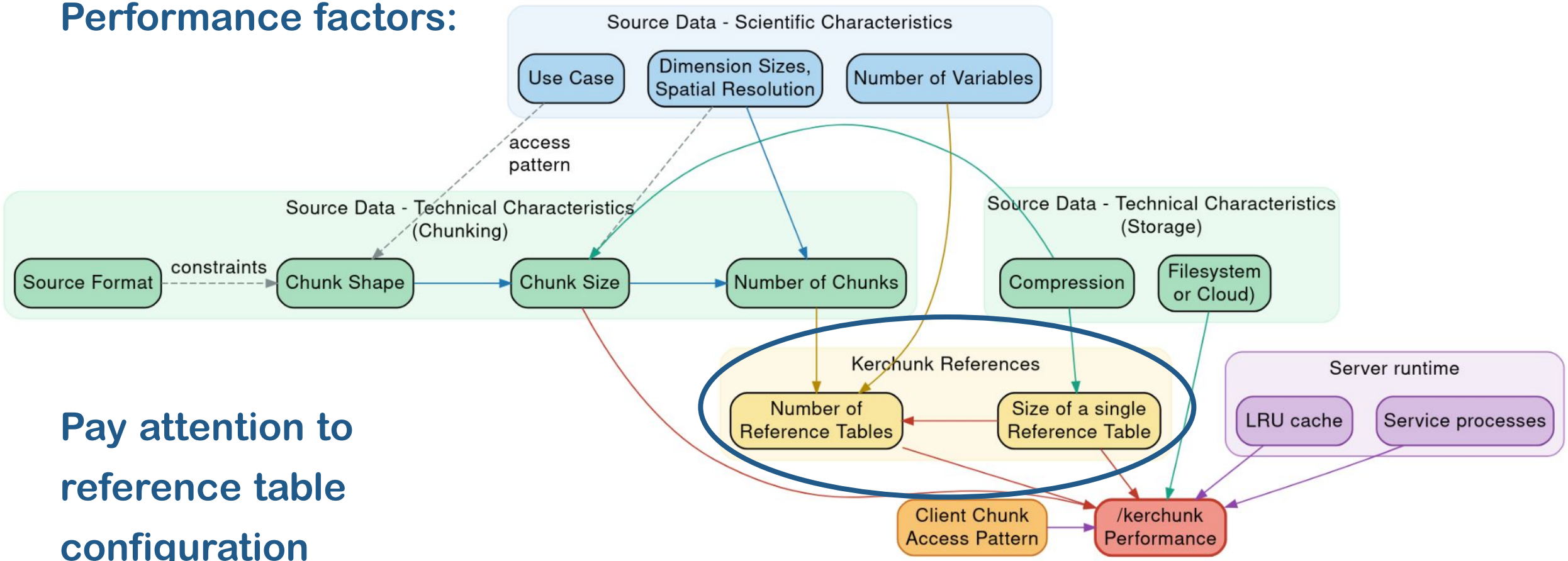
## Performance factors:




https://km-scale-cloud.dkrz.de

# Good Virtualization, bad virtualization

## Performance factors:




## Pay attention to reference table configuration

https://km-scale-cloud.dkrz.de

# A next-gen infrastructure component

Scalable!

→ 10PB with 64GB memory + a kerchunk API to avoid a memory bottleneck.

Data-as-a-Service

→ Dask recipes can be evaluated lazily on the fly to save storage space!

Makes data usable

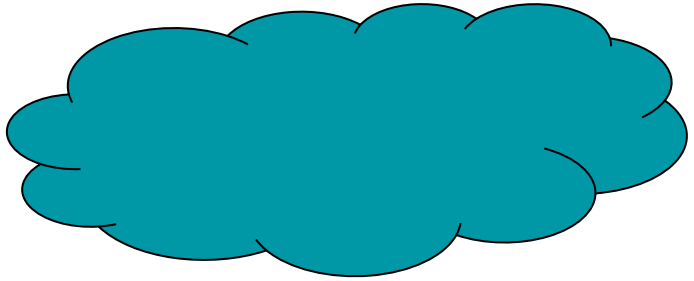
→ Lossy compression for bandwidth limited communities.

An Upgrade for NetCDF data server

→ provides uniform zarr (v2) from any resource: cacheable chunks, no redundant meta data (100MB per lat, lon and bounds variable for each file).

https://km-scale-cloud.dkrz.de

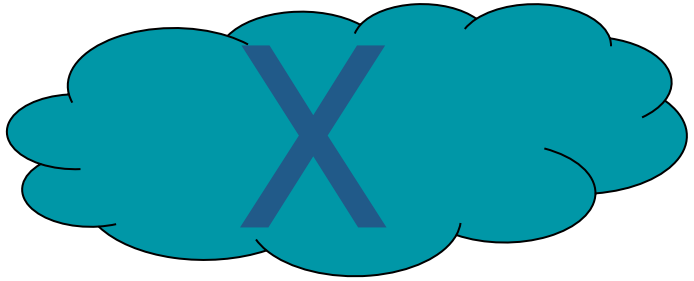
# Cloud object storage



| Feature | User and provider advantage |
| --- | --- |
| Fully open/no-auth access possible | Enables quick sharing of data |
| Independent from HPC | Increased availability |
| No filesystem. | No namespace conflicts when sharing scripts. Rather easy to scale the cloud storage. |

https://km-scale-cloud.dkrz.de

# Cloud object storage



But… (the system admin perspective)

- DKRZ uplink not sufficient to serve all interested users.
- Find a system maintainer
- No High-performance: We pay a lot for Lustre
- Proxies cannot handle too many chunks, especially when writing
- No good chunking possible without knowing access patterns

https://km-scale-cloud.dkrz.de

# Xpublish

eerie.cloud
data server

- creates http-accessible zarr-datasets for each opened xarray dataset. It maps dataset information like the dask chunks to the zarr data model and creates http endpoints for all chunks.
- allows to use a server-side dask cluster for customized processing on-the-fly
- based on fastAPI which enables to easily add plugins (stac, intake)

https://km-scale-cloud.dkrz.de