Data near processing support for climate data analysis

Stephan Kindermann, Carsten Ehbrecht
Deutsches Klimarechenzentrum (DKRZ)
Overview

- **Background / Motivation**
  - Climate community data infrastructure
  - Data processing near data centers needed
- **A component system for processing services**
- **A specific service example**
  - Code packaging and deployment
  - Deployment at Data Center / HPC Center / Home Institute / Cloud Infrastructure
- **Summary and Outlook**
Background: Climate Model Data Processing


Main driver for climate data infrastructure development: Intercomparison Projects

Climate Model Intercomparison Projects (CMIPs):
- CMIP3: ~ 35 TB
- CMIP5: ~ 3 PB = 100x CMIP3
- CMIP6:~ xx PB (> 10x CMIP5)

ESGF / IS-ENES Infrastructure

Data Processing:
- „download and process at home“ no longer feasible
- Data near processing
- Flexible approach ( ... science clouds are coming ... )

ESGF (meta-)data services
compute services

Home Institute
Community Portal

”Climate open science cloud“
Motivation

**Wanted:** A modular climate data processing solution

- Open interfaces
- No re-invention of the wheal: Build on stable open source approaches
- Modular, flexible installation, configuration and deployment system

**Approach:** An integration solution (*birdhouse*) with an extensible set of processing and data management services (*birds*)

- Based on OGC WPS services (+ other OGC service components)
- Flexible installation and deployment (conda, docker)
- Re-usable data management components (ESGF, cloud, thredds data sources)
The Birdhouse approach
Processing Approach – Example bird in birdhouse

Uniform set of packaging recipes
• Maintained on github
  https://github.com/bird-house/conda-recipes
• Available on binstar
  https://anaconda.org/birdhouse/packages

Environments
Libraries
Source code

(Executable) QA components
• qa_dkrz
• cf_checker
• cdo_info

Data quality assurance (QA) service
E.g. hummingbird
Processing Approach – Example bird in birdhouse

Environments
- Default anconda channel
- Birdhouse channel
- IOOS (U.S. Integrated Ocean observing System) channel

Libraries
- Netcdf4
- Udunits
- python

Source code
- qa_dkrz
- cf_checker
- cdo

(Executable) QA components
- qa_dkrz
- cf_checker
- cdo_info

Data quality assurance (QA) service
- E.g. hummingbird

Stephan Kindermann (DKRZ)
Processing Approach – Example bird in birdhouse

Packaging of components to OGC WPS service
- Recipes again hosted on github
- Include docker target

Environments
Libraries
Source code

(Executable) QA components
- qa_dkrz
- cf_checker
- cdo_info

Data quality assurance (QA) service
E.g. hummingbird
Processing Approach – Example bird in birdhouse

Client Interfaces:
- Ipython notebooks
- Birdhouse GUI
- Birdhouse command line

Environments
- Libraries
- Source code

(Executable) QA components
- qa_dkrz
- cf_checker
- cdo_info

Data quality assurance (QA) service
- E.g. hummingbird
Processing Approach – Example bird in birdhouse

? Real big climate data analysis?

→ Climate (meta-)data handling components / services are needed

(Executable) QA components
- qa_dkrz
- cf_checker
- cdo_info

Data quality assurance (QA) service
E.g. hummingbird

Client Interfaces (GUI, cmd, jupyter notebook,..)
Processing Approach – Example bird in birdhouse

? Real big climate data analysis?

→ Climate (meta-)data handling components / services are needed

→ Adhere to same birdhouse principles (recipes, packaging, distribution,..)

(Executable) QA components

- qa_dkrz
- cf_checker
- cdo_info

Data quality assurance (QA) service
E.g. hummingbird

Environments
Libraries
Source code

Netcdf4
Udunits
.. python
qa_dkrz

cdo

Client Interfaces
(GUI, cmd, jupyter notebook,..)
The Birdhouse approach
Status and Outlook

- **Birdhouse** provides modular system to develop and deploy web processing services
  - HPC center, Data center, (cloud) service provider, scientist
    - code, recipes: [https://github.com/bird-house](https://github.com/bird-house)
    - binstar channel: [https://conda.anaconda.org/birdhouse](https://conda.anaconda.org/birdhouse),
    - Docker hub: [https://hub.docker.com/u/birdhouse](https://hub.docker.com/u/birdhouse)
    - documentation: [http://birdhouse.readthedocs.org](http://birdhouse.readthedocs.org)
    - Demo installation: [http://mouflon.dkrz.de](http://mouflon.dkrz.de)

**Concrete deployment plans:**
- DKRZ: generic data services, e.g. quality control
- DKRZ, IPSL: ESGF data processing
- DKRZ, IPSL, BADC: ESGF data processing for Copernicus

**Integration plans:**
- ESGF: integration with other ESGF OGC WPS deployments at PCMDI, NASA, ..
- EUDAT: collaboration in context of EUDAT generic execution framework (GEF)
- ENVRI+: cross-community harmonization of OGC-WPS processing approaches
Thank You!

Questions?

Info / Contact:
• http://birdhouse.readthedocs.org
• kindermann@dkrz.de, ehbrecht@dkrz.de