**WPS based processing services for the Copernicus Climate Change Service**

A. Stephens, S. Kindermann, S. Denvil, W. Som de Ceuff  
Centre for Environmental Data Analysis (CEDA), German Climate Compute Centre (DKRZ), Institut Pierre Simon Laplace (IPSL), Royal Netherlands Meteorological Institute (KNMI)

**REFERENCES**

3. ESGF, https://esgf.llnl.gov/

**ARCHITECTURE OF COMPUTE NODE**

- Software Components of WPS Service
  - A WPS request (HTTP GET/POST) comes from a WPS client.
  - The Nginx/Gunicorn combination delegates the request to the PyWPS WSGI application.
  - Gunicorn - spawns several workers to use the available CPUs on a single compute node.
  - PyWPS - Python implementation of OGC Web Processing Standard.
  - Supervisor - used to start/stop and monitor services.
  - Processing outputs and status documents are web accessible by the Nginx file-server.
  - Token based access control (using OAuth) for WPS service.
  - WPS Processes are defined for project analysis toolbox, like C3S MAGIC diagnostics.
  - Processing Backend has read-only access to the climate data pool on file-system with CMIP5 climate model projections and observational data.
  - Using PyWPS scheduler extension (Slurm, GridEngine) to run process on a compute-cluster for scalability.

- Climate model projections and observational data.
- Providing the required data and services for global climate projections to the Climate Data Store (CDS) of the Climate Change Service (CDS) portal hosted at ECMWF, UK.
- Data Node - Consisting of vanilla Earth System Grid Federation (ESGF) index and data node.
- Compute Node - Providing compute facilities using the Web Processing Service (WPS) standard interface.
- Processing Backend - External software toolboxes to analyse model climate projections.
- Climate Model Projections (CMIP5, CORDEX) in filesystem cache.
- Quality Control -Climate Model Projections are selected for C3S and quality checked.
- Replication - Using Synda Python library for managing data movement.

**Service Interfaces exposed to Climate Data Store (CDS)**

- Web Processing Service (WPS) - standard interface for processing.
- OpenDAP - remote data access interface for NetCDF files.
- ESG search - adapted Solr search interface for ESGF data discovery.

**Federated C4CDS Nodes**

- Geographically distributed and highly available set of data and compute services.
- Federated between the leading European institutes: CEDA, IPSL and DKRZ.
- Using load-balancing across sites / failover strategy.
- All 3 sites of the same replicated local data pool.
- All 3 sites have the same (exact version) software stack using a common software deployment (SDDS).
- CEDA hosts the main node, IPSL and DKRZ take over service when needed.

**CLIMATE PROJECTIONS FOR THE CLIMATE DATA STORE (C4CDS)**

- Providing the required data and services for global climate projections to the Climate Data Store (CDS) of the Climate Change Service (C3S) portal hosted at ECMWF, UK.
- Data Node - Consisting of vanilla Earth System Grid Federation (ESGF) index and data node.
- Compute Node - Providing compute facilities using the Web Processing Service (WPS) standard interface.
- Processing Backend - External software toolboxes to analyse model climate projections.
- Climate Model Projections (CMIP5, CORDEX) in filesystem cache.
- Quality Control - Climate Model Projections are selected for C3S and quality checked.
- Replication - Using Synda Python library for managing data movement.

**Software Components of WPS Service**

- A WPS request (HTTP GET/POST) comes from a WPS client.
- The Nginx/Gunicorn combination delegates the request to the PyWPS WSGI application.
- Gunicorn - spawns several workers to use the available CPUs on a single compute node.
- PyWPS - Python implementation of OGC Web Processing Standard.
- Supervisor - used to start/stop and monitor services.
- Processing outputs and status documents are web accessible by the Nginx file-server.
- Token based access control (using OAuth) for WPS service.
- WPS Processes are defined for project analysis toolbox, like C3S MAGIC diagnostics.
- Processing Backend has read-only access to the climate data pool on file-system with CMIP5 climate model projections and observational data.
- Using PyWPS scheduler extension (Slurm, GridEngine) to run process on a compute-cluster for scalability.

**COPERNICUS CLIMATE CHANGE SERVICE**

- Copernicus is the European Union’s earth observation programme.
- Data from multiple sources: earth observation, satellites and in situ sensors.
- Thematic areas: land, marine, atmosphere, climate change, emergency management, security.
- Users: policymakers and public authorities.

**CLIMATE DATA STORE (CDS)**

- A climate data store will contain the geophysical information needed to analyse the climate change indicators in a consistent and harmonised way.
- This will combine the functions of a distributed data centre with a set of services and facilities for users and content developers.
- The store will provide data resources and computing facilities that can be utilised, for example, to develop improved climate reanalyses and seasonal forecasts.

**REPRESENTATIVE SERVICES**

- Climate Model Projections (CMIP5, CORDEX) in filesystem cache.
- Quality Control - Climate Model Projections are selected for C3S and quality checked.
- Replication - Using Synda Python library for managing data movement.

**SERVICE INTERFACES EXPOSED TO CLIMATE DATA STORE (CDS)**

- Web Processing Service (WPS) - standard interface for processing.
- OpenDAP - remote data access interface for NetCDF files.
- ESG search - adapted Solr search interface for ESGF data discovery.

**FEDERATED C4CDS NODES**

- Geographically distributed and highly available set of data and compute services.
- Federated between the leading European institutes: CEDA, IPSL and DKRZ.
- Using load-balancing across sites / failover strategy.
- All 3 sites of the same replicated local data pool.
- All 3 sites have the same (exact version) software stack using a common software deployment (SDDS).
- CEDA hosts the main node, IPSL and DKRZ take over service when needed.

**ARCHITECTURE OF COMPUTE NODE**

- Software Components of WPS Service
  - A WPS request (HTTP GET/POST) comes from a WPS client.
  - The Nginx/Gunicorn combination delegates the request to the PyWPS WSGI application.
  - Gunicorn - spawns several workers to use the available CPUs on a single compute node.
  - PyWPS - Python implementation of OGC Web Processing Standard.
  - Supervisor - used to start/stop and monitor services.
  - Processing outputs and status documents are web accessible by the Nginx file-server.
  - Token based access control (using OAuth) for WPS service.
  - WPS Processes are defined for project analysis toolbox, like C3S MAGIC diagnostics.
  - Processing Backend has read-only access to the climate data pool on file-system with CMIP5 climate model projections and observational data.
  - Using PyWPS scheduler extension (Slurm, GridEngine) to run process on a compute-cluster for scalability.

- Climate Model Projections (CMIP5, CORDEX) in filesystem cache.
- Quality Control - Climate Model Projections are selected for C3S and quality checked.
- Replication - Using Synda Python library for managing data movement.

**C3S MAGIC - CLIMATE DATA ANALYSIS**

- Developed by KNMI, eScience Center, DLR and others.
- Used by C4CDS as Processing Backend for CMIP5 climate model projections.
- To calculate standardized characteristics from available climate model output.
- ESMValTool - To develop and deliver an enhanced version of the ESMValTool software.
- Metrics - computes and displays a wide set of performance metrics and diagnostics.
- Multi-model products - To combine the climate information generated by various climate models into a single estimate of any future climate signal.
- Climate index time-series - To compute single-model and multi-model time series of climate indices.
- Tailored products - To assure that specific needs of envisaged end users in the selected economic sectors are facilitated by the software.

**SDDS - SOFTWARE DEPLOYMENT DEPENDENCY SOLUTION**

- Manage and deploy Software for C4CDS Compute Nodes
  - Requirement - To deploy codes from external projects, such as C3S MAGIC / ESMValTool, into the C4CDS Compute Node.
  - SDDS - Consists of a software environment and application, managed through a Github repository, which includes a basic template (contributed by Birdhouse) of a working WPS service (PyWPS).
  - Conda - The template uses a Conda "environment" to record the software dependencies and to build a reproducible software installation.
  - Docker - Used to provide the Compute Node through containers.
  - Ansible - Ansible and Buildout are used to setup a WPS (PyWPS) with all services (Supervisor, Gunicorn, Nginx) and configuration files.
  - SDDS is used to set-up C4CDS Compute Nodes for CMIP5 (global) and CORDEX (regional) climate projections with a specific analysis toolbox.

**NEXT STEPS**

- Further integration of MAGIC codes.
- Roll-out of C4CDS at all three sites.
- Improved SDDS - using a template generator, replacing Buildout by Ansible, deployment in Docker Cluster.
- Using ESGF OAuth service for secure tokens.