Automating Data Synchronization, Checking, Ingestion and Publication for CMIP6

Ag Stephens & Alan Iwi
(STFC Centre for Environmental Data Analysis)
Thanks to Emma Hibling & Mark Elkington
(Met Office)
Ingest-publish workflow: Requirement

**Data Provider**
- Check facets/values against defined vocabularies;
- Check content against MIP tables;
- Check spatial/temporal characteristics;
- Define version.

**Holding Area**
- Create version directory;
- Move files into versioned directory structure;
- Use standard tool to generate mapfiles;
- Store mapfiles in a standardised structure at all nodes;
- Should be able to re-generate all THREDDS records from the mapfiles.

**Data Archive**
- Generate mapfiles
- Use standard tool to generate mapfiles;
- Store mapfiles in a standardised structure at all nodes;
- Should be able to re-generate all mapfiles from the archive.

**Metadata**
- Generate THREDDS record
- Should be able to re-generate all THREDDS records from the mapfiles.

**Data Node**

**Index Node**
- Publish to ESGF Search
- This is the part that makes the data visible to portals and other search tools linking to ESGF.

**SYNCHRONISE**
For different datasets (ESGF or other) we can imagine there are different processes.

Common pattern:
- Each task is isolated.
- "UNDO" behaviour may be desirable (e.g. "remove files", "unpublish")
- Tasks may be managed by:
  - Different (Unix) users
  - With different access levels
  - On different servers.
MOHC-to-ESGF pipeline - CMIP6

Write model outputs

MASS (tape store)

CEDA (& ESGF)

Message Queue (RabbitMQ)

pull and ingest data

With special thanks to
Emma Hibling and Mark Elkington (Met Office)
CEDA REceive-to-Publish Pipeline (CREPP)

We have called it "CREPP" - currently an internal ("cedadev") GitHub project.

Client-server architecture:

- "Server" is actually just a DB.
- "Clients" are any number of Controllers on any number of machines.
CREPP: Key concepts

- **Dataset ID** (ESGF) is the unit of granularity across the system, e.g.:

  cordex.output.CAS-44.MOHC(ECMWF-ERAINT.evaluation.r1i1p1.
  MOHC-HadRM3P.v1.mon.clt.v20150608

- All Controller actions should be **atomic** wherever possible. This will maximise the chances of “DO” and “UNDO” being possible for each Process Stage.
The Data Model is built in Django...each type of box is a relational DB table.
The Data Model is built in Django...each type of box is a relational DB table.
Data Model

The Data Model is built in Django...each type of box is a relational DB table.

Process Chain

- Process Stage (1)
- Process Stage (2)
- Process Stage (3)

Status (1)

Status (2)

Status (3)

Dataset

File (1)
- Checksum
- Symlink

File (2)
- Checksum
- Symlink

File (3)
- Checksum
- Symlink

Global Settings

Event

On change...
## "DO" and "UNDO" actions

<table>
<thead>
<tr>
<th>Action Type</th>
<th>QC</th>
<th>Ingest</th>
<th>Publish (TDS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DO</strong></td>
<td>Run QC</td>
<td>Create directories; move files</td>
<td>Publish to THREDDS</td>
</tr>
<tr>
<td><strong>UNDO</strong></td>
<td>-</td>
<td>Move files back to cache; remove directories</td>
<td>Unpublish from THREDDS</td>
</tr>
</tbody>
</table>
So how does it work?

Controller: Sync

Controller: QC

Controller: Ingest

Controller: Publish

I've added a dataset

Database

QC needed?

CEDA disk

Met Office

Great, I'll add them to the db

Can I ingest some data?

Are we ready for ESGF?
Key components

Controller: Sync

Controller: QC

Controller: Ingest

Controller: Publish

[Diagram showing key components such as databases, RabbitMQ, and ESGF, with labels for each controller.]
A Process Chain

Controller: Sync

Controller: QC

Controller: Ingest

Controller: Publish

CEDA disk

RabbitMQ

Met Office

ESGF

National Centre for Atmospheric Science

National Centre for Earth Observation
Is there any work for me?

I've added a dataset
I've added a dataset

Controller: Sync

Controller: QC

Controller: Ingest

Controller: Publish

Great, I'll add them to the db

Met Office

Database

CEDA disk

ESGF

National Centre for Atmospheric Science

National Centre for Earth Observation
Can I ingest some data?
Controller: Sync

Controller: QC

Controller: Ingest

Controller: Publish

Ready to publish?
Is there any work for me?

I've added a dataset

QC needed?

Ready to publish?

Can I ingest some data?

Great, I'll add them to the db
Example Process Chain: CMIP6

<table>
<thead>
<tr>
<th>#</th>
<th>Process Stage</th>
<th>&lt;user&gt;@&lt;host&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Message Queue</td>
<td>cedauser@mass-cli1</td>
</tr>
<tr>
<td>2</td>
<td>QC</td>
<td>badc@ingest1</td>
</tr>
<tr>
<td>3</td>
<td>Ingestion</td>
<td>controller: badc@ingest1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>workers: badc@ingest_cluster</td>
</tr>
<tr>
<td>4</td>
<td>Mapfile generation</td>
<td>badc@ingest1</td>
</tr>
<tr>
<td>5</td>
<td>CEDA metadata records</td>
<td>badc@ingest1</td>
</tr>
<tr>
<td>6</td>
<td>ES-DOC record generation</td>
<td>badc@ingest1</td>
</tr>
<tr>
<td>7</td>
<td>ESGF publication: DB</td>
<td>root@esgf-data1</td>
</tr>
<tr>
<td>8</td>
<td>ESGF publication: TDS (without reinit)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ESGF publication: TDS main catalog</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ESGF publication: index</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Consume RabbitMQ Message</td>
<td></td>
</tr>
</tbody>
</table>

Many Process Stages are re-usable. The Controller can be re-deployed in a Process Stage of multiple Process Chains (for different projects).
Technology choices

Python
PostgreSQL
Django
RabbitMQ
Apache
Bootstrap
Django front-end

The CEDA Dataset Pipeline App

Many datasets have an ingest "pipeline" that requires multiple processing stages that run on various servers under different user IDs. This app provides a front-end to manage and monitor the processes as each dataset travels through the system.

The first project to test this system is "CMIP6-MOHC" (Met Office Hadley Centre climate simulations for the 6th Coupled Model Intercomparison Project). Select listings from the Views menu above to monitor existing datasets being processed. If you log in you will also be able to pause or resume the system and run other management functions.

Welcome to our superb app, it will revolutionise your life. Try some links:

- View Datasets
- View Files
- View Chains
- View Events
Events view - allows real-time monitoring

Events
An Event is logged each time a Controller has completed (i.e. succeeded or failed to process) a Dataset. This page allows you to view Events in the system based on a set of search criteria.

Show Search Filters

<table>
<thead>
<tr>
<th>Time</th>
<th>Dataset</th>
<th>Process Stage</th>
<th>Message</th>
<th>Action Type</th>
<th>Succeeded?</th>
<th>Withdrawn?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug. 18, 2016, 10:55 a.m.</td>
<td>cmip5.output1.MOHc.HadGEM2-ES.rcp45.day.atmos.cfDay.r1i1p1.v20120114</td>
<td>Publish</td>
<td>do</td>
<td>SUCCEEDED</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Aug. 18, 2016, 10:55 a.m.</td>
<td>cmip5.output1.MOHc.HadGEM2-ES.rcp45.day.atmos.cfDay.r1i1p1.v20120114</td>
<td>Ingest</td>
<td>do</td>
<td>SUCCEEDED</td>
<td>False</td>
<td></td>
</tr>
<tr>
<td>Aug. 18, 2016, 10:55 a.m.</td>
<td>cmip5.output1.MOHc.HadGEM2-ES.rcp45.day.atmos.cfDay.r1i1p1.v20120114</td>
<td>QC</td>
<td>do</td>
<td>SUCCEEDED</td>
<td>False</td>
<td></td>
</tr>
</tbody>
</table>
Special action: "withdraw"

If the MOHC spots a problem with a dataset they may change the state (via a RabbitMQ message) from "available" to "withdrawn".

This triggers action at CEDA:

1. If already ingested/published:
   - hide?/unpublish
2. If not ingested/published:
   - **DO NOT** ingest/publish.
   - Acknowledge/consume the message with "available" state.
Integration issues
Integration with CIM2

- We intend to adopt the **cdf2cim** tool in the publication workflow to extract *Simulation* records.
- These will be automatically generated and pushed to the ES-Doc server.
Will CREPP handle replication?

NO:

• Replication nodes are being developed to work with Synda (using GridFTP where possible).
• We expect a different set of recipes/rules to be managing replication.

(or) YES:

• Replication is a set of tasks on different servers that (can) use the ESGF Dataset as their unit of granularity.
• The publish/unpublish components could be managed using CREPP.

Need to understand more about Synda post-processing workflows before we decide on this.
Recovery response

1. The (only) database goes offline:
   - All Controllers \textit{wait...fail...undo...stop.}
   - Manual recovery to backup db.
   - Re-start all Controllers.

2. Individual Controller (or server it runs on) fails:
   - Some datasets are in "claimed" state
   - Remove claims
   - Let them be re-run by new instance of the Controller

3. Urgent software upgrade required
   - Switch on Global Pause
   - Current tasks will run to completion; then all will pause; it is safe to stop all and roll out new software before re-starting
• Code base developed
• Being tested on operational platform - with first MOHC test simulations
• Individual Controllers being written to handle specific process stages
• Needs to be ready soon!
Further information

- Centre for Environmental Data Analysis
  - [http://www.ceda.ac.uk](http://www.ceda.ac.uk)
  - support@ceda.ac.uk
- CREPP code (currently internal to CEDA):
  - [https://github.com/cedadev/crepp](https://github.com/cedadev/crepp)
- Met Office pipeline (climate-dds) code (internal):
  - [https://code.metoffice.gov.uk/trac/cdds/](https://code.metoffice.gov.uk/trac/cdds/)

If you are interested in finding out more please contact me on:

ag.stephens@stfc.ac.uk
## CREPP Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue</td>
<td>RabbitMQ instance of a queue of Dataset held in MASS with a Met Office status associated with them.</td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td>A message in the Queue that specifies the Met Office status related to a single Dataset.</td>
<td>Status can be: available</td>
</tr>
</tbody>
</table>
# CREPP Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>A set of files that have a complete ESGF DRS description including the version component.</td>
<td>Note that this term has a specific meaning throughout the system.</td>
</tr>
<tr>
<td>Controller</td>
<td>A process running on a node that communicates with the DB and manages Workers running locally or remotely.</td>
<td>A Controller can run as a daemon process, might be invoked by cron or other means. The key aspects are that it routinely polls the DB for its next set of tasks and manages Workers to perform Tasks.</td>
</tr>
</tbody>
</table>
## CREPP Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Meaning</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worker</td>
<td>A process running on a node that performs a distinct task and then terminates.</td>
<td>Workers may be invoked locally or via a scheduler (e.g. on the “Ingest” cluster).</td>
</tr>
<tr>
<td>Task</td>
<td>A single complete Action undertaken by a Worker when operating on a single Dataset at a given Process Stage.</td>
<td>Each completed Task will result in the DB being updated. Where successful, this will update the status to trigger the next Controller. An entry will also be made in the Event Log.</td>
</tr>
<tr>
<td>Term</td>
<td>Meaning</td>
<td>Comments</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Event Log</td>
<td>A table in the DB that records ALL outcomes from Tasks per Dataset.</td>
<td></td>
</tr>
<tr>
<td>Process Chain</td>
<td>An ordered set of Process Stages.</td>
<td>Multiple ESGF “projects” may use the same chain and a single project might use multiple chains for different data providers.</td>
</tr>
<tr>
<td>Process Stage</td>
<td>A component of the processing chain that is managed by a Controller.</td>
<td>E.g. “Run QC”, “Create mapfiles”, etc.</td>
</tr>
<tr>
<td>Action Type</td>
<td>The attribute of the Task that specifies the type of behaviour.</td>
<td>Can be: “Do”, “Undo”</td>
</tr>
</tbody>
</table>