eePEcAn

Integrating Google Earth Engine with the PEcAn project

Cesar Luis Aybar Camacho

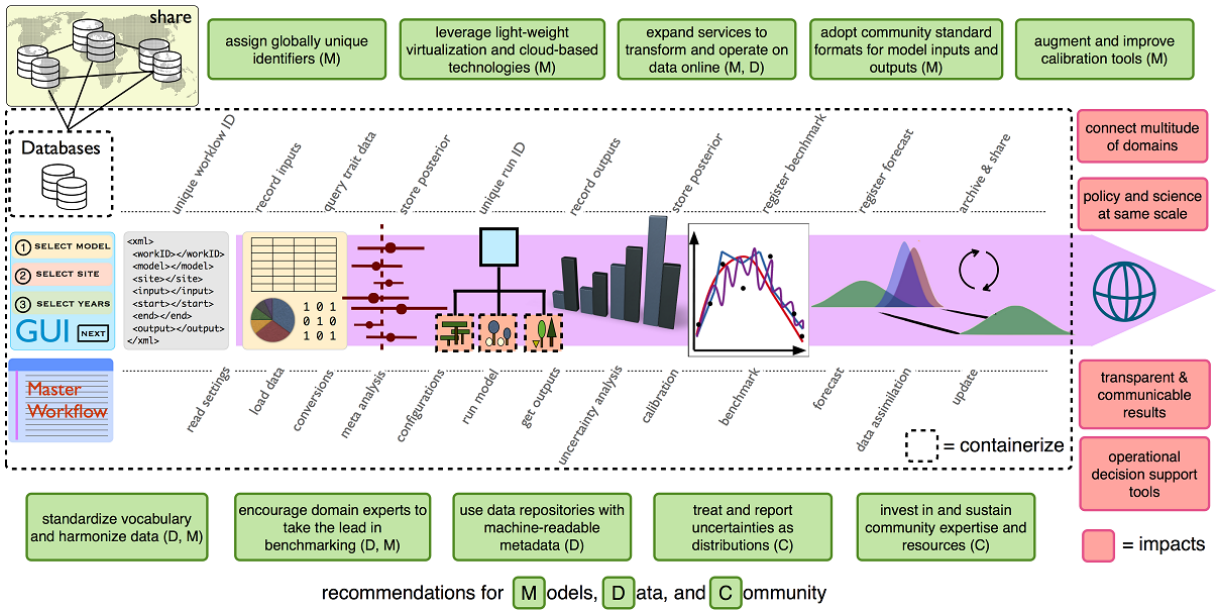
Github repository
https://github.com/csaybar/eePEcAn

Year of publication
2020

What is PEcAn?

Ecosystem models are essential for the development of scientific and operational applications, which allow to understand the terrestrial biosphere and, for instance, forecasting changes in the carbon cycle and its impact on natural and human systems. The amount of data being collected and produced is increasing on daily basis as we enter the "big data" era, but only a fraction of this data is being used to constrain models (Cowdery, et. al, 2018). Predictive Ecosystem Analyzer (PEcAn) is a framework, which is developed mainly in R, for ecosystem modeling that offers researches the opportunity to create 100% reproducible and transparent data analysis pipelines. Currently, PEcAn is comprised of:

- An application program interface (API) that provides a friendly-user I/0 interface to interact with 11 different ecosystem models.
- An uncertainties data analysis module. web-based user interface and visualization tools (that use shiny).
- An extensible collection of modules to handle specific types of analyses, model-data syntheses, and data processing.



How run PEcAn?

There are three principal ways to install PeCAN in a system.

- Virtual machine
- PEcAn Docker
- PEcAn OS specific installation

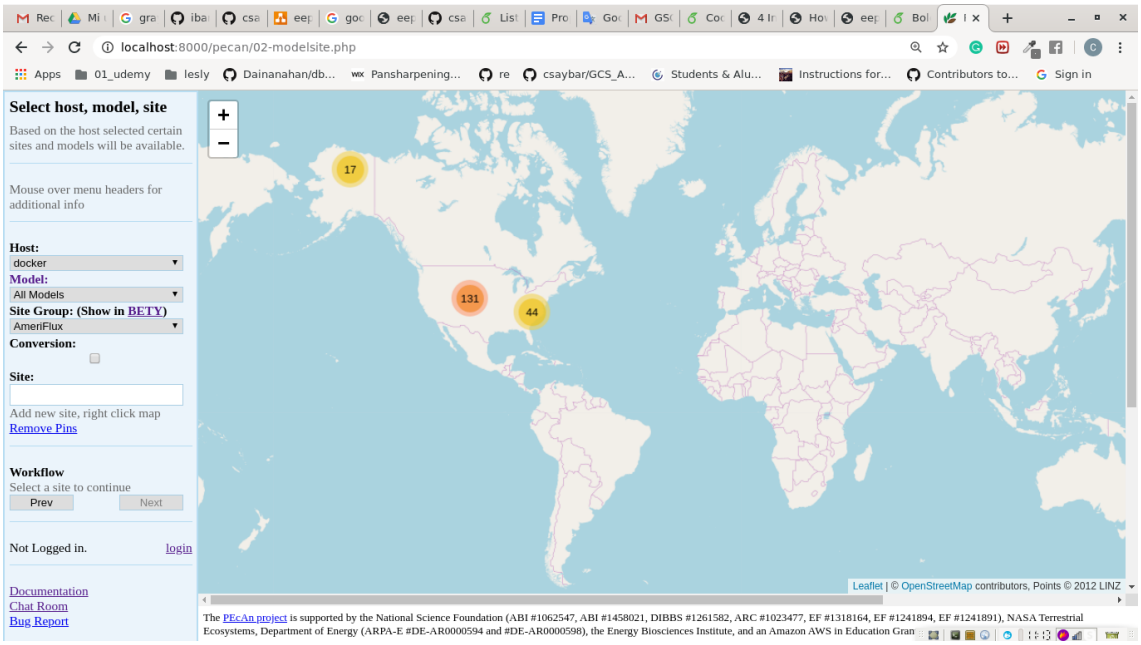
The most intuitive and user-friendly form to install PEcAn is using docker. All you have to do is create a *docker-compose.yml* file, adapt it according to the documentation, and then run:

```
docker-compose -p pecan up -d
```

```
aybarpc01@debian:~/pecan$ docker-compose up -d
pecan_rabbitmq_1_ee5d598039c6 is up-to-date
pecan_portainer_1_c9501b5c2bcd is up-to-date
pecan_thredds_1_f837b6b81128 is up-to-date
pecan_docs_1_5619389a697e is up-to-date
pecan_minio_1_29e13dd9acbe is up-to-date
Starting pecan_rstudio-nginx_1_182e7ddf49b6 ...
pecan_rstudio_1_2777e7c88156 is up-to-date
pecan_traefik_1_d7d3e64a903c is up-to-date
pecan_postgres_1_f4e7bf5abc40 is up-to-date
pecan_sipnet_1_30801943739b is up-to-date
pecan_monitor_1_9a7852d17305 is up-to-date
pecan_ed2_1_a4b5a9020bc3 is up-to-date
pecan_maespa_1_1add93f388be is up-to-date
pecan_bety_1_d8bd092c79e6 is up-to-date
pecan_web_1_dabf6edba4ef is up-to-date
Starting pecan_rstudio-nginx_1_182e7ddf49b6 ... done
aybarpc01@debian:~/pecan$
```

If all of the containers started successfully, you should be able to access the various components from a browser via the following URLs (if you run these commands on a remote machine replace localhost with the actual hostname).

- PEcAn web interface (running models) – http://localhost:8000/pecan/ (NOTE: The trailing backslash is necessary.)
- PEcAn documentation and home page – http://localhost:8000/
- BETY web interface – http://localhost:8000/bety/
- File browser (minio) – http://localhost:8000/minio/
- RabbitMQ management console (for managing queued processes) – http://localhost:8000/rabbitmq/
- Traefik, webserver showing maps from URLs onto their respective containers – http://localhost:8000/traefik/
- Monitor, service that monitors models and shows all models that are online as well as how many instances are online and the number of jobs waiting. The output is in JSON – http://localhost:8000/monitor/



I/O PEcAn

The PEcAn system is configured using a XML file, often called *pecan.xml*. You can find a complete list of all configurable parameters here. The input data is used is used in **data assimilation, calibration and validation**. The principal parts of the PEcAn framework are:

- TRAIT -> Query the trait database for data and priors.

- META -> Run the PEcAn meta.analysis.
- CONFIG -> Write model specific configs.
- MODEL-> Start ecosystem model runs.
- OUTPUT -> Get results of model runs.
- ENSEMBLE -> Run ensemble analysis on model output.
- FINISHED -> Pecan workflow complete.

2 GEE sync PEcAn - Project Idea

Establishing GEE and PEcAn communication. Google Earth Engine (GEE) is a cloud service that makes remote sensing data available for Earth System science. GEE has an API through which certain geopspatial analyses could be performed. We want to establish an efficient GEE-PEcAn link to ingest the outputs of such analyses into the PEcAn workflow seamlessly. PEcAn team will provide the initial example code that runs on the GEE servers, the GSoC participant will build the link that pulls the results from GEE to PEcAn, as well as submitting the code remotely to the GEE servers through the PEcAn workflow.

3 eePEcAn: Implementation plan

eePecAn: Module to Integrate Google Earth Engine and PecAn
eePecAn is a three-part project:

1. **Handling Authentication and Initialization using R:** Google Earth Engine currently does not support R, which is the main language in which PecAn is developed. So the first step is to create a synchronization between the Earth Engine Python API and R. For this step will be used **reticulate** very similar to what was seen rgee with the difference that all credentials will be previously stored in the file **pecan.xml** (the user could change their credentials to preference) and then be reconfigured the /config of the system.

2. **Extract time series by points(eePecan_extract)**: The second part looks for a robust module for obtaining time series. The idea is to improve and simplify the rgee::ee_extract function so that it is not limited to only 5000 elements per request. In order to guarantee the efficiency and speed of the process. *eePecan_extract* will be carried out using ee.Join objects. The function will return a data.frame and would be as follows:

```
library(PEcAn.all)
eePecan_extract(
  ImageCollection = "MODIS/006/MOD13A2",
  sf_geometry = my_points,
  startdate = "2000-01-01",
  lastdate = "200-12-31",
)
```

   In addition to the implementation, I will add different unit tests, in order to guarantee a proficient performance.

3. **Improvement of the shiny app to display PEcAn outputs**: The last GSoC, Shuang Lu enhance the PEcAn shiny app to visualizing model output data alongside external data. I intend to give the user the possibility of adding external data only by adding the **GEE collection name, the band name**

and the correction scale. With only these arguments the time series will appear automatically (the date range will be obtained from the file **eePEcAn.xml**).

```
<run>
  <site>
    <id>772</id>
    <met.start>2002-01-01 00:00:00</met.start>
    <met.end>2005-12-31 00:00:00</met.end>
  </site>
  <inputs>
    <met>
      <id>5000000005</id>
    </met>
  </inputs>
  <start.date>2002/01/01</start.date>
  <end.date>2005/12/31</end.date>
</run>
```

4 Timeline

Before April 30:

- To familiarize myself completely with the PEcAn infrastructure.
- Learn more about docker compose.
- Study more about PostgreSQL and how it is sync with docker.

April 30 – June 01 (Before the official coding time)

- To do self coding to sync shiny and Google Earth Engine.
- During this period I will remain in constant touch with my mentor and the PEcAn community. I will remain active on slack and Mailing lists to discuss and finalize the modifications (if any) that need to be realized.
- With the help of my mentor, I will become absolutely clear about my future goals, the Earth Engine - PEcAn implementations that need to be done as well as the approach that I will follow to integrate my functions with the PEcAn project.

June 01 – June 29 (Official coding period starts)

- Implementation of Authentication and Initialization of Google Earth Engine using R inside the PEcAn ecosystem.
- Implementation of **eePecan_extract**.
- Testing of all my code using testthat.

June 30 – August 10

- Integration between **eePecan_extract** and the shiny app to display PEcAn outputs.
- Test and document the existing code more thoroughly.
- Further refine tests and documentation for the whole project and PR.

A buffer of two weeks has been kept for any unpredictable delay.

5 Personal details

Full Name: Cesar Luis Aybar Camacho

Age: 25 years

Sex: Male

Study level: Erasmus Graduate Student at the CDE program 2020-2022.

Why PEcAn?: PEcAn is one of the few projects in Earth Science that is fully intended to be transparent and reproducible. I personally believe that the use of technology such as distributed version-control system, docker containers, and CI/CD are the right way for doing the disruptive and high-impact science. I am definitely interested in learning from top-level researchers that use this stack on their daily basis.

Why this project?: Google Earth Engine along with tensorflow are my favorite technologies, after spending more than a year learning about them. I have never used them in a tangible project with real results. I firmly believe that the GEE - PecAn project is the perfect opportunity to put everything I have learned into practice.