Pangeo

Traitement des données PR/PF ARGO

PANGEO

A community platform for Big Data geoscience

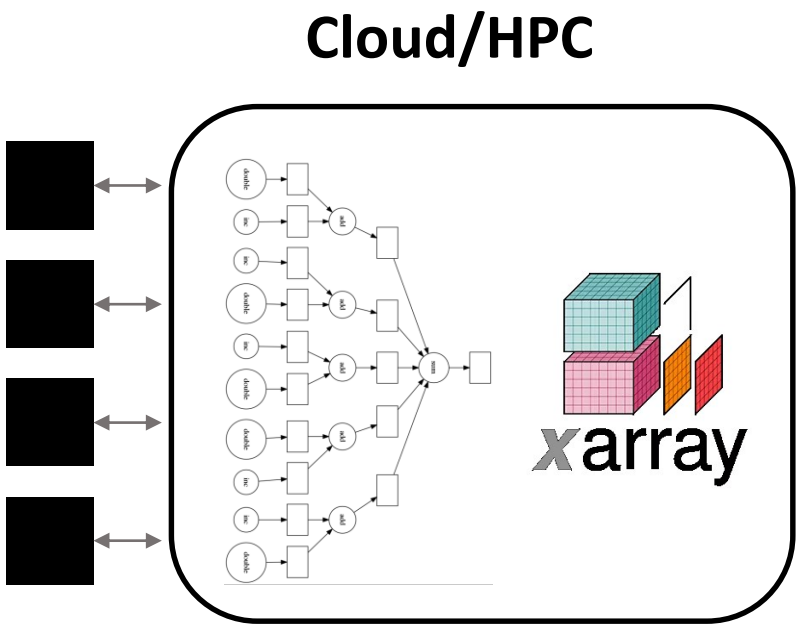
Sommaire

Pile Pangeo


Iris
TensorFlow
cartopy
GCM
DASK
xarray
matplotlib
scikit
NumPy
IPython
jupyter
python

Architecture Pangeo

Données, prêtes à l'analyse, stockées et cataloguées sur un système de stockage distribué accessible à l'échelle globale (p. ex. S3, GCS)



Système de calcul parallèle construit sur Kubernetes ou HPC.
Dask dit aux nœuds ce qu'ils doivent faire.

Jupyter pour un accès interactif sur des systèmes distants.

Xarray fournit des structures de données et une interface intuitive pour interagir avec les ensembles de données.

JupyterHub

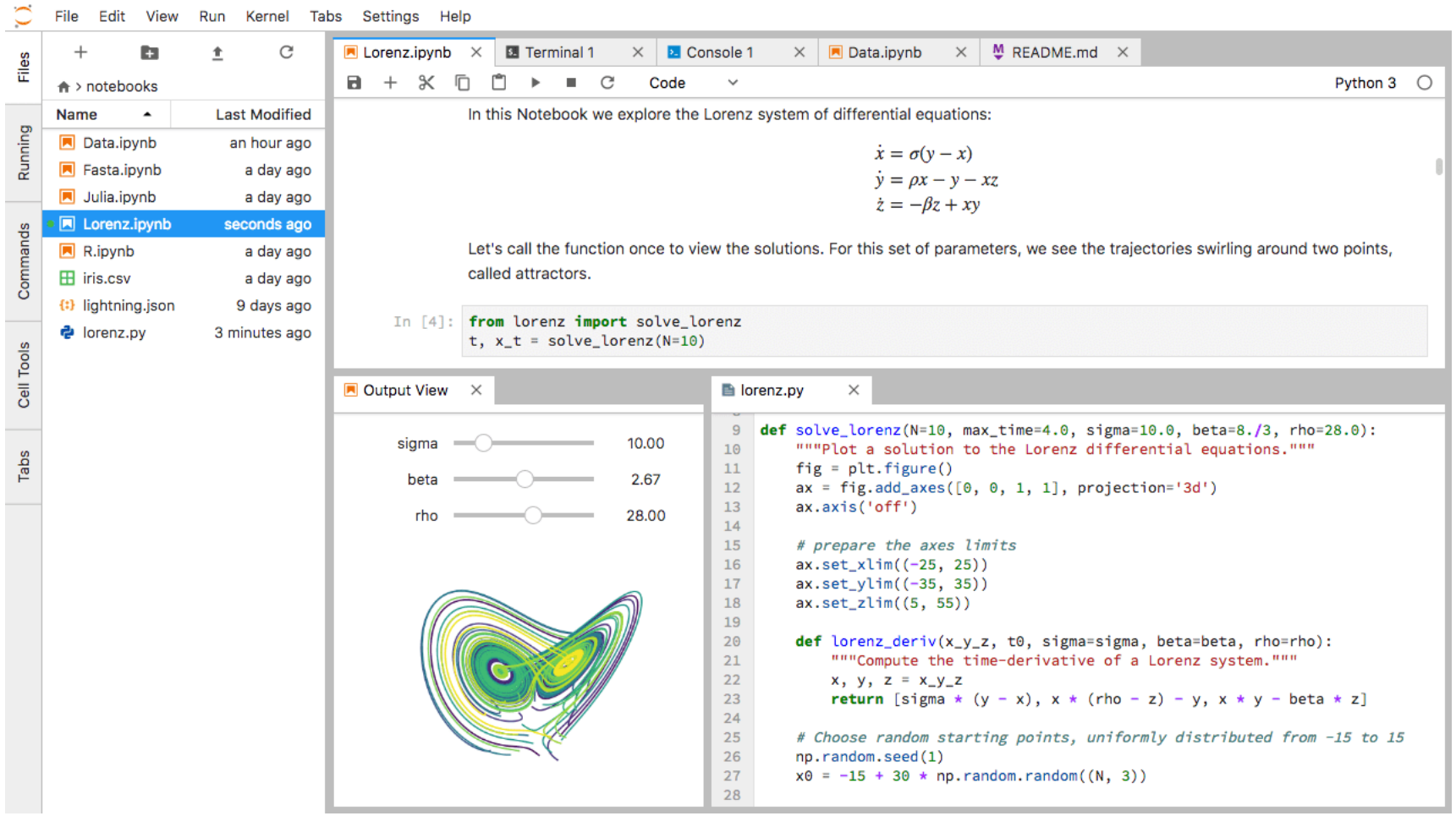
JupyterHub apporte la puissance des blocs-notes à des groupes d'utilisateurs.

Il permet aux utilisateurs d'accéder à des environnements et des ressources informatiques sans leur imposer des tâches d'installation et de maintenance.

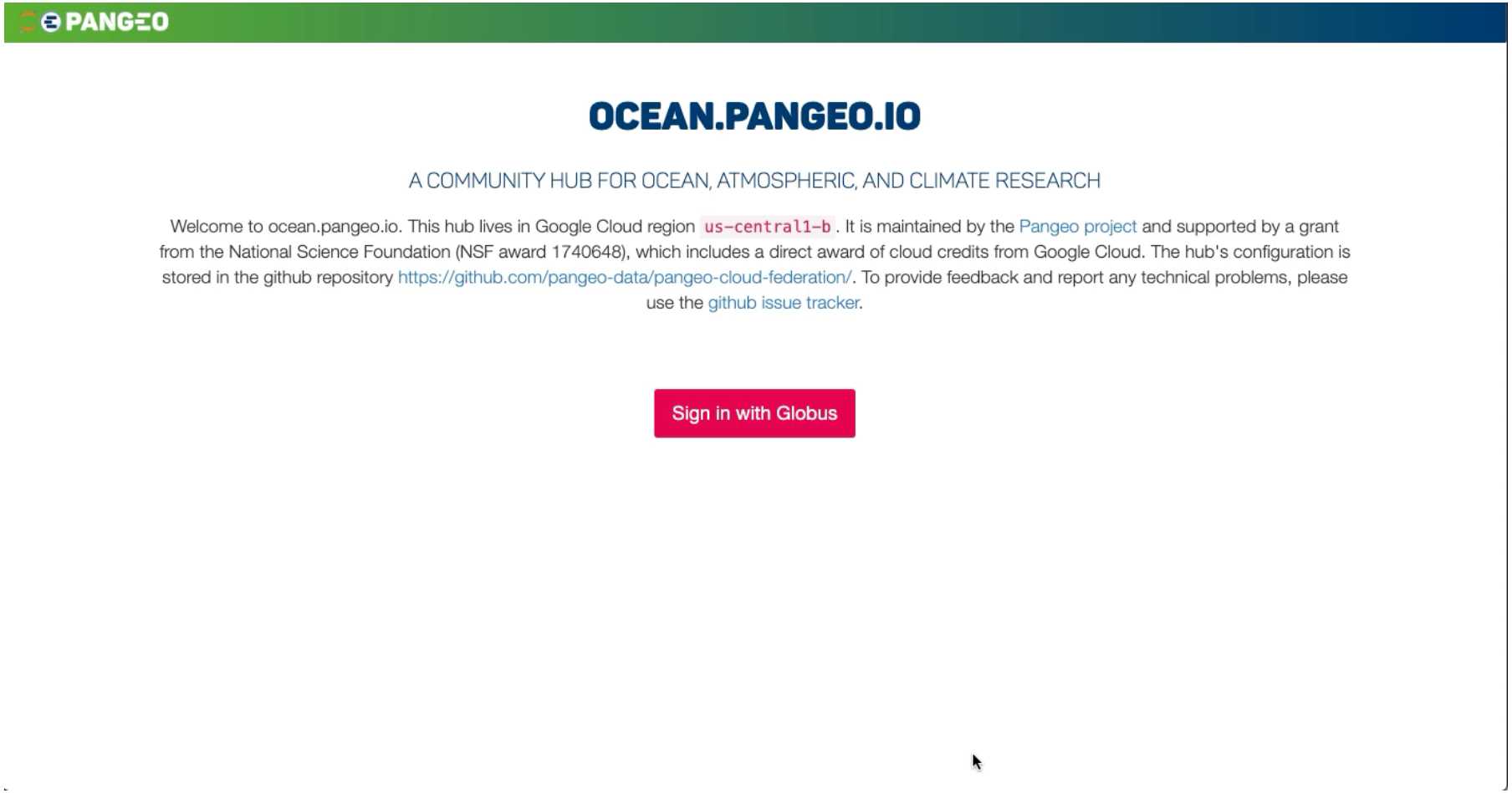
Les utilisateurs - y compris les étudiants, les chercheurs et les spécialistes des données - peuvent effectuer leur travail dans leurs propres espaces de travail sur des ressources partagées qui peuvent être gérées efficacement par les administrateurs système.

JupyterHub fonctionne dans le nuage ou sur votre propre matériel, et permet de servir un environnement préconfiguré. Il est personnalisable et évolutif, et convient aux petites et grandes équipes, aux cours universitaires et aux grandes infrastructures.

JupyterHub


File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
notebooks
Name
Last Modified
Data.ipynb an hour ago
Fasta.ipynb a day ago
Julia.ipynb a day ago
Lorenz.ipynb seconds ago
R.ipynb a day ago
iris.csv a day ago
lightning.json 9 days ago
lorenz.py 3 minutes ago
Lorenz.ipynb
Terminal 1
Console 1
Data.ipynb
README.md
Code
Python 3
In this Notebook we explore the Lorenz system of differential equations:
$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$
Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.
In [4]:
from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
Output View
sigma 10.00
beta 2.67
rho 28.00
lorenz.py
def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
"""Plot a solution to the Lorenz differential equations."""
fig = plt.figure()
ax = fig.add_axes([0, 0, 1, 1], projection='3d')
ax.axis('off')
# prepare the axes limits
ax.set_xlim((-25, 25))
ax.set_ylim((-35, 35))
ax.set_zlim((5, 55))
def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
"""Compute the time-derivative of a Lorenz system."""
x, y, z = x_y_z
return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
# Choose random starting points, uniformly distributed from -15 to 15
np.random.seed(1)
x0 = -15 + 30 * np.random.random((N, 3))

JupyterHub


PANGEO
OCEAN.PANGEO.IO
A COMMUNITY HUB FOR OCEAN, ATMOSPHERIC, AND CLIMATE RESEARCH
Welcome to ocean.pangeo.io. This hub lives in Google Cloud region us-central1-b. It is maintained by the Pangeo project and supported by a grant from the National Science Foundation (NSF award 1740648), which includes a direct award of cloud credits from Google Cloud. The hub's configuration is stored in the github repository https://github.com/pangeo-data/pangeo-cloud-federation/. To provide feedback and report any technical problems, please use the github issue tracker.
Sign in with Globus

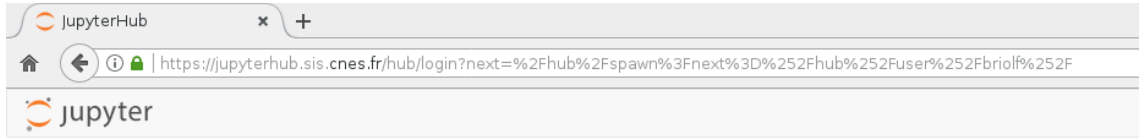
JupyterHub

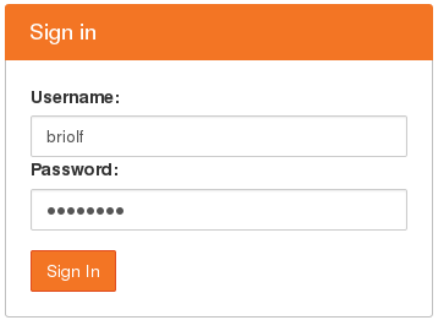
JupyterHub est déployé sur HAL

L'ouverture à l'Internet est en cours

Disponible depuis SSH/Firefox

Contacter Gérald si vous voulez l'utiliser





Binder

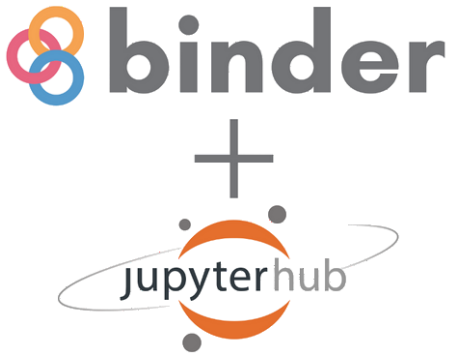
Binder vous permet de créer des environnements informatiques personnalisés qui peuvent être partagés et utilisés par de nombreux utilisateurs distants.

Il est alimenté par BinderHub, qui est un outil open-source qui déploie le service Binder dans le cloud.

Un tel déploiement est disponible, sur https://binder.pangeo.io/, et son utilisation est gratuite.

Les contributions récentes du CNES:

- https://github.com/CNES/pangeo-pyinterp
- https://github.com/CNES/pangeo-pytide
- https://github.com/fbriol/pangeo-argo



Binder

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Security | Insights | Settings

Branch: master ▾ **pangeo-argo** / binder / **Dockerfile**

fbriol first commit

1 contributor

2 lines (1 sloc) | 47 Bytes

```
1
2  FROM pangeo/pangeo-notebook-onbuild:2019.09.23
```

fbriol / **pangeo-argo**

Unwatch ▾ 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Security | Insights | Settings

Branch: master ▾ **pangeo-argo** / binder / **environment.yml**

Find file | Copy path

fbriol new try d148371 3 days ago

1 contributor

7 lines (6 sloc) | 78 Bytes

Raw | Blame | History

```
1  name: pytide
2  channels:
3    - conda-forge
4  dependencies:
5    - pyinterp
6    - pyarrow
```

README.md

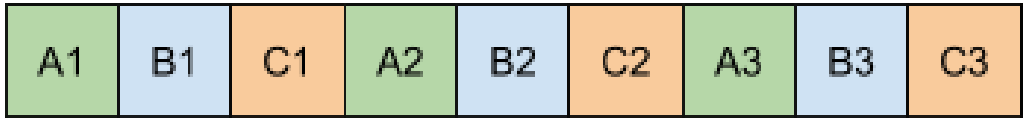
launch binder

https://binder.pangeo.io/v2/gh/fbriol/pangeo-argo/master

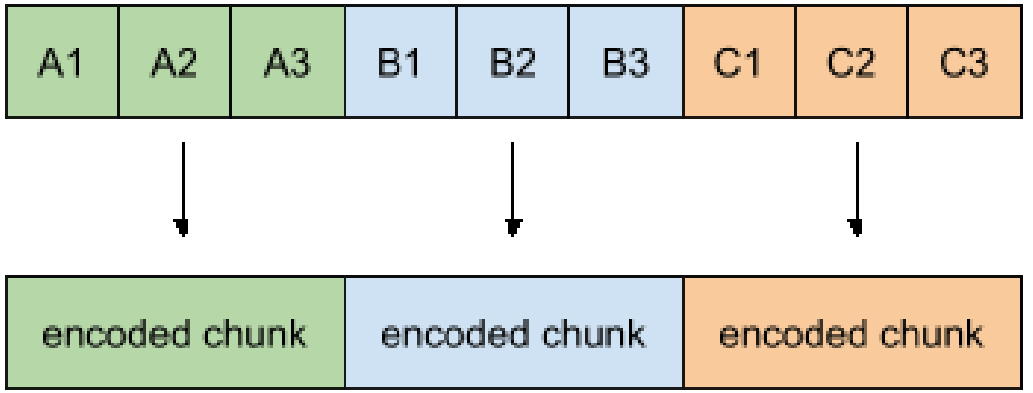
Données tabulaires: stockage colonne

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

Stockage en lignes



Stockage en colonnes



Stockage colonnes

Limite les E/S aux données réellement nécessaires :

- Charger uniquement les colonnes auxquelles il faut accéder.

Gain de place :

- La disposition en colonnes compresse mieux
- Encodages spécifiques.

Active les moteurs d'exécution vectorisée.

- Les opérations sur les vecteurs sont effectuées avec un minimum d'instructions machine (x20)

Parquet : le format colonne sur disque

Le format Apache Parquet est un standard pour stocker des données tabulaires qui peuvent être lues et écrites à partir des langages de programmation les plus courants.

Ce stockage en colonnes offre les avantages suivants :

- La compression par colonne est efficace et permet d'économiser de l'espace de stockage.
- Des techniques de compression spécifiques à un type peuvent être appliquées, car les valeurs des colonnes ont tendance à être du même type.
- Les requêtes qui récupèrent des valeurs de colonnes spécifiques n'ont pas besoin de lire la totalité des données brutes, ce qui améliore les performances.
- Différentes techniques d'encodage peuvent être appliquées à différentes colonnes.

Arrow: le format colonne en mémoire

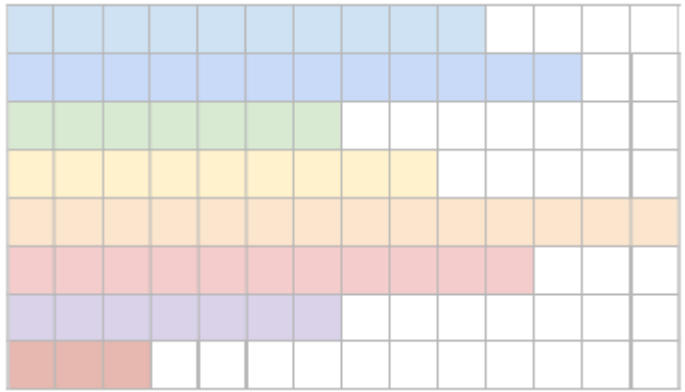
- Bien documenté et compatible avec tous les langages
- Conception pour tirer profit des caractéristiques modernes des CPU
- Intégrable dans les moteurs d'exécution, les couches de stockage, etc.
- Interopérable
- Structure de données imbriquée
- Maximiser le débit du CPU : pipelines, SIMD, localité de cache, etc.
- E/S dispersion/rassemblement

Les données argo

Ces données sont stockées dans des fichiers NetCDF sous la forme de vecteurs et matrices

- `char PSAL_QC(N_PROF, N_LEVELS) ;`
  - `PSAL_QC:long_name = "quality flag" ;`
  - `PSAL_QC:conventions = "Argo reference table 2" ;`
  - `PSAL_QC:_FillValue = " " ;`

Le nombre de niveaux est différents entre les capteurs: le stockage matriciel est inadapté

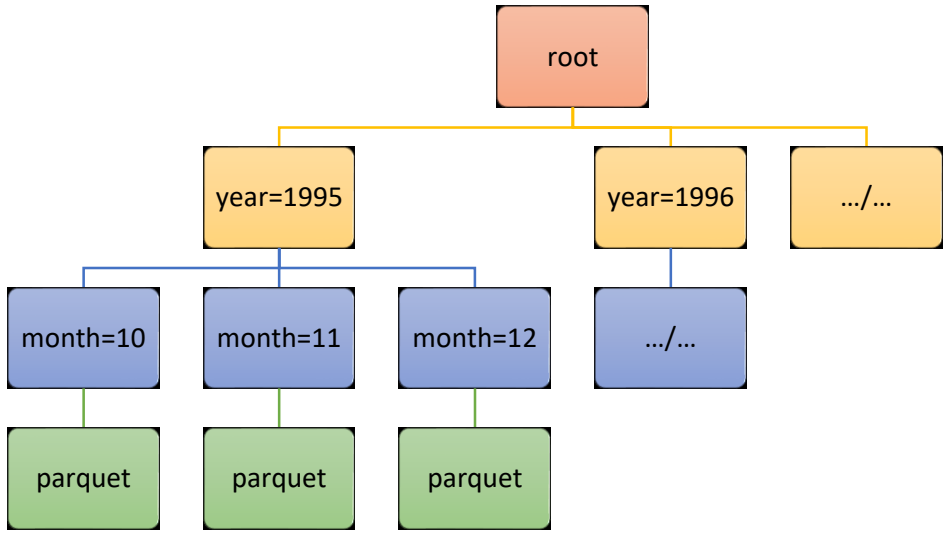


Les données argo

Le format de stockage tabulaire est plus adapté ↦ Parquet

Arrow permet d'enregistrer des listes dans les fichiers Parquet

Arrow permet de hiérarchiser l'information (filtres)



Conversion des données NetCDF vers les « DataFrame »

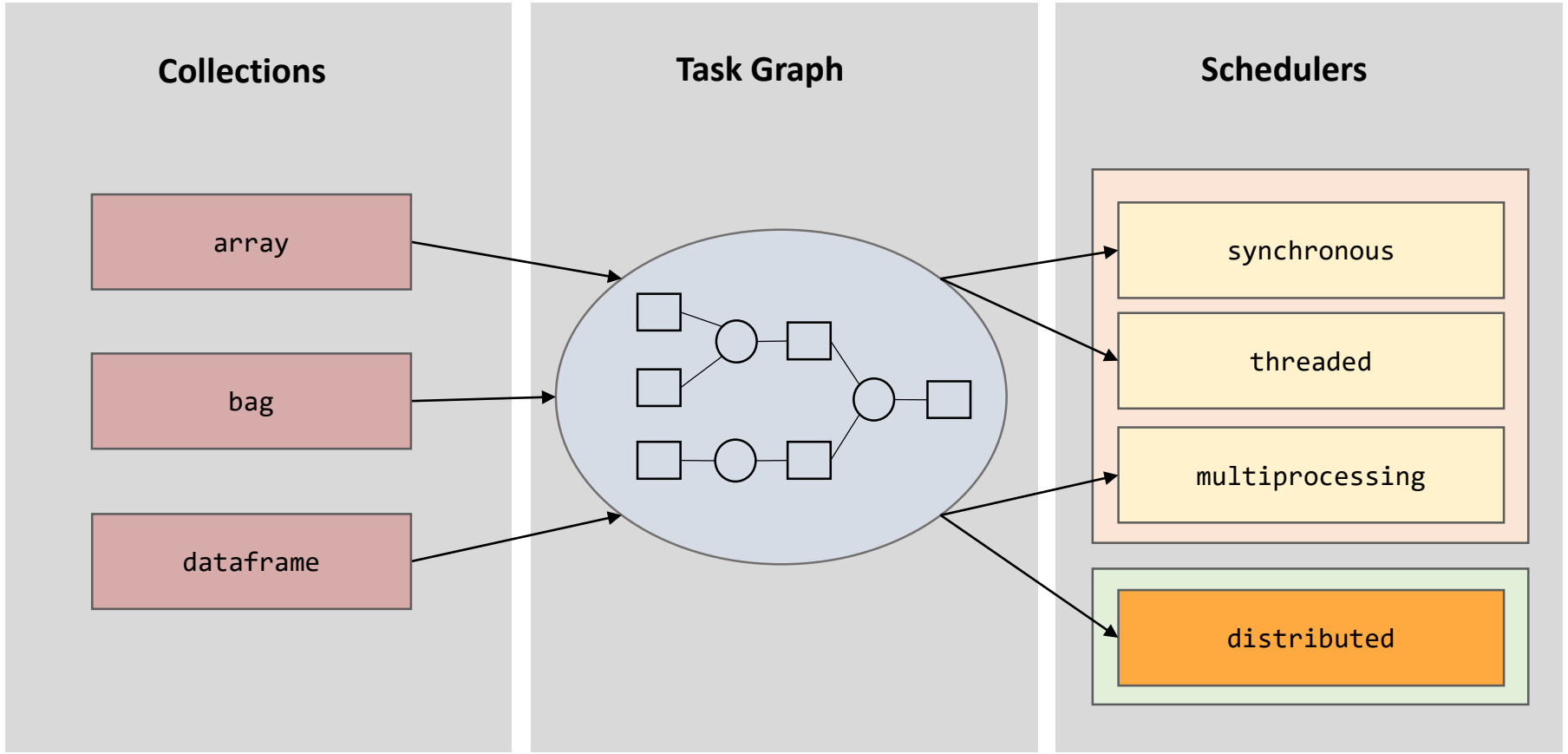
La difficulté consiste à créer le schéma de la table. D'autant plus lorsque l'on ne connait pas les cas d’usage sur les données !

Il faut faire attention aux conventions de représentations des valeurs de remplissage, des types de données, du format des dates et de la représentation des chaines de caractères.

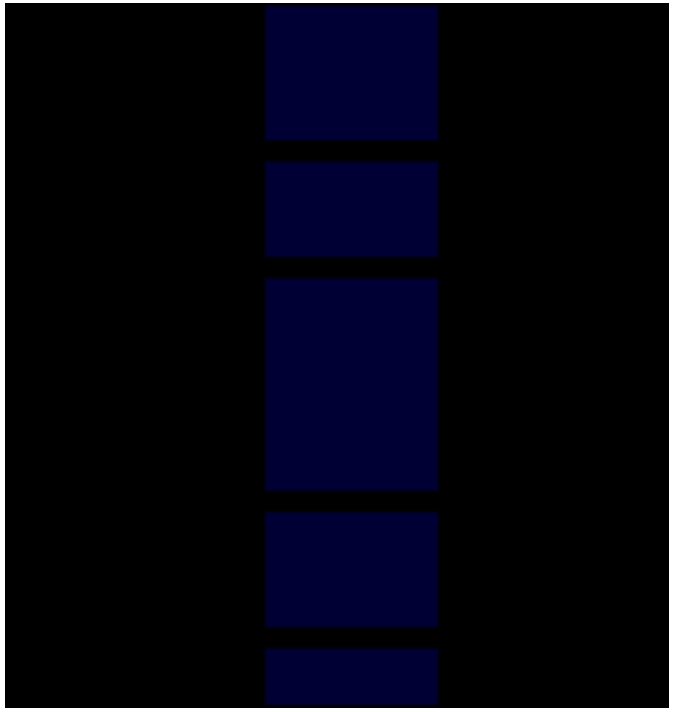
La conversion des vecteurs dans les tables est le point le plus difficile, mais c'est fait !

L’écriture dans les fichiers Parquet est presque transparente pour l’utilisateur

Le calcul : « dask.DataFrame »



dask.DataFrame



Un Dask DataFrame est un grand DataFrame parallèle composé de plusieurs DataFrame Pandas plus petits, répartis le long de l'index.

Ces Pandas DataFrames peuvent vivre sur disque pour des calculs plus volumineux que la mémoire sur une seule machine, ou sur plusieurs machines différentes dans un cluster.

Une opération Dask DataFrame déclenche de nombreuses opérations sur les Pandas DataFrames constitutives.

Performances

Les performances ont été mesurées sur une base de données des flotteurs PR/PF de 1995 à juin 2018.

Les données ont été découpées par jour.

La base compte près de 2 millions d'enregistrements pour un volume de 5.6 G (94 G au format NetCDF).

Compter le nombre d'enregistrements dans l'intégralité de la base prend 1min.

Effectuer une sélection géographique:

- Sur une année : 16 secondes
- Sur l'intégralité de la BD: 1mn 30

Sélectionner des flotteurs sur leurs identifiants:

- Sur une année : 13 secondes
- Sur l'intégralité de la BD: 1min

Calculer une anomalie (différence entre deux vecteurs):

- Sur une année : 40 secondes
- Sur l'intégralité de la BD: 4min

Interpolation de la SLA à partir de fichiers NetCDF:

- Sur une année : 1min 30 secondes
- Sur l'intégralité de la BD: 37 min

Interpolation de la SLA à partir de fichiers Zarr:

- Sur une année : 20 secondes
- Sur l'intégralité de la BD: 2 min 40 secondes