



DATA
TERRA



ODATIS

Benchmark : Analyse de l'accès et du subsetting de données via différents protocoles

Jeudi 10 octobre 2024

Gwenaël CAËR



contact@odatis-ocean.fr | www.odatis-ocean.fr

Introduction

1. Les Objectifs

- Sur un même jeu de données:
 - Étudier différents **formats** de données
 - *NetCDF / Zarr / Kerchunk*
 - Étudier différents **protocoles** d'accès à la donnée
 - *POSIX / S3 / FTP / HTTP / OPeNDAP*
 - Étudier différentes **opérations**
 - *Chargement / Lecture / Subsetting*
- *Gitlab* : https://gitlab.ifremer.fr/odatis/benchmarks/data_access

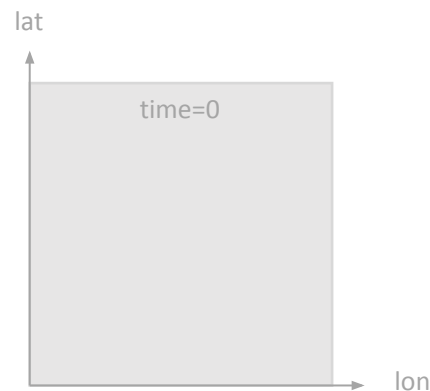
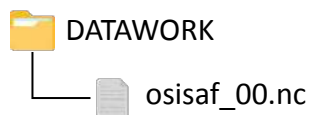
2. Le Dataset

OSISAF : Global Metop Sea Surface Temperature (fichiers de 12h)
(time, longitude, latitude).

Les formats

1. NetCDF (Network Common Data Form)

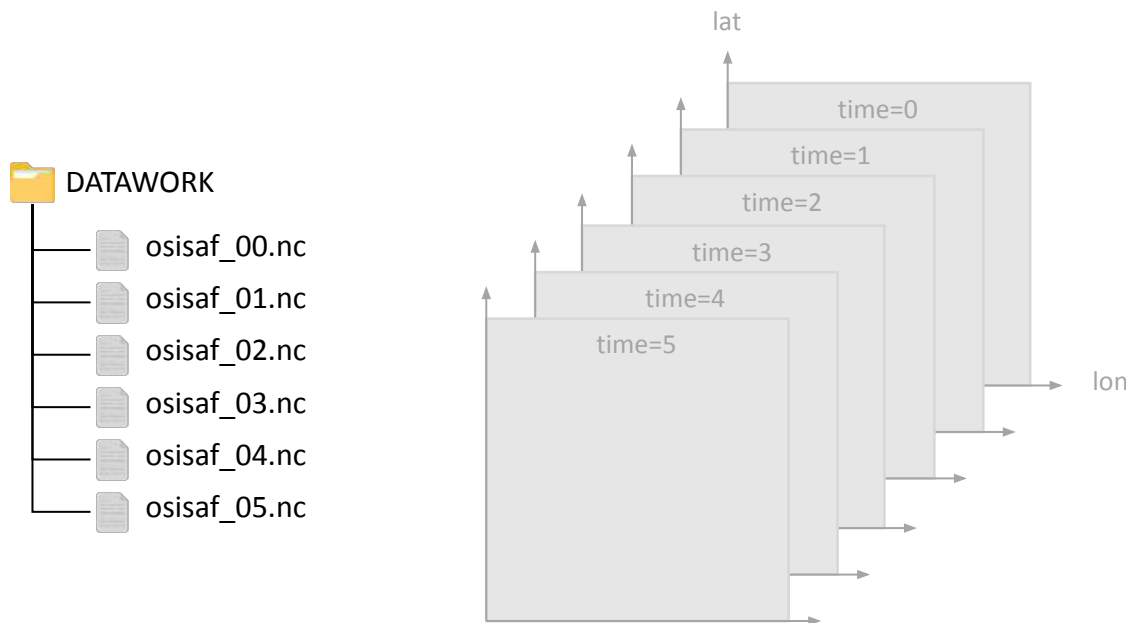
NetCDF est un format de fichier largement utilisé pour stocker des données scientifiques **multidimensionnelles**, il est auto-descriptif contenant les données mais aussi les métadonnées.



Les formats

1. NetCDF (Network Common Data Form)

NetCDF est un format de fichier largement utilisé pour stocker des données scientifiques **multidimensionnelles**, il est auto-descriptif contenant les données mais aussi les métadonnées.



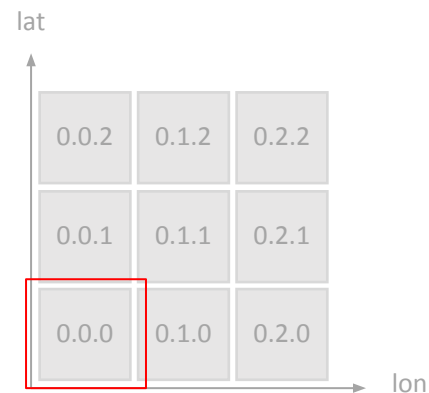
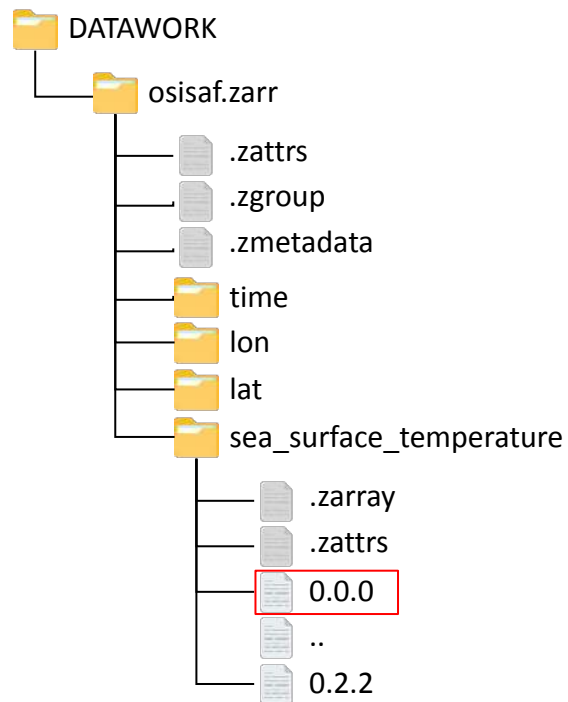
Les formats

2. Zarr

Zarr est un format moderne et flexible pour le stockage de données scientifiques **multidimensionnelles**. Contrairement à NetCDF, qui repose sur des fichiers binaires monolithiques, Zarr est conçu pour un accès distribué et parallèle aux données.

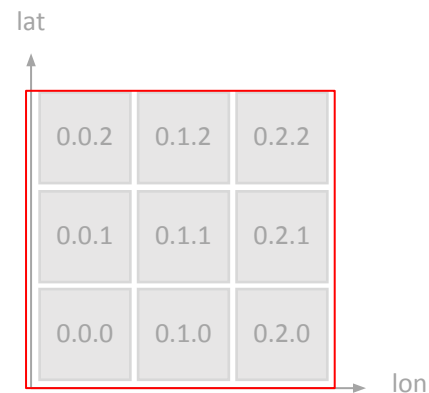
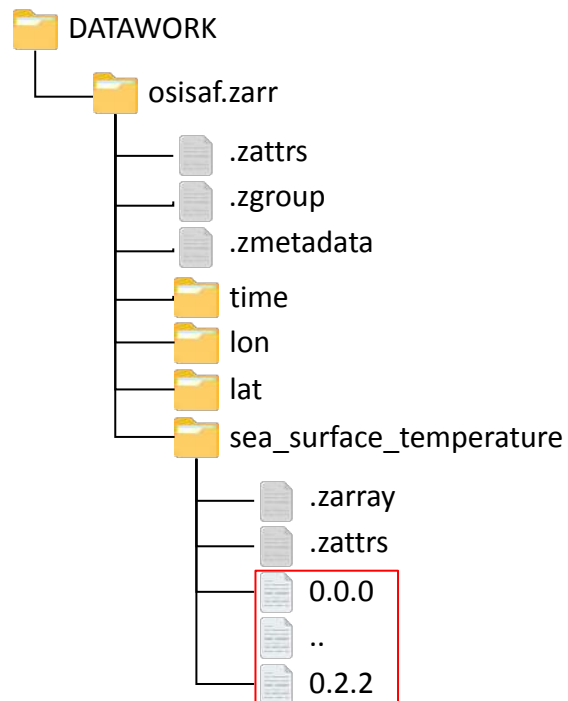
Les formats

2. Zarr



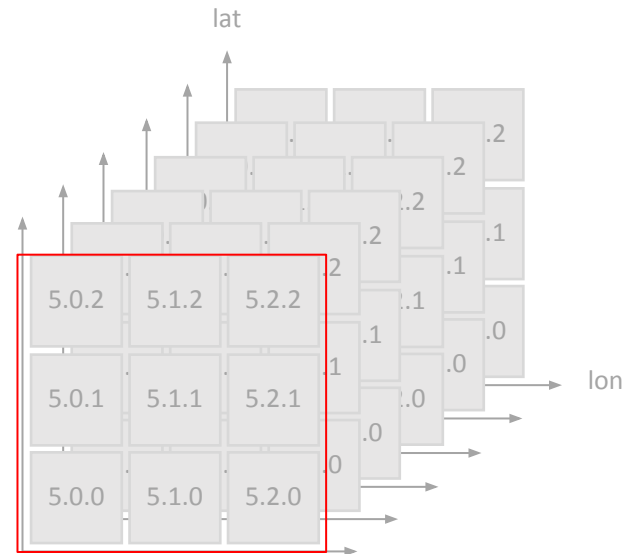
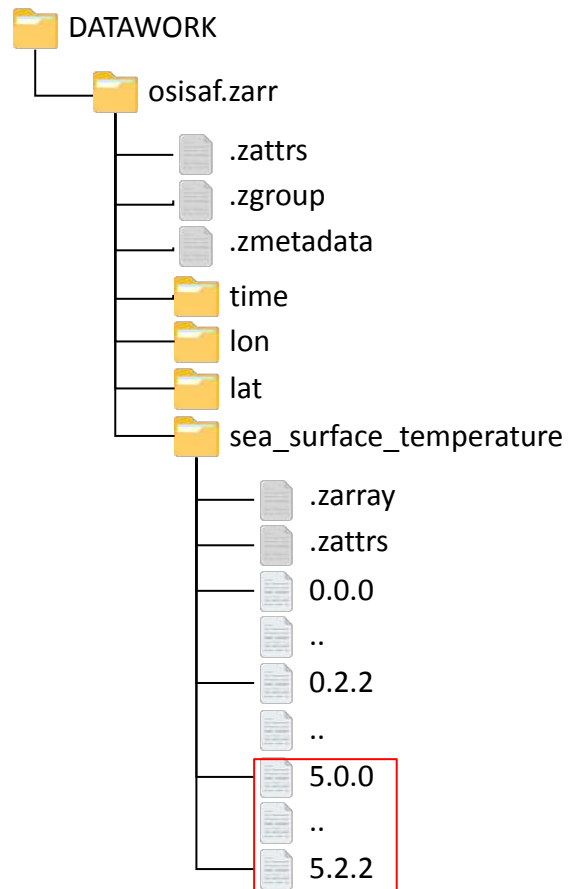
Les formats

2. Zarr



Les formats

2. Zarr



Les formats

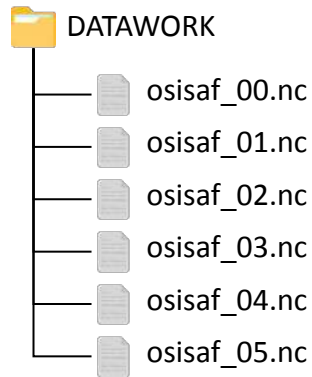
3. Kerchunk

Kerchunk est une bibliothèque conçue pour créer une émulation Zarr à partir de fichiers NetCDF.

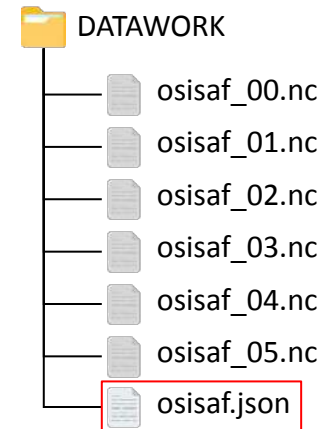
Il fonctionne en créant un index léger qui pointe vers les données réelles, sans avoir à les dupliquer.

Les formats

3. Kerchunk



```
{
  "version": 1,
  "refs": {
    ".zattrs": "{...}",
    ".zgroup": "{...}",
    "lat/.zarray": "{...}",
    "lat/.zattrs": "{...}",
    "lat/0": [
      "/datawork/osisaf_00.nc",
      28985,
      2169
    ],
    "sea_surface_temperature/.zarray": "{...}",
    "sea_surface_temperature/.zattrs": "{...}",
    "sea_surface_temperature/0.0.0": [
      "/datawork/osisaf_00.nc",
      129644117,
      25920000
    ],
    "sea_surface_temperature/0.0.1": [
      "/datawork/osisaf_00.nc",
      155564117,
      25920000
    ],
    "sea_surface_temperature/0.1.1": [
      "/datawork/osisaf_00.nc",
      25964117,
      25920000
    ]
  ]
}
```



Les protocoles

Voici les protocoles étudiés:

- POSIX (local)
- S3
- FTP
- HTTP
- OPeNDAP

Les opérations

Voici les opérations effectuées:

- Chargement (*Lecture des métadonnées*)
- Lecture (*Lecture des données -> calcul de la moyenne*)
- Subsetting (*Lecture d'un sous ensemble de la donnée -> 10%*)

Benchmark (Résultats)

1. NetCDF - Les différents protocoles

Configuration	
Cache	False
Nombre de coeur	5
Chunks	{'time':1, 'lon': 3600, 'lat': 1800}
Slices	{'lon': slice(0,7200), 'lat':slice(0, 360)}
Protocole	POSIX

format	netcdf				
filesystem	posix	ftp	http	opendap	s3
load (9.4s)	1.00	7.09	1.17	0.75	1.50
read (3.6s)	1.00	3.48	7.96	11.16	27.46
subset (0.9s)	1.00	2.88	22.69	17.49	27.76

Valeurs normalisées (par NetCDF en local)

Benchmark (Résultats)

2. Local VS Distant

Configuration	
Cache	False
Nombre de coeur	5
Chunks	{'time':1, 'lon': 3600, 'lat': 1800}
Slices	{'lon': slice(0,7200), 'lat':slice(0, 360)}

filesystem	posix			s3			
	format	netcdf	zarr	kerchunk	netcdf	zarr	kerchunk
load (9.4s)		1.00	0.73	0.69	1.50	0.72	0.65
read (3.6s)		1.00	0.99	1.13	27.46	3.06	1.48
subset (0.9s)		1.00	1.56	1.51	27.76	6.79	3.72

Valeurs normalisées (par NetCDF en local)

Benchmark (Difficultés)

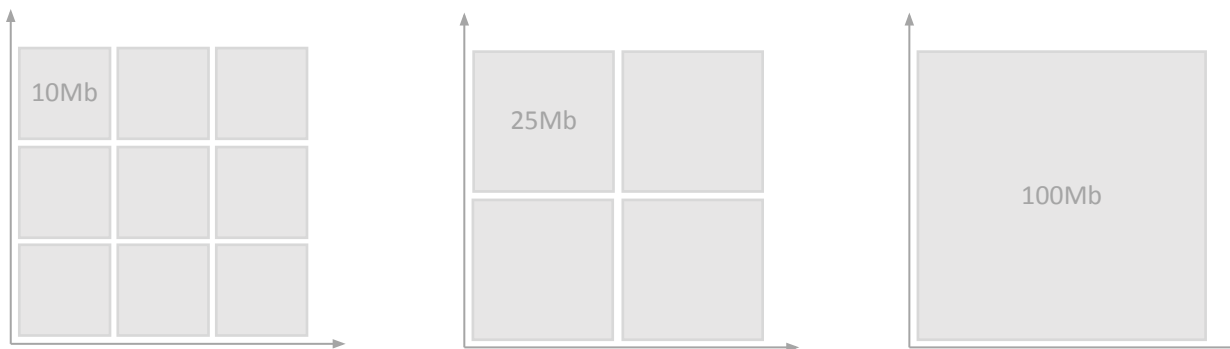
1. De nombreux paramètres

- Taille du dataset (30 fichiers de 200Mb = 6gb)
- Avec ou sans cache (local / serveur)
- Différentes méthodes pour accéder à la données (fsspec / xarray)
- Taille des chunks (de 10 à 100Mb)

Benchmark (Difficultés)

1. De nombreux paramètres

- Taille du dataset (30 fichiers de 200Mb = 6gb)
- Avec ou sans cache (local / serveur)
- Différentes méthodes pour accéder à la données (fsspec / xarray)
- Taille des chunks (de 10 à 100Mb)



Benchmark (Difficultés)

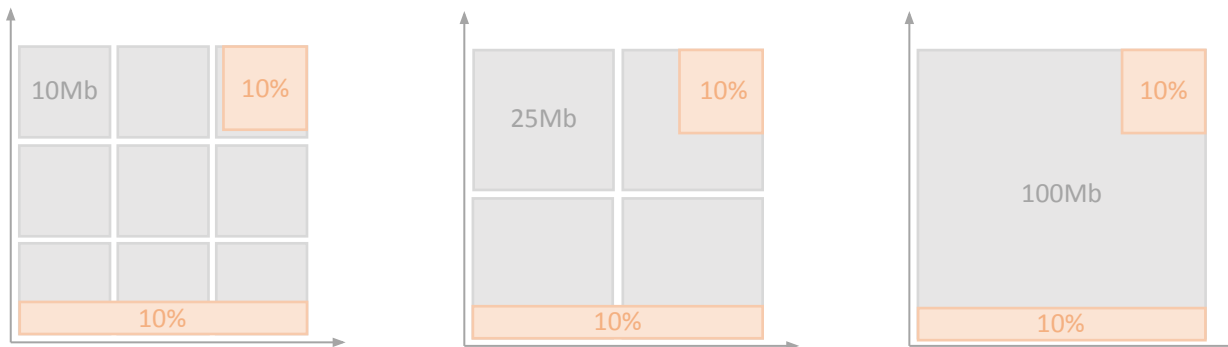
1. De nombreux paramètres

- Taille du dataset (30 fichiers de 200Mb = 6gb)
- Avec ou sans cache (local / serveur)
- Différentes méthodes pour accéder à la données (fsspec / xarray)
- Taille des chunks (de 10 à 100Mb)
- Ressources (CPU / thread) (2 et 5)
- Position du subset (10% - 1 seul chunk VS plusieurs chunk)

Benchmark (Difficultés)

1. De nombreux paramètres

- Taille du dataset (30 fichiers de 200Mb = 6gb)
- Avec ou sans cache (local / serveur)
- Différentes méthodes pour accéder à la données (fsspec / xarray)
- Taille des chunks (de 10 à 100Mb)
- Ressources (CPU / thread) (2 et 5)
- Position du subset (10% - 1 seul chunk VS plusieurs chunk)



Benchmark (Difficultés)

1. De nombreux paramètres

- Taille du dataset (30 fichiers de 200Mb = 6gb)
- Avec ou sans cache (local / serveur)
- Différentes méthodes pour accéder à la données (fsspec / xarray)
- Taille des chunks (de 10 à 100Mb)
- Ressources (CPU / thread) (2 et 5)
- Position du subset (10% - 1 seul chunk VS plusieurs chunk)
- Nombre d'itérations (entre 1 à 10)
- ...

2. De nombreux obstacles

- Processus très itératifs (de part le nombre de paramètres)
- Erreurs / Bug / Problèmes réseaux
- ...

Benchmark (Conclusion)

1. Les protocoles

Protocoles	Avantages	Inconvénients
POSIX	Rapidité (+++), possibilité de chargement et subsetting	
S3	Rapidité (++), possibilité de chargement et subsetting, tout format de fichier	Difficultés de prise en main
FTP	Dans certaine condition préférable de télécharger une fois, pour travailler en local.	Pas de chargement ou de subsetting, nécessite de télécharger TOUTE la donnée en locale.
HTTP	Rapidité (+), possibilité de chargement et subsetting (HTTP-RANGE?)	
OPeNDAP	Rapidité (+), possibilité de chargement et subsetting	

Benchmark (Conclusion)

2. Les formats

Protocoles	Avantages	Inconvénients
NetCDF	Chargement, subsetting	Moins optimisé pour le cloud
Zarr	Lecture/écriture en parallèle, chargement, subsetting	Contient de nombreux petit fichiers
Kerchunk	Avoir les avantages du zarr sur des fichiers NetCDF. Très utile pour les collections historiques au format NetCDF	Le fichier de référence JSON peut devenir très lourd s'il y a de nombreux chunk (possibilité d'utiliser un fichier parquet)



**DATA
TERRA**



ODATIS

Merci de votre attention



28/06/2024

contact@odatis-ocean.fr | www.odatis-ocean.fr