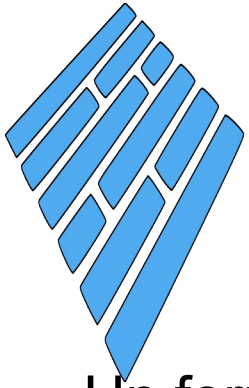
Stocker des séries temporelles au format Parquet

Charly Coussot, OSUG - Observatoire des Sciences de l'Univers de Grenoble, IRD

SIST, 2024



Parquet, c’est quoi ?

Un format de fichier de stockage de données orienté colonne optimisé pour stocker et lire des données massives.

Développé par Apache et conçu pour fonctionner efficacement avec des frameworks de systèmes de fichiers distribués comme Apache Spark, Hadoop, etc.

Caractéristiques :

- **Stockage orienté colonne**: permet compression et efficacité des requêtes
- **Évolutivité**: permet d’ajouter ou de supprimer des colonnes sans avoir à reconstruire le fichier existant
- **Interopérabilité** : supporté par de multiple langage et outils.

Le stockage des données dans Parquet

Date	Int	String
01/06/2024	0	not valid
02/06/2024	2	not valid
03/06/2024	1	OK
04/06/2024	1	OK

Stockage orienté ligne

01/06/2024	0	not valid	02/06/2024	2	not valid	03/06/2024	1	OK	04/06/2024	1	OK

Stockage orienté colonne

01/06/2024	02/06/2024	03/06/2024	04/06/2024	0	2	1	1	not valid	not valid	OK	OK

Stockage orienté colonne avec des groupes de lignes de taille 2

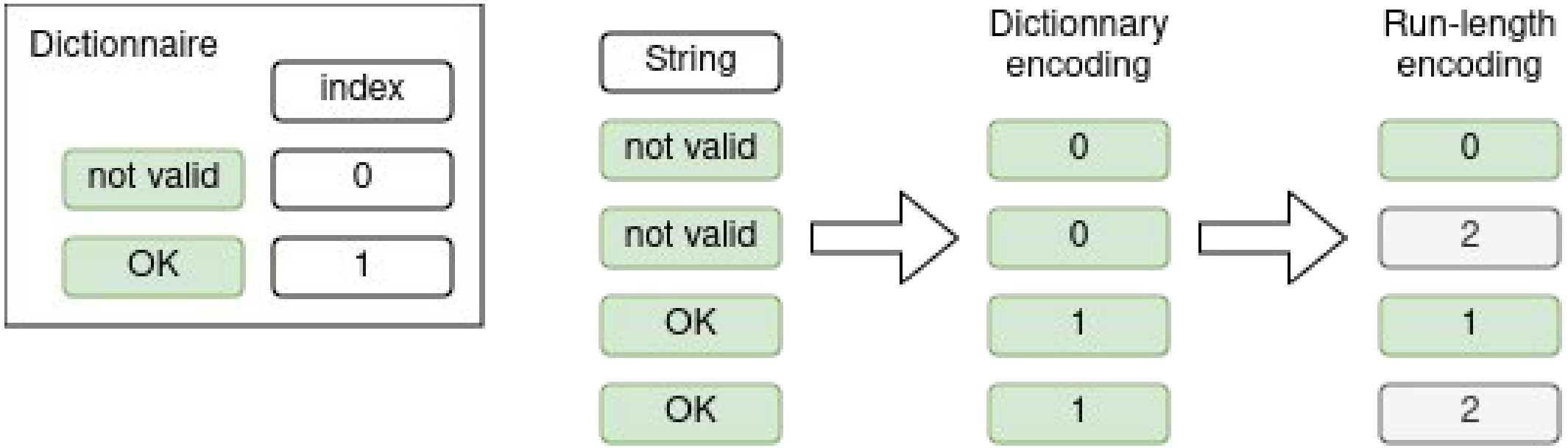
01/06/2024	02/06/2024	0	2	not valid	not valid	03/06/2024	04/06/2024	1	1	OK	OK

Pourquoi Parquet pour des séries temporelles ?

- **Compression efficace**
- Performance des requêtes
- Flexibilité du schéma
- Intégration

Pourquoi Parquet pour des séries temporelles ?

- **Compression efficace**



Pourquoi Parquet pour des séries temporelles ?

- Compression efficace
- **Performance des requêtes**
- Flexibilité du schéma
- Intégration

Pourquoi Parquet pour des séries temporelles ?

- Compression efficace
- **Performance des requêtes**

Requête: combien de valeur valides sont non-nulles ?

Date	Int	String
01/06/2024	0	not valid
02/06/2024	2	not vaild
03/06/2024	1	OK
04/06/2024	1	OK

Stockage orienté ligne

01/06/2024	0	not valid	02/06/2024	2	not valid	03/06/2024	1	OK	04/06/2024	1	OK

Pourquoi Parquet pour des séries temporelles ?

- Compression efficace
- **Performance des requêtes**

Requête: combien de valeur valides sont non-nulles ?

Date	Int	String
01/06/2024	0	not valid
02/06/2024	2	not vaild
03/06/2024	1	OK
04/06/2024	1	OK

Stockage orienté ligne

01/06/2024	0	not valid	02/06/2024	2	not valid	03/06/2024	1	OK	04/06/2024	1	OK

Stockage orienté colonne

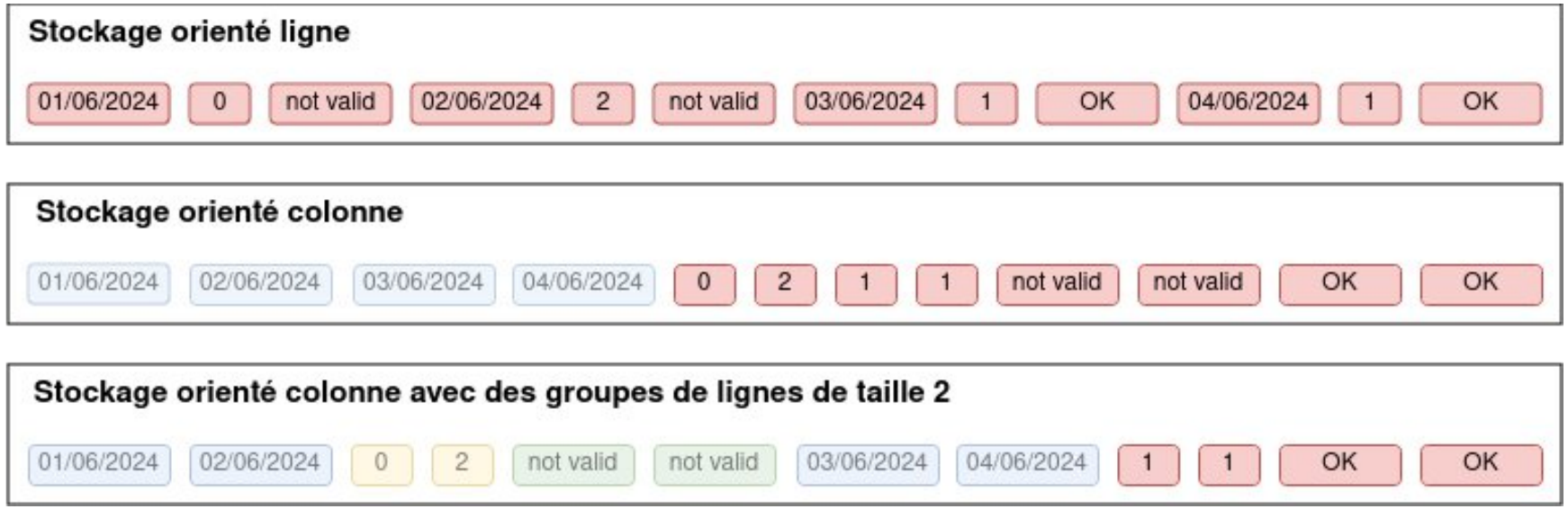
01/06/2024	02/06/2024	03/06/2024	04/06/2024	0	2	1	1	not valid	not valid	OK	OK

Pourquoi Parquet pour des séries temporelles ?

- Compression efficace
- **Performance des requêtes**

Requête: combien de valeur valides sont non-nulles ?

Date	Int	String
01/06/2024	0	not valid
02/06/2024	2	not vaild
03/06/2024	1	OK
04/06/2024	1	OK



Pourquoi Parquet pour des séries temporelles ?

- Compression efficace
- Performance des requêtes
- **Flexibilité du schéma**
- Intégration

Pourquoi Parquet pour des séries temporelles ?

- Compression efficace
- Performance des requêtes
- **Flexibilité du schéma**

Possibilité d’ajouter des colonnes sans avoir à reconstruire le fichier entièrement

Cas d’usage: ajout de nouveaux capteurs au fil du temps

Pourquoi Parquet pour des séries temporelles ?

- Compression efficace
- Performance des requêtes
- Flexibilité du schéma
- **Intégration**

Python: - **Pandas**: *pandas.read_parquet* et *pandas.DataFrame.to_parquet*.
- **PyArrow**: *pyarrow.parquet* pour des opérations plus avancées

R: - **Arrow Package**: *arrow::read_parquet* et *arrow::write_parquet*.

Java: - **Apache Parquet Library**: Pour lire et écrire des fichiers Parquet directement

Julia: - **Parquet.jl**: Pour lire et écrire des fichiers Parquet directement

Retour d’expérience

- Tests en écriture (avec une configuration minimale et donc pas mal de biais)
    - Plus rapide que d’écrire les données dans InfluxDB, Cassandra ou TimescaleDB
    - Satisfaisant en lecture (pas de requête complexe)

- Test en lecture : ~10ms pour lire un an de données au pas de temps 5 minutes dans un fichier

- Compression : 30 ans de données à 5min (chronique de précipitation au Sahel)
    - csv : ~ 220 Mo
    - parquet : ~ 145 Ko