Projet INF728

Recueil et requêtage sur des données de GDELT

Introduction

mongoDB

Techno choisie: MongoDB

Benchmark

Avantages

- Bonne flexibilité sur les requêtes une fois les documents insérés
- Alternative à Cassandra qui a été choisi par la majorité des groupes

Inconvénients

- Sharding lourd à mettre en place par rapport à Cassandra
- Requêtes plus difficiles sur les embedded documents

Code: https://github.com/jbSarda/INF728

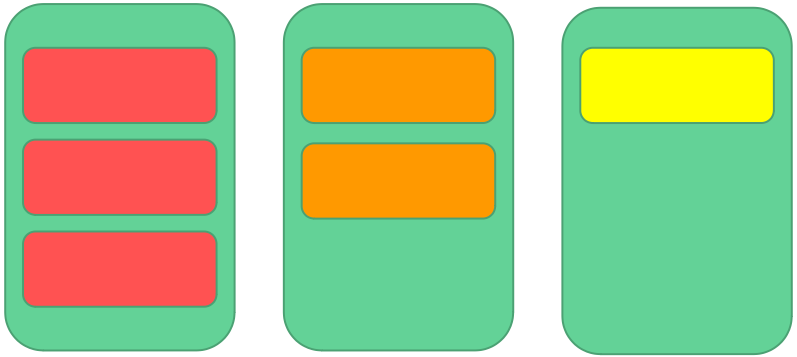
Sommaire

I. Structures matérielle et logicielle

Nomenclature

Replica-set Ensemble de machines qui contiennent toutes exactement les mêmes données

Shard Portion de l'ensemble des données stockées sur un réplica-set donné (et uniquement sur celui-ci)



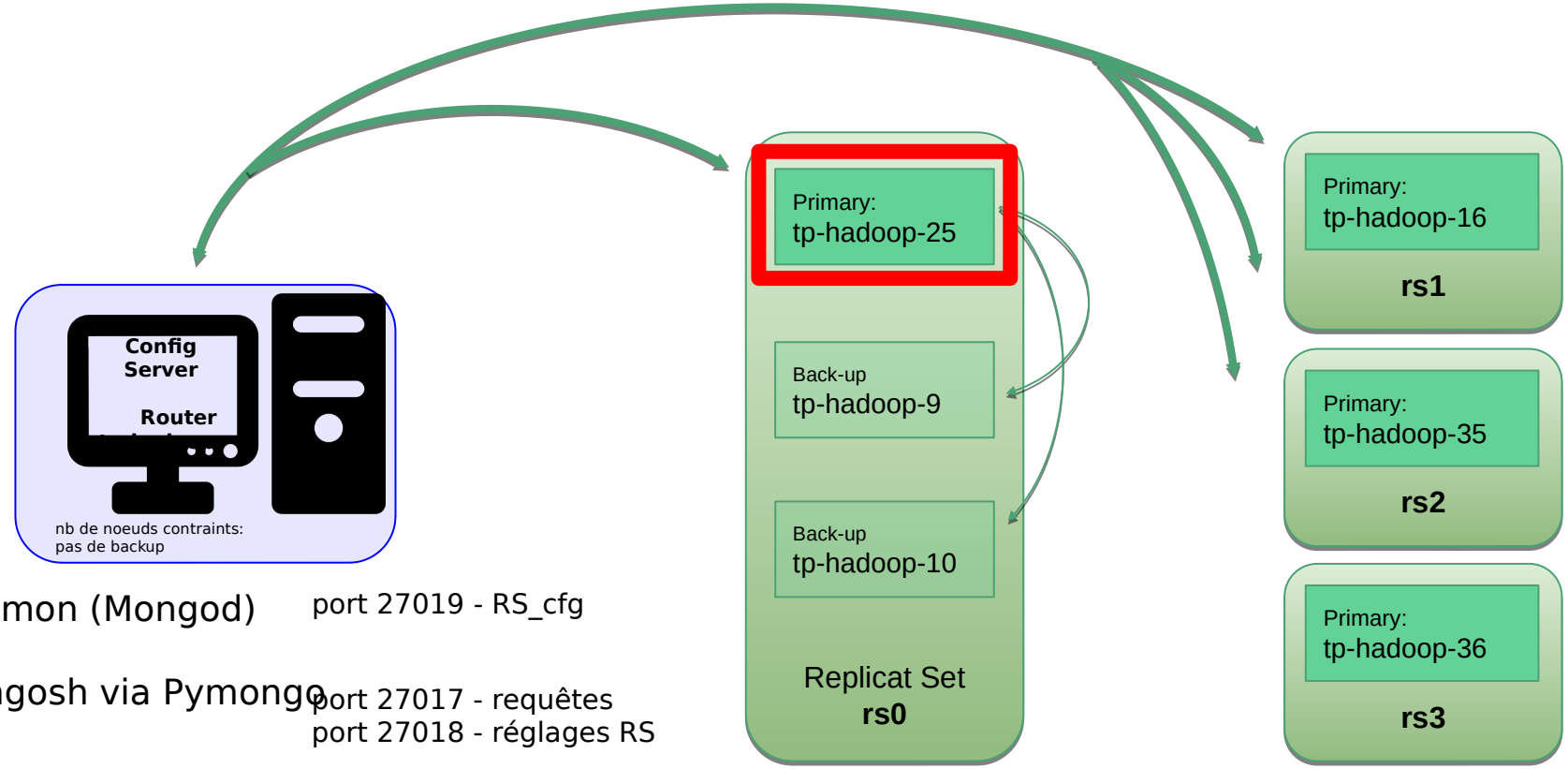
rs0 / shard0
réplication : x2

rs1 / shard1
réplication : x1

rs2 / shard2
pas de réplication

En pratique, au cours de cette présentation, les deux termes sont utilisés comme synonymes pour désigner les blocs de notre architecture

I. Structure matérielle et logicielle



Daemon (Mongod) port 27019 - RS_cfg

Mongosh via Pymongo port 27017 - requêtes
port 27018 - réglages RS

Récapitulatif

→ 1 config server : “annuaire” de la base

→ 1 routeur : “mongos”

→ 4 shards dont seulement 1 répliqué (rs0)

```
[direct: mongos] test> db.adminCommand( { listShards: 1 } )
{
  shards: [
    {
      _id: 'rs0',
      host: 'rs0/tp-hadoop-10:27018,tp-hadoop-25:27018,tp-hadoop-9:27018',
      state: 1,
      topologyTime: Timestamp({ t: 1644249582, i: 2 })
    },
    {
      _id: 'rs1',
      host: 'rs1/tp-hadoop-16:27018',
      state: 1,
      topologyTime: Timestamp({ t: 1644249608, i: 3 })
    },
    {
      _id: 'rs2',
      host: 'rs2/tp-hadoop-35:27018',
      state: 1,
      topologyTime: Timestamp({ t: 1644249619, i: 5 })
    },
    {
      _id: 'rs3',
      host: 'rs3/tp-hadoop-36:27018',
      state: 1,
      topologyTime: Timestamp({ t: 1644249703, i: 17 })
    }
  ],
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1644249706, i: 1 }),
    signature: {
      hash: Binary(Buffer.from("0000000000000000000000000000000000000000", "hex"), 0),
      keyId: Long("0")
    }
  },
  operationTime: Timestamp({ t: 1644249706, i: 1 })
}
```

II. Recueil et stockage de la donnée

II. Modélisation sous forme de documents

→ Documents : events

→ Embedded documents: articles

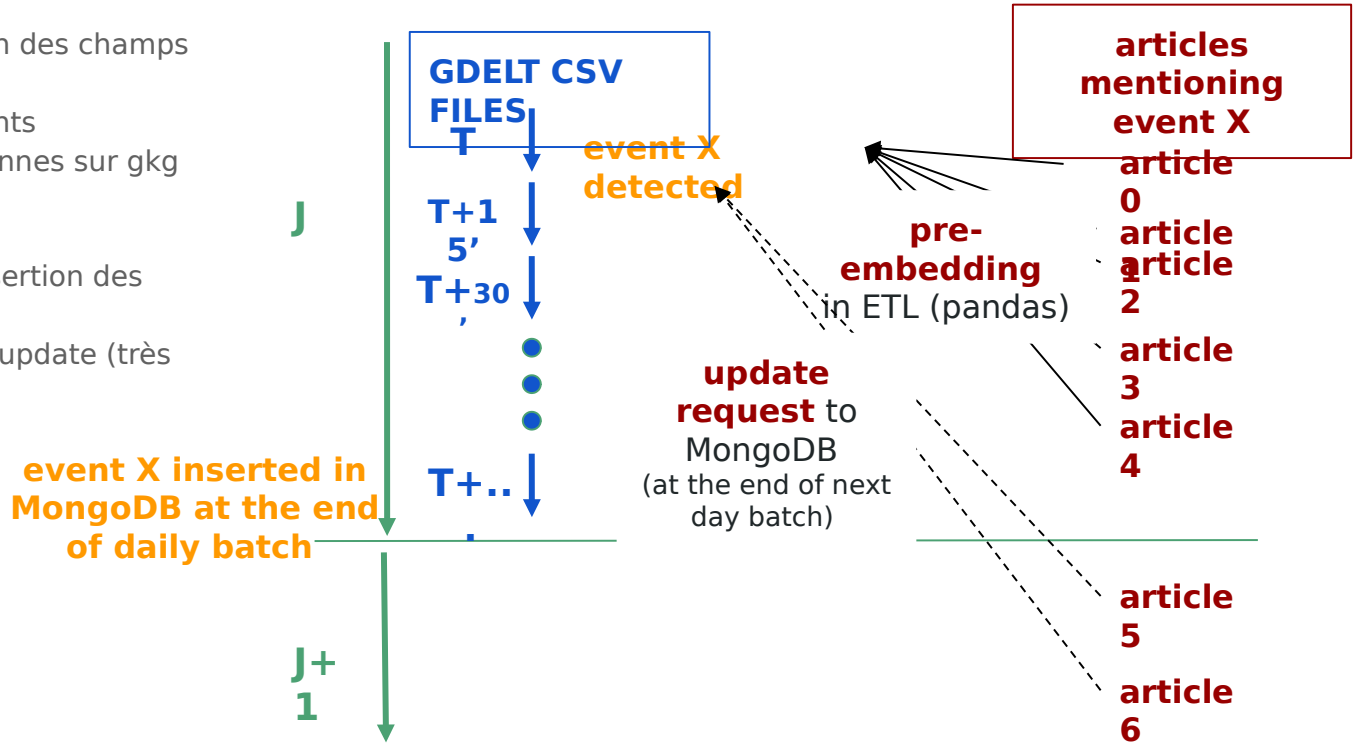
```
[direct: mongos] gdelt> db.evt.findOne()
{
  _id: ObjectId("61fc624a321004e857187507"),
  ID: 967254409,
  date: ISODate("2021-02-01T00:00:00.000Z"),
  country: 'AF',
  tone: -6.33484162895925,
  theme_base: 'Use conventional military force',
  theme_root: 'FIGHT',
  num_mentions: 4,
  num_sources: 1,
  act1_country: NaN,
  act2_country: 'AF',
  list_articles: [
    {
      ID: 'https://www.ebar.com/arts_&_culture/books/301570',
      date: ISODate("2021-02-01T00:00:00.000Z"),
      source: 'ebar.com',
      lang: 'eng',
      locs: [ 'AF' ],
      tone: '-6.19266055045872',
      persons: [
        'donovan russo',
        'frank paine',
        'anthony johnson',
        'steven cahill'
      ],
      org: [ 'seton hall university', 'young', 'yahoo' ]
    }
  ]
}
```

```
[direct: mongos] gdelt> db.evt.find({"ID": 963342007})
[
  {
    _id: ObjectId("61fc5a10f6993eea0e969cdc"),
    ID: 963342007,
    date: ISODate("2021-01-08T00:00:00.000Z"),
    country: 'US',
    tone: -6.69144981412639,
    theme_base: 'Make a visit',
    theme_root: 'CONSULT',
    num_mentions: 1,
    num_sources: 1,
    act1_country: 'US',
    act2_country: 'US',
    list_articles: [
      {
        ID: 'https://wsbs.com/pittsfield-man-faces-charges-arraigned-in-d-c-superior-court/',
        date: ISODate("2021-01-08T00:00:00.000Z"),
        source: 'wsbs.com',
        lang: 'eng',
        locs: [ 'US' ],
        tone: '-6.41509433962264',
        persons: [ 'david lester ross', 'andrew lelling' ],
        org: [ 'd c superior court', 'twitter', 'capitol police' ]
      },
      {
        ID: 'https://wupe.com/pittsfield-man-faces-charges-arraigned-in-d-c-superior-court/',
        date: ISODate("2021-01-08T00:00:00.000Z"),
        source: 'wupe.com',
        lang: 'eng',
        locs: [ 'US' ],
        tone: '-6.41509433962264',
        persons: [ 'david lester ross', 'andrew lelling' ],
        org: [ 'd c superior court', 'twitter', 'capitol police' ]
      },
      {
        ID: 'https://www.iberkshires.com/story/63903/Pittsfield-Man-Arrested-After-Riot-in-U.S.-Capitol.html',
        date: ISODate("2021-01-08T00:00:00.000Z"),
        source: 'iberkshires.com',
        lang: 'eng',
        locs: [ 'US' ],
        tone: '-9.40438871473354',
        persons: [ 'andrew e lelling', 'david lester ross', 'andrew lelling' ],
        org: [ 'police department' ]
      }
    ]
  }
]
```

II. ETL : script python utilisant pandas

- Récupération et transformation des champs utiles
  → réduire la taille des documents
  11 colonnes sur events, 9 colonnes sur gkg

- Batchs journaliers avec pre-insertion des articles dans les events
  → limiter les requêtes de type update (très longues)



II. ETL : performance en écriture et volume chargé

→ Extrait de logs d'insertion

Temps d'insertion :

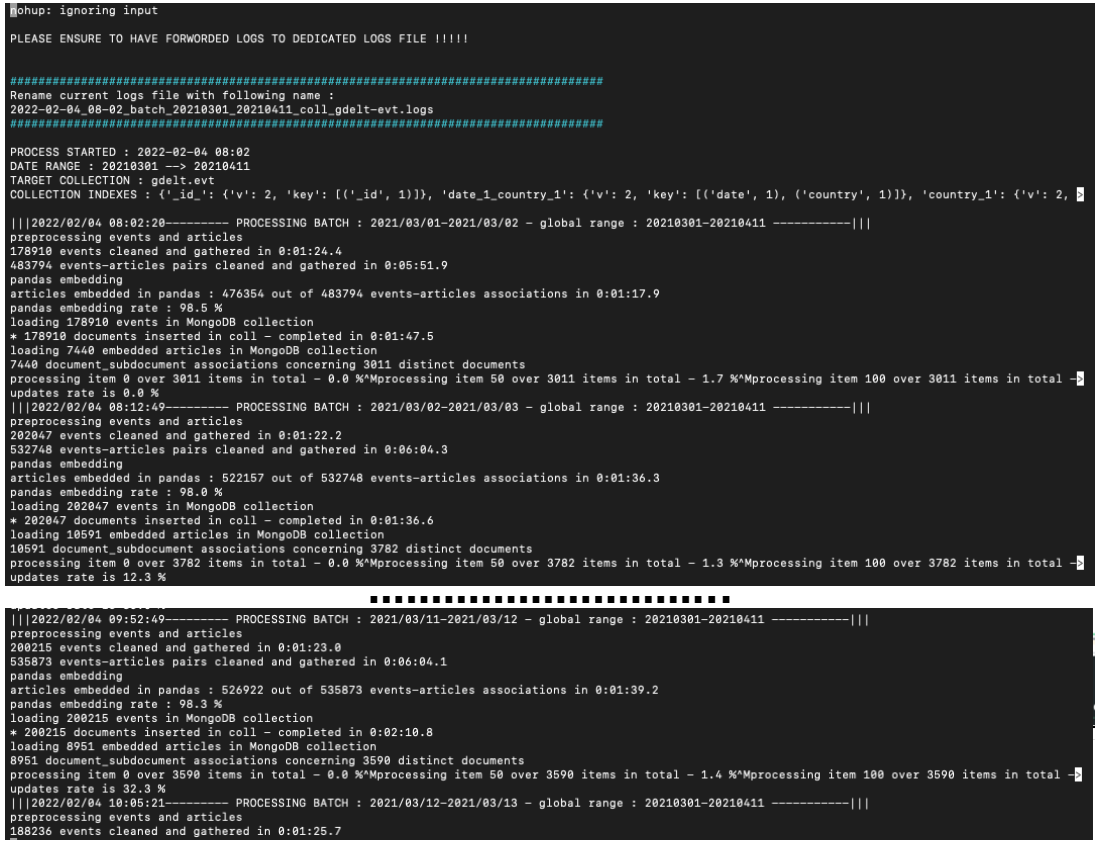
- 6 min / jour si 100k events
- 10 min/jour si 200k events
- env. 4h/mois

→ Volume de données chargées dans MongoDB :

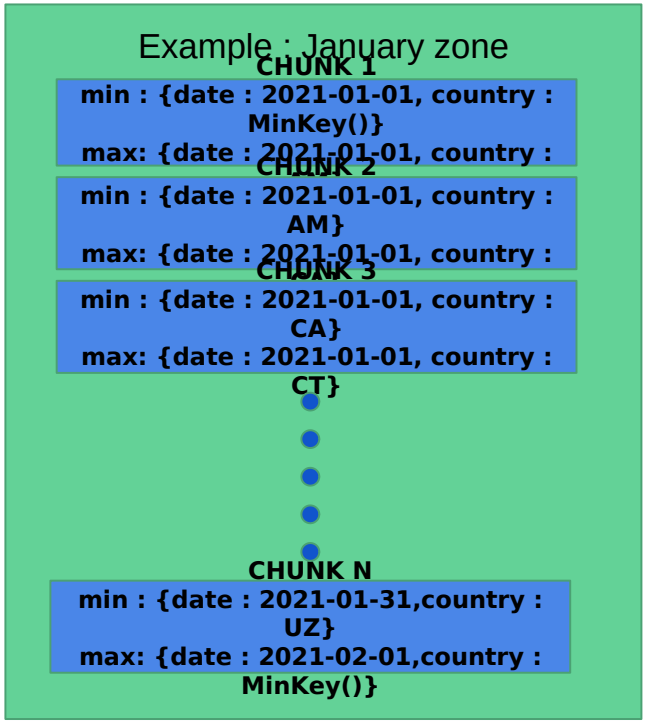
8 mois de données

= 40 Mio d'évènements

= 15 Go de données par machine



II. Structure du stockage dans MongoDB



→ MongoDB organise les données en chunks de 64 Mo sur les RS / shards

→ Identification de la clé de sharding déterminante = optimisation du temps d'écriture et de lecture

~~*[date] ?*~~

[date ; evt_country]

→ Répartition des chunks entre les shards = limiter les transferts réseaux inutiles

~~*géré automatiquement par le load balancer ?*~~

prédéfini (zones par mois)

III. Conception et test des requêtes

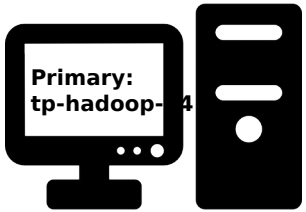
III. Conception et visualisation des requêtes

→ Connexion à la DB via PyMongo : jupyter notebook distant

→ Visualisation des résultats des requêtes par streamlit

Streamlit

jupyter



IV. Conclusion

Conclusion

Améliorations possibles

- Lenteur de l’ETL
  → goulot d’étranglement = capacité d’un unique shard
  → écrire en parallèle sur plusieurs shards
- Les requêtes sur les articles ne passent pas à l’échelle
  → nécessiter de créer une seconde collection

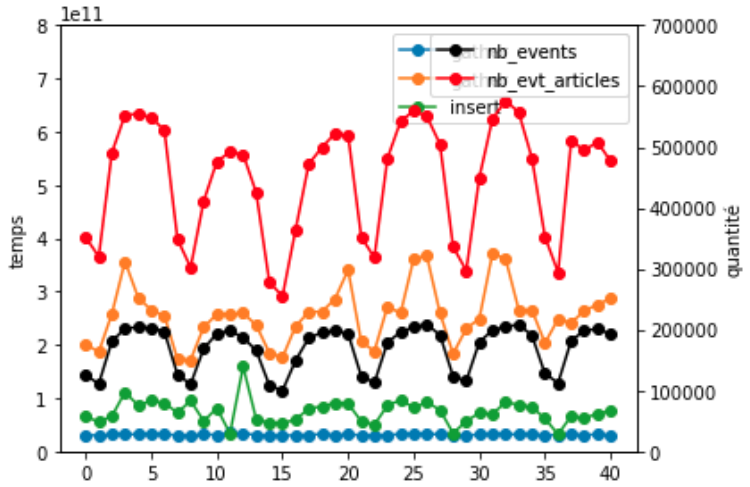
Point fort : flexibilité et capacité d’exploration des données (cf requête 4)

Point faible : duplication des articles (embedded) = lourdeur en mémoire

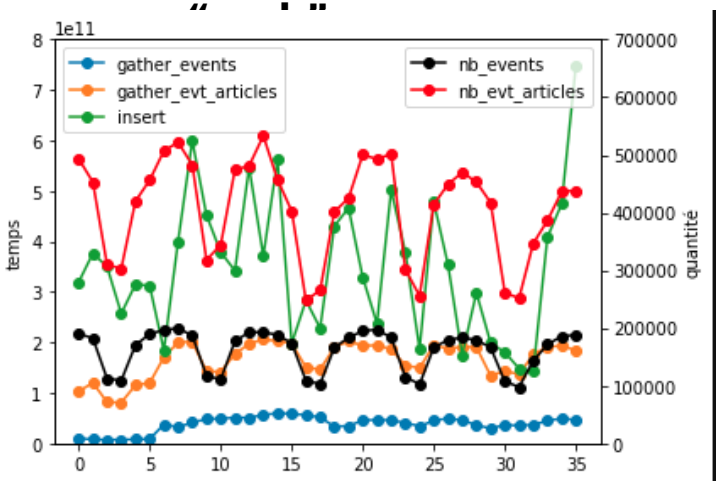
Questions ?

Analyse des logs d’insertion

Avant le



Pendant le



A partir de de mercredi, on voit que les temps d’insertion :

1) augmentent considérablement
2) ne dépendent plus du volume de données injecté mais de facteurs “externes”

→ surcharge du cluster OpenStack