

MongoDB et Elastic, meilleurs ennemis ?

#MongoElastic

Sébastien Prunier
Chris Woodrow



A propos de nous



Sébastien Prunier
Développeur@[Serli](#)

Craftman
MongoAddict



Chris Woodrow
Développeur@[Serli](#)

DataCurious
ElasticJunkie



Serli

Société de conseil et d'ingénierie

Développement, expertise, R&D, formation

70 personnes

Contribution à des projets OSS

Membre du JCP





Petit sondage ...





MongoDB

Base de données NoSQL

2007 par 10gen

C++

Orientée documents

Open Source (2009)

Licence AGPL





Elastic{Search}

Moteur de recherche distribué
2010 par Shay Banon
Java - Basé sur Lucene
Orientée documents
Open Source
Licence Apache



elastic



Projets types Mongo et Elastic

- Applications satellites
 - Vision 360°
- Business Intelligence
- Open data
- Moteur de recherche
- Backend mobile
- Objets connectés



Beaucoup de points communs

- Orientées documents (JSON)
- Distribuées
- Schemaless
- Indexes secondaires
- Aggrégations
- Recherche géographique



Mongo ou Elastic ?

La tronçonneuse ...



... ou la batte ?



Comment choisir ?





Popularité ?



db-engines.com

277 systems in ranking, June 2015

Rank	Jun 2015		DBMS	Database Model	Score		
	May 2015	Jun 2014			Jun 2015	May 2015	Jun 2014
1.	1.	1.	Oracle	Relational DBMS	1466.36	+24.26	-34.56
2.	2.	2.	MySQL	Relational DBMS	1278.36	-15.91	-31.20
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1118.05	-12.98	-105.74
4.	↑ 5.	4.	PostgreSQL	Relational DBMS	280.90	+7.39	+40.92
5.	↓ 4.	5.	MongoDB	Document store	279.05	+1.73	+47.61
6.	6.	6.	DB2	Relational DBMS	198.70	-2.35	+0.67
7.	7.	7.	Microsoft Access	Relational DBMS	146.49	+0.91	+4.13
8.	8.	↑ 9.	Cassandra	Wide column store	108.91	+2.36	+27.06
9.	9.	↓ 8.	SQLite	Relational DBMS	107.97	+2.81	+18.79
10.	10.	↑ 12.	Redis	Key-value store	95.49	+0.76	+30.36
11.	11.	↓ 10.	SAP Adaptive Server	Relational DBMS	88.18	+2.69	+7.48
12.	12.	↓ 11.	Solr	Search engine	81.24	-1.68	+13.36
13.	13.	13.	Teradata	Relational DBMS	73.09	+2.96	+8.02
14.	14.	↑ 17.	Elasticsearch	Search engine	70.09	+5.26	+36.14
15.	15.	15.	HBase	Wide column store	61.70	-0.01	+20.89
16.	16.	↓ 14.	FileMaker	Relational DBMS	52.25	-0.65	-1.53
17.	17.	↑ 18.	Hive	Relational DBMS	47.49	+2.92	+14.80
18.	18.	↑ 20.	Splunk	Search engine	41.24	+0.52	+15.56
19.	19.	↓ 16.	Informix	Relational DBMS	36.54	-0.20	+0.91
20.	↑ 21.	↑ 23.	SAP HANA	Relational DBMS	35.28	+2.49	+14.64
21.	↓ 20.	↓ 19.	Memcached	Key-value store	33.42	+0.31	+1.98
22.	22.	22.	Neo4j	Graph DBMS	30.50	+1.86	+8.00
23.	23.	↓ 21.	CouchDB	Document store	27.94	+0.77	+5.15
24.	24.	↑ 26.	Couchbase	Document store	24.95	-0.28	+9.17





Méthode pas très fiable ...





Démarrage rapide



Developer friendly

Download,
Unzip,
Run,
Play !





Vers le Minimum Viable Product





Installation d'un cluster Elastic



2 minutes

config/elasticsearch.yml

cluster.name: my_cluster

```
$ ./elasticsearch  
$ ./elasticsearch
```





Installation d'un replicaset Mongo

```
$ mongod --dbpath=~/mongo-data/0 --replSet rs0
$ mongod --dbpath=~/mongo-data/1 --replSet rs0 --port 27018
$ mongod --dbpath=~/mongo-data/2 --replSet rs0 --port 27019
```

```
rs.initiate()
rs.add("<hostname>:27018")
rs.add("<hostname>:27019")
```

mongo shell



Le sharding est une autre histoire ...





Noeud pour les tests en Java

```
Settings settings = ImmutableSettings.builder()
    .put("http.port", "9200")
    .put("network.host", "localhost")
    .put("path.data", "/tmp/es-local-data")
    .build();

Node node = NodeBuilder.nodeBuilder()
    .local(true)
    .data(true)
    .clusterName("elasticsearch")
    .settings(settings)
    .build();

node.start();
```





Indexation



Indexation dans Elastic

Basé sur un **index inversé**
mapping → **analyse** des données

Near Real-Time



Index Inversé

ElasticSearch (Lucene) : index inversé.

Doc #1 “La Vie de Brian”

Doc #2 “Le Sens de la vie”

- Stop words
- Minification
- ASCII folding
- Langages
- ...

Terme	Documents
brian	#1
sens	#2
vie	#2, #1

Sans full-text : **not_analyzed**



Near real-time?

Vous avez bien dit **Near** real-time?

Flush périodiques (**1s**) -> optimiser Lucene

Est-ce vraiment important ?





Indexation dans MongoDB

Requête possible sur un attribut même s'il n'est pas indexé (attention cependant aux perfs !)



mongoDB



Mise à jour dans Elastic

Réindexation totale du document
(même avec `_update!`)

Suppression du document puis recréation





Mise à jour dans MongoDB

`db.collection.update()`

Remplacement d'un document, ou modification partielle via un opérateur (**\$set, \$pull, \$push ...**)

*Options : **multi** (false par défaut), **upsert***



Relations



Disclaimer

Attention

Si vous devez faire des jointures :

- Soit vous avez mal modélisé vos documents
- Soit c'est légitime ... et vous avez choisi la mauvaise base !



Relations dans MongoDB

Deux possibilités :

- Référence à un `_id` d'un autre document
- Utilisation d'un objet `DBRef`

Dans les deux cas → plusieurs requêtes !



Relations dans MongoDB

Annonce pour la version 3.2 : **\$lookup (!?)**





Relations dans Elastic

Parent/Child

Un **document** est lié à son parent

Une solution au problème des **updates**

Impact sur les **perfs**

Contraintes techniques



Full text search



Je veu fair du fotbal



NoSQL Workshop Recherche Carte Statistiques API checkup

Rechercher une installation sportive

NoSQL Workshop Recherche Carte Statistiques API checkup

Rechercher une installation sportive

441000002 - Stade de Football

La Roche Corbinière 44690 Monnières
lat : -1.357052 ; lon : 47.131143
200 place(s) de parking

Equipements

#	Nom	Type	Famille	Activités
70083	Terrain de Football A	Terrain de football	Terrain de grands jeux	Football / Football en salle (Futsal)
70084	Terrain de Football B (ou d'honneur)	Terrain de football	Terrain de grands jeux	Football / Football en salle (Futsal)

441130009 - Stade de Football

Route de Puceul 44170 Nozay
lat : -1.625256 ; lon : 47.563547
250 place(s) de parking

Equipements

#	Nom	Type	Famille	Activités
72237	Terrain de Football A d'Honneur	Terrain de football	Terrain de grands jeux	Football / Football en salle (Futsal)
174206	Terrain de Football d'entraînement B	Terrain de football	Terrain de grands jeux	Football / Football en salle (Futsal)





MongoDB Full Text Search





Agrégations



Les agrégations

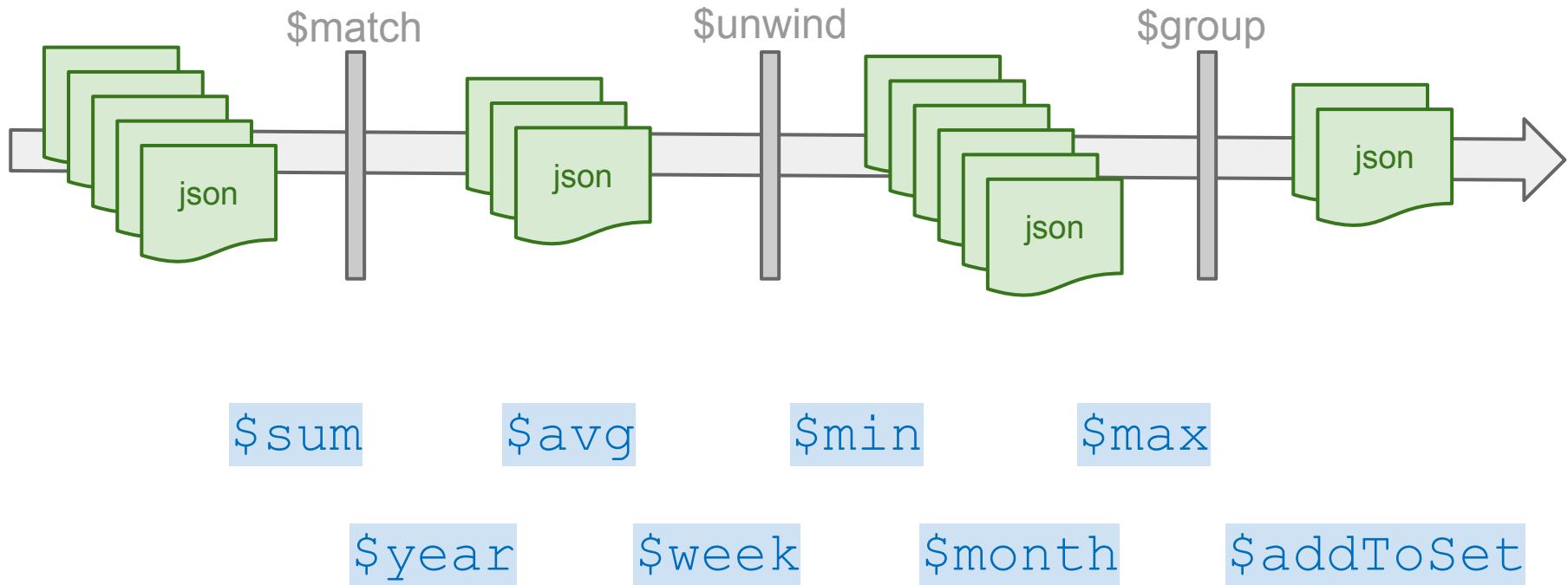
Pour compter, grouper, ... les données

- ▼ Étoiles
 - 1 étoile (3)
 - 2 étoiles (3)
 - 3 étoiles (4)
 - 4 étoiles (4)
 - non classé (3)
- ▼ Type d'établissement
 - Hôtels (16)
 - B&B / Chambres d'hôtes (1)
- ▼ Note des commentaires
 - Très bien : 8+ (7)
 - Bien : 7+ (16)
 - Agréable : 6+ (17)
- ▼ Équipement
 - connexion Wi-Fi au réseau local (17)
 - parking (17)
 - navette aéroport (1)





MongoDB aggregation pipeline

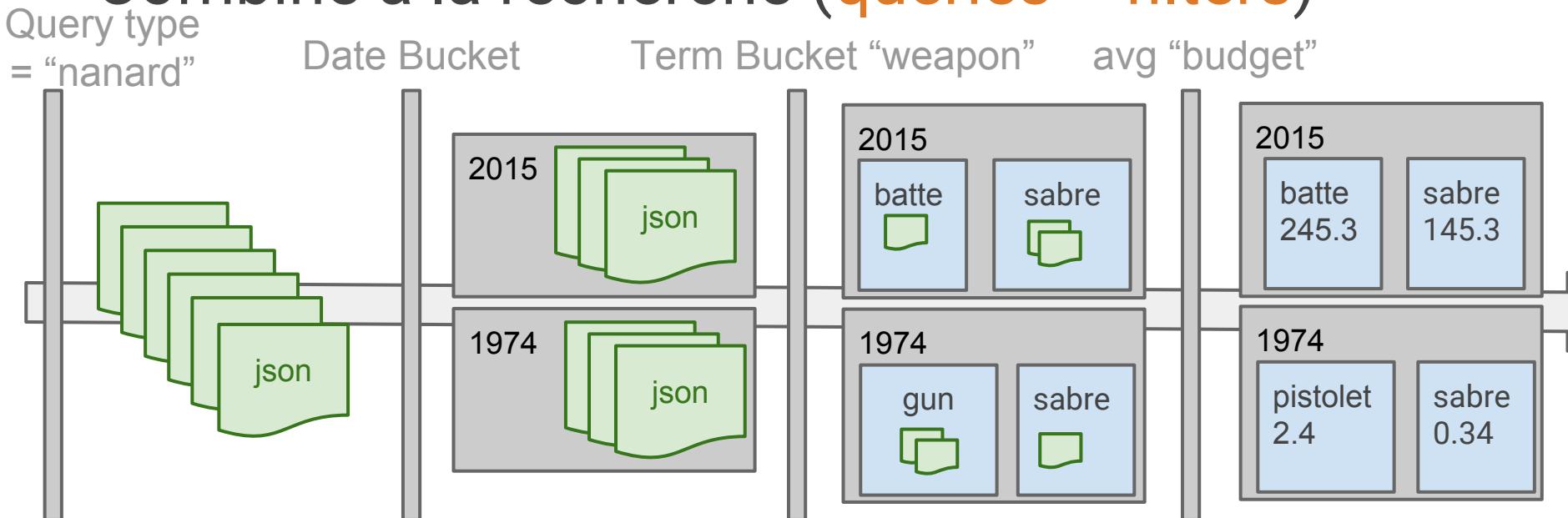




Les agrégations dans Elastic

Deux types d'agrégations : Bucket ou Metrics

Combiné à la recherche (**queries + filters**)





Stockage de fichiers



GridFS

Limite de BSON à 16 MB

Fichiers stockés en morceaux (**chunks**)

Attention à la façon dont le fichier est
reconstitué (**streaming** ?)

Pas de lien direct vers le fichier



Elastic : Attachment Type

Les fichiers sont stockés encodés en Base 64

Pas de streaming

Analyse avec Apache Tika

Par défaut 100000 caractères

Pas de lien direct vers le fichier

Limite théorique : 2Go



GridFS / Attachment Type





Pour stocker des fichiers...

Plusieurs solutions pour stocker beaucoup de fichiers

- HDFS, Ceph, ...
- Amazon S3, Google Cloud Storage, ...



Modèle distribué



RéPLICATION & SHARDING

RéPLICATION

duplication des données (redondance)

→ haute disponibilité

SHARDING

répartition des données sur plusieurs shards

→ scalabilité horizontale



Master

MongoDB et Elastic : Master Node

Si le master tombe, les noeuds restants élisent un nouveau master

Quorum de noeuds : Eviter le split-brain

En cas de partitionnement perte de Disponibilité



Sharding

Dans MongoDB : Configurable

Dans Elastic : By design mais limitant

Immutable

$\text{nbNoeudsMax} = \text{nbShards} \times \text{nbRépliques}$



Call me maybe : Kyle Kingsbury

Jepsen : outil de test des systèmes distribués
En cas de partionnement



<https://aphyr.com/tags/jepsen>



Call me maybe - Elastic

Problèmes de **durabilité**

Mettez les chances de votre côté !



Work In **Quick** Progress : <http://bit.ly/1FWvOr2>



Call me maybe - MongoDB

Problèmes de **cohérence**
(Stale Read / **Dirty** Read)



Certains bugs fixés, d'autres en cours (3.2)

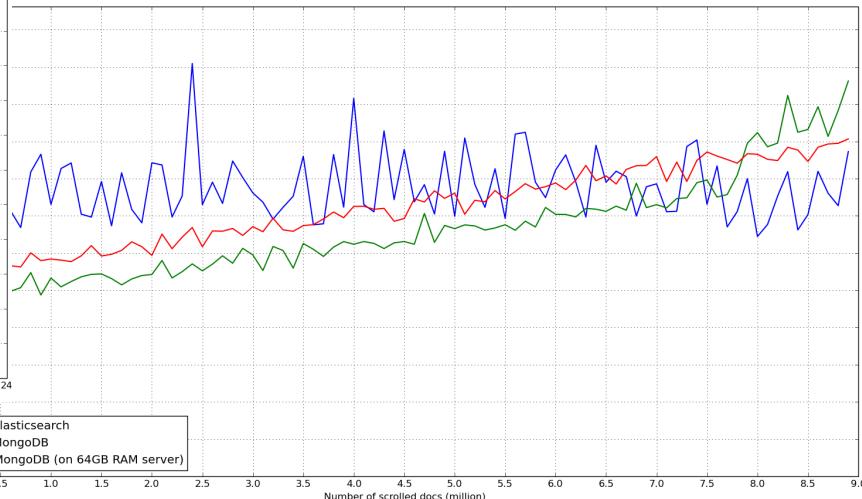
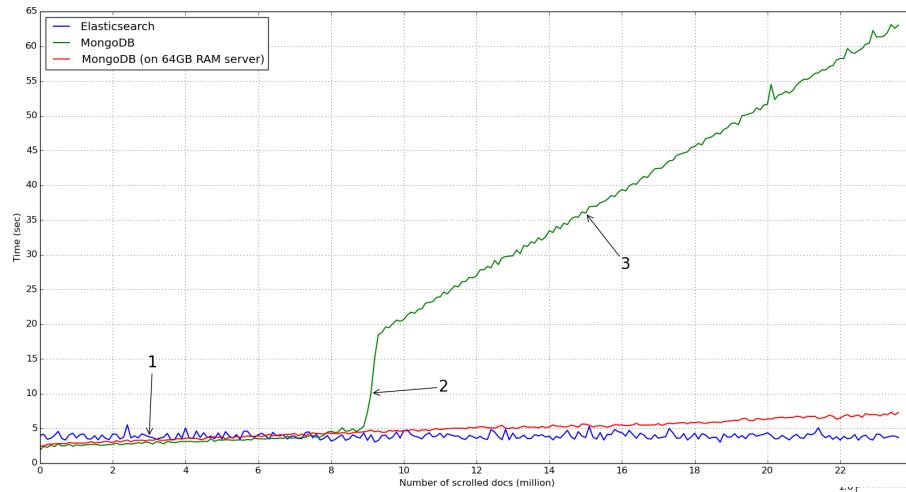


Performances



Benchmark

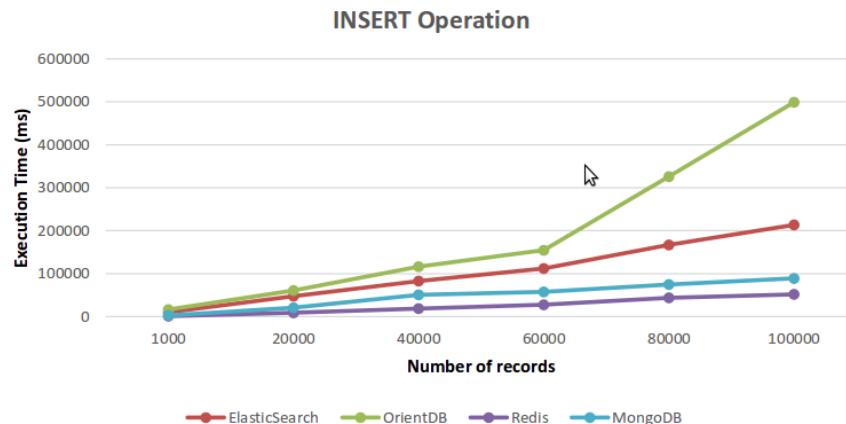
Un article sur **Quarkslab's blog** l'affirme, Elasticsearch est bien plus performant que MongoDB





Benchmark

Une étude de l'**Université de Zaria** est sans appel, **MongoDB** est nettement plus performant que **Elastic**





Benchmarks sucks !





Production



Objectif : Séduire les développeurs
Pas mal d"**"ajustements"** pour passer en prod





Systèmes vivants !

Configuration
Sécurisation
Monitoring
Tuning
Automatisation



Ecosystème



Ecosystème MongoDB

Drivers dans beaucoup de langages

Tooling en **Go** (v3)

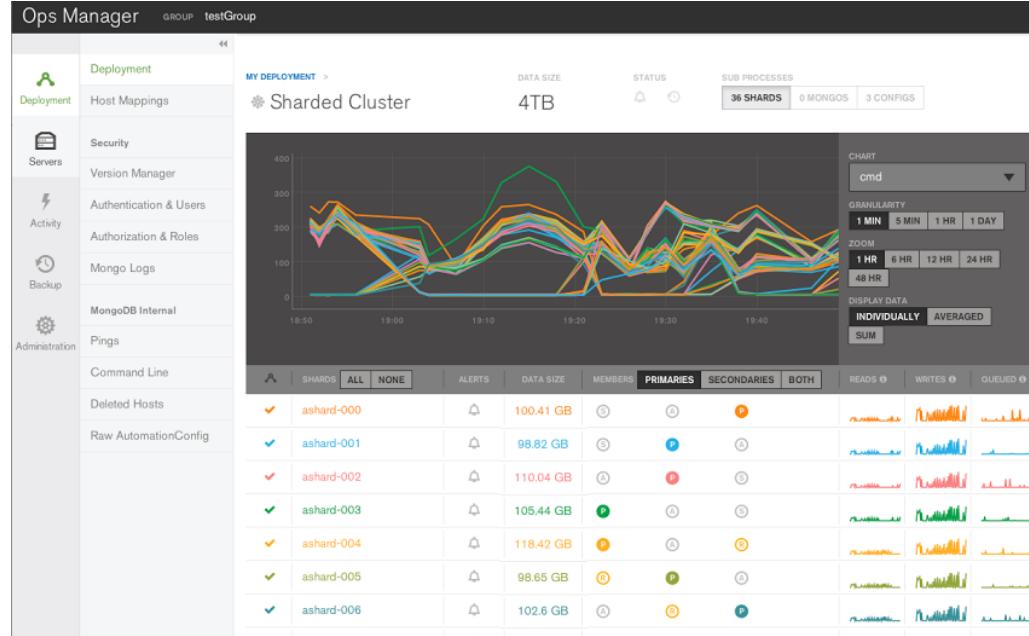
Mongo-Hacker (shell extension)

Beaucoup de **GUI** (mais pas forcément top)



Ecosystème MongoDB

MMS / OpsManager





Client Elastic

Client **natif** pour **Java** (protocole binaire)

API REST

Simple à mettre en oeuvre

Des clients pour tous les langages

Parfois les réponses sont un peu “verbeuses”





Ecosystème Elastic

Un écosystème très dense :

Kibana : Data Viz



Logstash : ETL

Shield : Sécurité

Marvel : Supervision



...





Cloud

Des offres SaaS pour les deux :

MongoDB

MongoLab, MongoSoup, MongoHQ

Elastic

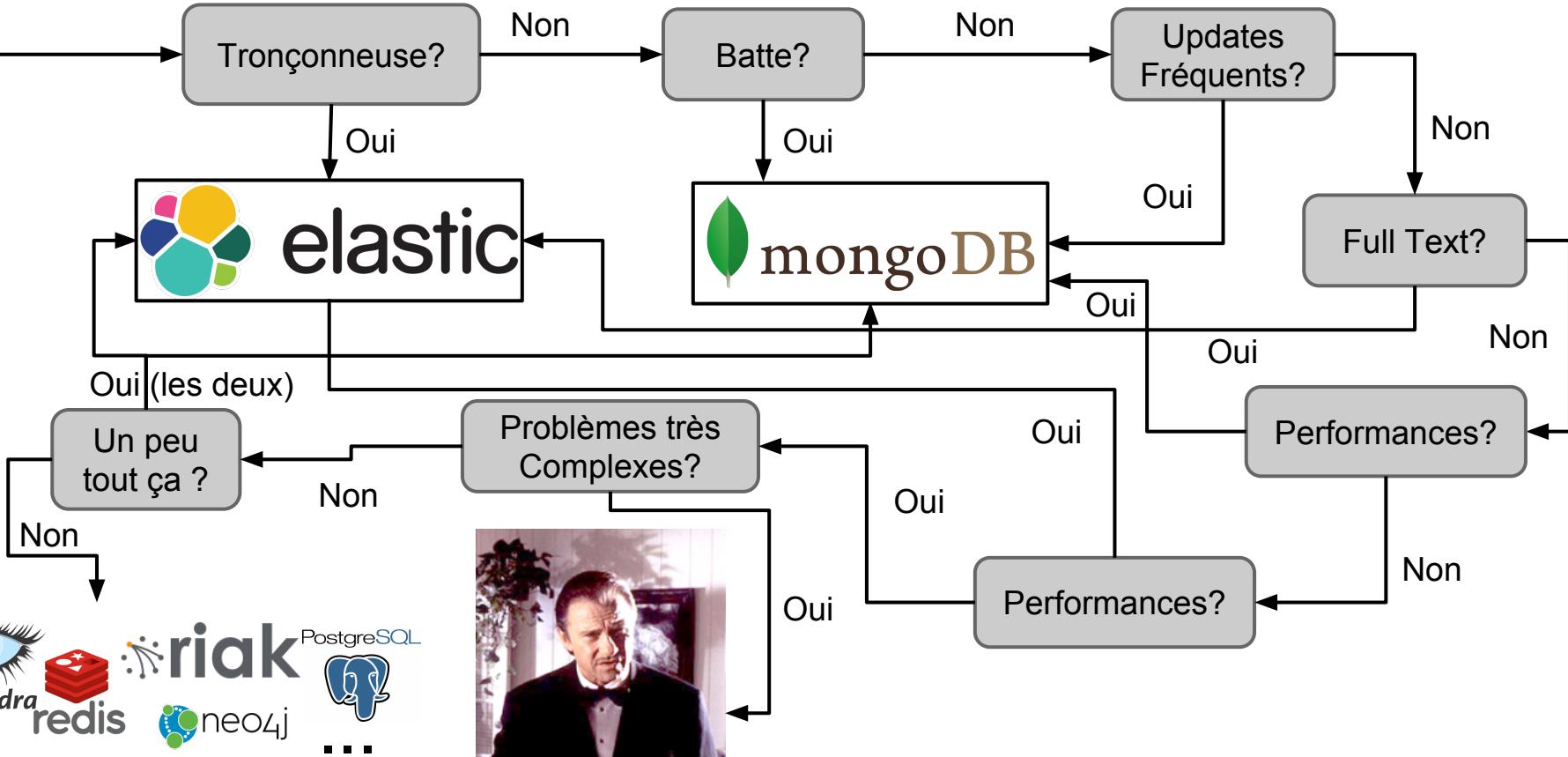
Found, Bonsai, Search



Faire un choix



Alors? MongoDB ou ElasticSearch?





Pour résumer

MongoDB

Update

Temps réél

Elastic

Fonctionnalités

Ecosystème



Les deux ?

Bénéficier des avantages des deux.
Minimiser les inconvénients de chacun.





Même paradigme

Deux bases orientées documents.
Pas besoin de remodéliser (généralement)





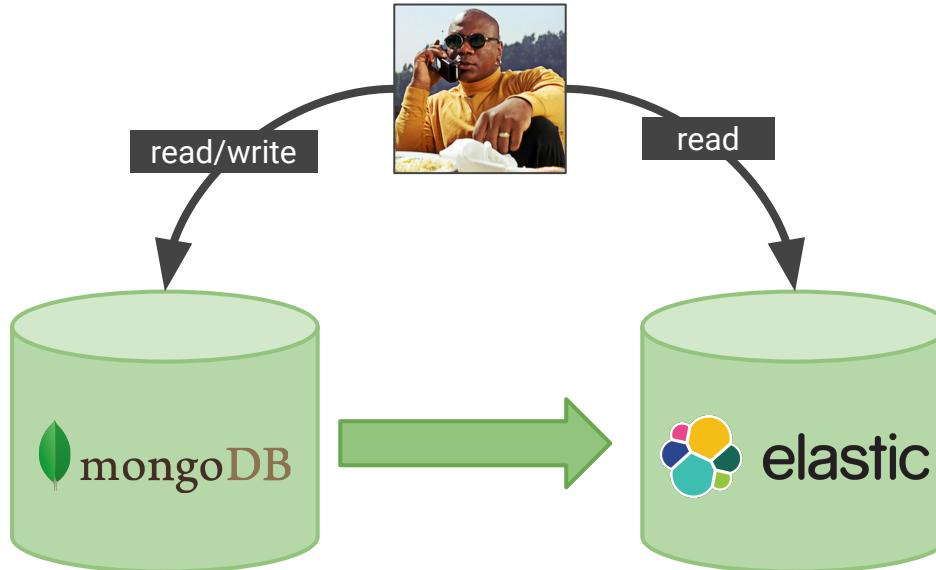
Architecture



Une solution

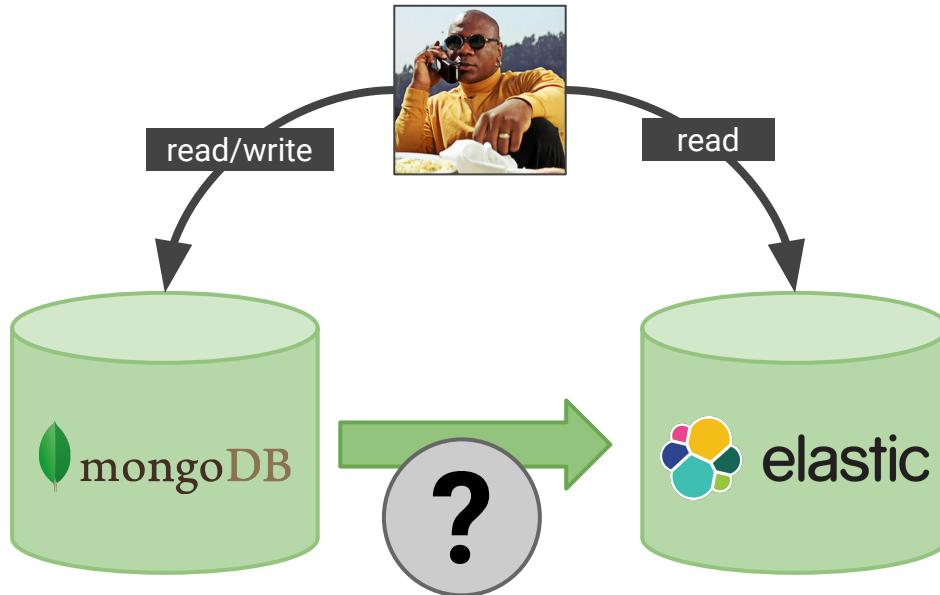
Ecrire dans MongoDB

Lire et rechercher dans Elastic





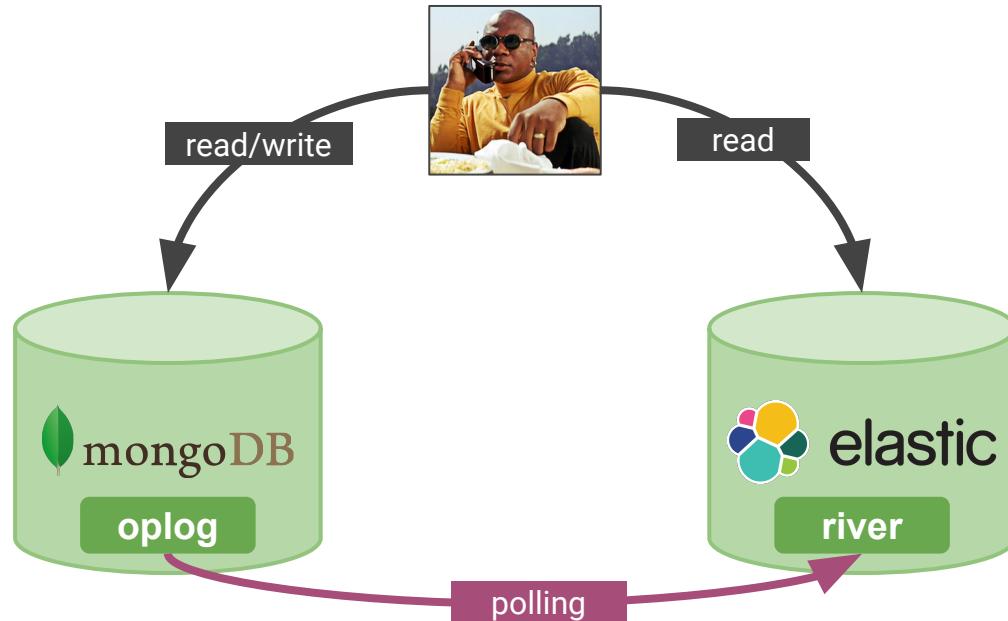
Mais ...



Comment synchroniser ?

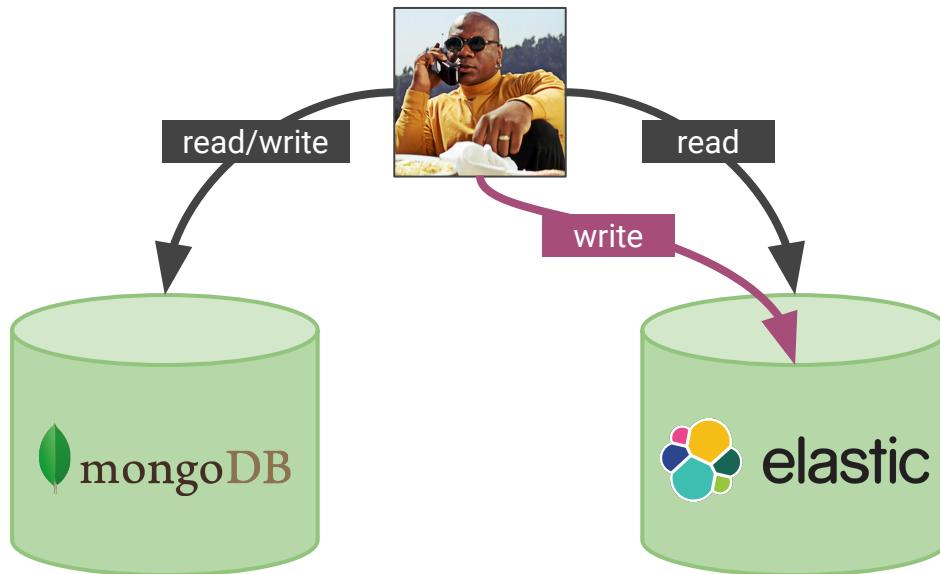


MongoRiver



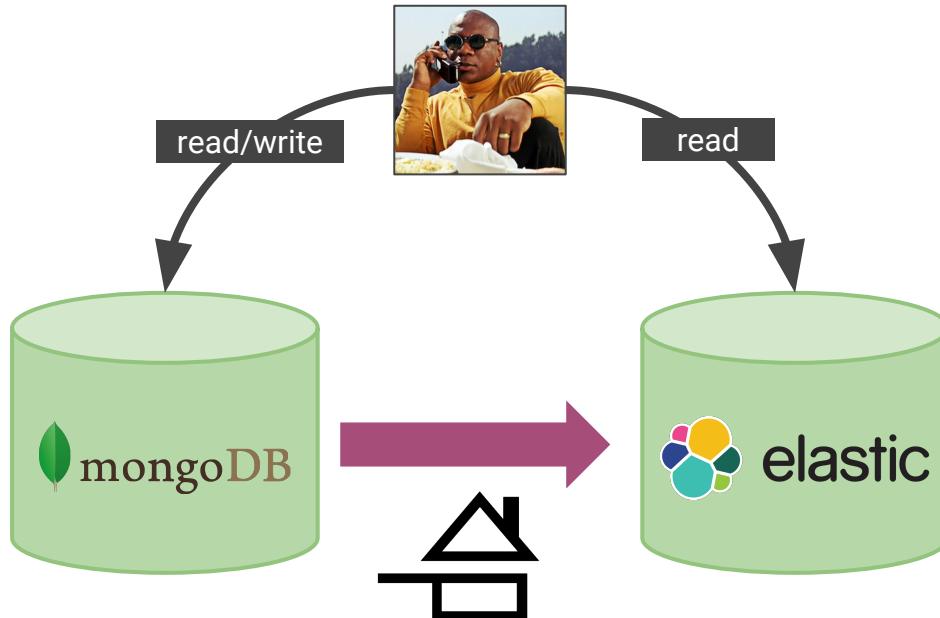


Ecrire dans les deux



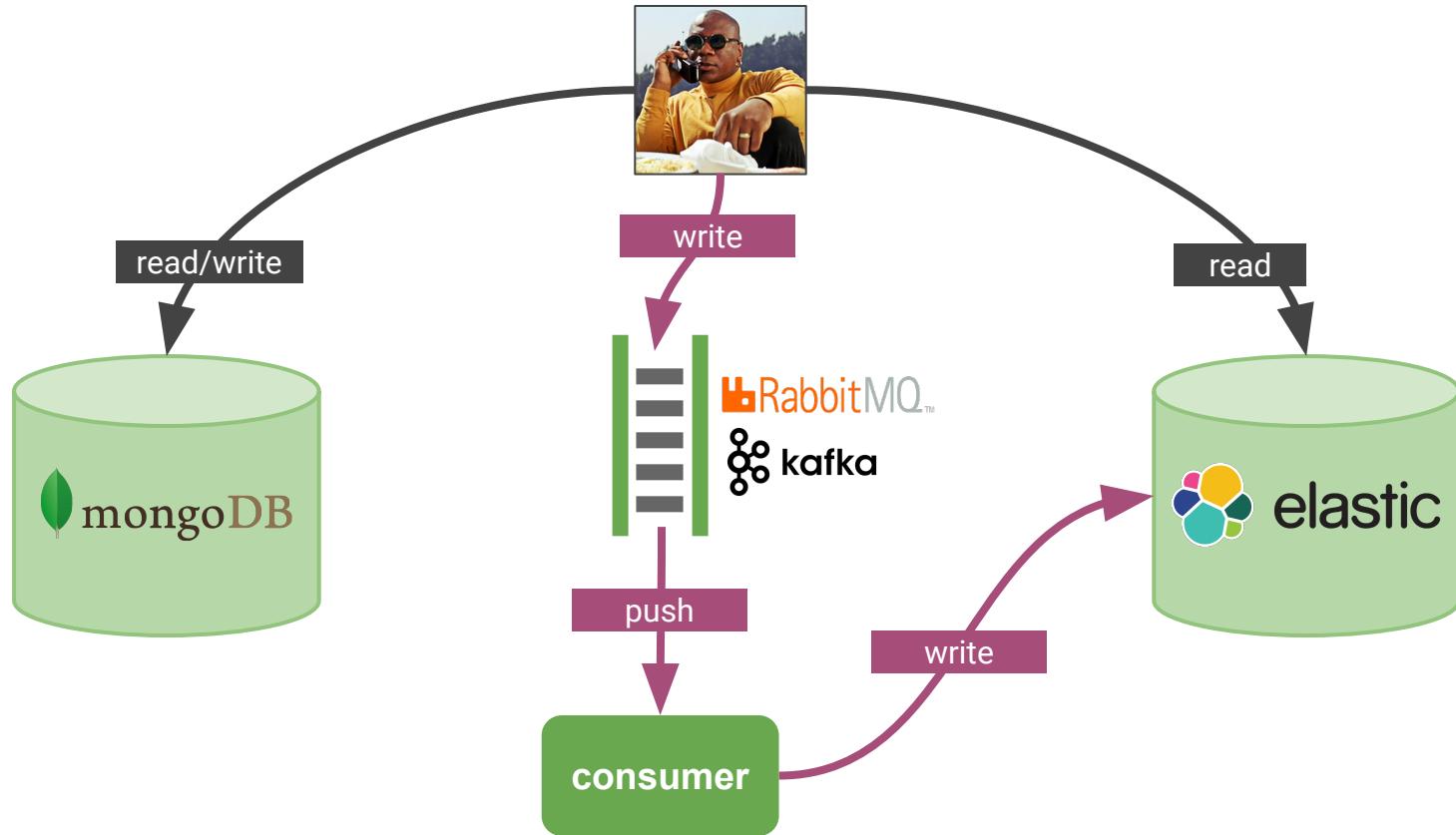


Batch maison





Message Queue





Conclusion

Le bon outil pour le bon
usage !



Merci !

