Rapport d'identification de ArangoDB

1. Editeur

ArangoDB Inc.
Sources (pour les questions 1 à 12) :
https://db-engines.com/en/system/ArangoDB
https://www.arangodb.com/why-arangodb/
https://www.arangodb.com/why-arangodb/cluster/
https://en.wikipedia.org/wiki/ArangoDB

2. Versions initiales (Nr. De version et date)

2011 (avec le nom de AvocadoDB), 2012 (sous le nom de ArangoDB).

3. Versions actuelles (Nr. De version et date)

3.6.0 sortie le 8 janvier 2020.

4. Modèles de données supportés(Clé/Valeur, Orienté document, Orienté Colonnes, Orienté Graphe)

Clé/Valeur, Orienté document, Orienté Graphe.

5. Gestion du schéma (sans schéma, dynamique, statique, mixte)

Sans schéma.

6. Support de SQL (DDL, DML)

Non.

7. Support d'index secondaires

Oui.

8. Langage de développement du sgbd nosql

C++.

9. Support / Pérennité (communauté , etc.…)

Communauté, ArangoDB Inc.

10. API Supportées

AQL(ArangoDB Query Langage)
Foxx Framework
Graph API (Gremlin)
GraphQL query language

HTTP API
Java & SpringData
JSON style queries
VelocyPack/VelocyStream

11. Théorème CAP (CP, AP, AC)

CP : Cohérence et tolérance au Partitionnement : tout le monde voit les mêmes données au même moment et résistance au pannes.

12. Méthode de partitionnement

Sharding (depuis version 2.0).

13. Méthode de réplication

Réplication synchrone et asynchrone. La **réplication synchrone** est utilisée entre les DB-Serveurs d'un ArangoDB Cluster. La **réplication asynchrone** est utilisée dans les autres cas. (https://www.arangodb.com/docs/stable/architecture-replication.html).

14. Concept de consistance

Utiliser la **réplication synchrone** garantie la consistance (https://www.arangodb.com/docs/stable/architecture-deployment-modes-cluster-architecture.html). Lorsqu'on utilise plusieurs bases de données à modèle unique, la consistance devient un problème. Mais avec ArangoDB, un seul back-end gère les différents modèles de données avec la prise en charge des transactions ACID. ArangoDB offre une forte consistance sur une seule instance lorsqu'il fonctionne en mode Cluster. (https://www.arangodb.com/why-arangodb/native-multi-model-database-advantages/).

15. Concept de durabilité

ArangoDB assure aux transactions les propriétés ACID ! (https://blog.eleven-labs.com/fr/introduction-a-arangodb-part-1/). Donc la durabilité.

16. Clés étrangères

Les clés étrangères n'existent pas dans ArangoDB, mais il est possible de définir des liens en utilisant les Joins ou les Edges. (https://stackoverflow.com/questions/26589705/arangodb-link-documents).

17. Support de références (REF)

Pas trouvé.

18. Licences et prix

Licence : Open Source (ApacheV2). Prix : Gratuit.

19. Différents types de versions (communautaire, entreprise, …)

Version communautaire, que l'on utilise. Ou version entreprise, sous licence commerciale, pour connaître le prix, il faut les contacter.

20. Audience dans le marché

Sur le classement général de DB-Engine, ArangoDB est 62ème.

51.	↑ 54.	↓ 50.	Ehcache	Key-value	6.23	-0.03	-0.36
52.	↑ 55.	↑ 54.	Microsoft Azure Search	Search engine	6.12	-0.10	-0.14
53.	↓ 50.	↓ 46.	Aerospike	Key-value, Multi-model	6.06	-0.88	-1.11
54.	↓ 52.	↓ 52.	Sphinx	Search engine	6.02	-0.46	-0.35
55.	↓ 53.	↓ 47.	Interbase	Relational	5.99	-0.47	-0.88
56.	↑ 60.	↑ 58.	SAP SQL Anywhere	Relational	5.51	+0.28	+0.20
57.	57.	↓ 55.	Kdb+	Time Series, Multi-model	5.37	+0.09	-0.23
58.	↓ 56.	↓ 56.	Ingres	Relational	4.97	-0.48	-0.53
59.	↓ 58.	↓ 57.	Derby	Relational	4.97	-0.28	-0.43
60.	↓ 59.	↓ 49.	Riak KV	Key-value	4.76	-0.48	-2.04
61.	↑ 62.	↑ 64.	ArangoDB	Multi-model	4.68	-0.20	-0.11
62.	↑ 63.	↑ 69.	Microsoft Azure Table Storage	Wide column	4.68	-0.17	+0.26
63.	↑ 65.	↑ 66.	Google Cloud Datastore	Document	4.66	-0.14	-0.06
64.	64.	↓ 60.	HyperSQL	Relational	4.62	-0.20	-0.54

Si on regarde le classement dédié aux graph DBS, il est 3ème.

Rank			DBMS	Database Model	Score		
May 2020	Apr 2020	May 2019			May 2020	Apr 2020	May 2019
1.	1.	1.	Neo4j	Graph	49.76	-1.05	-1.27
2.	2.	2.	Microsoft Azure Cosmos DB	Multi-model	30.68	-1.37	+3.08
3.	3.	↑ 4.	ArangoDB	Multi-model	4.68	-0.20	-0.11
4.	4.	↓ 3.	OrientDB	Multi-model	4.14	-0.38	-2.23
5.	5.	5.	Virtuoso	Multi-model	2.35	-0.27	-0.97
6.	6.	↑ 7.	Amazon Neptune	Multi-model	1.76	-0.04	+0.44
7.	7.	↓ 6.	JanusGraph	Graph	1.65	-0.13	+0.03
8.	8.	↑ 10.	GraphDB	Multi-model	1.19	+0.02	+0.15
9.	9.	9.	Dgraph	Graph	1.09	-0.03	+0.05

Si on regarde maintenant le classement dédié aux Key-Value Stores, il est 10ème.

Rank			DBMS	Database Model	Score		
May 2020	Apr 2020	May 2019			May 2020	Apr 2020	May 2019
1.	1.	1.	Redis	Key-value, Multi-model	143.48	-1.33	-4.93
2.	2.	2.	Amazon DynamoDB	Multi-model	64.72	+0.45	+8.78
3.	3.	↑ 4.	Microsoft Azure Cosmos DB	Multi-model	30.68	-1.37	+3.08
4.	4.	↓ 3.	Memcached	Key-value	23.93	-1.41	-4.97
5.	5.	5.	Hazelcast	Key-value, Multi-model	8.67	-0.49	+0.25
6.	6.		etcd	Key-value	7.28	+0.10	
7.	↑ 8.	↑ 8.	Ehcache	Key-value	6.23	-0.03	-0.36
8.	↓ 7.	↓ 6.	Aerospike	Key-value, Multi-model	6.06	-0.88	-1.11
9.	9.	↓ 7.	Riak KV	Key-value	4.76	-0.48	-2.04
10.	↑ 11.	↑ 11.	ArangoDB	Multi-model	4.68	-0.20	-0.11
11.	↓ 10.	↓ 10.	Ignite	Multi-model	4.56	-0.39	-0.49
12.	12.	↓ 9.	OrientDB	Multi-model	4.14	-0.38	-2.23
13.	13.	↓ 12.	Oracle NoSQL	Key-value, Multi-model	3.76	-0.05	+0.14

Il est également 10ème dans le classement dédié aux Documents stores.

Rank			DBMS	Database Model	Score		
May 2020	Apr 2020	May 2019			May 2020	Apr 2020	May 2019
1.	1.	1.	MongoDB	Document, Multi-model	438.99	+0.57	+30.92
2.	2.	2.	Amazon DynamoDB	Multi-model	64.72	+0.45	+8.78
3.	3.	↑ 4.	Microsoft Azure Cosmos DB	Multi-model	30.68	-1.37	+3.08
4.	4.	↓ 3.	Couchbase	Document, Multi-model	28.58	-1.83	-6.09
5.	5.	5.	CouchDB	Document	16.92	-0.85	-2.19
6.	6.	↑ 7.	Firebase Realtime Database	Document	13.12	+0.47	+1.84
7.	7.	↓ 6.	MarkLogic	Multi-model	10.96	-0.30	-3.09
8.	8.	8.	Realm	Document	8.38	-0.16	+0.74
9.	9.	↑ 10.	Google Cloud Firestore	Document	6.34	-0.27	+1.35
10.	10.	↑ 11.	ArangoDB	Multi-model	4.68	-0.20	-0.11
11.	11.	↑ 13.	Google Cloud Datastore	Document	4.66	-0.14	-0.06

Et enfin, dans le classement dédié aux Search Engines, il est 7ème.

Rank May 2020	Rank Apr 2020	Rank May 2019	DBMS	Database Model	Score May 2020	Score Apr 2020	Score May 2019
1.	1.	1.	Elasticsearch	Search engine, Multi-model	149.13	+0.22	+0.51
2.	2.	2.	Splunk	Search engine	87.75	-0.33	+2.51
3.	3.	3.	Solr	Search engine	52.58	-1.01	-8.22
4.	4.	4.	MarkLogic	Multi-model	10.96	-0.30	-3.09
5.	↑ 6.	↑ 6.	Microsoft Azure Search	Search engine	6.12	-0.10	-0.14
6.	↓ 5.	↓ 5.	Sphinx	Search engine	6.02	-0.46	-0.35
7.	7.	7.	ArangoDB	Multi-model	4.68	-0.20	-0.11
8.	8.	8.	Algolia	Search engine	4.46	+0.01	+0.31
9.	↑ 11.	↑ 10.	Google Search Appliance	Search engine	2.59	-0.03	-0.38
10.	↓ 9.	↑ 11.	Amazon CloudSearch	Search engine	2.59	-0.11	-0.22
11.	↓ 10.	↓ 9.	Virtuoso	Multi-model	2.35	-0.27	-0.97

21. Tables et tables filles

Pas de table.

22. Clé avec major et minor key

Pas vu.

23. Gestion des utilisateurs

Oui. (https://www.arangodb.com/arangodb-user-management/).

24. Gestion des droits

Oui. (https://www.arangodb.com/arangodb-user-management/).

25. Gestion des namespaces ou databases

La gestion des bases de données utilise l'interface HTTP de ArangoDB.

Cette interface fournit des opérations pour créer, lister et supprimer des bases de données. Ces dernières sont mises en correspondance avec les méthodes du standard HTTP (POST, DELETE et GET).

Note : Ce système de management est uniquement accessible via la base de données par défaut (_system).

Détails :

- La méthode **GET** (suivie d'une route spécifique de l'utilisateur) permet de récupérer les informations d'une base de données ou encore d'afficher la liste des bases de données accessibles à l'utilisateur.
- La méthode **POST** permet de créer une base de données. Elle nécessite un objet JSON composé de propriétés spécifiques (nom, options, utilisateurs). La réponse est un objet JSON contenant l'attribut *result* égal à *true* or *false*.
- La dernière méthode **DELETE** permet de supprimer une base de données existante.

Source : https://www.arangodb.com/docs/stable/http/database-database-management.html

26. Systèmes d'exploitation supportés

ArangoDB est disponible sur les principaux systèmes d'exploitation : Windows, MacOS et Linux.

Ajoutons que pour ce dernier, des images officielles sont disponibles pour les distributions les plus populaires (Ubuntu, ArchLinux, Debian, Fedora…) mais il existe aussi des images non officielles fournies par la communauté.

Il comprend aussi les systèmes Kubernetes et Docker.

Source : https://www.arangodb.com/docs/stable/installation.html

27. Disponible en mode DBaas

ArangoDB est disponible en mode DBaas (base de données en tant que service) à travers sa plateforme ArangoDB Cloud appelée ArangoDB Oasis.

Note : Le choix du fournisseur de services en ligne reste libre selon la liste suivante : Google Cloud Platform, Amazon Web Services, Microsoft Azure.

Source : https://www.arangodb.com/docs/stable/oasis/

28. Support du Map/Reduce

Il n'y a pas de support intégré pour le Map/Reduce.

Notes : Cependant, ArangoDB propose une API pour les méthodes de Map/Reduce définies par l'utilisateur et il semble que le Map/Reduce puisse être réalisée avec des procédures stockées en Javascript.

Source : https://db-engines.com/en/system/ArangoDB

29. Lien vers la documentation technique y compris les API

https://www.arangodb.com/docs/stable/index.html

30. Typage (none, static, dynamique)

AQL (ArangoDB Query Language) supporte les types de données primitives d'une valeur et les types de données composées comprenant plusieurs valeurs.

Voici la liste :

- Null
- Boolean
- Number
- String
- Array (list)
- Object (document)

Source : https://www.arangodb.com/docs/stable/aql/fundamentals-data-types.html

31. Applications communautaires l'utilisant

Voici une courte liste de projets communautaires autour d'ArangoDB :

a) Plugins/Connecteurs :

- ArangoDB Kubernetes Operator (https://github.com/arangodb/kube-arangodb)
- Spring Data ArangoDB (https://github.com/arangodb/spring-data)
- arangodb-spark-connector (https://github.com/arangodb/arangodb-spark-connector)

- guacaphant (https://github.com/deusdat/guacaphant)

b) Projets:

- waller (https://github.com/deusdat/waller) une librairie Clojure pour utiliser Ragtime.
- LAS2peer-AnnotationService (https://github.com/rwth-acis/LAS2peer-AnnotationService) un service de stockage d'annotations Java.
- Graphiti (https://github.com/sleepycat/graphiti) bibliothèque Ruby permettant de faire communiquer une classe de hachage à une vase de données ArangoDB.

Source : https://www.arangodb.com/projects-and-integrations/

32. Domaines d'applications

Le multi-modèle natif d'ArangoDB est utilisé pour divers projets dans des secteurs et des entreprises de toutes tailles : « Single View Of Everything », cybersécurité, détection de fraude, moteur de recommandations ou encore gestion et surveillance de réseau.

Voici quelques exemples :

- Refinitv: "single view of everything" rapide et sécurisée
- Liaison's Health Platform: solution de Data Platform as a Service (dPaas) pour la santé
- Gestion des certificats numériques et des clés cryptographiques
- Triton : Apprentissage du code avec ArangoDB aux étudiants

Une liste complète est disponible :

Source : https://www.arangodb.com/why-arangodb/case-studies/

33. Architecture du moteur No SQL

L'architecture du moteur NoSQL est une architecture en cluster qui se base sur un modèle CP master/master sans point de défaillance unique.

CP pour faire référence au CAP théorème c'est-à-dire qu'en présence d'une partition réseau la base de données met en avant la cohérence interne devant la disponibilité.

Master/Master signifie que les clients peuvent envoyer des requêtes à un nœud arbitraire tout en ayant accès à la même vue.

Pour finir, l'expression sans point de défaillance unique signifie que le cluster peut continuer à servir des requêtes même si une machine tombe en panne.

Un cluster ArangoDB est constitué de plusieurs instances ArangoDB qui communiquent entre elles au sein d'un réseau. Chaque instance jour un rôle différent :

- Agent : il peut y en avoir un ou plusieurs par cluster et forment l'Agence. Cette Agence est un magasin de clés/valeur basé sur un nombre impair d'instances ArangoDB exécutant le *Raft Consensus Protocol* et qui stocke la configuration du Cluster. Elle effectue une élection de leaders et fournit des services de synchronisation pour l'ensemble du Cluster.

- Coordinateur : il est accessible de l'extérieur du réseau et communique avec les clients. Il coordonne les tâches du Cluster (services Foxx, exécution de requêtes) et sait où sont stockées les données ce qui permettra d'optimiser la localisation des requêtes fournies par les utilisateurs. Il y a souvent plus d'un coordinateur.
- Serveur DB : c'est ici que les données sont stockées. Il héberge des « shards » de données. Il est soit dirigeant (leader) soit suiveur (follower) d'une shard de données. Ces shards sont accessibles indirectement via les coordinateurs par les utilisateurs.

En résumé, le moteur NoSQL est organisé en Shard. Ces shards sont stockés dans des ServeursDB contenant les données. Une instance ArangoDB contient une ou plusieurs bases de données. Une base de données comprend 0, 1 ou plusieurs collections. Une collection comprend 0,1 ou plusieurs documents.

Notes : Il est possible d'utiliser une instance unique d'ArangoDB (comme nous allons le faire au cours de ce TP) autonome, sans réplication, sans possibilité de basculement et non en tant que regroupement de nœuds. (Cluster).

2 autres environnements sont disponibles : Master/Slave et Active Failover.

Sources : https://www.arangodb.com/why-arangodb/cluster/
https://www.arangodb.com/docs/3.6/architecture-deployment-modes-cluster-architecture.html
https://www.arangodb.com/docs/stable/architecture-deployment-modes-cluster-sharding.html

34. Montée en charge

ArangoDB étant une base de données distribuée prenant en charge plusieurs modèles de données elle est *scalable horizontalement* c'est-à-dire en utilisant de plus en plus de serveurs (ajout de nœuds).

Selon ses développeurs cette approche est préférable pour ArangoDB car elle permet d'améliorer les performances et la capacité tout en assurant sa résilience à la réplication et au basculement automatique (fail-over).

ArangoDB est aussi *scalable verticalement* en utilisant des serveurs de plus grande capacité car il n'y a pas de limitation intégrée (le serveur utilisera automatiquement plus de threads si plus de CPUs sont présents).

Source : https://www.arangodb.com/docs/stable/scaling.html

35. Gestion de la disponibilité

Un cluster peut toujours être lu à partir d'une collection si des shards deviennent non disponibles. Les données d'une shard non disponible ne sont pas accessible mais la lecture d'autres shards restera possible.

Source : https://www.arangodb.com/docs/3.6/architecture-deployment-modes-cluster-sharding.html#high-availability

36. Procédure d'installation

WINDOWS :

2 méthodes possibles :

- De façon automatisée avec NSIS Installer (assisté (GUI) ou en ligne de commande)

Le répertoire d'installation par défaut est C:\Program Files\ArangoDB-3.x.x

Des options d'installations sont disponibles au choix (conserver une sauvegarde des données, ajouter des exécutables au chemin d'accès, créer une icône de bureau…)

Pour lancer le serveur il suffit de lancer l'exécutable arangod.exe situé dans le sous dossier *usr/bin* (ArangoDB is ready for business. Have fun ! certifie l'installation). Ce dernier est accessible à l'adresse suivante : *http://127.0.0.1:8529/*

- Manuellement avec une archive ZIP (installation avec XCopy)

LINUX :

1) Télécharger le package de la distribution Linux appropriée.
2) Suivre les instructions du gestionnaire de package.
3) Installer ArangoDB avec yum, aptitude, urpmi ou zypper.

macOS :

Plusieurs méthodes sont possibles :

- Homebrew
- DMG Package
- Tar.gz Archive

Plus d'informations sont disponibles : https://www.arangodb.com/docs/stable/installation.html