NoSQL Database option to store FHIR messages

Cassandra VS MongoDB

What are FIHR messages?

FHIR stands for Fast Healthcare Interoperability Resources. It is a draft standard describing data formats and elements known as "resources" and an API for exchanging electronic health records. It uses a choice of JSON or XML for data representation and according to www.healthecare-informatic.com one of its goals is to facilitate interoperation between healthcare systems, so that it is easier to deliver health care information to healthcare providers and individuals on many different devices such as tablets, phones and computers.

Furthermore, FHIR does not provide a document centric approach, as it directly exposes data elements as services, this means that basic elements can be reclaimed and manipulated/worked on via their own resource URLs.

What is NoSQL?

NoSQL database is a non-relational or not only SQL and largely distributed database system that enables rapid and analysis of large volume, different data types. They provide scalability, availability and fault tolerance being key factors and seek to solve big data performance issues that relational databases weren't designed to address.

What is MongoDB?

MongoDB is a free open source cross platform document oriented database program. It is a NoSQL database and it is not the traditional table based relational database. Instead it uses a JSON-like document calling the format BSON. The benefit of this NoSQL database integration is that it makes the integration of data in certain types of applications easier and faster, but this means that the data is stored in an unstructured manner.

What is Cassandra?

Apache Cassandra, similarly to MongoDB is also a free NoSQL open source cross platform distributed database management system. It is designed to handle large amount of data across many commodity servers. Cassandra offers no single point of failure with high availability which means that if part of a system fails it will not stop the entire system from working. In contrast to MongoDB Cassandra is "Column-Based" and not "Document-Based".

Difference between Column-Based and Document-Based Database

- **Column-Based**

  They store data in related rows however they translate the data structure into a format that can be stored (serialisation) into columns. The benefit of this is that it allows for very quick querying and processing of data while storing data that is somewhat related. In other terms, each row (addressed by a key) contains one or more "columns". Columns

themselves are key-value pairs and they are ordered and stored according to their keys (names).

- **Document-Based**

  Stores data typically in “documents”, usually XML or JSON documents, they are usually schemaless which means the data is unstructured which means you can have any data that you want in them and change them on the fly. Documents contain key value pairs, which can contain any sort of value, array or even another document. Documents allow you to store your data how you see useful and it has a fast writing and good query times based on indexing. The biggest advantage however is the schema flexibility.

Advantages and Disadvantages of MongoDB

Pros:

- Schema-less, if you have a flexible schema, this is ideal for a document store like MongoDB
- Ease of scale out. Scale reads by using replica sets
- MongoDB is free and ideal for running on cheap commodity kit
- Document based database (simple JSON data). Also complex data can be modelled very easily because of the rich semantics of JSON documents.

Cons:

- Data size is higher, for example each document has field name stored it
- Less flexibility with querying (i.e no JOINS)
- Aggregations/data analysis is not fast, however if required Hadoop may be needed to be added to the mix
- Less up to date information available

Advantages and Disadvantages of Cassandra

Pros:

- Cassandras write speed is very fast for the amount of write volume it can handle
- It includes multi data centre replication, this replication will copy the information to any number of instances of the Cassandra process
- JVM based, easy integration with other JVM based applications such as Hadoop
- Includes Cassandra Query Language for querying Cassandra. It is a subset of SQL.

Cons:

- No Ad-Hoc queries, this means you have to model your data around the queries you want to surface
- No Aggregations, since it is a key-value store, things like SUM, MIN, MAX, AVERAGE are incredible resource intensive
- Unpredictable performance as Cassandra has many background tasks that are not scheduled by the user

- Its offered data structures does not include JSON, so using it to manipulate large amounts of JSON data can be very difficult

Which Database is the most suitable?

I believe for the system engineering project, MongoDB will be the best choice. Since I believe we will be dealing with JSON data, MongoDB is an excellent choice as it uses BSON to store the data. BSON is the binary encoding of JSON-like documents that MongoDB uses when storing documents in collections. It further adds support for data types like Date and binary that aren't supported in JSON. Furthermore, it has deep query-ability; it supports dynamic queries on documents using a document based query language that's almost as powerful as SQL.

Additionally it is schema-less, in which one collection holds different documents and number of fields content and size of the documentation can differ from one document to another. MongoDB is also easy to scale and it uses internal memory for storing the working set which means faster access of data.

MongoDB on Azure

Microsoft azure is a cloud platform that enables you to quickly build, deploy, and manage application across a network of Microsoft-managed data centres. It enables you to create any applications using any programming language of your choice, tools and frameworks.

You are able to deploy MongoDB directly on Azure using the Azure VM (Azure Infrastructure-as-a-service)

Useful Links:

MongoDB Inc, MongoDB Ecosystem, 'Microsoft Azure', Documentation, 2017 [Online]. Available at: https://docs.mongodb.com/ecosystem/platforms/windows-azure/

Iain Foulds, Ralph Squillace, Install MongoDB on a Windows VM in Azure, 2017 [Online]. Available at: https://docs.microsoft.com/fr-fr/azure/virtual-machines/windows/classic/install-mongodb

MongoDB Inc, MongoDB Documentation, 2017 [Online]. Available at https://docs.mongodb.com/

FIHR, FIHR Release 3, Documentation Index, 2011 [Online]. Available at: https://www.hl7.org/fhir/documentation.html