Dissertation Project

A Comparison of NoSQL and Indexing Solutions for Big Data

Author: Callum George William Guthrie

Supervisor: Dr. Albert Burger

Heriot Watt University

Edinburgh, Scotland

A dissertation submitted in partial fulfilment of the requirements for the degree of Bachelor of Science.

May 2016

Contents

Chapter 1

Introduction

The era of Big Data is upon us bringing with it a range of new challenges ”Without big data, you are blind and deaf and in the middle of a freeway.” (Moore, 2012) The importance which accompany these challenges encouraged the formulation of new approaches for cleaning, processing and using these enormous amounts of data. (Section 2.1)

Data is being collated and stored every second of every day and the value of doing so has never been greater. Billion dollar companies such as Google and Amazon dominate the market in data collection and pride themselves in knowing everything about everything. Former CEO of Google, Eric Schmidt famously said in 2010 ”We know where you are. We know where you've been. We can more or less know what you're thinking...” (Schmidt, 2010) Thus the power of data collection has led to the development of a range of technologies designed to meet the needs of Big Data.

The purpose of the following research, by way of investigation, is to deliver an insightful examination of a subset of new technologies which deliver high performance querying of large datasets. The ultimate aim of the research is to gain an understanding of these technologies and achieve a level of mastery which permits a thorough scrutiny of their application to Big Data.

There are a number of different indexing solutions available however for the means of this project to encapsulate a comprehensive examination a focus will be on leading NoSQL solutions, modern search and analytics engines and for comparative reasons a conventional

relational database management system. The following technologies to be used for the project:

1. MongoDB (Section 3.1.1.1)
2. Neo4j (Section 3.1.2.1)
3. Apache Cassandra (Section 3.1.3.1)
4. MySQL (Section 3.1.4.1)

1.1 Objectives

The key objectives and main intended outcomes for the project are:

- Investigate the strengths and weaknesses of the functionality each technology provides.
- Compare and contrast the analytical capabilities of each technology by way querying prototype models.
- Conduct a comparative analysis to investigate the scalability of each technology.

1.2 Project Intentions

Choosing an indexing solution for querying Big Data sets can be difficult as there are so many options with limits on each respective functionalities. Taking the A & O into consideration the project intends to identify the best overall performing indexing solution for this dataset -

EXPAND

1.2.0.1 Previous Study

INCOMPLETE

Chapter 2

Background

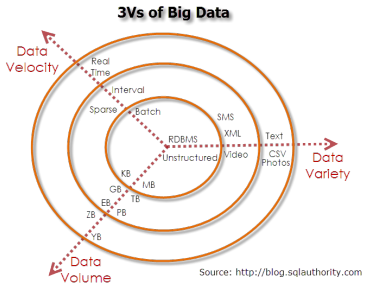
To deliver an all round level of comprehension the following section discusses generic terms which are used throughout the project such as Big Data (Section 2.1) and Extract Transform Load (Section 2.2)

2.1 Big Data

Big Data is a broad evolving term bound to a complex and powerful application of analytical insight which over recent years has had a variety of definitions. In simplistic terms Big Data can be described as extremely large datasets that may be studied computationally to reveal patterns, trends, and associations for ongoing discovery and analysis.

2.1.1 3vs Model

In 2001, Gartner analyst Doug Laney delivered the original 3vs model which categorises big data in to three dimensions; Volume, Variety and Velocity. The characteristics of each property are defined as **Volume** - The size of the generated data is required to assess whether the dataset is in fact 'big' enough to be categorised as Big Data.



Variety - The type of content, and an essential fact that data analysts must know. This helps people who are associated with and analyze the data to effectively use the data to their advantage and thus uphold its importance.**Velocity** - In this context, the speed at which the data is generated and processed to meet the demands and the challenges that lie in the path of

growth and development. **Variability** - The inconsistency the data can show at times?-which can hamper the process of handling and managing the data effectively. **Veracity** - The quality of captured data, which can vary greatly. Accurate analysis depends on the veracity of source data. Laney's 2001 publication *3D data management: Controlling data volume, variety and velocity* is still widely recognised today as the expansion of all three properties encapsulate the challenges currently faced of big data management.

2.2 Extract Transform Load

A basic definition of the Extract Transform Load (ETL) process is pulling data out of one database, refactoring the composition of the data and putting the data into another database. - EXPAND

2.2.1 Process

ETL is a three step procedure which combines database functions into one tool. - EXPAND

2.2.1.1 Extract

is the first step in the ETL procedure in which data is read from a database. The Extract step covers the data extraction from the source system and makes it accessible for further processing. The main objective of the extract step is to retrieve all the required data from the source system with as little resources as possible. The extract step should be designed in a way that it does not negatively affect the source system in terms or performance, response time or any kind of locking.

2.2.1.2 Transform

where the extracted data is manipulated from its previous state and converted into another database format. The transform step applies a set of rules to transform the data from the source to the target. This includes converting any measured data to the same dimension (i.e. conformed dimension) using the same units so that they can later be joined. The transformation step also requires joining data from several sources, generating aggregates,

generating surrogate keys, sorting, deriving new calculated values, and applying advanced validation rules.

2.2.1.3 Load

completes the three step procedure and is where data is written into the target database. During the load step, it is necessary to ensure that the load is performed correctly and with as little resources as possible. The target of the Load process is often a database. In order to make the load process efficient, it is helpful to disable any constraints and indexes before the load and enable them back only after the load completes. The referential integrity needs to be maintained by ETL tool to ensure consistency.

2.2.2 Tool Implementation

https://jena.apache.org/ EXPAND

Chapter 3
Technical Discussion

The focus of this technical investigation was to develop my knowledge of the NoSQL and indexing solutions examined in the project and to gain insight into the subject matter by reflecting on previously conducted research. Prior to this research, my understanding of the functionality provided by Neo4j and Apache Cassandra was minimal. Throughout my university degree I have undertaken modules which have stipulated a working knowledge of MySQL as a prerequisite therefore my comprehension of MySQL is proficient. - EXPAND

3.1 Database Classification

One of the first decisions to be made when when selecting a database is the characteristics of the data you are looking to leverage. (Dash, 2013) There are a multitude of options available with many different classifications. - EXPAND

3.1.1 Document-Oriented Database

Document-orientated database (DODB) are designed for storing, retrieving and managing document files such as XML, JSON and BSON. The documents stored in a DODB model are data objects which describe the data in the document, as well as the data itself. - EXPAND

3.1.1.1 MongoDB

One of the most popular NoSQL technologies is MongoDB. MongoDB is an open source cross-platform DODB. The premise for using MongoDB is simplicity, speed and scalability (MongoDB White Paper, 2015). Its ever growing popularity, specifically amongst

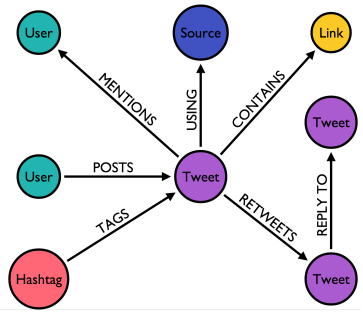
programmers, stems from the unrestrictive and flexible DODB data model which gives you the ability to query on all fields and boasts instinctive mapping of objects in modern programming languages. (MongoDB White Paper, 2015) The database design of MongoDB is based on the JSON file format named BSON.

NOTES A record in MongoDB is stored in collections. A collection is a grouping of MongoDB documents. Within this free flowing environment documents can become as sophisticated and complex as required; information about a document record can be sub categorised by the integration of nested data. *NOTES*

EXPAND - FUNCTIONALITY?

3.1.2 Graph-Orientated Database

A graph-oriented database (GODB), is a form of NoSQL database solution that uses graph theory to store, map and query relationships. A graph is a collection of nodes connected by relationships. "Graphs represent entities as nodes and the ways in which those entities relate to the world as relationships." (Robinson, Webber and Eifrem, 2015) The formation of the graph database structure is extremely useful and eloquent as it permits clear modelling of a vast and often ecliptic array of data types. (Robinson, Webber and Eifrem, 2015) An example of data represented in a graph structure is the Twitter relationship model.



Figure[ADD FIGURE REF] illustrates the nodes involved in a standard tweet and the relationship link between them. The labeled nodes indicate the various operations which are involved in one the tweet. One interpretation of the [FIGURE REF] example is that a user posts a tweet, using the Twitter App which mentions another user and includes a hashtag and

link. - REVISE

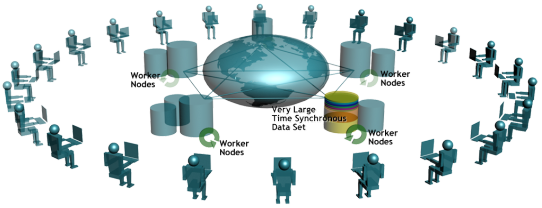
3.1.2.1 Neo4j

Neo4j is an open-source NoSQL GODB which imposes the Property Graph Model throughout its implementation. The team behind the development of Neo4j describe it as an ”An intuitive approach to data problems”(Neo4j web ref). One of the reasons in which Neo4j is favoured predominantly amongst database administrators and developers is its efficiency and high scalability. This is in part due to its compact storage and memory caching for the graphs. ”Neo4j scales up and out, supporting tens of billions of nodes and relationships, and hundreds of thousands of ACID transactions per second.”(Neo4j web ref) - FINISH WRITNG UP NOTES

EXPAND - FUNCTIONALITY?

3.1.3 Distributed Database

A distributed database (DDB) comprises of two or more data files located at different sites and servers on a computer network. (DD ref) The advantage of using a DD is that as the database is distributed, multiple users can access a portion of the database at different locations locally and remotely without obstructing one another's work. It is pivotal for the DD database management system to periodically synchronise the scattered databases to make sure that they all have consistent data. (DD ref) For example if a user updates or deletes data in one location is is essential this change is mirrored on all databases. This ability to remotely access a database from all across the world lends itself to not only multinational companies for example but also startup businesses which recruit the expertise of others from various locations.



3.1.3.1 Apache Cassandra

WRITE UP NOTES

3.1.4 Relational Database

A relational database (RDB) is a collection of data items organised as a set of tables, records and columns from which data can be accessed or reassembled in many different ways. (RDB ref) The connected tables are known as relations and contain one or more columns which comprise of data records called rows. Relations can also be instantiated between the data rows to form functional dependencies called keys which are classified as : (RDB ref 2)

- One to One: One table record relates to another record in another table.
- One to Many: One table record relates to many records in another table.
- Many to One: More than one table record relates to another table record.
- Many to Many: More than one table record relates to more than one record in another table.

3.1.4.1 MySQL

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). -FINISH WRITING UP NOTES EXPAND - FUNCTIONALITY?

3.2 Previous Research - Change Heading

NoSQL has become a bit of a ”...ubiquitous and possibly overused term.” (Lerman, 2011)

3.2.1 TBC

Chapter 4

Dataset Background

The dataset used in the prototype applications is a real world dataset taken from the biological environment. The data is constructed by an ontology derived from the combined research projects undertaken on the e-Mouse Atlas Project (EMAP) by Dr Duncan Davidson and Professor Richard Baldock.

The name EMAP carries a certain amount of ambiguity as it is the name of the project that developed the anatomy, and is also the name of the anatomy itself. Therefore with the motivation of clarity, I will refer to the project that developed the anatomy as e-Mouse Atlas (EMA) and the name of the anatomy as EMAP. Inspired by the findings of Theiler (1989) and Kaufman (1992), EMA uses embryological mouse models to provide a detailed map of mouse development. The EMAP has a developed collection of three dimensional computer models of mouse embryos at the consecutive stages of growth generation with anatomical domains joined by an ontology of anatomical names. The main deliverable of the EMA resource is to provide a comprehensive visualisation of the post-implantation of mouse development and to induct an investigation of the gene expression in the post-implantation mouse embryo. EXPAND

4.1 Discussion

The EMA ontology has several different branch deliverables, each provides an alternative aspect of the evolution of a mouse embryo. The branches which will be utilised for this

research project are the timed stage specific structure, EMAP and the aggregated non stage specific e-mouse Atlas Project Abstract (EMAPA) which are respectively discussed below [REFERENCE]. The EMA dataset's were chosen as the source of data for this research as it is a freely available, rich and substantial data source.

4.1.1 EMAP

The devised EMAP ontology was originally developed to deliver a structured and controlled vocabulary of stage-specific anatomical structures for the developing laboratory mouse. As the EMA research has progressed, the ontology has followed suit, and is continually under development. The current ontology is in scope for a forthcoming release. Based on the timed component stage-specific Theiler development of a mouse embryo, the EMAP dataset combined the stage identifier and the anatomical name based on the researched information for the respective development stage. The timed components are regarded as the main aspect of the ontology and the abstract, non-stage-specific terms came as a secondary protocol.

A hierarchical structure for each of the Theiler stages has been developed and are presented in separate directed acyclic graphs. This data is available in an obo-formatted file [REFERENCE]

The intended outcome of this version of the EMAP was to provide information about the shape, gross anatomy and detailed histological structure of the mouse.

4.1.2 EMAPA

The EMAPA structure is a refined and developed non-stage specific representation of the mouse anatomy ontology. This enhanced version of the ontology is now considered as the primary EMA anatomy ontology and will form the basis of the dataset for this research. As with the EMAP structure the EMAPA is available in

EXPAND

4.2 Data Sources

WRITE UP NOTES

4.2.1 EMA Database

WRITE UP NOTES

4.2.2 OBO

WRITE UP NOTES

4.2.3 OWL

WRITE UP NOTES