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

Contents 1

1 Introduction 2
1.1 Objectives . . . 3
1.2 Motivation . . . 3

2 Literature Review 4
2.1 Big Data . . . 4
2.1.1 5vs Model . . . 4
2.2 Extract Transform Load . . . 5
2.2.1 Process . . . 5
2.2.2 Tool Implementation . . . 6
2.3 NoSQL . . . 6
2.4 Database Classification . . . 7
2.4.1 Distributed Database . . . 7
2.4.2 Document-Oriented Database . . . 8
2.4.3 Graph-Orientated Database . . . 8
2.4.4 Relational Database . . . 9
2.5 Proposed Technologies . . . 9
2.5.1 MongoDB . . . 9
2.5.2 Neo4j . . . 10
2.5.3 Apache Cassandra . . . 10
2.5.4 MySQL . . . 10
2.6 Dataset . . . 10
2.6.1 EMAP . . . 10
2.6.2 EMAP anatomy . . . 11
2.6.3 EMAPA anatomy . . . 12
2.6.4 EMAGE anatomy . . . 13
2.7 Data Sources . . . 13
2.7.1 EMA Database . . . 13
2.7.2 OBO . . . 13
2.7.3 OWL . . . 13

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 2.5.1)

2. Neo4j (Section 2.5.2)
3. Apache Cassandra (Section 2.5.3)
4. MySQL (Section 2.5.4)

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 Motivation

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

Chapter 2

Literature Review

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 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 2.1.1) *FIX*

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 5vs Model

In 2001, Gartner analyst Doug Laney delivered the original 3vs model which define the challenges and opportunities which have arisen by the increase in data volumes. Laney categorises big data in to three dimensions; Volume, Velocity and Variety, with the increase of each encapsulating the challenges currently faced today of big data management. The

characteristics of each property are defined as **Volume** - Refers to the bast amounts of data generated every second. With the creation and storage of large quantities of data, the scale of this data becomes progressively vast. **Velocity** - Refers to the speed at which new data is generated and the speed at which data moves around emphasising the ”timeliness” of the big data. In order to fully profit from the commercial value of big data, data collection and data examination must be conducted promptly . **Variety** - This characteristic alludes to the various types of data we can now use; semi-structured and unstructured. Examples being ”audio, video, webpage and text as well as traditional structured data” (Chen, Mao, Zhang and CMeung REF).

The amount of data being produced has dramatically increased from when Laney first brought us the 3vs model in 2001. Big Data is a term becoming more and more common in business and society. Overcoming obstacles and implementing effective, actionable Big Data strategies is key for successful big data management. IBM has introduced a fourth V, Veracity **Veracity** Data inconsistencies and incompleteness result in data uncertainty and unreliability. This creates a new challenge; keeping data organised. (Big Data ref)

The final and considered by many to be the most important V of big data is Value. ”All the volumes of fast-moving data of different variety and veracity have to be turned into value” (IBM ref) One of the biggest challenges faced by organisations is having the ability to turn data into value. **Value** Refers to ones ability to turn big data into something useful and which adds value to your organisation.

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

2.3 NoSQL

NoSQL is labeled as a next generation database known to most as ”Not only SQL” (NOSQL REF). This definition however insinuates its defiance against the once industry standard SQL.

It was originally developed in 1998 by Carlo Strozzi; a member of the Italian Linux society, with the intention of being a non-relational, widely distributable and highly scalable database. Strozzi named the database management system NoSQL to merely state it does not express queries in the traditional SQL format. Sadalage and Fowler believe the definition we commonly refer NoSQL as comes from a 2009 conference in San Fransisco held by Johan Oskarsson, a software developer. Sadalage and Fowler recall Oskarssons desire to generate publicity surrounding the event and in an attempt to do so devised the twitter hashtag "NoSQL Meetup". The main attendees at the conference debrief session were Cassandra, CouchDB, HBase and MongoDB and so the association stuck. (Sadalage and Fowler ref)

There are are number of key features which encompass the NoSQL framework and encapsulate the essence of its popularity. Many of the NoSQL databases boast their capacity of working in a cluster environment - a cluster being two or more connected computers working collaboratively. Thus delivering a range of options for consistency and distribution (Sadalage and Fowler ref) (Section 2.4.1).

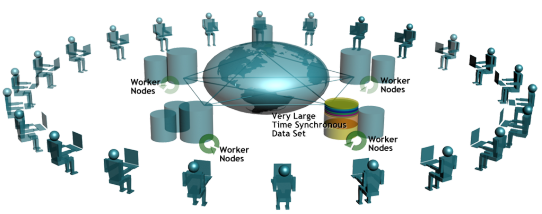
NoSQL solutions are not bound by a definitive schema structure. This permits the ability to freely adapt database records or add custom fields for example without considering structural changes. This is extremely effective when dealing with varying data types and data sets, in comparison to the traditional relational database model which when tackling this issue often resulted in ambiguous field names. (Sadalage and Fowler ref)

2.4 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

2.4.1 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



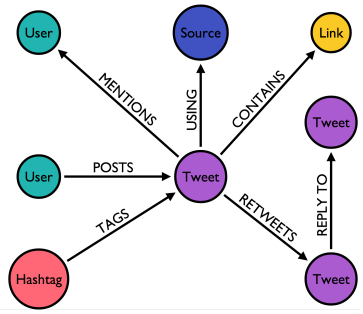
users can access a portion of the database at
different locations locally and remotely without obstructing one another's work. It is pivotal for the DD database management system to periodically synchronise the scattered databases to make sure that they all have consistent data. (DD ref) For example 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.

2.4.2 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

2.4.3 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

2.4.4 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.

2.5 Proposed Technologies

The technologies being used in this project are outlined below. EXPAND

2.5.1 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*

2.5.2 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

2.5.3 Apache Cassandra

WRITE UP NOTES

2.5.4 MySQL

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL).

2.6 Dataset

The dataset used as a resource to populate the database solutions is called EMAP; a freely available anatomical ontology of the developmental stages of mouse embryos. The EMAP dataset was chosen for this project as my supervisor has much experience in the field and would be able to assist me with any queries I had regarding the data. The size and granularity of the EMAP dataset also meets the criteria which will be required to test the database solutions (Section REQUIREMENTS), explore the limitations of each database comparatively and pose insight into the overall performance of each database.

2.6.1 EMAP

The *Edinburgh Mouse Atlas Project* (EMAP) is an ongoing research project to develop a digital atlas of mouse development. The objective of the EMAP is to implement a digital model of mouse embryos for each time stage in development (EMAP Ref). The collated model embryo data is then used to form a database from which further research can be conducted

and experiments can be mapped.

Each time step in the digital model are named *Theiler Stages* inspired by the research conducted by Karl Theiler (REF). A Theiler stage defines the development of a mouse embryo by the form and structure of organisms and their specific structural features. (Theiler Ref) There are 26 individual Theiler Stages which define the growth and evolution of the mouse embryo. The Theiler Stage scheme comprises of both the anatomical developmental stage definition and the estimated length of time since conception. Each Theiler Stage also provides a brief description of the anatomy and any significant changes between the current and previous stage.

Theiler proposed using this scheme as embryos at the same developmental age can have evolved at different rates and therefore exhibit different structural characteristics. (REF Theiler 1989) The EMAP has developed a collection of three dimensional computer models which illustrate and summarise each Theiler Stage. (EMAP website link).

The anatomy generated at each Theiler Stage has an associated ontological representation. Each provides an alternative aspect of the evolution of a mouse embryo which corresponds with a respective Theiler Stage. The abbreviated term EMAP carries a certain amount of ambiguity as it is refers to the name of the project, and one of the stages in the implemented anatomy. (Ref Ken thesis) For the purpose of this project the main anatomy to be utilised is the aggregated non stage specific Edinburgh Mouse Atlas Project Abstract (EMAPA) anatomy. The reasoning behind this and comparative ontological anatomies such as EMAP and Edinburgh Mouse Atlas of Gene Expression (EMAGE) are discussed below.

2.6.2 EMAP anatomy

The EMAP anatomy ontology was originally developed to deliver a controlled vocabulary of stage-specific anatomical structures for the developing laboratory mouse. As the EMAP research has progressed, the ontology has followed suit, and is continually under development.

The original EMAP anatomy ontology consists of a series of relational components organised in a hierarchical tree structure which utilise ”part-of” relations and subdivisions which encompass each Theiler Stage.(EMAP REF) The intention behind the implementation of the original ontology structure was to ”...describe the whole embryo as a tree of anatomical structures successively divided into non-overlapping named parts”. (EMAP REF)

Each of the Theiler Stage components has an appropriately named term label, known as the *short name* which describes each respective component. Each Theiler Stage also has a *full name* which comes in the form of the components entire hierarchical path (EMAP REF). Neither the *full* nor *short* anatomical name of each component are required to be distinct and can appear in several Theiler Stages. Therefore to avoid ambiguity each component can be addressed by a unique identifier. The unique identifier is in the form of the relevant anatomy followed by a number (EMAP:number). For example ”choroid plexus” is the short name of TS20/mouse/body region/head/brain/choroid plexus and has a has unique identifier of EMAP:4218.

The EMAP hierarchical structure facilitates the need for basic ”data annotation and integration” however a combination of the lack of hierarchical views, missing or poorly represented Theiler Stages and label name ambiguity exposed the limitations of the EMAP structure. As a result the need for a hybrid ”abstract” version of the anatomy was identified and subsequently developed; EMAPA. (EMAP REF) Thus the EMAPA anatomy will be the main data source for this work.

2.6.3 EMAPA anatomy

The EMAPA is a refined and algorithmically developed non-stage specific anatomical ontology *abstract* representation of the EMAP anatomy. The EMAPA implementation replaces the EMAP hierarchical tree structure for a *directed acyclic graph* structure; a graph in which it is impossible to start at some vertex v and follow a sequence of edges that eventually loops back to v again.(GRAPH REF) Thus enabling the ability to represent multiple parental relationships and other forms of ”is-a” relations where appropriate. (EMAP REF)

Each anatomical component in the EMAPA anatomy is identified as a single term, coupled with the appropriate start and end Theiler Stage at which the component is considered to be present in the developing embryo. (EMAP REF) With the aim of enhancing user experience, the EMAPA anatomy implements an alternative naming convention from the EMAP anatomy replacing full path names for components to *”print names”*. Using the above example for comparison, ”EMAP:4218” in the EMAP anatomy becomes ”TS20 brain choroid plexus” in EMAPA. This naming convention supplements the requirement of uniqueness and is easily comprehensible.

The research surrounding the EMAPA resource is continually being developed, thus the

growth of the project as a whole is progressively increasing with the richness of data at the heart.(EMAP REF) The EMAPA ontology is now available in a standard ontological format developed by the Open Biological Ontologies (OBO) consortium (Section 2.7.2).This enhanced version of the ontology is now considered as the primary EMAP anatomy ontology and thus will be the main source for the work on this project.

2.6.4 EMAGE anatomy

2.7 Data Sources

WRITE UP NOTES

2.7.1 EMA Database

WRITE UP NOTES

2.7.2 OBO

WRITE UP NOTES

2.7.3 OWL

WRITE UP NOTES