Initial Plan

Building a Twitter Clone with Neo4J

CM3203 - One Semester Individual Project

Credits: 40

School of Computer Science

Cardiff University, 2021

Author: Kiril Engovski

Supervisor: Alia I Abdelmoty

Moderator: Yipeng Qin

Contents

1 Introduction **1**

2 Project Description **1**
2.1 Overall Aims 1
2.2 Motivation and Background 1

3 Project Aims and Objectives **3**

4 Work Plan **4**
4.1 Weekly activities and deliverables 4
4.2 Gantt Chart 8
4.3 Risk Plan 9

References **10**

1 Introduction

This report contains the initial plan for my One Semester Individual Project module at Cardiff University.

2 Project Description

2.1 Overall Aims

The aim of the project is to build a Twitter clone using graph database technology as a store for the social network data and demonstrate two main ideas:

1. How the graph data model can support the representation of social network data and relationships, specifically Twitter;
2. How the graph database technology can handle the storage and manipulation of large dynamic data sets, such as the ones present in social networks.

The demonstration is to be delivered by building a Twitter clone using Neo4J as the graph database technology and then evaluating the application against existing similar approaches including the actual Twitter application. The aim of the evaluation is to consist of variety of analysis on performance, scalability, reliability and suitability of the database technologies.

In addition to that, upon completion of the main objective described above, I will aim for extending the Twitter application by introducing extra feature of systematically identifying states and subjective information on Tweets i.e. messages posted by users of the application. With that, I will aim for demonstration of how the graph data model can support the analysis and reasoning of the data stored. Here, NLP methods will be employed to understand semantic aspects of the data, for example, sentiment analysis and event detection.[1]

2.2 Motivation and Background

With the ongoing Industry revolution (Industry 4.0), more and more emerging technologies are introduced to enhance everyday activities ranging from manufacturing, health, traffic and education to smart houses and gaming [1–3]. The combination of these emerging technologies creates a mesh of relationships between all kinds of devices. If we try to imagine the network, for example within a social media or even just a smart house, we can see a lot of lines (relationships) between users in the case of social media to represent who follows who or between the remotes of the people who live in the house and all the smart devices inside it.

Mentioning those lines, or relationships, or connections, we all get an image of a graph. Graphs have been around at least 300 hundred years back [4], however graph databases only became popular in the last decade, meaning that most of the industry today is not built with graph databases. If the easiest way to imagine the things today is as graphs, doesn't it make sense for it to be the best way for storing the things and the relationships between them?

Along with the emerging technologies, social network sites have become one the main sources of communication and exchanging thoughts in the recent years as social media and user-generated content phenomena grew [5]. For two users to communicate, they need to be connected; Being

[1] This objective is not within the main scope of the project and is to be considered only if the other tasks go as planned or there is a change within the project scope.

friends on Facebook or following each other on Twitter are both types of connections i.e. relationships between users. By saying "Kiril follows Martin", we state that there is a relationship between Kiril and Martin, and the relationship is "follows". Users can communicate by posting messages. If Kiril posts something on Twitter for example, then Kiril is connected to this post as he is the author of it and therefore we can call the relationship between Kiril and the post "posted". Representing those two main concepts of social networks using a graph is very simple and can be seen on Figure 1 [6].

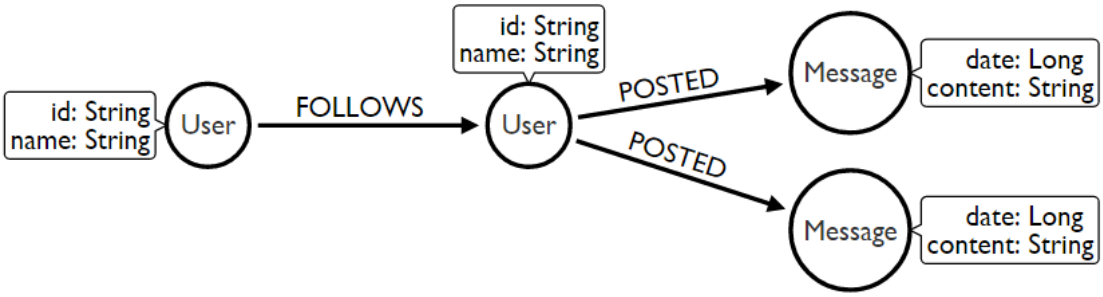


Figure 1: Social network graph representing "follows" and "posted" relationships

From Figure 1 we can see that it is easy to see who follows who, who posted what and interestingly, what the followers of a user posted, by following the arrows. Comparing this to MySQL tables with one to many relationships where the relationships such as "follows" are identified by looking up user ids, seems simpler and easier to interpret.

This idea of seeing everything as relationships between nodes aligns well a structure of social network because each connection, whether it is between a user and post or between two users, can be seen as a relationship of a specific type. The simplicity of representing those ideas as graphs along with the natural one to one translation from drawn graphs with nodes and arrows into graph technologies such as Neo4J graph databases is what motivates me to try and re-build one of the most popular social networks, Twitter, with its data mainly stored in a graph database.

Furthermore, one of the great features on Twitter is the "Who to follow" box, which is a list of recommended accounts on Twitter that the logged in user might like to follow. For new users on Twitter, this could be very important in order for their social network to grow, which is why I identify it as one of its key features. The algorithm behind it is complex and ranks recommendation based on range of factors from address and location to followers of followers. Comparing many factors and looking at chains of followers makes this kind of analysis not only slow but actually impossible using relational databases (where graphs can provide sub-second results) [7]. Graph databases, in particular Neo4J is suitable for such complex recommendations according to Nir and Henry, Soltuion Engineers at Neo4J [7] because of its:

- Explainability: Easy to understand by non-technical users even when it's complex.
- Rapid development: Neo4j is schemaless, and you can refactor your graph to easily consider new data or different access patterns.
- Personalization and Contextualization: Neo4J uses storage layer that connects data from multiple silos. Having quick access to everything you know about a user or product allows you to make more relevant recommendations.

- Performance: Neo4j's native graph engine uses index-free adjacency to eliminate the need for complex joins and enable real-time traversals.
- Graph Algorithms: Allowing searches of similarity, community detection and others who apply to general graphs.

Taking the ability of graph databases a step further, we can transform Twitter into a more powerful engine which looks what happens inside the relationships and analyse posted messages to detect attributes such as Tone and Topics of the messages using NLP. Neo4j is able to connect bodies of text and establish context as to how they relate to each other. This also applies to words, sentences and documents. Such relationships are very useful when drawing inferences and insights quickly from the text at scale, which makes Neo4j suitable for NLP [8].

3 Project Aims and Objectives

- Find out the requirements of building a Twitter clone by studying the existing application and similar approaches.
- Determine the required components of the system and derive a design for the overall architecture.
- Investigate how the graph database model and the database technology can be used to implement the identified components (interface, data storage and data manipulation).
- Design an overall schema of the Neo4J graph database to be used a store as well as specific schemas to represent key components of the system.
- Build the Twitter clone application with Neo4j graph database as a store for the social network data with interactive user interface.
- Find and download suitable sample data set to populate the database and demonstrate the functionality of the application.
- Investigate NLP (Natural Language Processing) approaches within Neo4J databases and determine possible approaches that can support the analysis and reasoning of the data stored in the graph database of the Twitter clone.
- Designa and build NLP component over the developed application and demonstrate that the graph data model can support understanding semantic aspects of the data, for example, sentiment analysis and event detection.
- Test the developed system and NLP components using a representative sample data set and process, filter and visualize the results from the tests.
- Critically evaluate and appraise the developed system against other existing solutions that use alternative database technologies for the social network data.

4 Work Plan

4.1 Weekly activities and deliverables

Week 1 (1 Feb - 7 Feb)

- Studying Twitter as a social media, its main and additional functionalities to determine the main system components that build Twitter;
- Background research and reading on Graph Databases for social media networks, in particular looking for existing applications of Twitter Clone and studying the main concepts of those to gain understanding of how the main system components are implemented;
- Writing initial plan for the project including its Aims and Objectives and a Work Plan for the rest of the semester.
- **Deliverables:**
  - Initial Plan Report
  - Supervisor meetings schedule

Week 2 (8 Feb - 14 Feb)

- Studying further the found existing applications both based on Neo4J graph database and Twitter clones with other databases as their main store for the social network data;
- Analysing and studying the approach and the shared code on GitHub of one existing similar approach using Neo4J by MAX DE MARZI called "Building a Twitter Clone with Neo4J" [9]
- Investigating further the abilities of graph database to identify possible graph database schemas;
- Studying how Natural Language Processing techniques are used with graph databases to determine ways of implementing it to the Twitter clone as a tool for analysis and reasoning of the data stored to understand semantic aspects of the data;
- Create a design of the graph database schema to be used for the application;
- Analysing options and deciding on frameworks to be used for the application's back end and front end compatible with Neo4J and in particular the schema plan;
- Installing required software and libraries for the application while taking note of the same as a guide for running it on different machines.
- **Deliverables:**
  - Graph database schema showing main components of the application and the relationships between them;
  - User interface desing
  - Draft or final graph database model
  - Project repository set up for development

Week 3 (15 Feb - 21 Feb)

- Setting up Neo4J database based on the schema design;
- Creating UML diagrams to represent the relationships of components of the system to be built;
- Start building the Twitter clone application's front and back end with the identified frameworks;
- Study further some of the frameworks where required for better implementation of the components;
- Connecting the graph database with the other ends of the application.
- Progress review with supervisor
- **Deliverables:**
  - UML diagrams showing system components relationships
  - A simple application draft but with successfully established connection between all ends of the application
  - Summarized points and decisions from the progress review meeting

Week 4 (22 Feb - 28 Feb)

- Building the main components of the Twitter clone;
- Creating a simple user interface to allow user interaction with the application;
- Writing Cypher queries for data retrieving to mimic the behaviour of Twitter.
- **Deliverables:**
  - A Twitter Clone prototype with basic social media functionalities allowing user interaction

Week 5 (1 Mar - 7 Mar)

- Executing functionality and performance tests on the built application while detecting errors areas of improvements;
- Developing bug fixes;
- Improving the performance of the application by writing more optimal Cypher queries to do faster reads from the graph database and re-writing pieces of code in a more optimal and structured way;
- Improving the user interface to enhance the user interaction with the system.
- **Deliverables:**
  - A fully functional and optimised Twitter clone with Neo4J database limited on features

Week 6 (8 Mar - 14 Mar)

- Second progress review with supervisor, taking notes and later summarizing the review;
- Upon deciding, begin working on the chosen path: whether to expand on existing Twitter features (depending on how far they are developed by this week) to get a better clone or start building a new feature, not available on the original Twitter application which is systematically identifying states and subjective information on posted messages i.e. Tweets (explained with more detail within the Overall Aims section);
- Research NLP techniques compatible with Neo4J to achieve the desired feature;
- + Begin building NLP data set for training [2].
- **Deliverables:**
  - Summarized points and decisions from the progress review meeting
  - New features prototypes or drafts

Week 7 (14 Mar - 21 Mar)

- + Complete NLP data set if NLP approach is chosen for the new feature;
- Continue building the new features for more complete and efficient Twitter clone;
- Search for suitable data to populate the application with;
- + Studying further NLP techniques with Neo4J and working preparing to implement the same for identifying subjective information about relationships and messages posted;
- **Deliverables:**
  - More complete drafts or prototypes of the new features/NLP data set
  - Merging plan

Week 8 (22 Mar - 28 Mar)

- Complete the new features to be added to the Twitter clone
- + Train the application with the NLP data set prepared earlier;
- Run unit tests;
- Merge the successful newly developed components to the functional Twitter clone;
- Deal with bugs that occur during merge;
- Prepare test cases for the application overall;
- Begin testing the application;
- **Deliverables:**

[2] The + is before tasks that are related to the NLP approach and could potentially be withdrawn from the plan later

  - Complete Twitter clone application with Neo4J as a store for the social data network + trained with NLP to detect tone and topics of posted messages.
  - Test case.

Week 9 (19 Apr - 25 Apr)

- Finish the tests prepared;
- Populate the application with the downloaded sample data set;
- Prepare and run different evaluation scripts to test the suitability and behaviour for the built application and other existing similar approaches, focusing on the database technology;
- Summarize, visualize and comment on the outcome to evaluate the built application and database technology against the existing approaches.
- **Deliverables:**
  - Evaluation scripts
  - Evaluation summary with results from the tests including any figures used

Week 10 (26 Apr - 2 May)

- Create structure for the final report;
- Begin report writing by looking through project diary.
- **Deliverables:**
  - Around 50% of the final report is written.

Week 11 (3 May - 9 May)

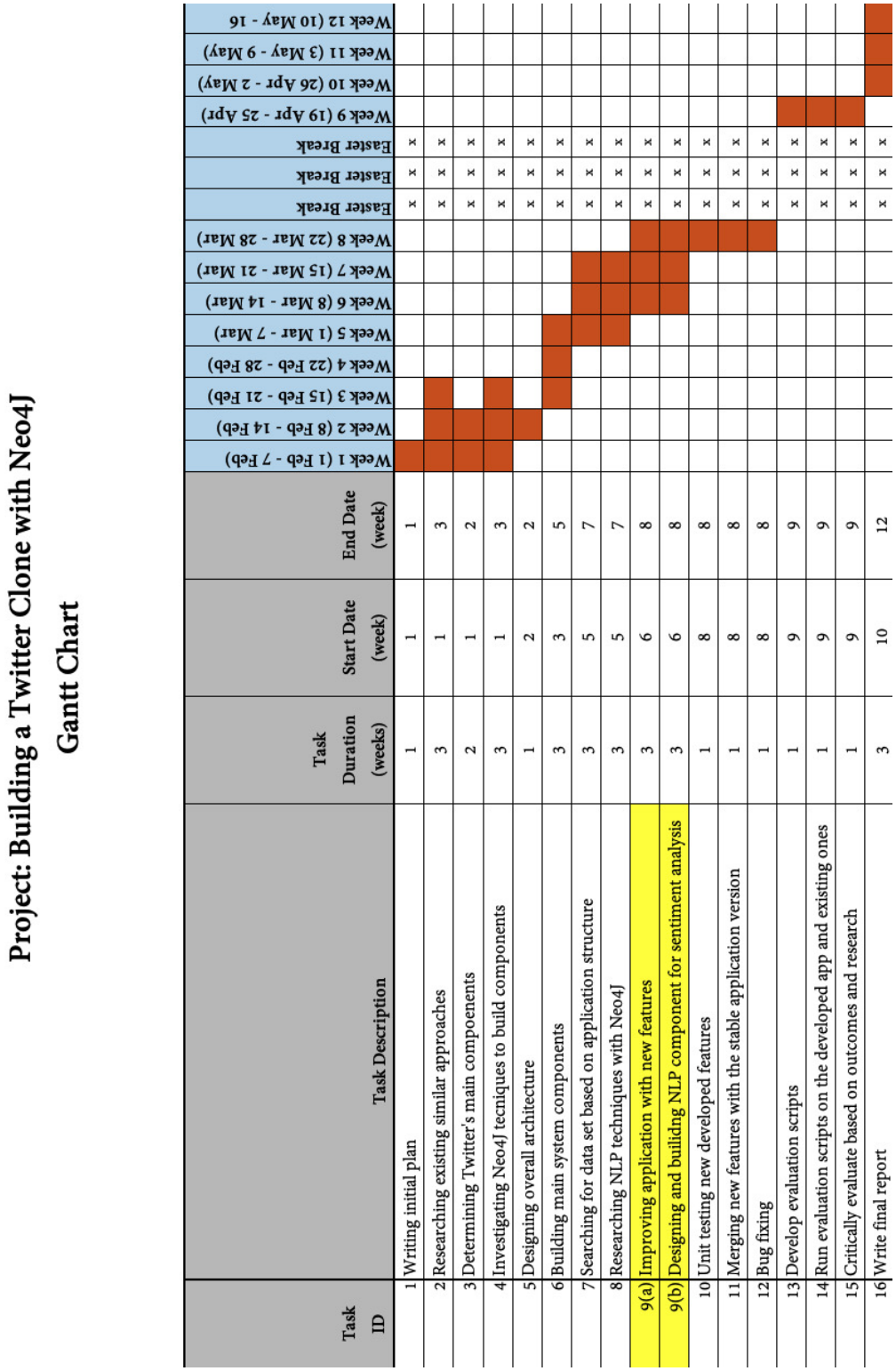
- Continue writing the final report.
- **Deliverables:**
  - Near complete version of the final report.

Week 12 (10 May - 17 May)

- Finalize final report;
- Prepare submission files and submit;
- **Deliverables:**
  - Complete version of the final report
  - Final report submission with supporting files (14 May 2021)

Supervisor catch-up meetings are scheduled to happen every week on Friday. Two in-depth review meetings are scheduled for Week 3 and Week 6. Others might be added and the current could be rescheduled later.

4.2 Gantt Chart



Project: Building a Twitter Clone with Neo4J

Gantt Chart

Task ID	Task Description	Task Duration (weeks)	Start Date (week)	End Date (week)	Week 1 (1 Feb - 7 Feb)	Week 2 (8 Feb - 14 Feb)	Week 3 (15 Feb - 21 Feb)	Week 4 (22 Feb - 28 Feb)	Week 5 (1 Mar - 7 Mar)	Week 6 (8 Mar - 14 Mar)	Week 7 (15 Mar - 21 Mar)	Week 8 (22 Mar - 28 Mar)	Easter Break	Easter Break	Easter Break	Week 9 (19 Apr - 25 Apr)	Week 10 (26 Apr - 2 May)	Week 11 (3 May - 9 May)	Week 12 (10 May - 16
1	Writing initial plan	1	1	1									x	x	x				
2	Researching existing similar approaches	3	1	3									x	x	x				
3	Determining Twitter's main compoenents	2	1	2									x	x	x				
4	Investigating Neo4J tecniques to build components	3	1	3									x	x	x				
5	Designing overall architecture	1	2	2									x	x	x				
6	Building main system components	3	3	5									x	x	x				
7	Searching for data set based on application structure	3	5	7									x	x	x				
8	Researching NLP techniques with Neo4J	3	5	7									x	x	x				
9(a)	Improving application with new features	3	6	8									x	x	x				
9(b)	Designing and builidng NLP component for sentiment analysis	3	6	8									x	x	x				
10	Unit testing new developed features	1	8	8									x	x	x				
11	Merging new features with the stable application version	1	8	8									x	x	x				
12	Bug fixing	1	8	8									x	x	x				
13	Develop evaluation scripts	1	9	9									x	x	x				
14	Run evaluation scripts on the developed app and existing ones	1	9	9									x	x	x				
15	Critically evaluate based on outcomes and research	1	9	9									x	x	x				
16	Write final report	3	10	12									x	x	x				

Figure 2: Gantt chart

4.3 Risk Plan

The following section contains analysis of identified risks visualised on risk map.

Risk	Probability of Occurrence	Impact on Project	Mitigation Response
Inadequate time to complete the desired build version with features	Medium	Medium	Follow my plan, discussing any issues with the supervisor, refining some tasks and working over easter
Inadequate time to complete Project Report	Low	High	Follow work plan and begin writing on time and update the project diary continuously
Loss of Data	Low	High	Create back-ups of the data and the project diary regularly
Illness	Medium	Medium	Following coronavirus guides, informing the supervisor on time if I am unable to work for certain time period

Figure 3: Project Risk Map

References

[1] C. Themelis and J.-A. Sime, "From video-conferencing to holoportation and haptics: How emerging technologies can enhance presence in online education?" in *Emerging technologies and pedagogies in the curriculum*. Springer, 2020, pp. 261–276.

[2] C.-W. Lung, F.-L. Wu, F. Liao, F. Pu, Y. Fan, and Y.-K. Jan, "Emerging technologies for the prevention and management of diabetic foot ulcers," *Journal of tissue viability*, vol. 29, no. 2, pp. 61–68, 2020.

[3] W. A. Rogers and C. B. Mayhorn, "Technology in everyday," *Gerotechnology: Research and practice in technology and aging*, p. 1, 2004.

[4] N. Biggs, E. K. Lloyd, and R. J. Wilson, *Graph Theory, 1736-1936*. Oxford University Press, 1986.

[5] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of computer-mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[6] (2016, October). [Online]. Available: https://maxdemarzi.com/2016/10/28/news-feeds/

[7] (2019, September). [Online]. Available: https://neo4j.com/blog/intelligent-recommendation-framework/

[8] (2019, August). [Online]. Available: https://neo4j.com/blog/accelerating-towards-natural-language-search-graphs/

[9] (2017, March). [Online]. Available: https://maxdemarzi.com/2017/03/30/building-a-twitter-clone-with-neo4j-part-one/