A Neo4j database for analysis of inter-domain routing in the Internet

Alan Tang, Jesus College

Originator: Timothy Griffin

21 October 2016

Project Supervisor: Timothy Griffin

Director of Studies: Cecilia Mascolo

Project Overseers: Jon Crowcroft & Dr Thomas Sauerwald

Introduction and Description

The Border Gateway Protocol (BGP) is a network protocol designed to exchange routing and reachability information between autonomous systems. Large organizations and ISPs can control the routing and configuration of their own networks, but as the Internet is composed of many autonomous systems connected together, there needs to be a single protocol between autonomous systems to guide routing policies across the globe. BGP fulfills this role.

Given the importance of this protocol, creating a means to easily analyze and query the data would certainly be beneficial. The goal of this project is to design and implement a way of storing BGP data into a graph database, in particular Neo4j. It will develop a model that will capture the relationship between autonomous systems and the changes that occur over time. The project will also use Cypher, Neo4j's query language, to develop a library of queries to extract data from the database.

A graph database provides a natural starting point for modeling this data. Network data inherently has a graph structure. In comparison to a traditional relational database, graph databases provide a more efficient and natural means of following relationships between objects. Neo4j is a popular example of a graph database. It comes with drivers for some common programming languages, which allow Cypher commands to be called through programs.

Large amounts of BGP data is available from RIPE NCC website [1]. This will be used as input for the project. The raw data contains BGP routing and update information which can be parsed with existing tools such as bgpdump. The data captures information about wars, natural disasters, or other events which have affected the routing of the internet. The project aims to provide a new means of looking and analyzing these.

Declaration of Resources

When convenient, work will be done on my laptop (MacBook Pro, Intel i7). I accept full responsibility for this machine and I have made contingency plans to protect myself against hardware and/or software failure. Should this computer fail or should I require more disc space, the MCS computer will be used. The data will be backed up to a hard disk on a weekly basis and may be put on the cloud as a secondary backup.

Starting Point

Programs such as bgpdump and bgpstream exist to convert binary BGP data into a human readable form. These will likely be used in lieu of writing my own utility. The Neo4j database software which will be central to the project is freely available from its website [2].

BGP data is also available online from the RIPE NCC website. This data is freely available for research and has been used for a number of articles [3]. Other organizations such as CAIDA [4] and Dyn Research [5] have looked at this type of data before, and their analyses and results serve as an inspiration for the project.

Substance and Structure of the Project

The core of the project consists of creating a database for modeling BGP data. This will require careful design in order to ensure efficiency of looking up data. The design is likely the trickiest part, in particular, designing a way of modeling the time component of updates. To ensure that the system has adequate performance, the project will require understanding of the Neo4j database system as well as performance of the queries written in the Cypher language. This will require research into the operation of BGP and Neo4j. It will also require learning the Cypher querying language for retrieving data from Neo4j.

The project will consist of the following sections:

1. The design and implementation of the core database system that allows for efficient viewing of the available BGP data.

2. The creation of scripts to transfer existing BGP data into the database. Depending on the design of the database, this may include transforming or performing calculations to the data.

3. The creation of Cypher queries to view the data in the database. It should allow for basic information on snapshots of the data such as looking up paths at a particular moment in time. If possible, it should allow for route changes over time.

4. Evaluation of the speed, simplicity, and functionality of querying data.

5. A look at case studies which illustrate the usefulness of this system.

Initially the database will only model data from a single collector. As an extension, it can be made to incorporate data from several collectors at the same time. Another extension would be to optimize the performance to be as fast as possible.

Success Criteria

The following goals should be achieved: A model for representing BGP data is designed and implemented with Neo4j. It should incorporate both routing and temporal data.

- There is a means of reading all existing BGP data and storing it into the database.
- A library of at least 5 common or useful operations is written to query the data
- The performance of the system is efficient enough to be used with large datasets.

Timeline and Milestones

17 Oct - 31 Oct

Read about BGP and Neo4j. Learn to use the Cypher query language and write simple queries to improve understanding.

1 Nov - 14 Nov

Begin design of the database. Careful consideration should be made about the design as that will affect the efficiency of the project. Create small prototypes to test out certain ideas of the design.

15 Nov - 2 Dec

Finalize the design of the database and begin implementation. A functional means of converting BGP data into input for the database will need to be implemented in some form. It may not completely handle all the data, but it must run.

Christmas Vacation

Catch up on work from Michaelmas if necessary. If not necessary, continue adding features to the project such as incorporating new data and writing Cypher queries for snapshots.

16 Jan - 30 Jan

Finish the implementation of the system. Continue writing Cypher queries for the data. It should be fully functioning and presentable. Write the progress report and prepare a presentation.

31 Jan - 13 Feb

Begin evaluation of the project. Fix any remaining issues. All the code for the project (other than tests) should be completed. Begin work on the dissertation by setting up the outline and making a draft of the introduction chapter.

14 Feb - 27 Feb

Continue work on evaluation of program. Make a rough draft of the preparation and implementation chapters of the dissertation.

28 Feb - 13 Mar

Finish evaluation if not done, and write the evaluation and conclusion chapters of the dissertation. A rough draft of the dissertation should be completed.

Easter Vacation

Catch up on work if the unexpected difficulties come up or I fall behind on the schedule. If all the goals have been reached, work on possible extensions and edit the dissertation.

25 Apr - 19 May

Continue to edit the dissertation if needed. Work on extensions if time allows. Ideally submit early.

References

[1] RIPE Network Coordination Centre. RIS Raw Data. `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data`. 20 October 2016.

[2] Neo4j: The World's Leading Graph Database. https://neo4j.com/. 20 October 2016.

[3] RIPE Network Coordination Centre. Articles with RIS Analysis. `https://www.ripe.net/analyse/archived-projects/ris-tools-web-interfaces/articles-with-ris-analysis`. 20 October 2016.

[4] CAIDA: Center for Applied Internet Data Analysis. `http://www.caida.org/home/`. 20 October 2016.

[5] Dyn Research. `http://research.dyn.com/`. 20 October 2016.