Analytics of Publication Data with Graphulo - Entwurfsdokument

Alexander Kern Alexander Möller

24. Mai 2017

1 Aufgabenstellung

Das Ziel der Praktikumsaufgabe ist das Kennenlernen der Bibliothek Graphulo. Dabei handelt es sich um eine Java-Implementierung des GraphBLAS-Standards. Dieser umfasst grundlegende Matrixoperationen, mithilfe derer Entwickler Graphalgorithmen umsetzen können. [Graphulo]
Graphulo verwendet die verteilte Key-Value-Datenbank Accumulo als Datenspeicher. Die Datenbank basiert auf dem Design von Googles BigTable und nutzt Technologien wie das Hadoop File System und Zookeeper. [Accumulo]
Beispielhaft soll im Rahmen des Praktikums der DBLP-Datensatz, zur Verfügung gestellt von der Universität Trier, untersucht werden. DBLP steht dabei für „Digital Bibliography & Library Project“und umfasst Informationen über eine Vielzahl von Veröffentlichungen aus dem Bereich der Informatik. [DBLP]

2 Umsetzung

Die Umsetzung umfasst zwei verschiedene Schwerpunkte.
Einerseits soll eine Top-Level-Evaluation erfolgen. Diese soll anhand der Analyse von Merkmalen, der Pflege des Produktes und Qualität der Dokumentation sowie einer Überprüfung der Verbreitung untersucht werden, inwieweit sich Graphulo für den praktischen Einsatz eignet und ob schon ein Einsatz erfolgt.
Zur Erfüllung dessen sollen einerseits die Erfahrungen aus dem Praktikum selbst, sowie die Recherche in gängigen Foren, hierfür wurden Stackoverflow und Quora ausgewählt, auf GitHub selbst und ein Vergleich mit anderen Graphdatenbanken erfolgen.

Außerdem ist es ein Ziel, Analysen auf dem DBLP-Datensatz durchzuführen. Der erste Schritt dafür ist, eine lokale Accumulo-Instanz zu installieren und lauffähig zu machen.
Anschließend erweitern wir den [DBLP-Parser] um eine Schnittstelle, die geparste Daten in ein von Graphulo nutzbares Format konvertiert und in einer Accumulo-Instanz speichert.
Graphulo bietet neben Basismatrixoperationen bereits einige beispielhafte Graphalgorithmen an. Zuerst möchten wir versuchen, mithilfe dieser den Datensatz zu analysieren. Weitergehend möchten wir selbst einen Graphalgorithmus auf Basis der GraphBLAS-Operationen implementieren. Hierbei entschieden wir uns für einen Algorithmus zum Finden von Connected Components.

3 Aktueller Stand

Aufbauend auf der Installation von Hadoop, ZooKeeper und Accumulo auf einer virtuellen Maschine entstanden bereits Skripte, die den Prozess weitestgehend automatisieren. Diese sind auch für spätere Projekte die Accumulo benötigen einsetzbar und bieten die Möglichkeit beispielsweise ein Docker-Image mit einer lokalen Accumulo-Installation zu erstellen.
Die Schnittstelle für Java ist bereits umgesetzt und erstellt eine Adjazenzmatrix mit den Kollaborationen verschiedener Autoren.
Die Top-Level-Evaluation ist begonnen, ihr aktueller Stand findet sich in der Präsentation für das erste Testat.

4 Geplantes Ergebnis

Bis zum zweiten Testat soll die Top-Level-Evaluation weiter ausgebaut werden. Das technische Ergebnis stellt eine einfache Java-Schnittstelle für Accumulo dar, die Daten in einem für Graphulo nutzbaren Format akzeptiert. Diese soll der DBLP-Parser einsetzen. Außerdem entsteht eine Implementierung eines Connected-Components-Algorithmus, der in Graphulo einsetzbar sein soll.
Neben der eigenen Implementierung eines Graphen möchten wir auch die von Graphulo mitgelieferten Algorithmen einsetzen und prüfen, welche Informationen mit diesen aus dem DBLP-Datensatz gewonnen werden können.

Literatur

[Graphulo] `http://graphulo.mit.edu`, 19.05.2017

[Accumulo] `https://accumulo.apache.org`, 19.05.2017

[DBLP] `http://dblp.uni-trier.de`, 19.05.2017

[DBLP-Parser] `https://github.com/ScaDS/dblp-parser`, 19.05.2017