Dr. Konrad Völkel
Heinrich-Heine-Universität Düsseldorf
Wintersemester 2024/25

Data Science 2
Blatt 8, Abgabe am 11.12.2024 um 12:00

Bitte geben Sie eine PDF-Datei ab und ein Notebook oder einzelne Python-Skripten. Aus gegebenem Anlass: bitte stellen Sie vor der Abgabe sicher, dass Ihr Code auch wirklich funktioniert, indem Sie diesen in einen separaten Ordner verschieben und dort erneut ausführen bzw. das Notebook mit 'Restart Kernel and run all cells' ausführen (und danach speichern), bevor Sie den PDF-Export starten.

Aufgabe 1. *Sie haben bereits auf dem 2002er-Google-WebGraph `web-Google.txt.gz` eine explorative Analyse durchgeführt und mit NetworkX den 'pagerank' ausrechnen lassen (für einen Teilgraph).*

Implementieren Sie PageRank (zunächst ohne jede Form von Parallelisierung, wahlweise mit Numpy). Vergleichen Sie auf Teilgraphen, so wie dem vom vorigen Übungsblatt, Ihr Ergebnis mit dem PageRank, den NetworkX berechnet. Verfahren Sie analog mit dem größeren `wiki-topcats.txt.gz`-Datensatz.

Aufgabe 2. *Beschleunigen Sie Ihre PageRank-Implementierung durch Parallelisierung. Dazu können Sie entweder den vertrauten Ansatz mit MapReduce über MRJob verwenden, oder aber mit dem Paket `ipyparallel` mehrere Kerne auf dem selben Rechner verwenden. Ein Unterschied zwischen beiden Ansätzen ist, dass bei IPyParallel mehrere Kerne einen Speicher nutzen können, während im MapReduce-Paradigma ein verteiltes Dateisystem zugrundeliegt, in dem jeder Worker potentiell eigenen Speicher nutzt, der für andere nicht zugänglich ist. Man kann aber auch mit IPyParallel über Netzwerke Cluster bilden (und das ist die wesentliche Anwendung).*

Eine Einstiegsdokumentation in IPyParallel ist hier zu finden:
ipyparallel.readthedocs.io

Überlegen Sie, ob eine dünn besetzte Datenstruktur für den Graph geeignet sein könnte.

Vergleichen Sie Ihre parallelisierte Implementierung mit der 'langsamen' sowie der von NetworkX auf hinreichend kleinen Graphen (so wie in Aufgabe 1 und dem vorigen Übungsblatt), um herauszufinden, ob Sie den Algorithmus korrekt implementiert haben.

Aufgabe 3. *Reflektieren Sie - was war schwierig bei den Aufgaben 1 und 2? Was lief anders, als gedacht/geplant? Können Sie eine (oder mehrere) Erkenntnisse formulieren, die Sie aus dieser Übung mitgenommen haben?*

*Tipp: Wenn Ihnen die Implementierung mit Parallelisierung nicht gelingt, können Sie sich noch mit Kommiliton*innen austauschen und die fertige Aufgabe 2 eine Woche später erneut einreichen. Versuchen Sie trotzdem erstmal, so weit zu kommen, wie möglich.*

Bonusaufgabe: Finden Sie heraus, wie die Laufzeit ihrer parallelisierten Implementierung mit der Eingabegröße (Anzahl Knoten im Graph) skaliert, indem Sie sich darüber theoretisch Gedanken machen und diese empirisch mit wenigstens 4 Messungen prüfen. Erstellen Sie daraus eine Prognose, wie lange mit Ihren eigenen Rechenresourcen die Berechnung des PageRank auf der Wikipedia dauern würde. Dabei können Sie selbst ein Kriterium festlegen, wann Ihnen die Approximation des PageRank gut genug ist (sodass man abbrechen kann mit der Berechnung). Wenn Sie verrückt genug sind, prüfen Sie Ihre Hypothese durch Berechnung des PageRank auf dem Wikipedia-Link-Graphen (Achtung: der ist groß). Welche 10 Seiten sind die 'wichtigsten'?