Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel, H. Hatzel, J. Gresens

Übungsblatt 9 zur Vorlesung
Big Data Analytics
im WiSe 2015/2016
Abgabe: 18.12.2015, 390 P

Sollten Probleme auftauchen, wenden Sie sich bitte an die Mailingliste der Veranstaltung:

`bd1516@wr.informatik.uni-hamburg.de`

1 Datenimport nach HBase (150 P)

Sie sollen im folgenden die absoluten Worthäufigkeiten aus der Datei `/home/bigdata/wiki-clean--frequency-tiny.csv` (enthält die ersten 100 Zeilen der großen `wiki-clean-frequency.csv`) platzsparend in HBase ablegen.
Erstellen Sie hierfür in HBase zuerst in der `hbase shell` eine Tabelle um die Daten nach dem folgenden Schema zu speichern:

- Tabellename: bd1516_nachname1_nachname2_nachname3
- Zeile: Wort
- Spalten: article:articleA, articleB, articleC, ...
- Zelle: Worthäufigkeit von diesem Wort in verschieden Artikeln

Importieren Sie dann die Daten indem Sie sie mit Python einlesen und mithilfe des Pakets Starbase in HBase speichern.

1.1 Hinweise

Eine umfangreichere Einführung zu Starbase finden Sie unter `http://blog.cloudera.com/blog/2013/10/hello-starbase-a-python-wrapper-for-the-hbase-rest-api/`.

1.1.1 Codegerüst für Python

```python
#/usr/bin/env python3

from starbase import Connection

# connect to HBase
c = Connection('127.0.0.1','11111')

# choose table to work with
t = c.table('users')

# get all columns of row 'cdickens' and print it
row = t.fetch('cdickens')
print(row)

# put value 'bar' in cell 'cdickens','personalDet:foo'
t.insert('cdickens', {'personalDet': {'foo':'bar'}})
```

Abgabe:

`1-hbase-import.py` Ihr Skript zum Import der Worthäufigkeiten.

2 HBase als Datenquelle (120 P)

Im Folgenden sollen die in HBase liegenden Daten visualisiert werden. Wie auf Blatt 4 sollen Sie nun verschiedene Wordclouds erstellen:

- Visualisieren Sie für vier beliebig gewählte Worte deren Häufigkeit in den importierten Artikeln (d.h. die Wordcloud zeigt Artikeltitel in denen die Wörter vorkommen an).
- Visualisieren Sie für vier beliebig gewählte Artikel die Häufigkeit der im Artikel enthaltenen Wörter.

Messen und vergleichen Sie die Laufzeiten beider Vorgänge, was fällt ihnen auf?

2.1 Hinweise

Nutzen Sie erneut Starbase zum Zugriff auf die Daten in HBase. Unter Umständen kann es sinnvoll sein, die Daten in einem anderen Format einzulesen und zu verwenden.

Abgabe:

`2-gen-word-wordclouds.py`	Ihr Python-Skript zum Generieren der (Artikel-)Wordclouds für Wörter.
`2-gen-article-wordclouds.py`	Ihr Python-Skript zum Generieren der Wordclouds für Artikel.
`2-wordclouds.pdf`	Ihre Wordclouds.

3 HDFS REST API (120 P)

In dieser Aufgabe sollen Sie exemplarisch an WebHDFS die Nutzung einer REST API üben. Sie finden die API-Spezifikationen von WebHDFS im Netz unter `http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html`.
Verwenden sie zunächst `curl` um die folgenden Operationen durchzuführen:

- Ordnerinhalt ausgeben
- Datei erstellen
- Dateiinhalt ausgeben
- Datei löschen
- Ordner umbenennen
- Ordner löschen

Dokumentieren Sie Ihre `curl`-Aufrufe sowie die Rückgaben.
Erstellen Sie ein Python-Programm, das Ihnen den Zugriff auf ihre Dateien im HDFS im Stil von einem FTP-Clienten ermöglicht, d.h. ein interaktive Operation mit entfernten Dateien. Die UI Ihres Programms sollte sich nach Möglichkeit am Original orientieren (siehe `https://www.cs.colostate.edu/helpdocs/ftp.html`). Implementieren Sie mindestens die folgenden `hadoop fs` Kommandos: ls, put, get, mkdir, rmdir und rm [-r]. Zusätzlich implementieren Sie ein lokales `ls` und `chdir`.

3.1 Hinweise

Nutzen Sie falls nötig `http://httpbin.org/` um den Aufbau und Inhalt Ihrer HTTP-Requests zu sehen. Wir haben die Authentifikation in HDFS abgestellt, d.h. Sie müssen für alle Schreib-Operationen den Query Parameter `user.name` entsprechend dem Benutzernamen setzen.

3.1.1 Codegerüst für Python

```python
#!/usr/bin/env python3
import requests

def read_test():
    # reading
    ## via url
    url = 'http://10.0.0.61:50070/webhdfs/v1/user/gresens/data?op=OPEN'
    r = requests.get(url)
    print(r)
    print(r.text)

def run_cmd(cmd):
    """Translate command into REST API call and process the response."""
    pass

def repl():
    while True:
        try:
            cmd = input('>>> ')
        except EOFError:
            print('exit')
            break
        if cmd in ['exit', 'bye']:
            break
        elif cmd == '':
            continue
        else:
            run_cmd(cmd)

if __name__ == '__main__':
    read_test()
    repl()
```

Abgabe:

`3-curl.txt`	Ihre `curl`-Aufrufe und Rückgaben.
`3-hdfs-ftp.py`	Ihr in Python geschriebenes HDFS FTP Programm.

4 Bonusaufgabe Recherche: RESTful Services (30 P)

Recherchieren Sie ein wenig nach interessanten BigData Services, welche sich über REST API nutzen lassen. Finden Sie bereits bestehende Tools, die eine REST API nutzen. Haben sie Ideen für weitere Anwendungen?

Abgabe:

`4-Recherche.pdf`	Ihre Recherche-Ergebnisse.