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

Übungsblatt 11 zur Vorlesung
Big Data Analytics
im WiSe 2015/2016
Abgabe: 15.01.2016, 420 P

Bitte dokumentieren Sie die benötigte Bearbeitungszeit für die einzelnen Aufgaben in `bearbeitungszeit.txt`. Bitte denken Sie auch daran uns Feedback zur Veranstaltung zu geben:

`http://goo.gl/forms/B01IIDHvW2`

Bei der Abgabe von Source Code kommentieren Sie diesen bitte.

1 Storm Grundlagen (180 P)

In dieser Aufgabe sollen Sie eine Storm-Topologie entwickeln, die für Nachrichten aus einem sozialen Netzwerks einen Score (im übertragenen Sinne bspw. ein Bedrohungsgrad) berechnet.
Hierfür sind zwei verschiedene Spouts vorgegeben, die die Nachrichten bzw. Metadaten der jeweiligen Nachrichten ausgeben. Der (durchaus etwas naive) Ansatz einen Score zu berechnen ist zuerst für jede Nachricht die Worthäufigkeiten zu ermitteln und dann diese mit dem Score des Worts zu multiplizieren und für die Nachricht aufzusummieren.
Loggen sie alle Nachrichten, deren Score größer als zwei ist im HDFS.
Der Score der einzelnen Wörter finden sie in der Datei `hdfs:///user/bigdata/wordscores.txt`. Lesen Sie diese in Ihrer Storm-Topologie direkt vom HDFS ein.
Zeichnen Sie einen Graphen zur Übersicht über Elemente in Ihrer Topologie und die einzelnen Eingabe bzw. Ausgabe-Tupel der einzelnen Bolts.

1.1 Hinweise

Im Archiv zur Aufgabe finden Sie das Gerüst für einige Klassen. Nutzen Sie die gegebene Konfiguration für Maven, der Befehl `mvn package` kompiliert ihre Klassen und packt sie zusammen mit den Abhängigkeiten in ein großes Java Archiv.
Sie finden weitere Beispiele unter `https://github.com/apache/storm/tree/master/examples/storm-starter/src/jvm/storm/starter`. Wir haben zusätzlich für Sie die PrinterBolt zur Verfügung gestellt, welche für das Debuging nützlich sein könnte.

Abgabe:

`1-topology.pdf`	Ihre Topologie mit Beschreibung der Ausgabe-Tupel.
`1-SocNet.tar.gz`	Ihr Archiv, das alle fürs Kompilieren erforderlichen Klassen und Dateien ihrer Topologie enthält.

2 Storm erweitert + DRPC (120 P)

Erweitern Sie ihre Storm Topologie aus Aufgabe 1 um einen Bolt, der für jeden Benutzernamen den Score seiner Nachrichten aufsummiert. Zusätzlich soll der Bolt via DRPC-Request dazu aufgefordert werden können eine Liste zurückzugeben, die für jeden Nutzen den aktuellen Gesamtscore enthält.

2.1 Hinweise

Sie finden ebenfalls DRPC-spezifische Beispiele unter `https://github.com/apache/storm/tree/master/examples/storm-starter/src/jvm/storm/starter`.

Abgabe:

`2-topology-extended.pdf`	Ihre erweiterte Topologie mit Beschreibung der Ausgabe-Tupel.
`2-SocNetExtended.tar.gz`	Ihr Archiv, das alle fürs Kompilieren erforderlichen Klassen und Dateien ihrer erweiterten Topologie enthält.

3 Einfache Topology mit Trident (120 P)

In dieser Aufgabe sollen Sie praktische Erfahrung bei der Nutzung der Trident-API sammeln. Nutzen Sie Trident um zwei DRPC-streams zu implementieren, welche für den als Argument übergebenen Satz 1) die Wortanzahl berechnet und 2) den Bedrohungs-Score berechnet (mit Hilfe der HDFS-Score Datei wie in 1. Aufgabe).
Testen Sie Ihre Topologien anhand eines sinnvollen Beispiels.

3.1 Hinweise

Sie finden verschiedene Beispiele unter `https://github.com/apache/storm/tree/master/examples/storm-starter/src/jvm/storm/starter/trident`.
Nutzen Sie die gegebene Konfiguration für Maven, `$ mvn package` kompiliert ihre Klassen und packt sie zusammen mit den Abhängigkeiten in ein großes jar-Archiv.
Nutzen Sie der Einfachheit halber die `LocalDRPC`-Klasse für die DRPC-Aufrufe.

Abgabe:

`3-topology-trident.pdf`	Ihre Trident-Topologien mit Beispieldaten.
`3-trident.tar.gz`	Ihr Archiv, das alle fürs Kompillieren erforderlichen Klassen und Dateien ihrer Topologie enthält.