HW #1 Spark SQL, Spark Streaming
데드라인: 4월 27일 오후 6시 (팀별 과제)
제출 이메일: ds2-ta@spl.snu.ac.kr

HW #1-1을 스트림 처리 수업 (4월 19일) 전에 다 하는 것을 추천

HW #1-1 Spark SQL (총 10점)

[과제 목표]
분산환경에서 SparkSQL을 실행하고, 그 실행과정을 이해

[숙제 환경 셋팅]
1. 실습시간에 만들고 'Stop'시킨 EC2 3대 (Master, Worker1, Worker2) 'Start'
2. 제공되는 'HW1_1-SparkSQL' notebook을 MASTER_PUBLIC_IP:8888 jupyter에 upload
3. HDFS 셋업 (notebook에서 커맨드 제공함)
4. Spark cluster 셋업 (notebook에서 커맨드 제공함)
5. 만약 이미 업로드가 안되었으면, Wikipedia dataset file 3개를 HDFS에 업로드 (notebook에서 커맨드 제공함)
6. WikipediaTable은 4개의 column이 있음 = (project, title, count, size)

[과제 내용]
다음의 결과를 수행하는 2개의 SparkSQL Query들을 작성하고 실행:

(Query #1) WikipediaTable에서 project='en' 에 해당하지만, title='Psicobloc'는 아닌, count들의 평균값
⇒ 결과 table column(1개): average_count

(Query #2) WikipediaTable과 다음의 테이블을 JOIN하여, 각 owner 당 count의 최대값
⇒ 결과 table column(2개): owner, max_count

OwnerTable

title	owner
Psicobloc	Bob
Video_game_content_rating_system	Bob
Radiohead	James
Carlos_Keller_Rueff	James

* 참고 - SparkSQL dataframe in-memory 생성 방법:
http://datasciencedeconstructed.com/python/2016/08/08/PySpark-create-dataframe-from-scratch.html

[제출 방식]
Spark Web UI 스크린샷을 담은 PDF를 이메일로 제출:
(1) SQL query (1점*Query2개)
(2) jupyter notebook에 보여지는 Query 결과값 (1점*Query2개)
(3) Spark Web UI 'SQL' tab 에서 보여지는 Query plan 그래프의 스크린샷 (1점*Query2개)
(4) Spark Web UI 'Jobs' tab 에서 보여지는 DAG Visualization 그래프의 스크린샷 (1점*Query2개)
(5) Spark Web UI 'Stages' tab에서 보여지는, 해당 Query에 대응하는 Job에 속하는 모든 Stage들의 'Summary Metrics for N Completed Tasks' 테이블의 스크린샷 (1점*Query2개)

HW #1-2 Spark Streaming (총 10점)

[과제 목표]
Spark streaming을 활용하여 Kafka에서 발생되는 movie 데이터 스트림을 슬라이딩 윈도우로 처리하는 스트림 애플리케이션 만들기

[숙제 환경 셋팅]

1. 실습시간에 만들고 'Stop'시킨 EC2 3대 (Master, Worker1, Worker2) 'Start'
2. 제공되는 'HW1_2-SparkStreaming' notebook을 MASTER_PUBLIC_IP:8888 jupyter에 upload
3. 1대의 Master, 2대의 Worker에서 돌아가는 Spark cluster 셋업 (notebook에서 커맨드 제공함)
4. 만약 이미 다운로드가 안되어있으면, Kafka connector에 필요한 jar파일을 다운로드 (notebook에서 커맨드 제공함)

[과제 내용]
TA가 제공한 Kafka Broker(IP Address: 147.46.216.122:9092, Topic: movie)로부터 JSON 데이터 스트림을 제공받아, 다음의 연산을 차례대로 수행한 후 데이터를 HDFS에 저장. (notebook내에서 코딩 가능)

1. Kafka로부터 JSON형식으로 된 movie 데이터 스트림 받아 이를 batch size가 5초인 stream으로 만들기 (1점)

2. 데이터 스트림을 python json라이브러리를 사용하여, python dict형식으로 파싱하기 (2점)

   데이터 스트림 예시: {"genre": "Drama", "year": 1968, "title": "The Killing of Sister George"}

3. filter연산을 이용하여 "genre"에 "Drama"가 포함되지 않은 데이터들만을 추출해내기. 대소문자는 구별하지 않음 (3점)

예시 1: {“genre”: “Comedy”, …, ...} 는 통과
예시 2: {“genre”: “Drama”, …, ...} 는 통과시키지 않음
예시 3: {“genre”: “Comedy, drama”, …, …}는 통과시키지 않음 (“drama”가 포함)

4. Size = 30초, Interval = 5초인 Window를 만든 후, 10년단위의 “year”값을 key로 하여 (ex: 1950년대, 2010년대) 각 년대의 영화가 Window안에 몇 편이 있는지를 계산하기 (3점)

예시: Window내부에 1956년, 1961년, 1969년, 1971년도에 만들어진 영화가 있으면, “(1950, 1), (1960, 2), (1970, 1)”의 결과를 출력

5. 계산된 결과를 pprint()를 이용하여 stdout으로 출력 (1점)

[제출 방식]
팀별로 코딩이 완료된 Notebook파일 (.ipynb) 1 copy를 이메일로 제출.
채점은 10점 만점으로 [과제 내용]에 있는 배점 기준에 따라 이루어짐.

HW #1-1 SparkSQL (총 10점) 답안 예시

실습시간에 실행한 다음의 Query를 사용했을때의 답안 예시

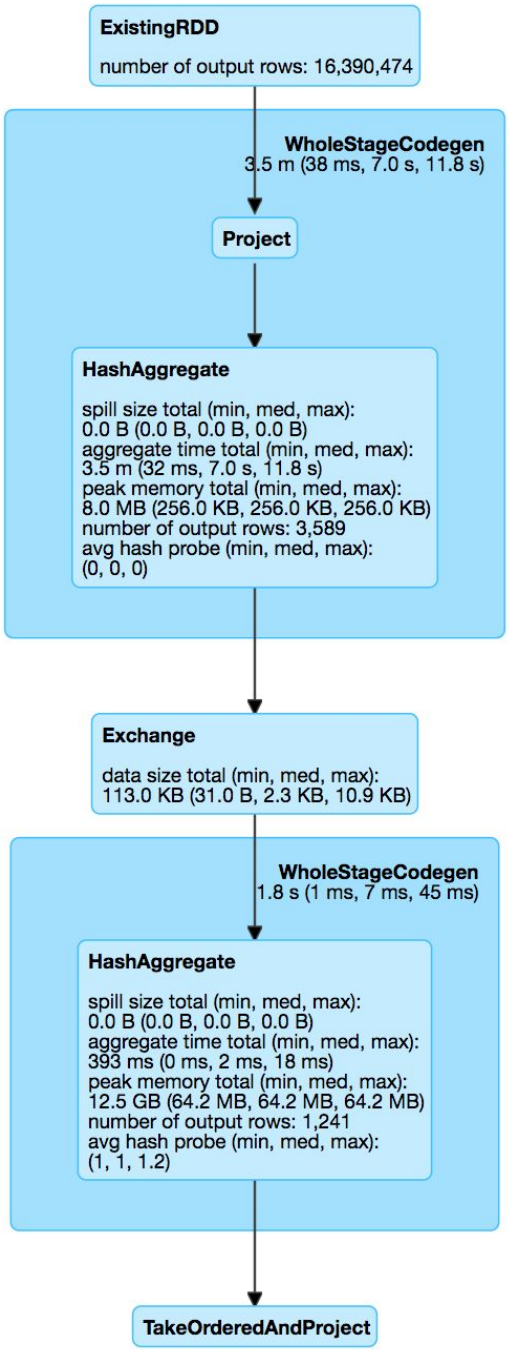
SELECT project, COUNT(title) as num_of_title FROM WikipediaTable
GROUP BY project ORDER BY num_of_title DESC

(1) SQL query
SELECT project, COUNT(title) as num_of_title FROM WikipediaTable
GROUP BY project ORDER BY num_of_title DESC

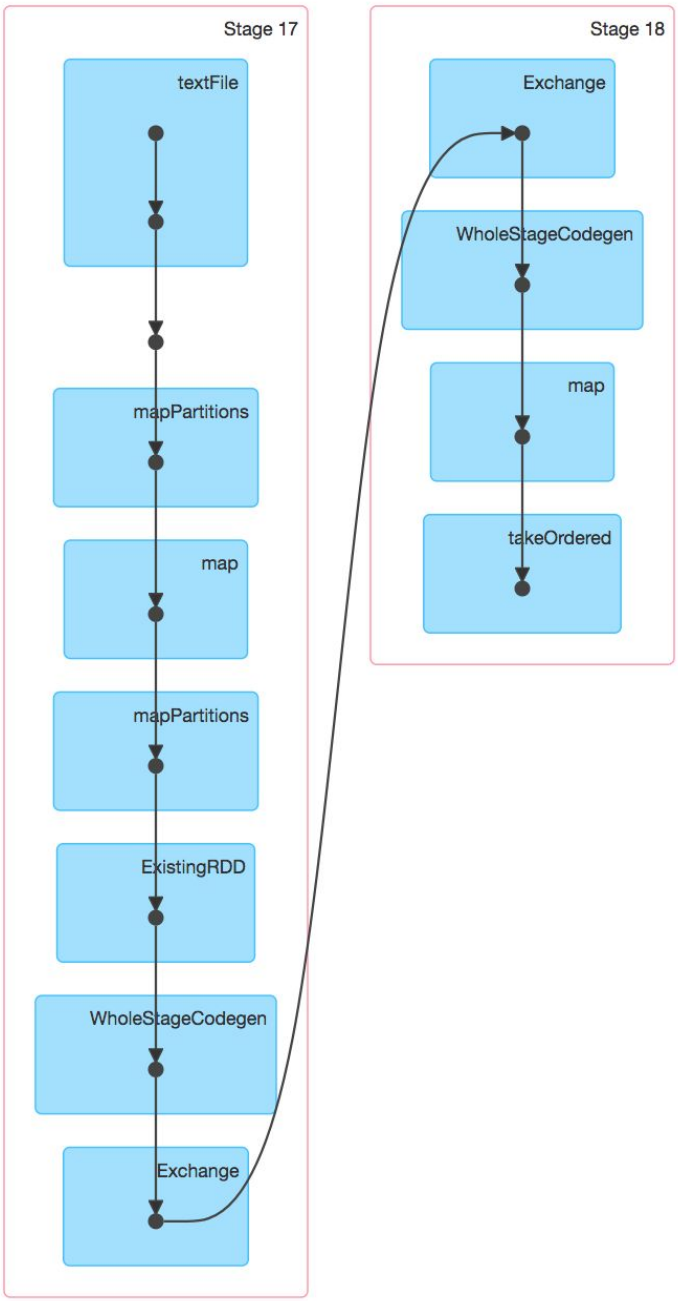
(2) jupyter notebook에 보여지는 Query 결과값

```
+---------+------------+
|  project|num_of_title|
+---------+------------+
|       en|     5843326|
|commons.m|      982905|
|       zh|      763713|
|       de|      696433|
|       fr|      692983|
|       ru|      642028|
|       ja|      610854|
|       es|      607773|
|   www.wd|      443484|
|       it|      356246|
|       pl|      310897|
|       pt|      286648|
|     en.d|      219174|
|       nl|      181850|
|       tr|      178015|
|       sv|      138963|
|       ko|      127152|
|       ar|      122273|
|       vi|      108635|
|       fa|      100613|
+---------+------------+
only showing top 20 rows
```

(3) Spark Web UI ‘SQL’ tab 에서 보여지는 Query plan 그래프의 스크린샷



(4) Spark Web UI ‘Jobs’ tab 에서 보여지는 DAG Visualization 그래프의 스크린샷



(5) Spark Web UI ‘Stages’ tab에서 보여지는, 해당 Query에 대응하는 Job에 속하는 모든 Stage들의 ‘Summary Metrics for N Completed Tasks’ 테이블의 스크린샷

<Stage 17>

Summary Metrics for 32 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	55 ms	4 s	7 s	10 s	12 s
GC Time	0 ms	18 ms	30 ms	47 ms	82 ms
Input Size / Records	227.0 B / 10	32.1 MB / 299251	32.1 MB / 530491	32.1 MB / 820128	32.7 MB / 1034473
Shuffle Write Size / Records	72.0 B / 1	1170.0 B / 17	4.5 KB / 73	9.9 KB / 189	15.0 KB / 347

<Stage 18>

Summary Metrics for 200 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	2 ms	5 ms	7 ms	10 ms	27 ms
GC Time	0 ms	0 ms	0 ms	0 ms	9 ms
Shuffle Read Size / Records	213.0 B / 3	745.0 B / 13	923.0 B / 18	1147.0 B / 22	1928.0 B / 53

HW #1-2 Spark Streaming (총 10점) 답안 예시

pprint()를 실행시켰을 때의 예상되는 결과 Output

```
-------------------------------------------
Time: 2018-04-12 07:30:15
-------------------------------------------
(1990, 3)
(1930, 2)
(1970, 2)
(1940, 3)
(2010, 1)
(1980, 1)
(1950, 1)

-------------------------------------------
Time: 2018-04-12 07:30:20
-------------------------------------------
(1920, 1)
(1990, 4)
(1930, 2)
(1970, 2)
(1940, 3)
(2010, 1)
(1980, 2)
(1950, 3)

-------------------------------------------
Time: 2018-04-12 07:30:25
-------------------------------------------
(1920, 2)
(1990, 4)
(1930, 2)
(1970, 3)
(1940, 4)
(2010, 1)
(1980, 2)
(1950, 5)
```