Zad 2

jak tłumaczyłam:
Spark job - praca Sparka
Spark action - akcja Sparka
Spark stage - etap pracy Sparka
Spark task - zadanie Sparka

Spark UI (User Interfaces)

tj (Jobs, Stages, Tasks, Storage, Environment, Executors, and SQL)
monitorują one status aplikacji Sparka, zużycie zasobów i konfiguracje Sparka.

1. zakładka Spark Jobs

1.1 Scheduling Mode - 3 tryby:
- Standalone mode
- YARN
- Mesos

1.2 Number of Spark Jobs jest równa liczbie akcji Sparka w aplikacji.
Na każdą pracę Sparka powinien przypadać co najmniej jeden etap pracy.

1.3 Number of Stages
Każda sekwencja transformacji może generować inną liczbę etapów prac Sparka

1.4 Description
Są linki do opisów z detalami prac Sparka, takich jak: Spark Job Status, DAG Visualization, Completed Stages

2. zakładka Stages

Możemy:
- wybrać Description danej pracy Sparka (pokazują się etapy dla tego danej pracy)
- na górze zakładki Spark Jobs wybrać opcję Stages (pokazują się wtedy wszystkie etapy prac w całej aplikacji)

W zakładce Stage tab jest strona, która pokazuje obecny stan wszystkich etapów dla wszystkich prac Sparka w aplikacji.

Liczba zadań (tasks) dla każdego etapu pracy to jest liczba partycji danych, na których będzie pracował Spark. Każde zadanie w etapie to będzie wykonanie przez Sparka tego samego, ale na różnych partycjach danych.

3. Tasks - są zlokalizowane na dole strony dla każdego etapu.

Na co zwrócić uwagę:
- input size - dla danego etapu
- Shuffle Write-Output - to jest etap zapisany słownie (?)

4. Storage

Pokazuje RDDs i DataFrames, które pozostały w aplikacji. W summary są: Storage levels, rozmiary i partycje dla wszystkich RDDs, w w details są rozmiary dla wszystkich partycji w RDDs lub DataFrames.

5. Environment

Ma pięć opcji:

- Runtime Information: zawiera runtime properties np. wersja Javy i Scali.
- Spark Properties: właściwości aplikacji takie jak ‘spark.app.name’ and ‘`spark.driver.memory`’.
- Hadoop Properties: właściwości związane z Hadoop i YARN. Uwaga: właściwość ‘`spark.hadoop`’ jest w ‘Spark Properties’.
- System Properties: szczegóły JVM.
- Classpath Entries: listuje klasy załadowane z innych źródeł, przydatne do rozwiązywania konfliktów między klasami.

Ogólnie w tej zakładce są wartości różnych zmiennych środowiskowych i zmiennych konfiguracji (?), włączając JVM, Sparka i właściwości systemowe.

6. Executors
   - zbiorcze informacje nt. wykonawców, które zostały utworzone dla aplikacji, włączając zużycie pamięci, zapełnienie dysku.
   W kolumnie Storage Memory jest ilość pamięci użytej i zarezerwowanej na tzw. caching (buforowanie danych).
   Są tam też informacje o wydajności (performance information)
7. SQL
   - jeśli aplikacja wykorzystuje zapytania SQL, wtedy w tej zakładce znajdują się informacje o czasie trwania, pracach Sparka, fizycznych i logicznych planach wykonania zapytania.
8. Structured Streaming Tab
   Jeśli wykorzystujemy prace Sparka w trybie “micro-batch“
   krótkie statystyki dla działających i zakończonych zapytań, najnowsze błędy dla zapytań zakończonych niepowodzeniem.
9. Streaming (DStreams) Tab
   Jeśli aplikacja korzysta ze Spark Streaming z API DStream. W zakładce pokazane jest opóźnienie i czas przetwarzania dla każdego “micro-batch” w potoku danych.
10. JDBC/ODBC Server Tab
    Jeśli Spark działa jako “distributed SQL engine” (silnik bazy danych?). Informacje o sesji i ukończonych operacjach SQL.

Zad 3

komendę groupBy(“name”).count() wykonałam w notatniku Ćwiczenia 3

Rezultatem jest plan fizyczny

wykonane zostało po kolei:

- FileScan - skanowanie pliku csv
- HashAggregate - podział na partycje - na każdej partycji wykonano partial_count
- Exchange hashpartitioning - Shuffle danych - przekazanie wyników działania partial_count na partycjach dalej, gdzie 200 to rozmiar jednej partycji
- HashAggregate - merge wyników z pomniejszych zestawów danych (z partycji) i wykonanie final count