

Program szkolenia:

Przetwarzanie danych Big Data z wykorzystaniem Apache Spark

Informacje:

Nazwa:	Przetwarzanie danych Big Data z wykorzystaniem Apache Spark
Kod:	BigDataML-BigData
Kategoria:	BigData, streaming i Machine Learning
Czas trwania:	2-3 dni
Forma:	40% wykłady 60% ćwiczenia

W trakcie szkolenia nauczymy się jak używać Apache Spark framework do szybkiego przetwarzania dużych ilości danych.

Kurs obejmuje wprowadzenie do architektury Apache Spark. Szkolenie może zostać przeprowadzone w języku Scala lub Python. Warsztaty obejmują kompleksowy proces tworzenie aplikacji Sparkowej - integrację ze źródłem, przetwarzanie danych, optymalizację procesu i zapis do bazy danych w środowisku Cloudowym.

Zobaczymy jak wygląda Spark API i będziemy pisać Spark Joby ilustrujące typowe jak i specyficzne problemy. Omówimy optymalizacje, najczęściej spotykane wyzwania i sposoby na ich pokonanie. Szkolenia skupia się głównie na części praktycznej

Szczegółowy program:

1. Wprowadzenie do przetwarzania danych z wykorzystaniem Apache Spark.

1.1. Geneza - najważniejsze zmiany, co nowego w obecnej wersji Apache Sparka, integracja z ClouDEM/Hadoopem

1.2. Wprowadzenie do API

1.2.1. RDD / Dataset / Dataframe

1.2.2. Główne cechy i różnice i porównanie wydajności

1.2.3. Rekomendacje i tips and tricks

1.3. Lazy Evaluation - Transformations and Actions

1.3.1. Jak zbudowany jest graf wykonywania transformacji

1.3.2. Kiedy graf wykonywania jest wykorzystywany

1.3.3. Jak re-używać wcześniej stworzone RDD z wcześniej wykonanymi transformacjami

1.4. Shuffling - Przesyłanie danych między maszynami

1.4.1. Które transformacje wymagają shufflingu (wide i narrow)

1.4.2. Koncept ReduceByKey i GroupByKey

1.4.3. Jak zminimalizować wpływ na wydajność aplikacji

1.5. Data Partitioning

1.5.1. Kiedy warto wykonać metodę repartition() a kiedy coalesce()

1.5.2. Zasady dzielenia wyników na partycje

1.5.3. Ilość/rozmiar partycji a wydajność przetwarzania

1.6. Podstawowa konfiguracja projektu bazującego na Apache Spark:

1.6.1. Jak skonfigurować skrypt uruchomieniowy joba

1.6.2. Jak pisać kod jobów aby był łatwo testowalny

1.7. Możliwości integracji Sparka z obecnymi rozwiązaniami (baza danych, hdfs, avro, text file, csv, json ...)

2. Architektura, integracja, najczęstsze problemy i optymalizacja aplikacji bazujących na Apache Spark

2.1. Architektura

2.1.1. Spark Driver, Worker i Executor

2.1.2. Job vs Stage vs Task

2.1.3. Jednostki przetwarzania i danych

2.1.4. Możliwości deploymentu

2.2. Testowanie Spark jobów

2.3. Joiny

2.3.1. Fizyczne typy joinów

2.3.2. Best practices

2.3.3. Wykorzystanie Join aby zminimalizować przesyłanie danych pomiędzy maszynami (reduce shuffling)

2.4. UDF - jak je konstruować i jaki mają wpływ na wydajność. Różnice między Dataframe i Dataset.

2.5. Optymalizacje jobów sparkowych i typowe problemy

2.5.1. Key-skew

2.5.2. OOM

2.5.3. Broadcast

2.5.4. Cache

2.5.5. Serialization

2.5.6. Jak dobierać rozmiar executorów

2.6. Interpretacja i optymalizacja planów zapytań

2.6.1. Poruszanie się po Spark UI

2.6.2. Jak sprawdzać ii na co zwrócić szczególną uwagę

2.7. Spark Catalyst i Tungsten