

NIERELACYJNE ROZWIĄZANIA BAZODANOWE

Wykład 2

Agenda



2

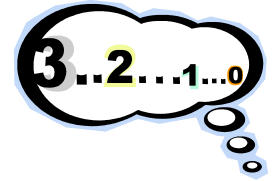
Narzędzia informatyczne do tworzenia,
przetwarzania oraz zarządzania zbiorami danych

Narzędzia ekosystemu Apache Hadoop

Narzędzia analityczne MapReduce

3

Narzędzia informatyczne do tworzenia, przetwarzania oraz zarządzania dużymi zbiorami danych



- ☐ Hadoop
- ☐ Spark
- ☐ Nutch
- ☐ Pig Latin
- ☐ Hive
- ☐ MapReduce
- ☐ NoSQL



Czym jest Hadoop?

5

- Platforma open source do przetwarzania danych
- Zadania:
 - ▣ Model programowania
 - ▣ Wsparcie dla algorytmów MapReduce
 - ▣ Model dystrybucji danych i przetwarzania równoległego
 - ▣ Zarządza klastrem

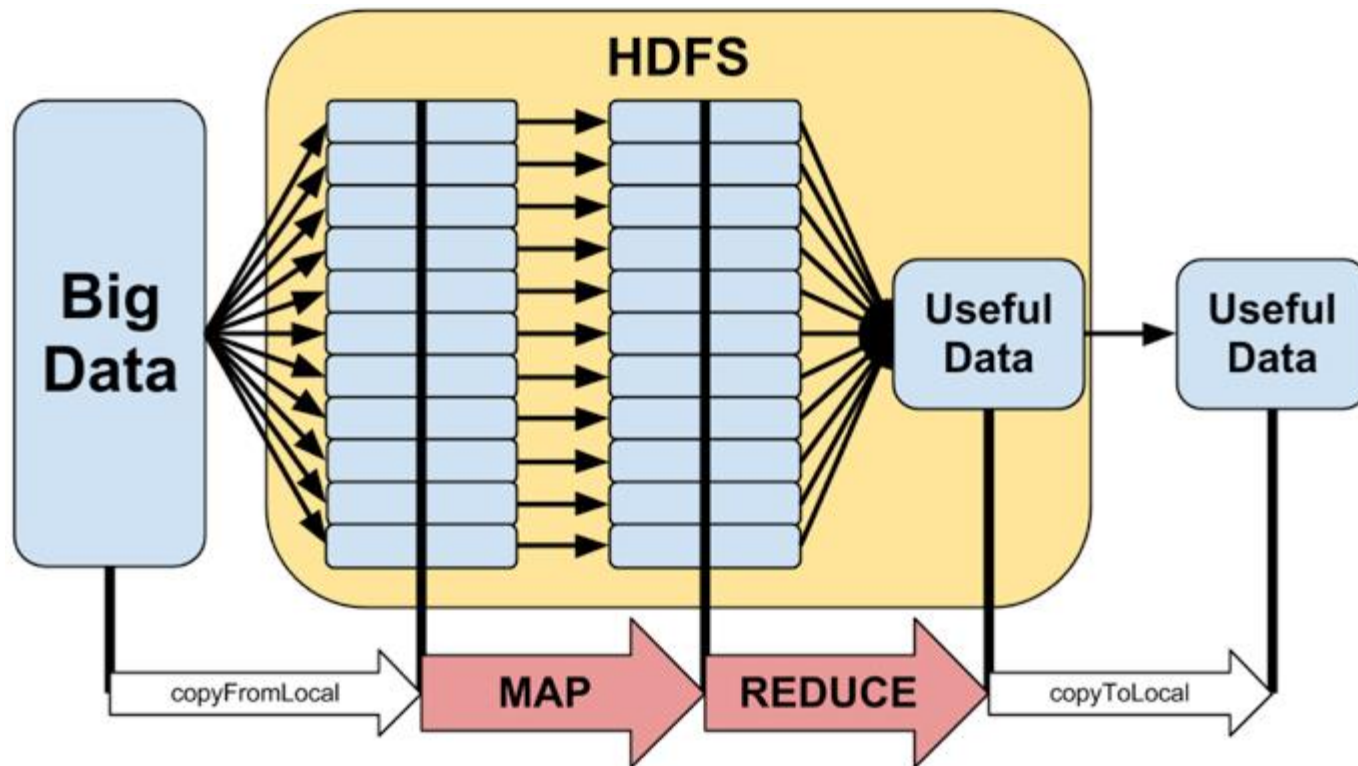
Komponenty

6

- Trzy aspekty oprogramowania:
 - ▣ **Hadoop Common**
 - ▣ **HDFS**
 - ▣ **MapReduce**

Jak działa HDFS?

7



Oprogramowanie

8

- Projekty Apache
- Projekty Non-Apache
- Dostawcy własnych implementacji Hadoop
- Dostawcy przyjaznych interfejsów Hadoop
- Projekt Hadoop as a Service

Projekty Apache Hadoop

- Magazyn danych (NoSQL)
 - ▣ HBase
 - ▣ Cassandra
 - ▣ Hive
- Analiza danych
 - ▣ Pig
 - ▣ Mahout
- Koordynacja i zarządzanie
 - ▣ Ambari
 - ▣ Zookeeper
- Narzędzia użytkowe
 - ▣ Avro

Bazy NoSQL

10

- Magazyn dokumentów
 - ▣ Standardowy format (XML, JSON itp.)
 - ▣ Klucze unikalne
- Grafowe
- Magazyny danych typu klucz-wartość (key-value)
 - ▣ Rekordy są parami klucz-wartość (definiowane rekursywnie)
 - ▣ Obsługują duże tabele

Model BigTable

11

- Magazyn pamięci rozwijany przez Google
- BigTable to rozproszona, stała wielowymiarowa posortowana kolekcja typu mapa (mapuje klucz wartość)

Przykład HBase

12

Implementacja BigTable, która używa silnika HDFS

Row Key	Time Stamp	ColumnFamily contents	ColumnFamily anchor
"com.cnn.www"	t9		anchor:cnnsi.com = "CNN"
"com.cnn.www"	t8		anchor:my.look.ca = "CNN.com"
"com.cnn.www"	t6	contents:html = "<html>..."	
"com.cnn.www"	t5	contents:html = "<html>..."	
"com.cnn.www"	t3	contents:html = "<html>..."	

Jak działa MapReduce?

13

- Algorytmy MapReduce
- Model programowania do równoległego przetwarzania danych
- Wiele języków programowania i platform może być używanych
- Dostarcza abstrakcyjny model dla programistów

Model programowania MapReduce

14

- Program MapReduce dokonuje transformacji z listy wejściowej do listy wyjściowej
- Dwa etapy:
 - ▣ map (in_key, in_value) ->
 - (out_key, intermediate_value) list
 - ▣ reduce (out_key, intermediate_value list) ->
 - out_value list

Etap pierwszy: map

15

- Źródło danych powinno być zapisane w rekordach (linie w plikach, wiersze bazy danych...)
- Każdy rekord musi posiadać klucz
- Rekordy są przetwarzane dzięki funkcji map jako pary klucz-wartość (plik, wiersze)
- map() dostarcza jednej lub wielu pośredniczących wartości wraz z kluczem wyjściowym pochodzącym z danych wejściowych
- Mapowanie identyfikuje wartości wejściowe posiadające tę samą charakterystykę za pomocą klucza wyjściowego

Ćwiczenie 1.

16

- Podaj przykładowe pary klucz-wartość.
- Znajdź dowolne źródło danych (np. lista produktów, oferty pracy) i podaj w jaki sposób można zapisać je jako pary klucz-wartość.

Funkcja reduce()

17

- Po zakończeniu fazy mapowania, wszystkie pośredniczące wartości dla określonego grupy kluczy wyjściowych jest łączone w listę
- Funkcja reduce() agreguje pośredniczące wartości w jedną lub więcej wartości końcowych dla tych samych kluczy pośredniczących
 - ▣ zwykle jest jedna wartość końcowa dla jednego klucz

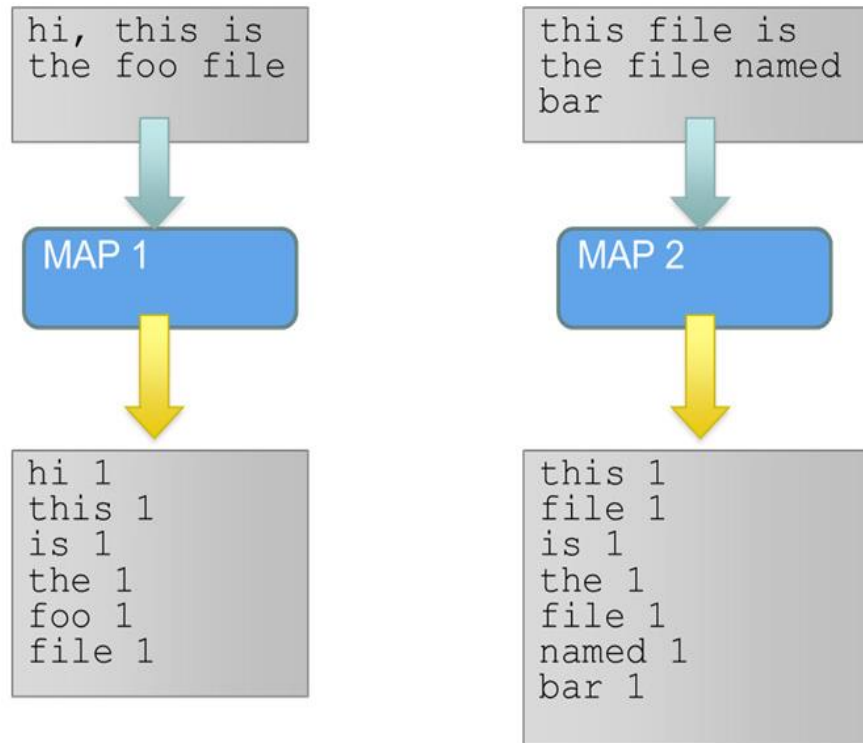
Typowe aplikacje MapReduce()

18

- Agregacja danych
- Analiza dziennika zdarzeń
- Statystyki
- Uczenie maszynowe

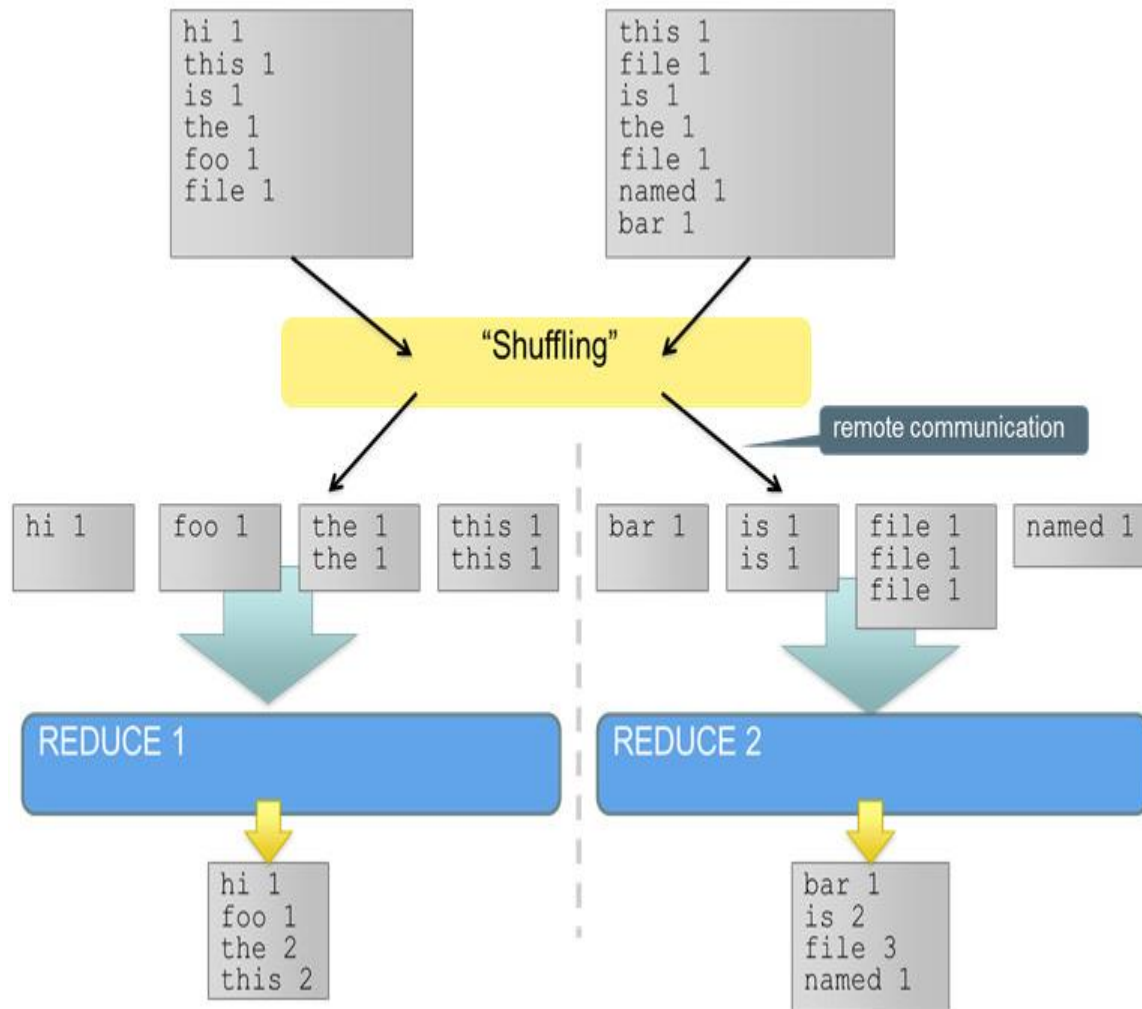
Przykład: liczenie słów - mapowanie

19



Przykład: liczenie słów - redukcja

20



Agregator z MapReduce

21

INPUT

```
D1,D2,D3,V
Alice, Angola, B, 4
Alice, Angola, A, 5
Alice, Brazil, A, 3
Alice, Angola, A, 4
Alice, Brazil, B, 8
Bob, Angola, A, 5
Bob, Brazil, A, 7
Bob, Brazil, B, 10
Bob, Angola, B, 20
```



MAP OUTPUT

```
Alice 4
Alice, Angola 4
Alice, B 4
Angola 4
B, Angola 4
B 4
Alice, Angola, B 4
Total 4

Alice 5
Alice, Angola 5
Alice, A 5
Angola 5
A, Angola 5
A 5
Alice, Angola, A 5
Total 5

...
```

Funkcja agregatora

22

INPUT

```
D1,D2,D3,V
Alice, Angola, B, 4
Alice, Angola, A, 5
Alice, Brazil, A, 3
Alice, Angola, A, 4
Alice, Brazil, B, 8
Bob, Angola, A, 5
Bob, Brazil, A, 7
Bob, Brazil, B, 10
Bob, Angola, B, 20
```



OUTPUT

```
Alice 24
Bob 42
Angola 38
Brazil 28
A 26
B 42

Alice, Angola 13
Alice, Brazil 11
Bob, Angola 25
Bob, Brazil 17

Alice, A 11
Alice, B 12
Bob, A 12
Bob, B 30

...

Total 61
```

Jak działa agregator?

23

- Mapowanie
 - ▣ KLUCZ jest połączeniem wymiarów w jednym wierszu $\langle D1 \mid D2 \mid \dots \mid Dn \rangle$
 - ▣ WARTOŚĆ jest miarą
 - ▣ Wyjściowe wiersze
 - Każdy wiersz powstaje dla każdej kombinacji wymiarów, gdzie znajduje się jeden lub więcej wymiarów
 - Każdy wiersz ma wpływ na ogólny wynik agregacji
- Redukowanie
 - ▣ Tak jak liczenie słów

Przykład programowania MapReduce

24

- `map(String input_key, String input_value):`
- *// input_key: numer wiersza*
- *// input_value: zawartość wiersza*
- `for each word in input_value:`
- `emit (word, "1");`
- `reduce(String output_key, Iterator intermediate_values):`
- *// output_key: słowo*
- *// output_values: lista wartości*
- `int result = 0;`
- `for each v in intermediate_values:`
- `result += ParseInt(v);`
- `emit(AsString(result))`

Przykład 2. W jaki sposób można mapować i redukować?

25

Wejściowy zbiór danych

Id biletu | Nr rezerwacji | Skąd | Cel | Nr lotu | Mile

Należy utworzyć następujący zbiór danych

Cel faktyczny | Liczba pasażerów

Przykład 2. W jaki sposób można mapować i redukować?

26

Wejściowy zbiór danych

ID statku | Długość | Szerokość | Czas

Jeden rekord dla każdego odczytu

ID statku | Źródło | Cel | Liczba pasażerów

Jeden rekord na statek

Zaprojektuj następujący zbiór

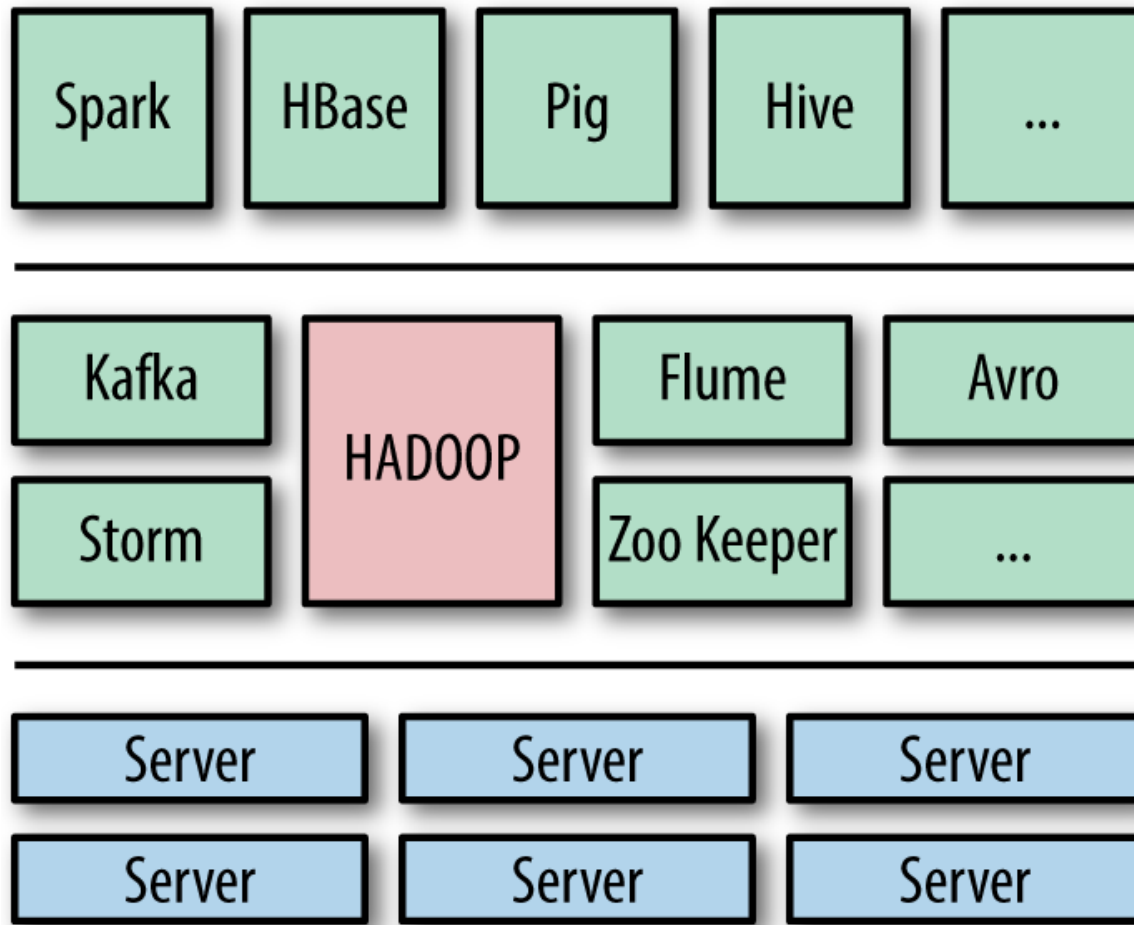
ID statku | Okres (dzień-noc) | Całkowity czas zatrzymania



Narzędzia ekosystemu Apache Hadoop

Architektura Apache Hadoop

28

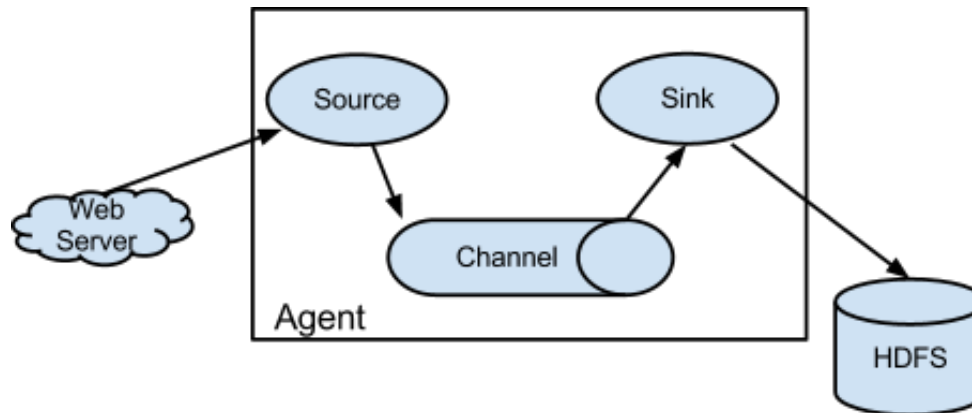




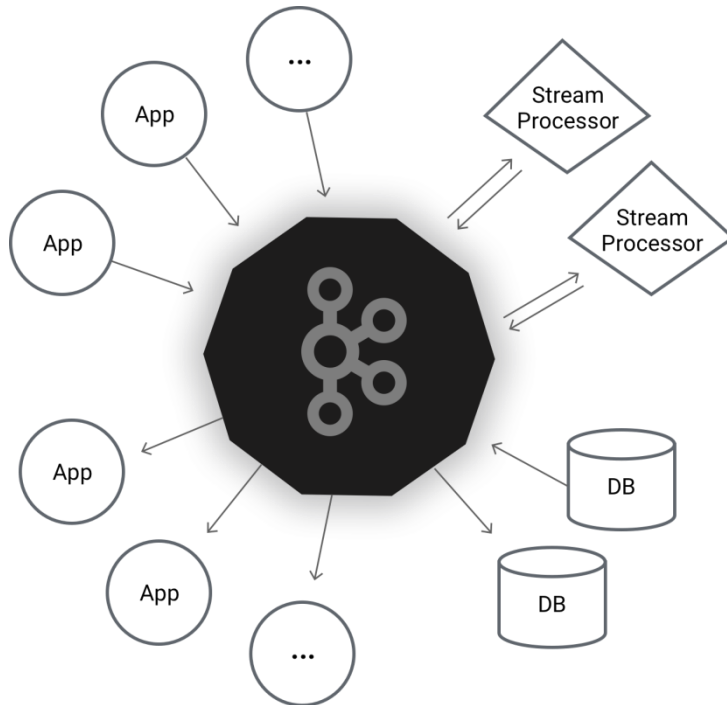
Apache Flume

29

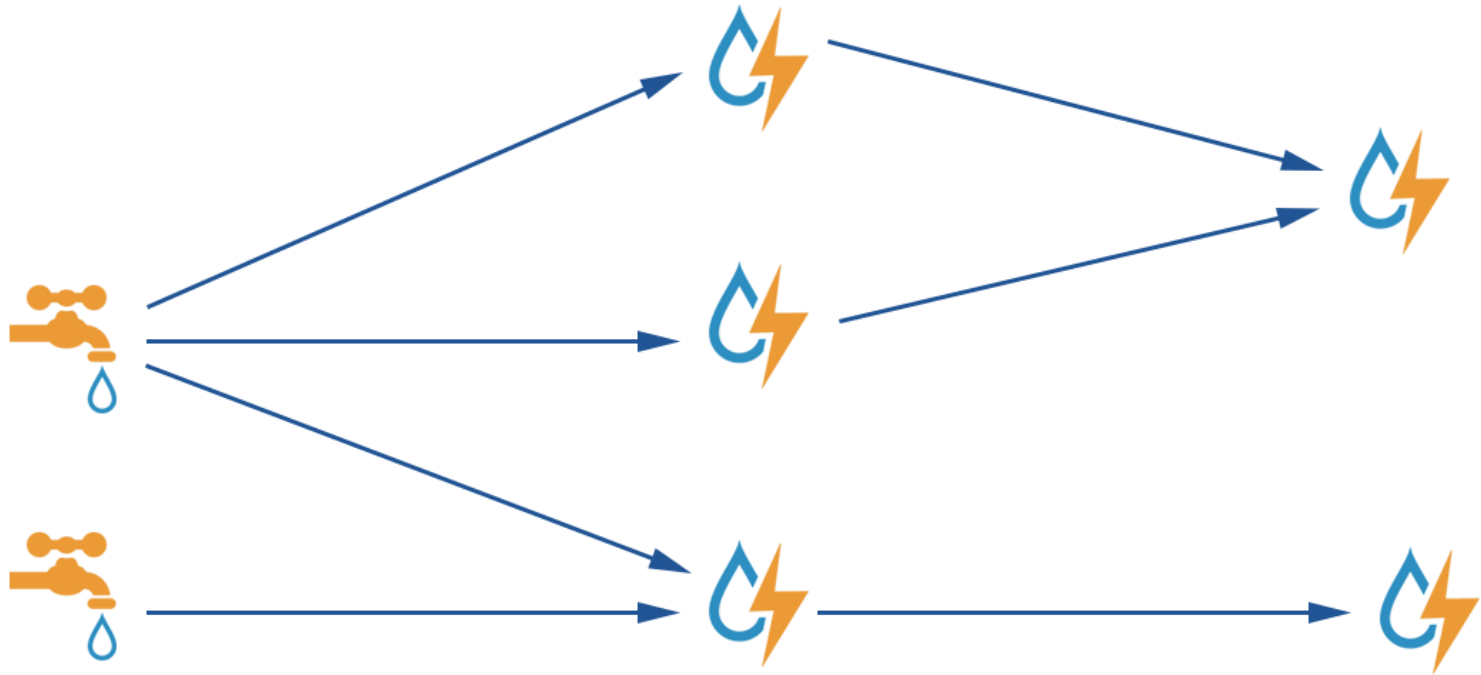
- Pobiera dane i składowuje w HDFS



- Zarządza danymi strumieniowymi (składowanie, przetwarzania, odczytywanie i zapisywanie)



- Przetwarzanie danych strumieniowych w czasie rzeczywistym

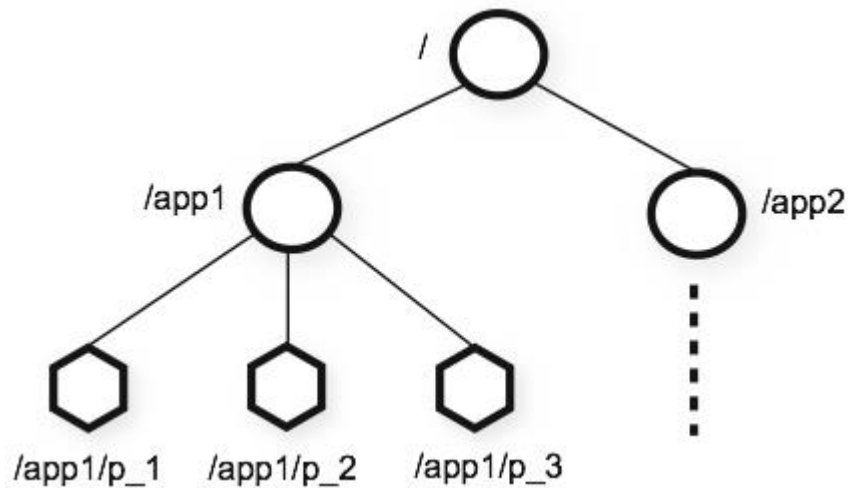




Apache ZooKeeper

32

- Zarządza serwerem Apache





□ System serializacji danych

```
{ "namespace": "example.avro",  
  "type": "record",  
  "name": "User",  
  "fields": [  
    { "name": "name", "type": "string" },  
    { "name": "favorite_number", "type": ["int", "null"] },  
    { "name": "favorite_color", "type": ["string", "null"] }  
  ]  
}
```

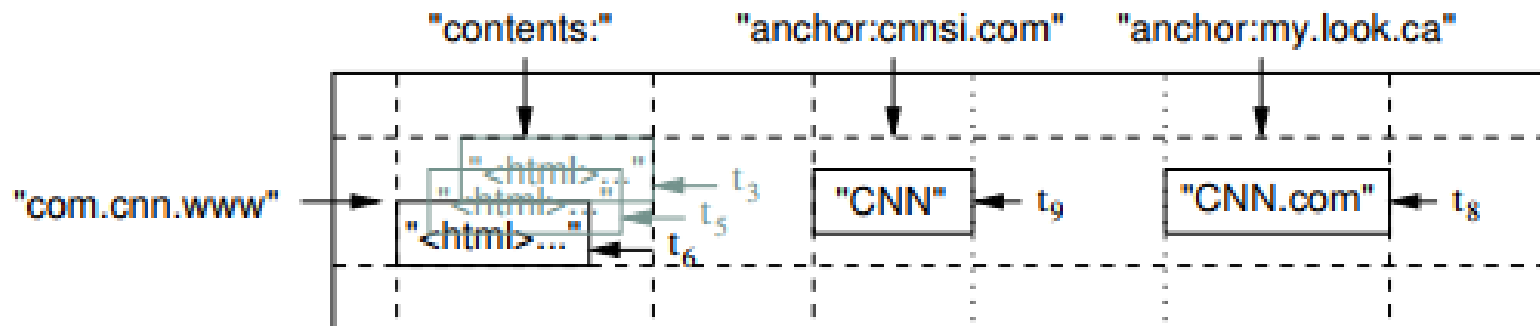


- System przetwarzania w czasie rzeczywistym, możliwość uruchamiania programów w językach Scala, Java, Python itp. na źródłach HDFS, Cassandra, HBase itp.

```
text_file = spark.textFile("hdfs://...")

text_file.flatMap(lambda line: line.split())
            .map(lambda word: (word, 1))
            .reduceByKey(lambda a, b: a+b)
```

- Baza danych NoSQL dla HDFS bazująca na modelu Google BigTable





Apache Hive

36

□ Hurtownia danych wykorzystująca SQL

```
CREATE TABLE page_view(viewTime INT, userid BIGINT,  
                        page_url STRING, referrer_url STRING,  
                        ip STRING COMMENT 'IP Address of the User')  
COMMENT 'This is the page view table'  
PARTITIONED BY(dt STRING, country STRING)  
STORED AS SEQUENCEFILE;
```

Apache Pig



37

□ Platforma do przetwarzania danych

```
A = LOAD 'student' USING PigStorage() AS (name:chararray, age:int, gpa:float);  
B = FOREACH A GENERATE name;  
DUMP B;  
(John)  
(Mary)  
(Bill)  
(Joe)
```

Ćwiczenia z MapReduce

Jak pracować z Hadoop?

39

- Użytkownik hduser
- Hadoop jest instalowany w:
 - ▣ /usr/local/hadoop
- Jak uruchomić MapReduce?
 - ▣ `hadoop jar ./share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar pi 2 5`

Jak utworzyć folder?

40

- ❑ `hadoop fs -mkdir -p /user/hduser/folder`
- ❑ W folderach (katalogach) przechowuje się zbiory danych w postaci plików.

Jak skopiować dane?

41

- Kopiowanie z lokalnego systemu plików do HDFS
 - ▣ `hadoop fs -copyFromLocal hdfsTest.txt hdfsTest.txt`
- lub prościej
 - ▣ `hadoop fs -put hdfsTest.txt`
- Skopiowanie ze zdalnego systemu plików do lokalnego
 - ▣ `hadoop fs -copyToLocal /user/hduser/hdfsTest.txt hdfsTest2.txt`

Jak wyświetlać pliki?

42

- Listowanie struktury danych:
 - ▣ `hadoop fs -ls`
- Wyświetlanie zawartości pliku:
 - ▣ `hadoop fs -cat /user/hduser/hdfsTest.txt`
- Usuwanie pliku
 - ▣ `hadoop fs -rm hdfsTest.txt`

MapReduce w 9 etapach



Etap 1: Przygotowanie systemu plików

44

1. Przejście do folderu

- **cd NAZWA_FOLDERU**
- np.
 - cd JM

2. Pobrać dane ze strony internetowej

- **wget -c STRONAWWW**
- np.
 - wget -c <http://wzr.pl/wydzial/index.php?i=26>

3. Zmienić nazwę pliku lokalnego na inputNAZWA

- **mv nazwaPliku inputNAZWA**
- np.
 - mv index.php\?i\=26 inputJM

Etap 2: Przygotowanie źródła danych w Apache Hadoop HDFS

45

4. Zalogowanie jako hduser (Hadoop User).

- **su hduser**

5. Utworzenie katalogu HDFS.

- **hadoop fs -mkdir -p /user/hduser/folder**
- np.
 - **hadoop fs -mkdir -p /user/hduser/sourceJM**

6. Skopiowanie pliku do HDFS.

- **hadoop fs -copyFromLocal plik folder/plik**
- np.
 - **hadoop fs -copyFromLocal inputJM sourceJM/inputJM**

Etap 3. Analiza WordCount

46

7. Weryfikacja dostępności pliku.

- **hadoop fs -ls folderName****hadoop fs -cat folderName/fileName**
- np.
- **hadoop fs -ls sourceJM****hadoop fs -cat sourceJM/inputJM**

8. Wykonanie analizy wordcount.

- **hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount inputFolder/inputFile outputFile**
- np.
- **hadoop jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.6.0.jar wordcount sourceJM/inputJM outputJM**

Etap 4. Weryfikacja rezultatów

47

9. Obserwuj rezultaty!

9.1. Sprawdzenie, czy plik wynikowy istnieje.

- **hadoop fs -ls**

9.2. Weryfikacja plików Hadoop.

- **hadoop fs -ls outputFolder**
- e.g.
- **hadoop fs -ls outputJM**

9.3. Weryfikacja rezultatów.

- **hadoop fs -ls outputFile**
- e.g.
- **hadoop fs -cat outputJM/part-r-00000**

Etap 5: Analiza rezultatów

48

Czy plik jest za duży?

9.4.
Skopiowanie
i analiza z grep.

- **hadoop fs -copyToLocal plikHDFS plikLokalny**
- Np.
 - `hadoop fs -copyToLocal /user/hduser/outputJM/part-r-00000 /home/hduser/outputJM.txt`

Pytanie

49

Oprogramowanie bazy danych to:

- ☐ HBase
- ☐ Pig
- ☐ Pig Latin
- ☐ Spark

