Apache Hadoop

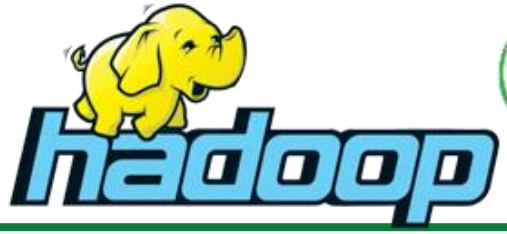
iBigWorld:
Innovations for Big Data in a Real World

Prof. dr. Dragan Stojanovic, **UNI**
Prof. dr. Natalija Stojanovic, **UNI**

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Hadoop w pigułce



- Ramy oprogramowania open-source (projekt Apache)
- Pierwotnie opracowany przez Yahoo (wywodzi się z Google)
- Cel: przechowywanie i przetwarzanie zbiorów danych na masową skalę
- Infrastruktura: klastry sprzętu typu commodity
- Rdzeń:
- HDFS, rozproszony system plików
- MapReduce, model programowania do przetwarzania danych na dużą skalę.
- Obejmuje szereg projektów pokrewnych (Ekosystem)
- Apache Pig, Apache Hive, Apache HBase itp.
- Używane produkcyjnie przez Google, Facebook, Yahoo! i wiele innych.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

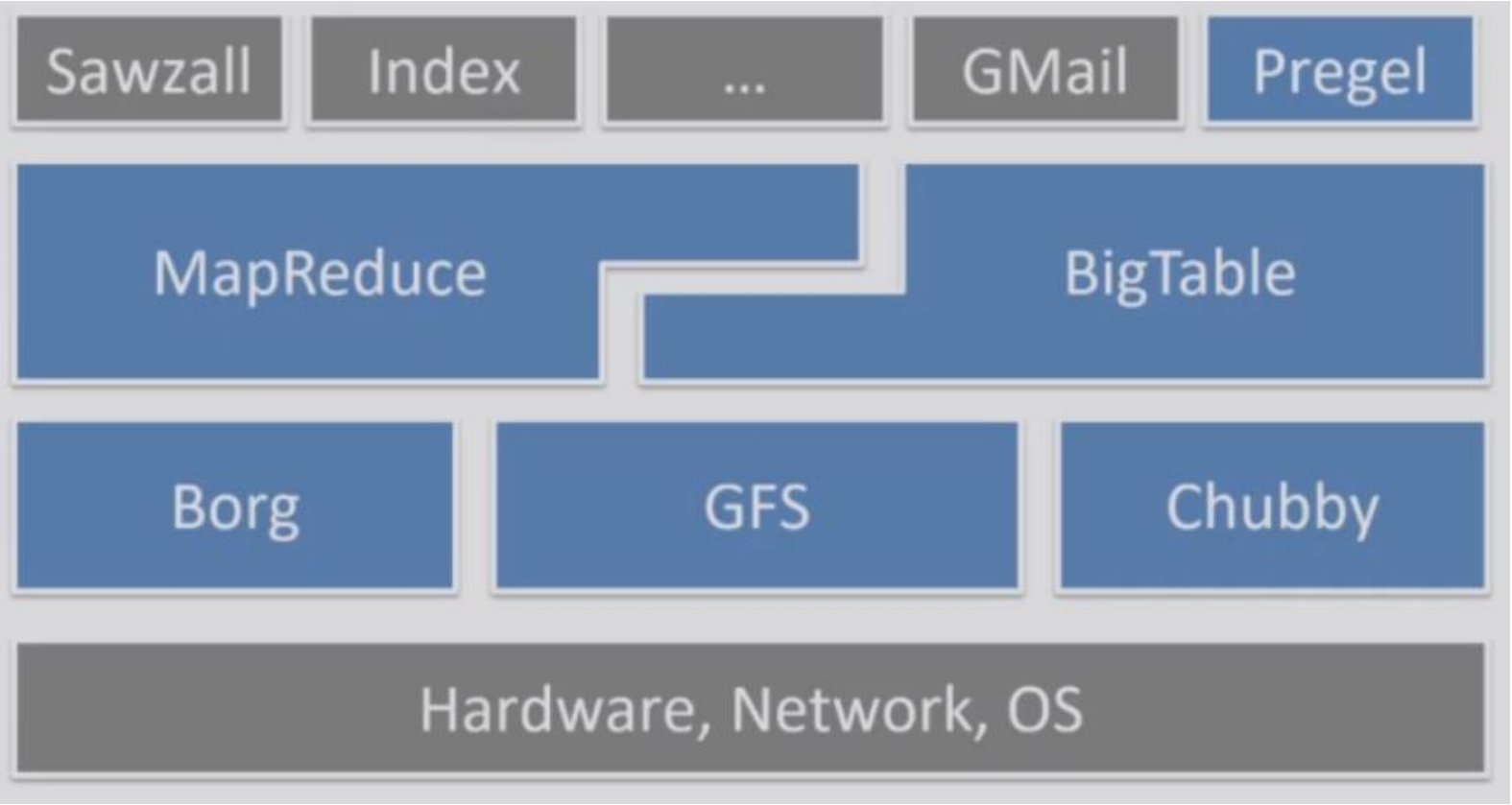
Hadoop: trochę historii

- 2003: Google publikuje artykuł o swojej architekturze klastrowej i rozproszonym systemie plików (GFS)
- 2004: Google publikuje artykuł o swoim modelu programowania MapReduce używanym na szczycie GFS
- napisany w C++
- closed-source, Python i Java API dostępne tylko dla programistów Google
- 2006: Apache & Yahoo! → Hadoop & HDFS (Doug Cutting i Mike Cafarella)
- open-source'owa implementacja Google MapReduce i GFS w Javie
- różnorodny zestaw API dostępny publicznie
- 2008: staje się niezależnym projektem Apache
- Yahoo! używa Hadoop w produkcji
- Dziś: platforma ogólnego przeznaczenia do przechowywania i analizy Big data
- Dystrybucje Hadoop od kilku dostawców, w tym EMC, IBM, Microsoft, Oracle, Cloudera, Hortonworks itp.
- wielu użytkowników (http://wiki.apache.org/hadoop/PoweredBy)
- badania i rozwój aktywnie trwają...

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

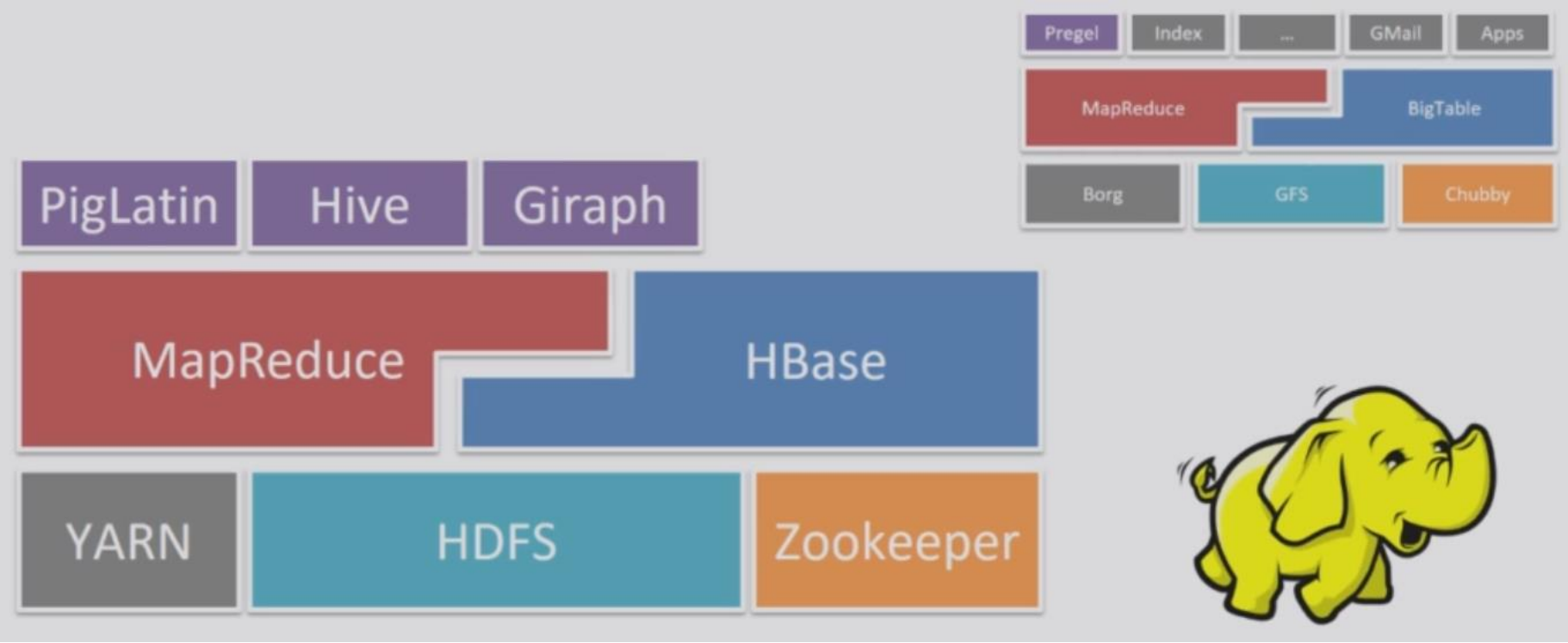
Google Stos dużych danych– 2010+



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

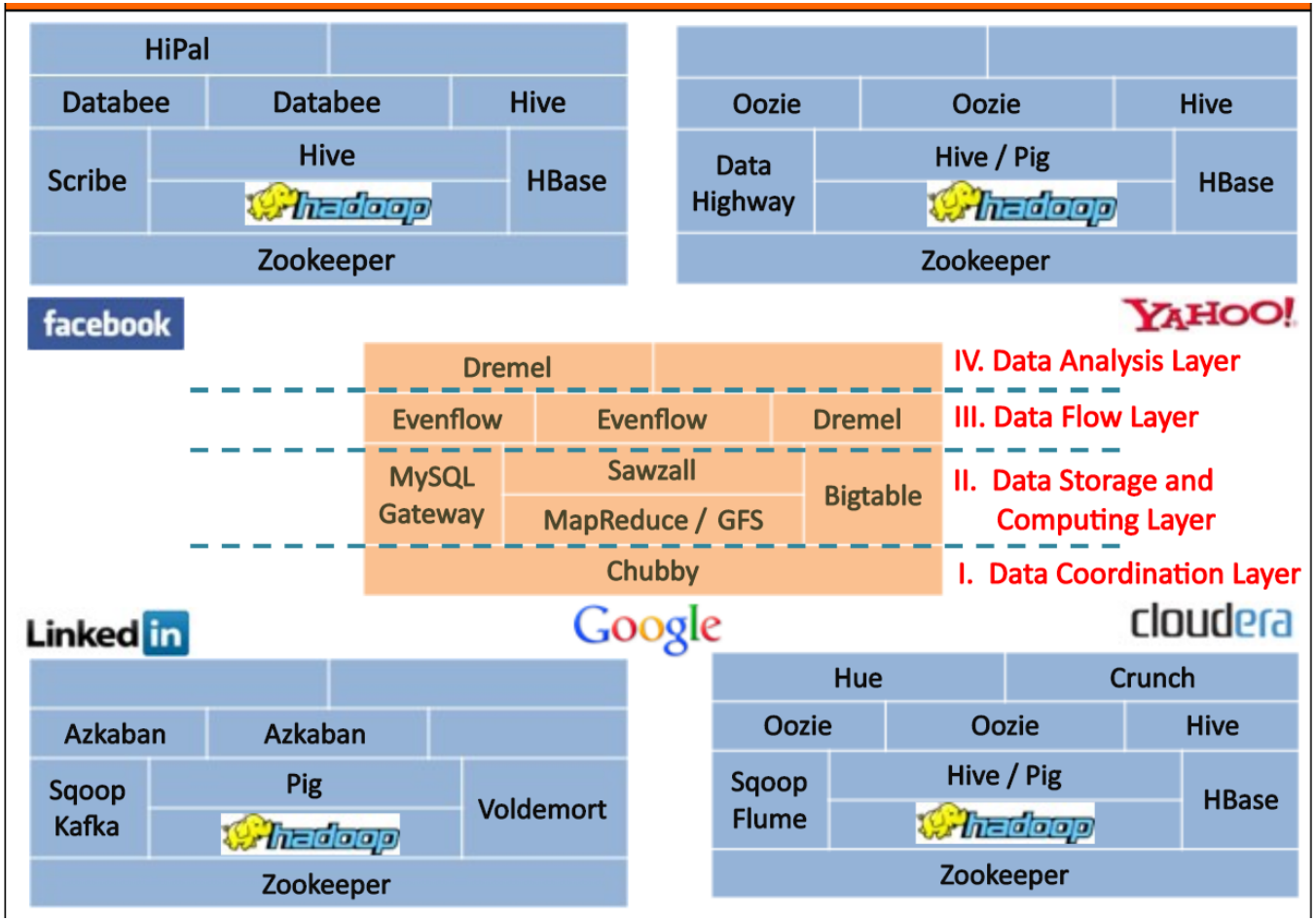
Apache Hadoop ecosystem



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

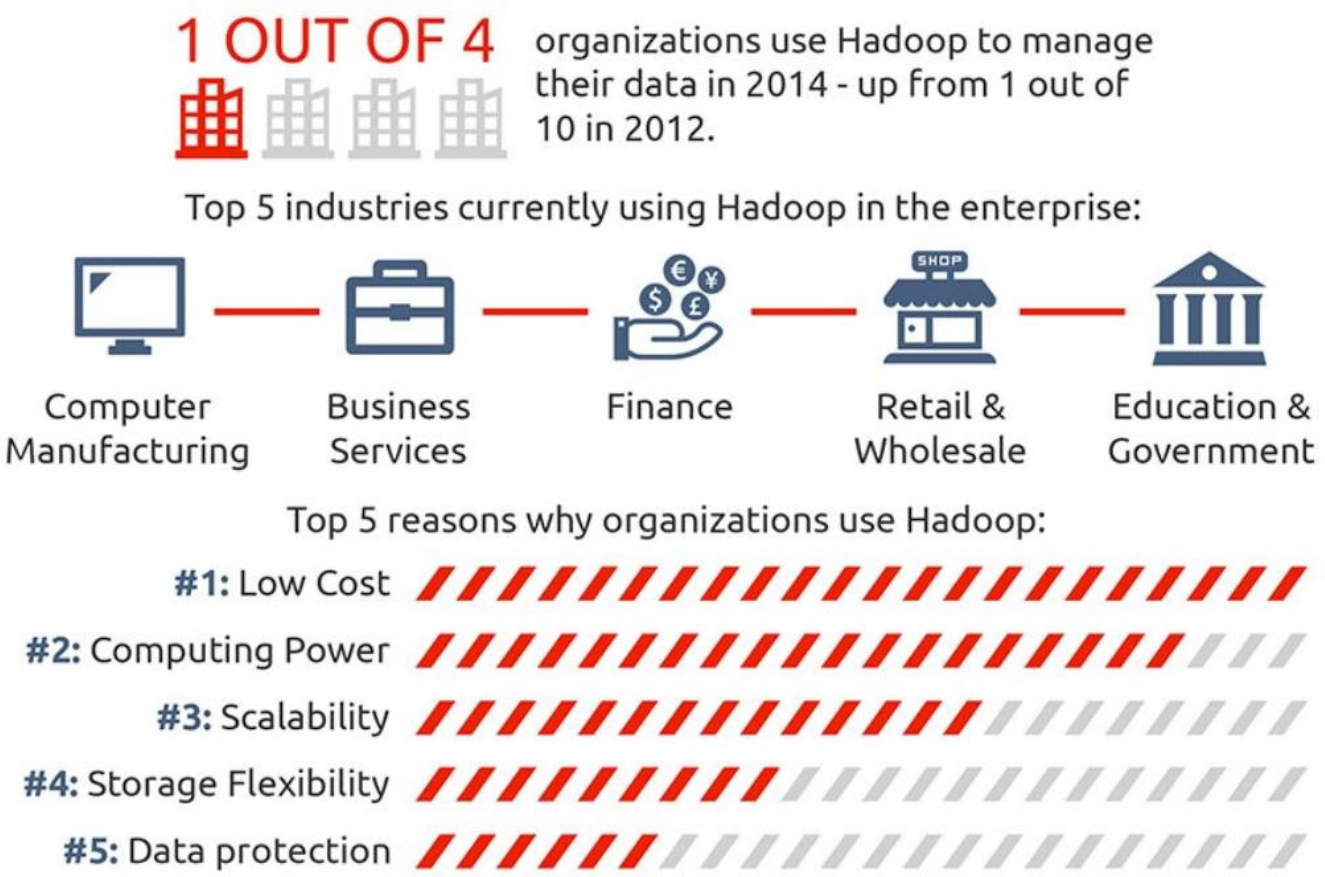
Hadoop software stacks



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Kto używa Hadoop i dlaczego?



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Cechy Hadoopa

61% of organizations plan to deploy Hadoop in the near future, or have already partially deployed it.

$50.2B Worldwide sales based on the Hadoop technology are forecasted to reach $50.2 billion by 2020.

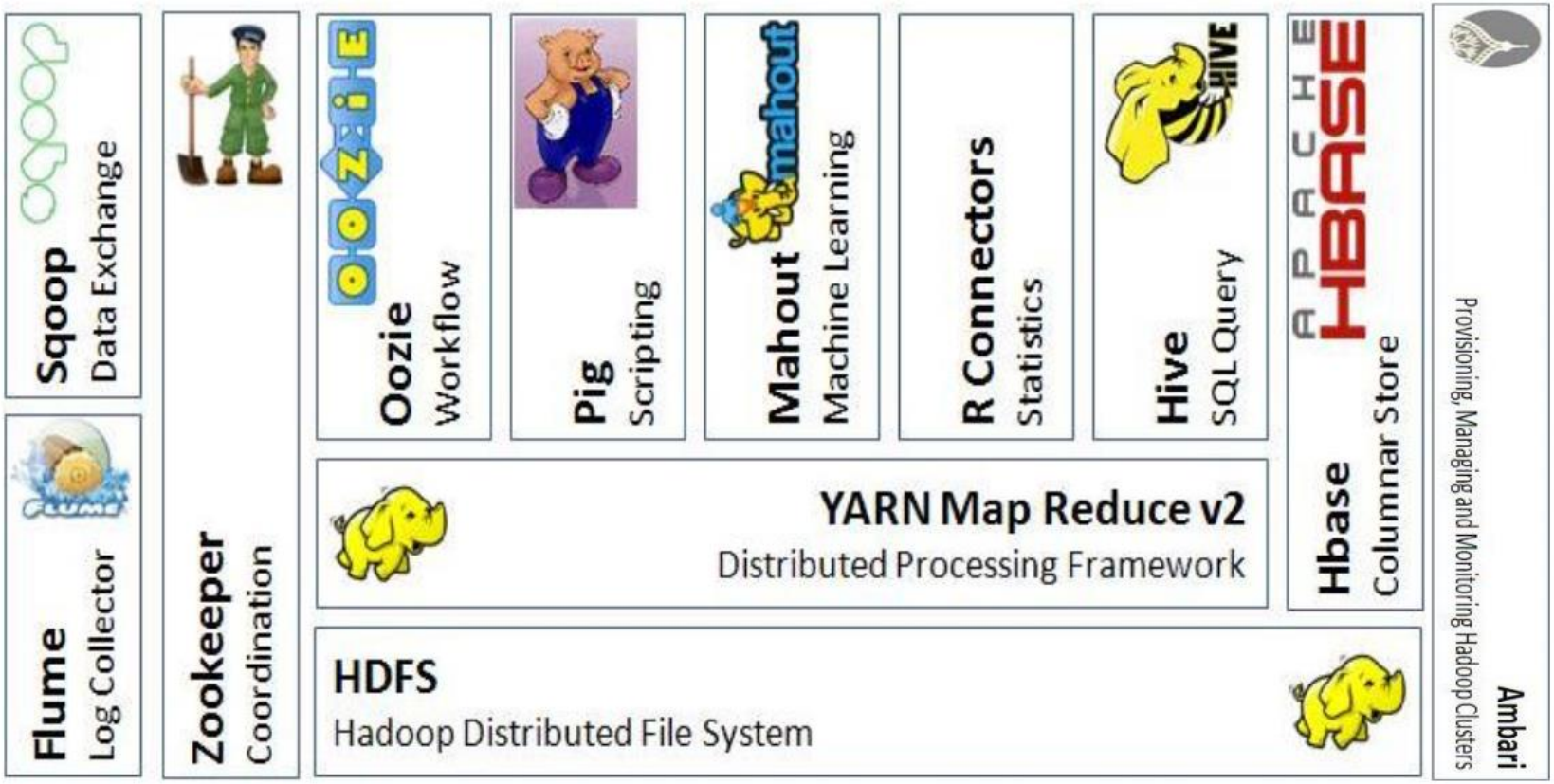
WHAT THE EXPERTS ARE SAYING

"The growing use of Apache Hadoop is fostering structured data growth, leading enterprises to understand how to reuse, repurpose and gain critical insight from data." - *Gartner*

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

The Hadoop ecosystem



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

The Hadoop ecosystem

- HDFS - rozproszony system plików
- Hadoop - framework programistyczny MapReduce
- HBase - Rozproszony, skalowalny, duży magazyn danych.
- Hive - hurtownia danych na szczycie Hadoop
- Pig - Platforma wysokiego poziomu dla Apache Hadoop
- Yarn - System zarządzania zasobami i harmonogramowania zadań
- Zookeeper - Rozproszona usługa koordynacyjna
- Flume - Rozproszona, niezawodna i dostępna usługa do efektywnego zbierania, agregowania i przenoszenia dużych ilości danych z dzienników.
- Oozie - system planowania przepływu pracy do zarządzania zadaniami Hadoop
- Mahout - rozproszony framework do uczenia maszynowego

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

HDFS & Hadoop

- HDFS
- Rozproszony system plików
- Serwery mogą ulec awarii, ale nie przerywają procesu obliczeniowego.
- Dane są replikowane z redundancją w całym klastrze.
- Hadoop/MapReduce
- Paradygmat programowania służący do wyrażania obliczeń rozproszonych na wielu serwerach.
- Siła napędowa dużej części dzisiejszego przetwarzania dużych danych.
- Stosowany również w innych środowiskach MPP i bazach danych NoSQL (np. Vertica i MongoDB)
- Ulepszona programowalność: Pig i Hive
- Poprawa dostępu do danych: HBase, Sqoop i Flume

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Co to jest MapReduce?

- Model programowania dla wyrażania rozproszonych obliczeń w masowej skali
- ramy wykonawcze do organizowania i wykonywania takich obliczeń
- równoległe wykonywanie różnych zadań,
- zapewniający redundancję i odporność na błędy.
- Inspirowany funkcjami map i reduce powszechnie stosowanymi w programowaniu funkcjonalnym
- Różne implementacje:
- Google,
- Hadoop,
- ...
- Google uzyskało patent na MapReduce.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Masowa skala: ile danych?

- Globalny wolumen danych w 2012 roku osiągnął:
- 3 Zettabajty ...
- 3 000 Exabajtów ...
- 3 000 000 Petabajtów ...
- 3.000.000.000 Terabajtów ...
- 3,000,000,000,000,000,000,000 bytes!
- Google przetwarza ponad 100 PB dziennie z 3M serwerów
- Facebook ma 300 PB danych użytkowników + 500 TB/dzień
- YouTube 1000 PB pamięci wideo
- 1.4B kart kredytowych w USA, 20B transakcji/rok
- Wielki Zderzacz Hadronów w CERN będzie generował 15 PB rocznie

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Przechowywanie i przetwarzanie danych

- Problem: Na przestrzeni lat pojemność pamięci masowej ogromnie wzrosła, ale szybkość dostępu nie nadąża

year: 1990
size: ~1.3GB
speed: 4.4 MB/s

5 mins

year: 2015
size: ~1TB
speed: 600 MB/s

30 minutes

year: 2014
size: ~1TB
speed: 100 MB/s

3 hours

Ronald Coase

If you torture the data long enough, it will always confess

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Dzisiejsze rozwiązanie: obliczenia klastrowe

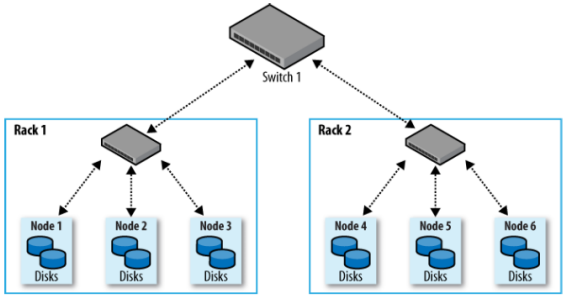
- 100 dysków twardych? 2 minuty na odczyt 1TB.
- Co w przypadku awarii dysków?
- Replikacja (RAID) ... lub
- Skalowanie w górę vs. Skalowanie w dół

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Obliczenia klastrowe

- Węzły obliczeniowe są przechowywane na stoj
- 8-64 węzłów obliczeniowych na stojaku
- Węzły obliczeniowe mogą znajdować się na wie
- Węzły w pojedynczej szafie są połączone siecią
- zazwyczaj gigabitowy Ethernet.
- Regały są połączone za pomocą innego poziomu sieci lub przełącznika
- Szerokość pasma komunikacji wewnątrz szafy jest zwykle znacznie większa niż komunikacji między
- Węzły obliczeniowe mogą ulec awarii! Roz
- Pliki są przechowywane redundantnie
- Obliczenia są podzielone na zadania



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Okucia towarowe

- Nie jesteś przywiązany do drogich, zastrzeżonych ofert jednego producenta
- Do budowy klastra można wybrać standardowy, powszechnie dostępny sprzęt od dowolnego z wielu dostawców.
- Towarowość ≠ Low-end!
- tanie komponenty o wysokiej awaryjności mogą być złudną oszczędnością
- ale drogie maszyny klasy bazodanowej nie wypadają dobrze na krzywej cena/wydajność

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Okucia towarowe (2)

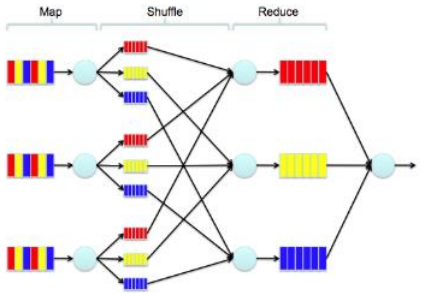
- Przykładowa typowa specyfikacja sprzętu commodity:
- Procesor 2 procesory czterordzeniowe 2-2,5GHz
- Pamięć 16-24 GB ECC RAM
- Pamięć masowa 4 × 1TB dyski SATA
- Sieć Gigabit Ethernet
- Yahoo! posiada ogromną instalację Hadoop:
- > 100 000 procesorów w > 40 000 komputerów
- Używany do wspierania badań nad systemami reklamowymi i wyszukiwaniem w sieci.
- Używany również do testów skalowania w celu wsparcia rozwoju Hadoop.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Nowy stos oprogramowania

- Nowe środowiska programistyczne zaprojektowane tak, aby ich równoległość nie pochodziła z superkomputera, ale z klastrów obliczeniowych
- Spód stosu: rozproszony system plików (DFS)
- Na szczycie DFS
- wiele różnych systemów programowania wysokiego poziomu
- DFS
- Mamy zwycięzcę: Hadoop
- System programowania
- Oryginalny: MapReduce



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

DFS: założenia

- Towarowy sprzęt
- Skalowanie "na zewnątrz", nie "w górę"
- Znaczące wskaźniki awaryjności
- Węzły mogą ulec awarii w czasie
- "Umiarkowana" liczba ogromnych plików
- Wielogigabajtowe pliki są powszechne, jeśli nie zachęcane.
- Pliki są zapisywane tylko raz, w większości dołączane do nich
- Być może współbieżnie
- Duże odczyty strumieniowe przy dostępie losowym
- Wysoka stała przepustowość przy niskich opóźnieniach

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

DFS: organizacja

- Pliki są dzielone na kawałki o wielkości zazwyczaj 64 megabajtów.
- Kawałki są replikowane w różnych węzłach obliczeniowych (zazwyczaj 3+)
- Węzły posiadające kopie jednego chunka znajdują się na różnych półkach.
- Rozmiar kawałków i stopień replikacji może być ustalony przez użytkownika
- Specjalny plik (węzeł główny) przechowuje, dla każdego pliku, pozycje jego kawałków.
- Węzeł główny jest sam w sobie replikowany
- Katalog dla systemu plików wie, gdzie znaleźć węzeł główny.
- Sam katalog może być replikowany.
- Uczestnicy korzystający z DFS wiedzą, gdzie są kopie katalogu.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Implementacje DFS

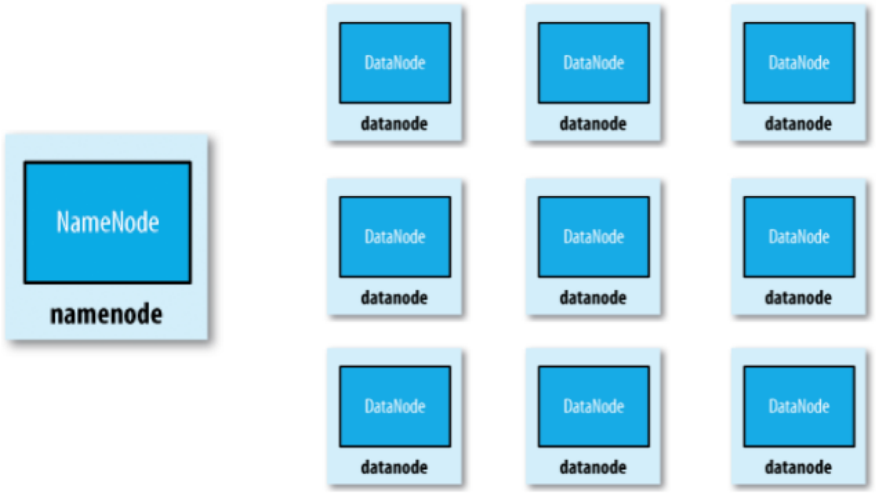
- Kilka rozproszonych systemów plików stosowanych w praktyce.
- Wśród nich:
- Google File System (GFS), pierwowzór tej klasy. Ostatnia wersja Google File System o nazwie kodowej Colossus została wydana w 2010 roku.
- Spectrum Scale - dawniej GPFS - General Parallel File System firmy IBM.
- CloudStore - open-source'owy DFS opracowany pierwotnie przez firmę Kosmix.
- HDF5 (Hierarchiczny Format Danych)
- S3 (Amazon EC2)
- Hadoop Distributed File System (HDFS), open-source'owy DFS używany z Hadoopem

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

HDFS concepcja

- HDFS ma architekturę master/slave
- Klaster HDFS posiada dwa rodzaje węzłów:
- NameNode: serwer główny, który zarządza przestrzenią nazw systemu plików i reguluje dostęp do plików przez klientów.
- Wiele DataNodes: zazwyczaj jeden na węzeł w klastrze, które zarządzają pamięcią masową dołączoną do węzłów, na których są uruchomione



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

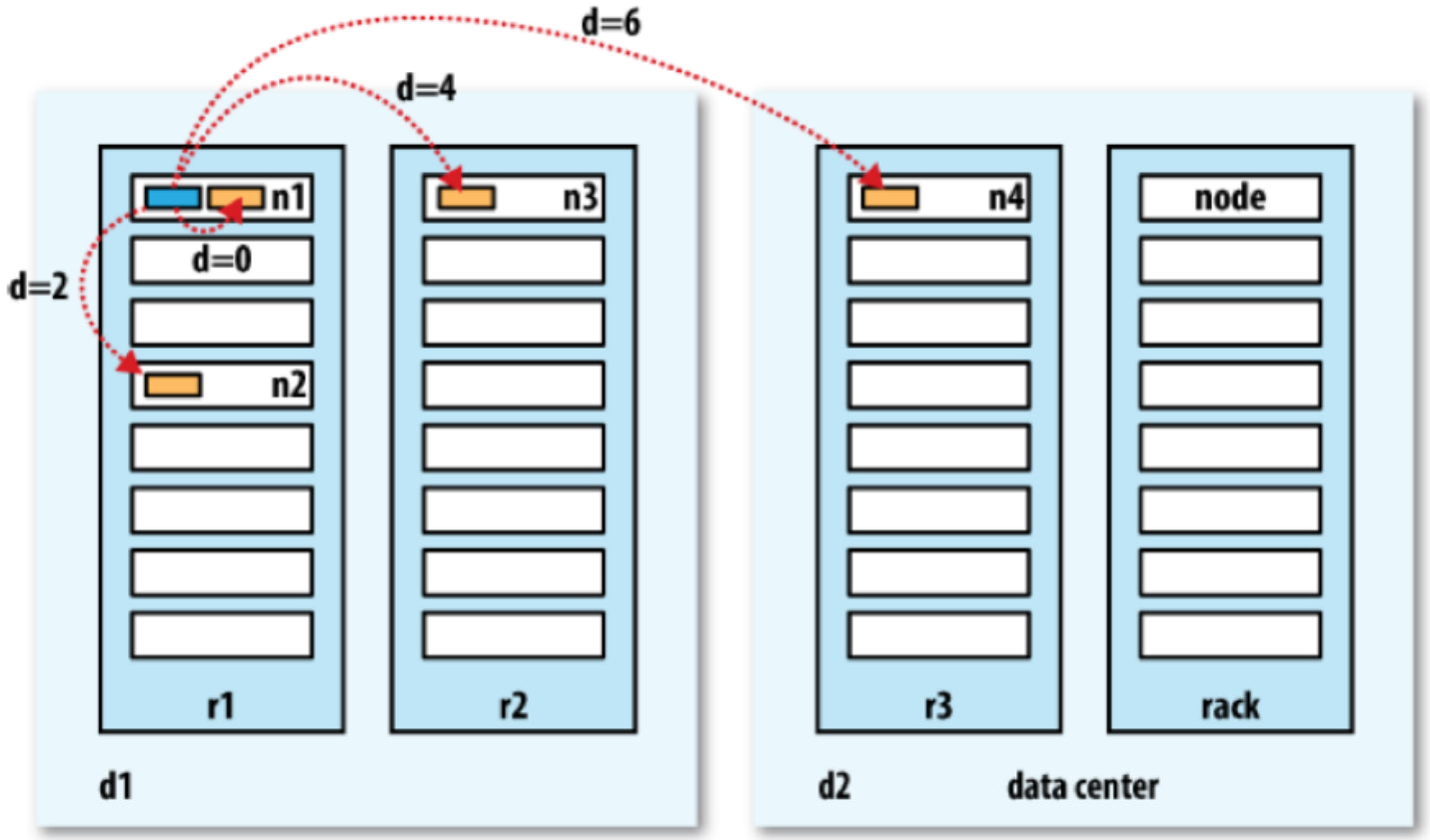
HDFS concepcja

- Węzeł nazw
- Zarządza drzewem systemu plików i metadanymi dla wszystkich plików i katalogów.
- Zna węzły danych, na których znajdują się wszystkie bloki dla danego pliku
- Węzły danych
- Po prostu przechowują i pobierają bloki kiedy są do tego upoważnione (przez klientów lub NameNode). Wewnętrznie, plik jest podzielony na jeden lub więcej bloków i te bloki są przechowywane w zestawie DataNodes .
- Bez węzła nazw HDFS nie może być używany.
- Maszyna NameNode jest pojedynczym punktem awarii dla klastra HDFS.
- Ważne jest, aby NameNode był odporny na awarie

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

HDFS - organizacja fizyczna



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

HDFS I/O

- Klient aplikacji, który chce odczytać plik (lub jego część) musi najpierw skontaktować się z węzłem NameNode, aby ustalić, gdzie są przechowywane dane.
- W odpowiedzi na żądanie klienta NameNode zwraca
- identyfikatory odpowiednich bloków
- lokalizację, w której przechowywane są bloki (tj. które DataNodes).
- Klient następnie kontaktuje się z DataNodes, aby pobrać dane.
- Bloki są przechowywane na standardowych, jednomaszynowych systemach plików
- HDFS leży na wierzchu standardowego stosu OS
- Ważna cecha projektu
- dane nigdy nie są przenoszone przez NameNode
- cały transfer danych odbywa się bezpośrednio pomiędzy klientami a DataNodes
- komunikacja z NameNode obejmuje jedynie transfer metadanych

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

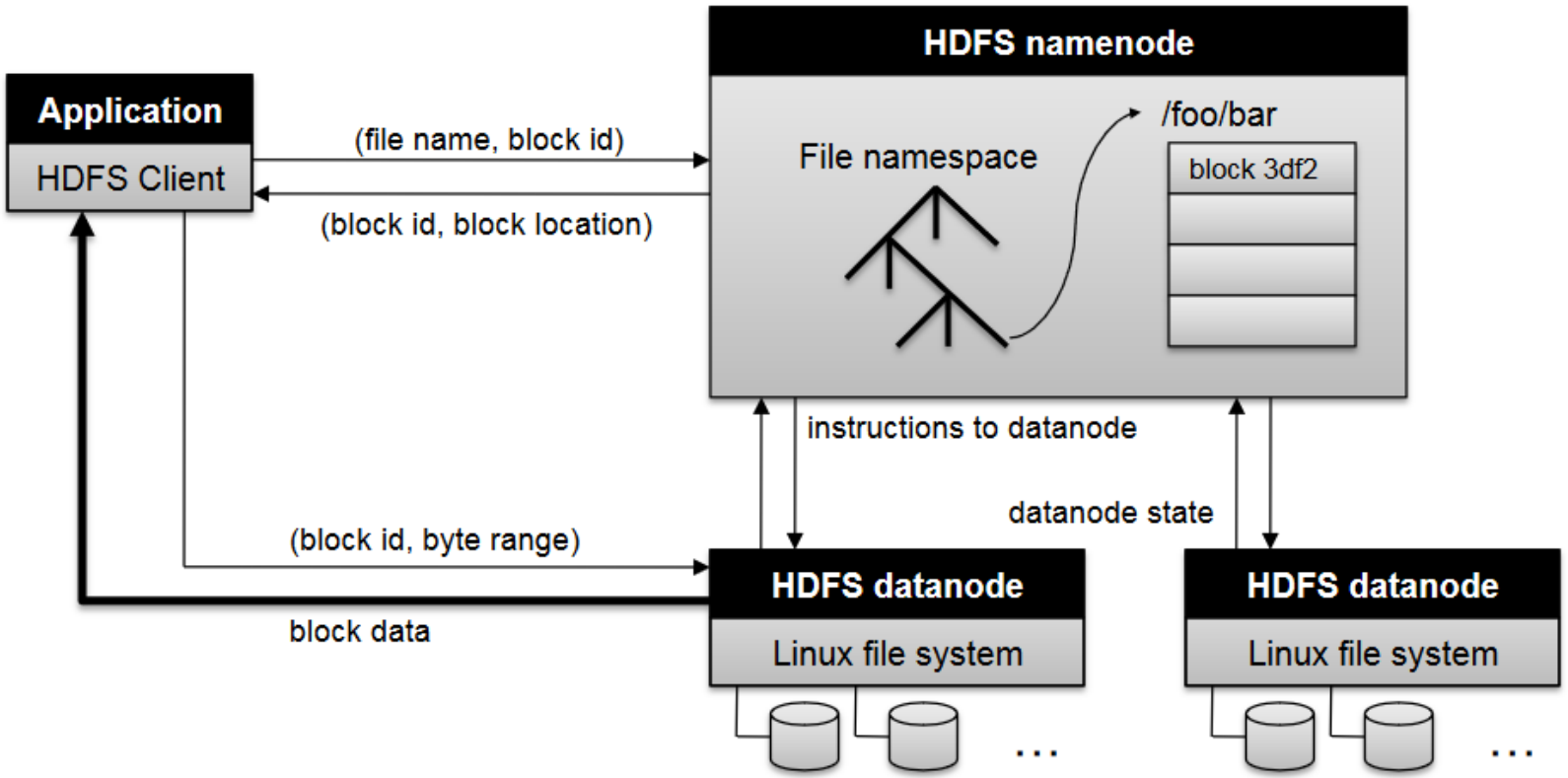
NameNode obowiązki

- Zarządzanie przestrzenią nazw systemu plików
- Przechowuje strukturę plików/katalogów, metadane, mapowanie plików na bloki, uprawnienia dostępu itp.
- Koordynacja operacji na plikach
- Kieruje klientów do DataNodes w celu odczytu i zapisu.
- Żadne dane nie są przenoszone przez NameNode
- Utrzymanie ogólnej kondycji
- Okresowa komunikacja z węzłami danych
- Replikacja i równoważenie bloków
- Zbieranie śmieci
-

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

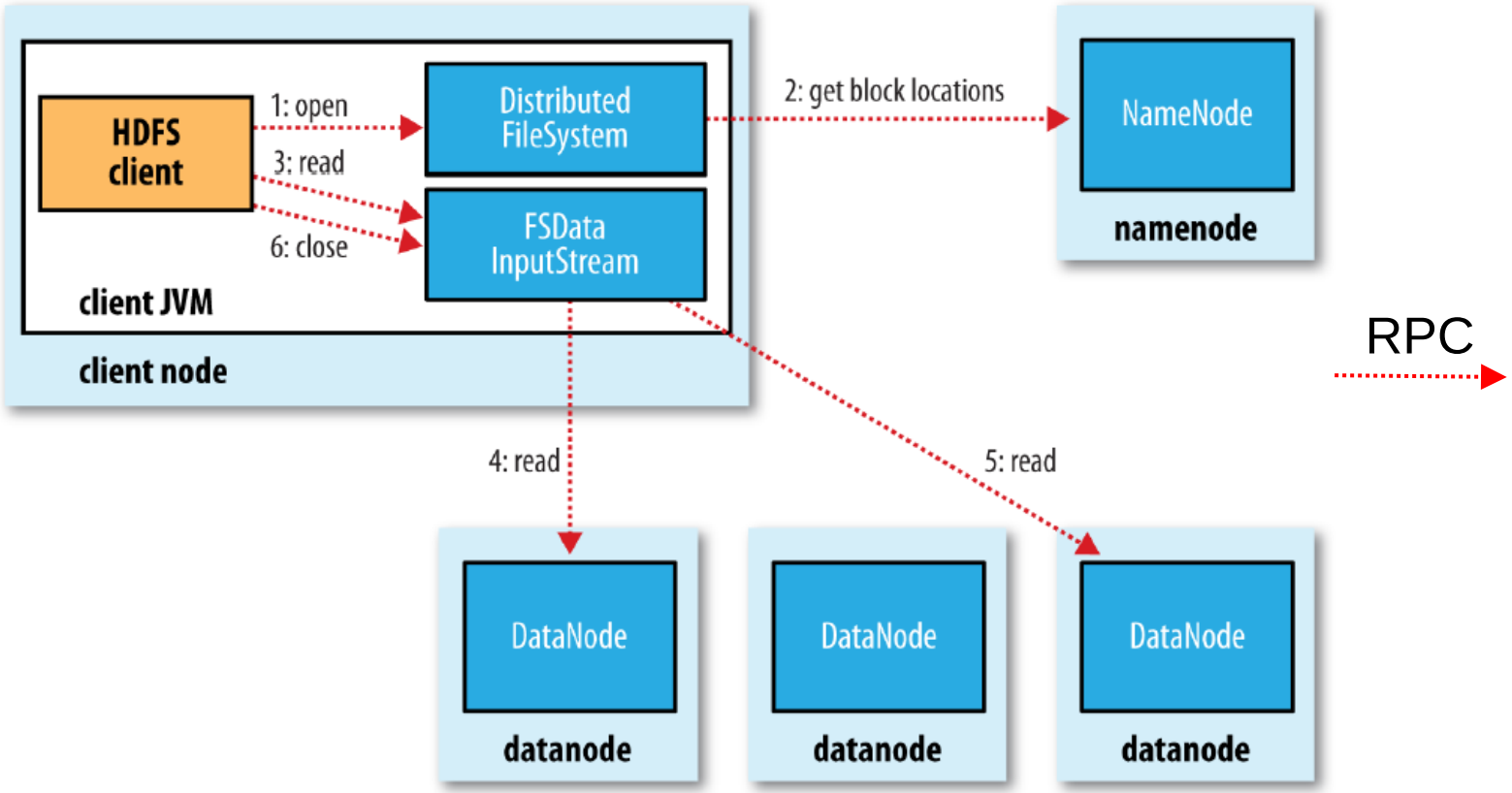
HDFS Architektura



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

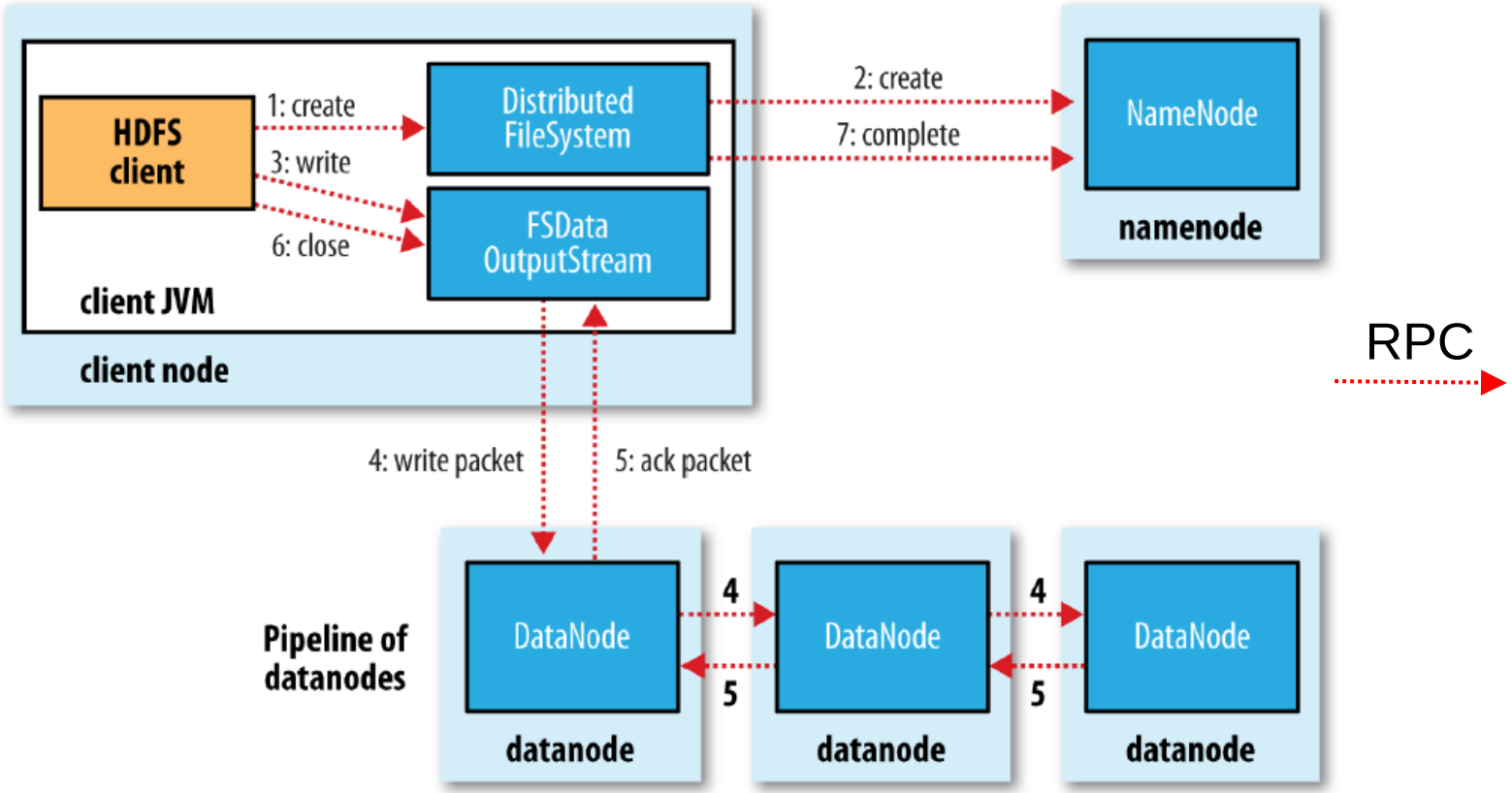
Odczyt danych z HDFS



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Zapisywanie danych do HDFS



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

MapReduce - obliczenia rozproszone na dużą skalę

- Ukryć szczegóły poziomu systemu przed programistami
- Programy współbieżne są trudne do zrozumienia i trudniejsze do debugowania; nie ma już warunków wyścigu, kontinuum blokady itp.
- Przenieś przetwarzanie na dane
- Klastry mają ograniczoną przepustowość
- Przetwarzaj dane sekwencyjnie, unikaj losowego dostępu
- Poszukiwania są kosztowne, przepustowość dysku jest rozsądna
- Oddzielenie co od jak
- Programista określa obliczenia, które muszą być wykonane
- Ramy wykonawcze ("runtime") zajmują się rzeczywistym wykonaniem.
- Skalowanie "na zewnątrz", nie "w górę" - Bezproblemowa skalowalność
- Centrum danych to komputer!

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Typowy problem z dużymi danymi

- Iteracja nad dużą liczbą rekordów równolegle
- Wyciągnij coś interesującego z każdej iteracji

Map

- Tasowanie i sortowanie wyników pośrednich różnych współbieżnych iteracji
- Agreguj wyniki pośrednie
- Generowanie ostatecznego wynik

Reduce

- Kluczowa idea: dostarczyć funkcjonalną abstrakcję dla tych dwóch operacji, Map i Reduce
- Korzenie w programowaniu funkcyjnym

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

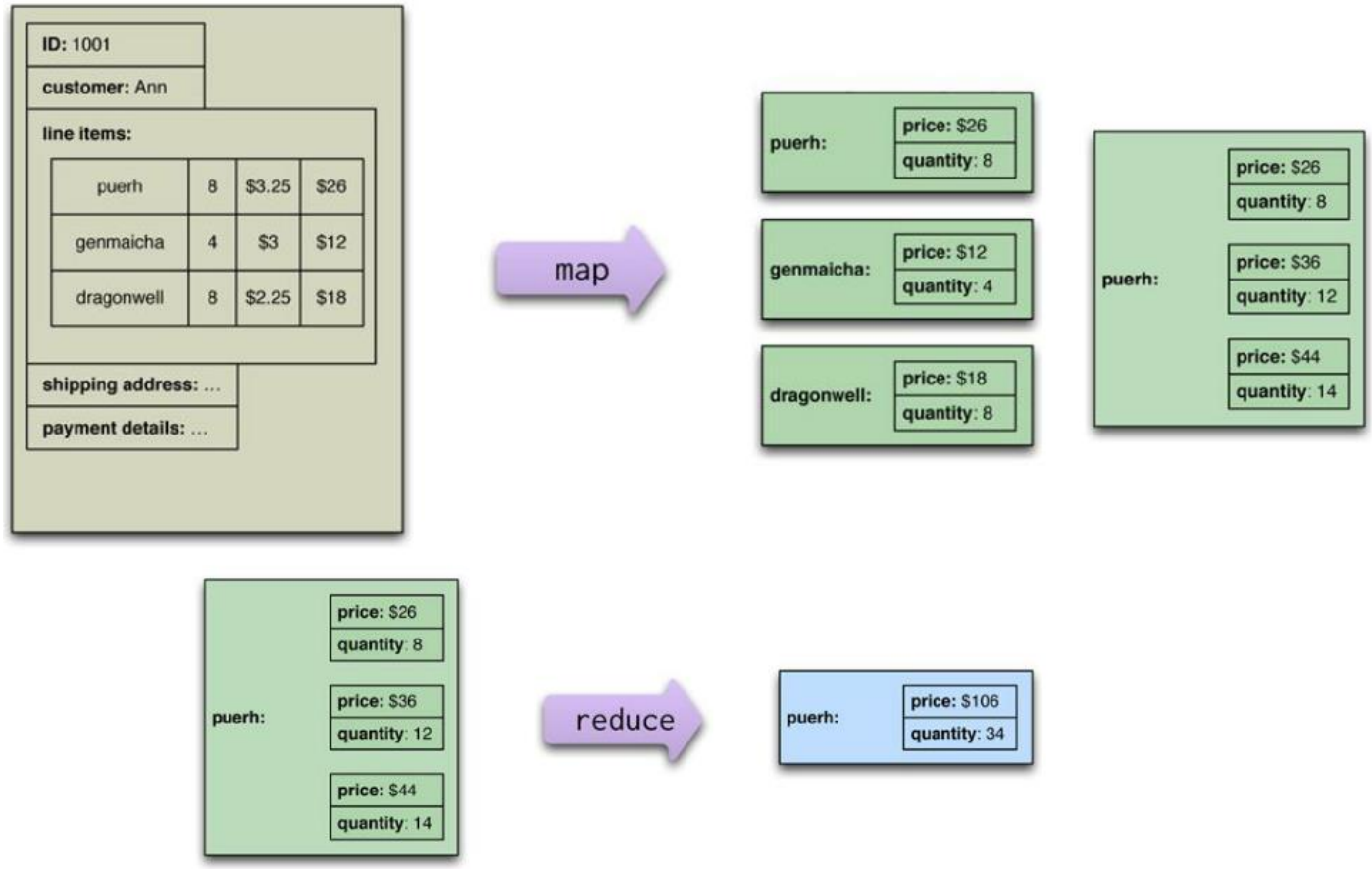
Model programowania MapReduce

- Map: przyjmuje jako dane wejściowe obiekt o kluczu (k,v) i zwraca zestaw par klucz-wartość: (k1,v1),(k2,v2),...,(kn,vn)
- Framework zbiera wszystkie pary z tym samym kluczem k i kojarzy z k wszystkie wartości dla k: (k,[v1,...,vn])
- Reduce: przyjmuje jako dane wejściowe klucz i listę wartości (k,[v1,...,vn]) i łączy je w jakiś sposób.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Przykład



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

Programowanie MapReduce

- Podstawowa struktura danych: pary klucz-wartość
- Programiści określają dwie funkcje:
- map (k1, v1) → [(k2, v2)]
- reduce (k1, [v1]) → [(k2, v2)]
- (k, v) oznacza parę (klucz, wartość)
- [...] oznacza listę
- zwykle klucze elementów wejściowych nie są istotne
- klucze nie muszą być unikalne: różne pary mogą mieć ten sam klucz
- Szkielet wykonawczy zajmuje się wszystkim innym!

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Program MapReduce

- Program MapReduce, określany jako zadanie, składa się z:
- kodu dla map i kodu dla reduce spakowanych razem
- parametrów konfiguracyjnych (gdzie leżą dane wejściowe, gdzie powinny być przechowywane dane wyjściowe)
- dane wejściowe, przechowywane w bazowym rozproszonym systemie plików
- Każde zadanie MapReduce jest dzielone przez system na mniejsze jednostki zwane zadaniami
- Zadania mapowania
- Zadania redukcyjne
- Wejście i wyjście zadań MapReduce są przechowywane na bazowym rozproszonym systemie plików

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Proces wykonania MapReduce

1) Pewna liczba zadań Map otrzymuje po jednym lub więcej kawałków danych.
2) Każde zadanie Map przekształca fragment w sekwencję par klucz-wartość.
3) Sposób tworzenia par klucz-wartość jest określony przez kod napisany przez użytkownika dla funkcji map.
4) Pary klucz-wartość z każdego zadania Map są zbierane przez kontroler główny oraz sortowane i grupowane według kluczy (Shuffle and sort).
5) Klucze są dzielone między wszystkie zadania Reduce, więc wszystkie pary klucz-wartość z tym samym kluczem trafiają do tego samego zadania Reduce.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

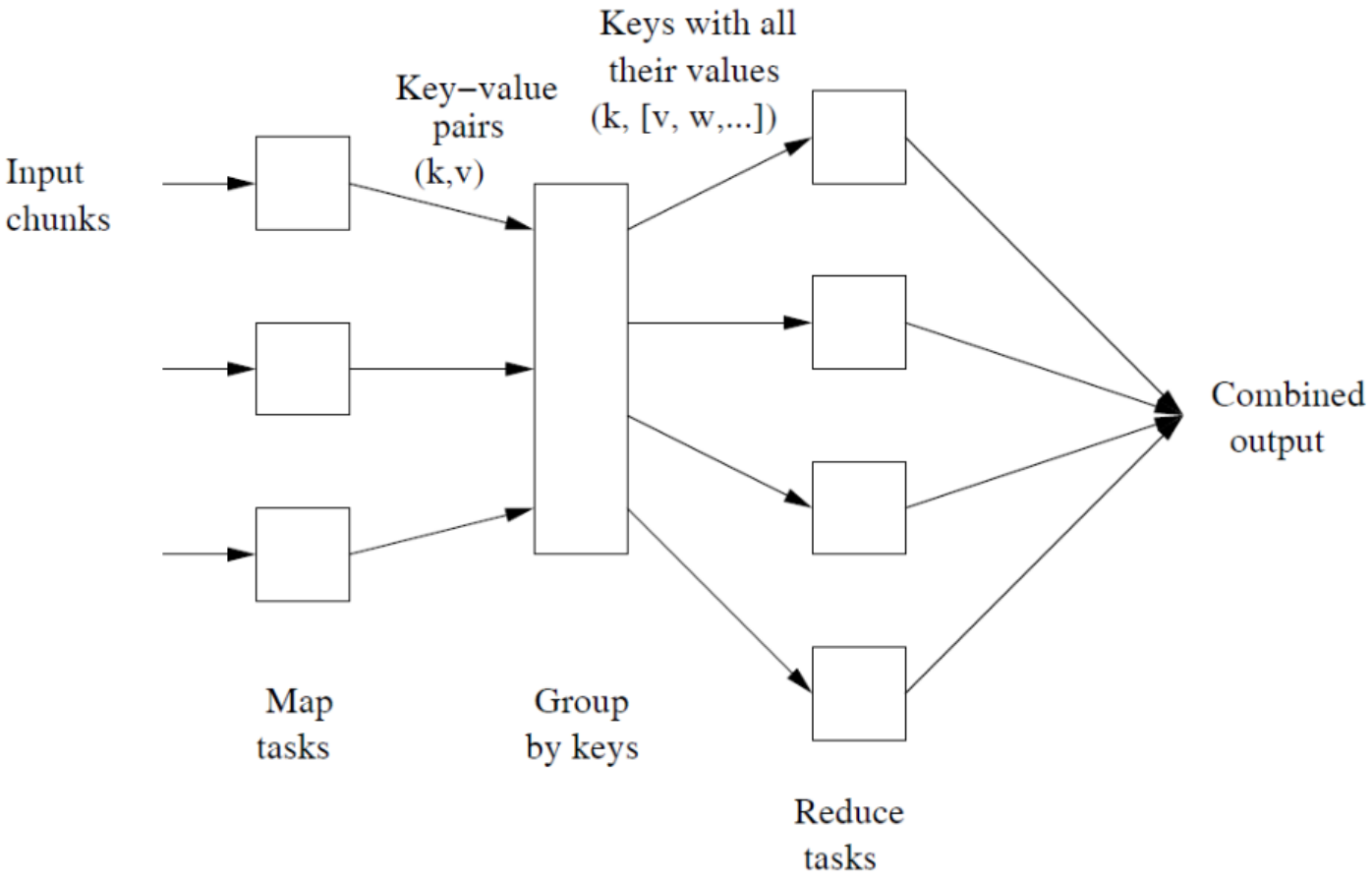
Proces wykonania MapReduce (2)

5) Zadania Reduce pracują na jednym kluczu w tym samym czasie i łączą wszystkie wartości związane z tym kluczem w jakiś sposób.
6) Zadania redukcyjne pracują na jednym kluczu w tym samym czasie i łączą wszystkie wartości związane z tym kluczem w jakiś sposób.
7) Wyjściowe pary klucz-wartość z każdego reduktora są trwale zapisywane z powrotem w rozproszonym systemie plików.
8) Wyjście kończy się w plikach r, gdzie r jest liczbą reduktorów.
9) Pliki r często służą jako wejście do kolejnego zadania MapReduce

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

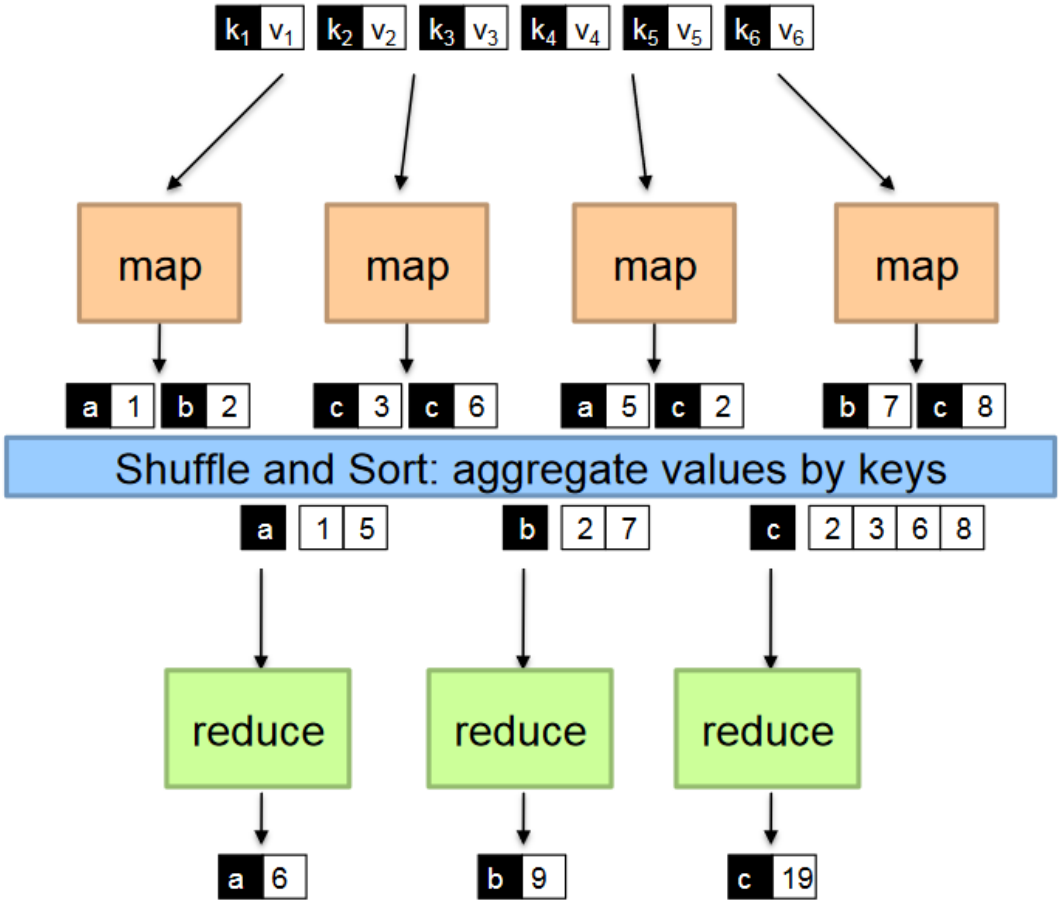
Proces MapReduce



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

Przykład ogólny



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

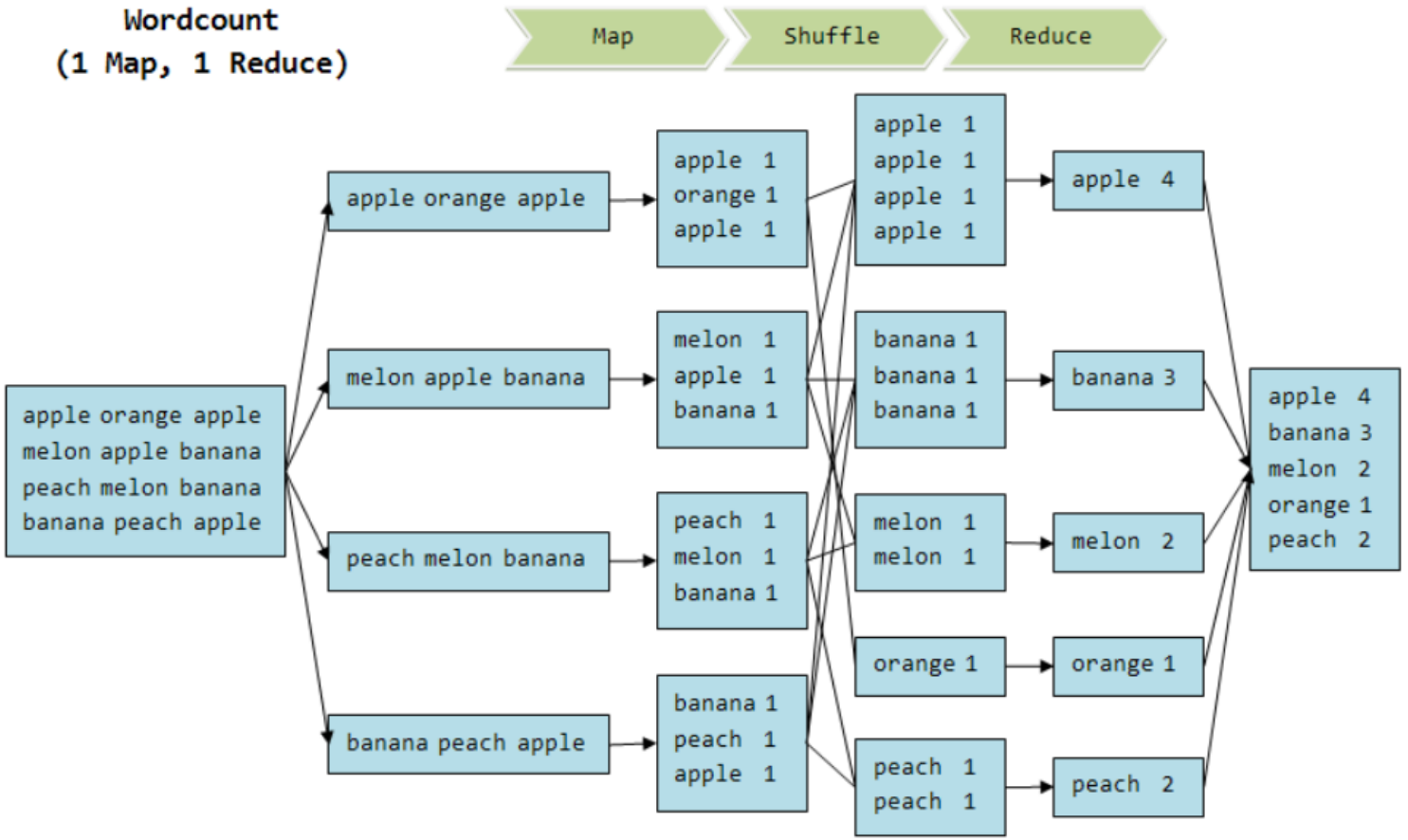
Praktyczny przykład: Word Count

- Problem: policzenie liczby wystąpień każdego słowa w zbiorze dokumentów.
- Wejście: repozytorium dokumentów, każdy dokument jest elementem
- Mapa: wczytuje dokument i wysyła ciąg par klucz-wartość, gdzie kluczami są słowa w dokumentach, a wartości są równe 1:
- (w1, 1), (w2, 1), . . , (wn, 1)
- Shuffle: grupuje po kluczach i generuje pary w postaci
- (w1, [1, 1, . . . , 1]) , . . . , (wn, [1, 1, ... , 1])
- Reduce: sumuje wszystkie wartości i emituje:
- (w1, k) , . . , (wn, l)
- Wyjście: (w, m) pary, gdzie w to słowo, które pojawia się co najmniej raz wśród wszystkich dokumentów wejściowych, a m to całkowita liczba wystąpień w wśród wszystkich tych dokumentów.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Liczba słów w praktyce



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

WordCount zastosowanie

```
Map(String docid, String text):
    for each word w in text:
        Emit(w, 1);

Reduce(String term, counts[]):
    int sum = 0;
    for each c in counts:
       sum += c;
    Emit(term, sum);
```


iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

WorldCount in Java

```
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
public class WordCount {
```


iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

WorldCount in Java (2)

```
// Map function
public static class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
     private Text word = new Text();
     public void map(LongWritable key, Text value, Context context)
             throws IOException, InterruptedException {
         // Splitting the line on spaces
         String[] stringArr = value.toString().split("\\s+");
         for (String str : stringArr) {
             word.set(str);
             context.write(word, new IntWritable(1));
         }
     }
}
```


iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

WorldCount in Java (3)

```
// Reduce function
public static class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable>{
    private IntWritable result = new IntWritable();
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
}
```


iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

WorldCount in Java (4)

```
    public static void main(String[] args)  throws Exception{
        Configuration conf = new Configuration();

        Job job = Job.getInstance(conf, "WC");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(MyMapper.class);
        job.setReducerClass(MyReducer.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```


iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

Przykład 2: Liczenie długości słowa

- Problem: policzenie ile słów o określonej długości występuje w zbiorze dokumentów.
- Wejście: repozytorium dokumentów, każdy dokument jest elementem
- Mapa: odczytuje dokument i emituje ciąg par klucz-wartość, gdzie kluczem jest długość słowa, a wartością samo słowo:
- (i,w1), . . , (j,wn)
- Tasowanie i sortowanie: grupuje według klucza i generuje pary w postaci
- (1, [w1, . . . , wk]) , . . , (n, [wr, ... , ws])
- Reduce: liczy liczbę słów w każdej liście i emituje:
- (1, l) , . . . , (p, m)
- Wyjście: (l, n) pary, gdzie l to długość, a n to całkowita liczba słów o długości l w dokumentach wejściowych.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

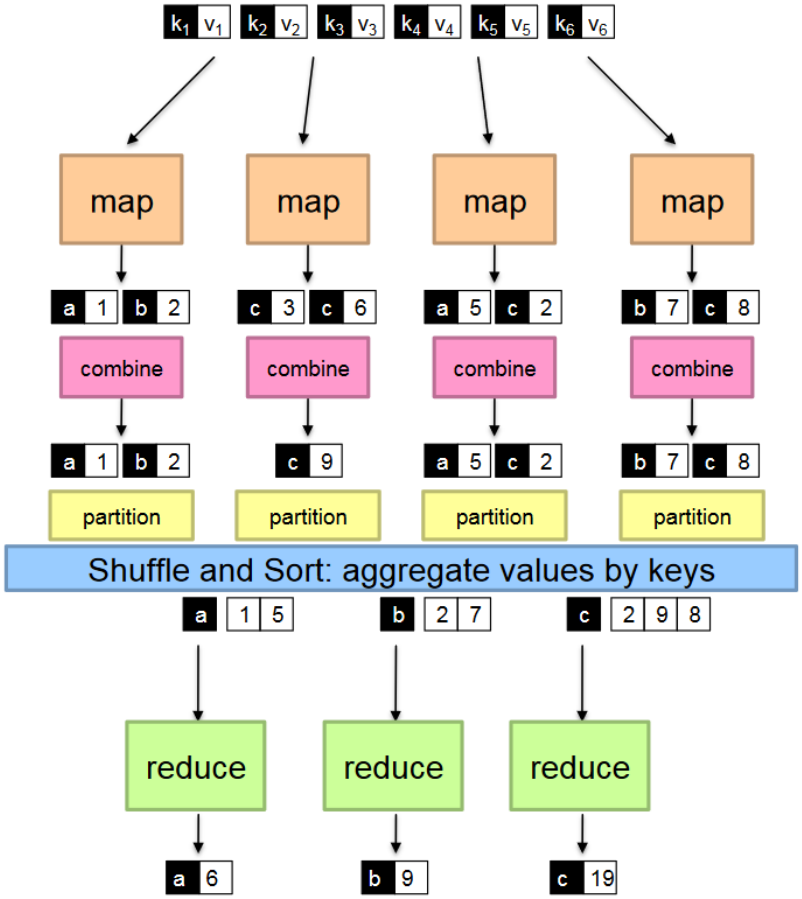
Wprowadzenie Kombinatoryki

- Kiedy funkcja redukcji jest asocjacyjna i komutacyjna, możemy przesunąć część tego, co robią reduktory do zadań mapy
- W tym przypadku stosujemy również sumator do funkcji mapy
- W wielu przypadkach ta sama funkcja może być użyta do łączenia jako ostateczna redukcja
- Tasowanie i sortowanie jest nadal konieczne!
- Zalety:
- zmniejsza ilość danych pośrednich
- zmniejsza ruch w sieci

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

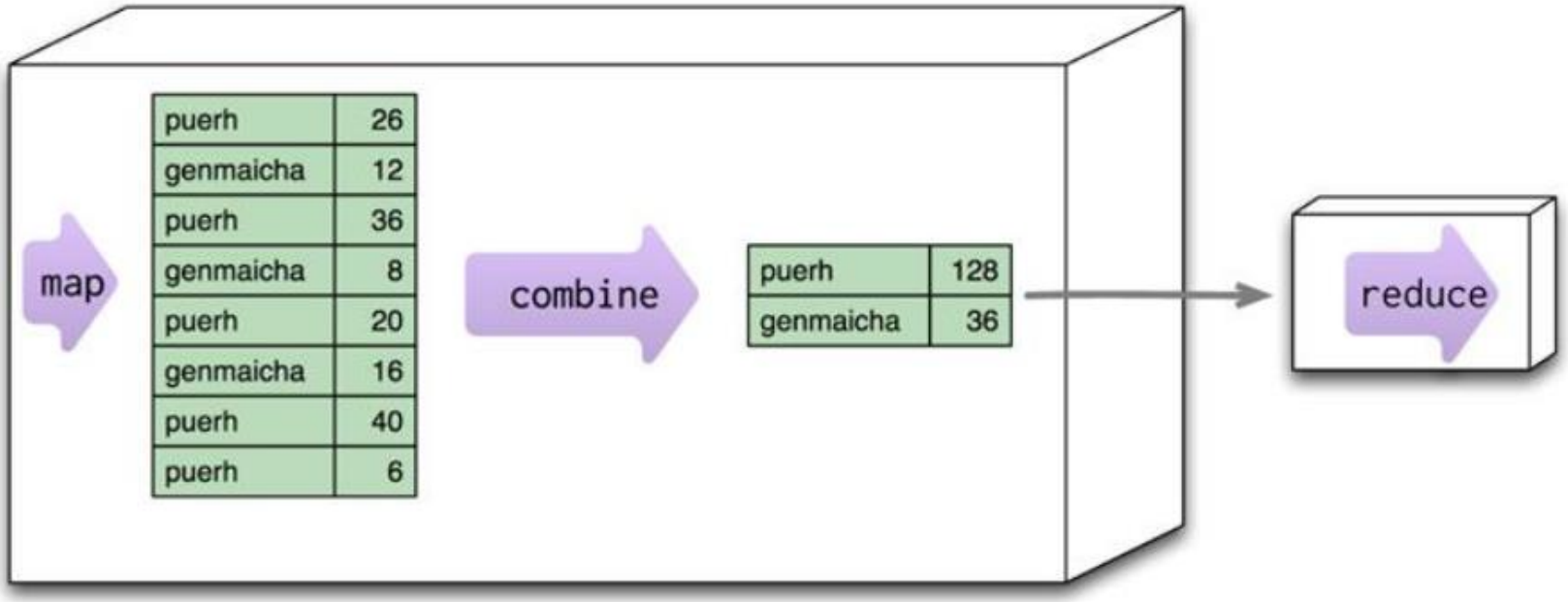
Sumator



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

Przykład kombinatora



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

Liczba słów z kombinatorami

- Wejście: repozytorium dokumentów, każdy dokument jest elementem
- Mapa: czyta dokument i emituje ciąg par klucz-wartość, gdzie kluczami są słowa w dokumentach, a wartości są równe 1:
- (w1, 1), . . . , (wn, 1)
- Combiner: grupuje po kluczach, sumuje wszystkie wartości i emituje:
- (w1, i), . . . , (wn, j)
- Tasowanie i sortowanie: grupuje według klucza i generuje pary w postaci
- (w1, [p, ... , q]) , ... , (wn, [r, ... , s])
- Reduce: sumuje wszystkie wartości i emituje:
- (w1, k) , . . , (wn, l)
- Wyjście: (w,m) pary, gdzie w to słowo, które pojawia się co najmniej raz wśród wszystkich dokumentów wejściowych, a m to całkowita liczba wystąpień w wśród wszystkich tych dokumentów.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Ile map i redukcji?

- Może być ustawiony w pliku konfiguracyjnym lub podczas wykonywania polecenia (dla map jest to tylko wskazówka)
- Liczba mapperów: napędzana przez liczbę bloków HDFS w plikach wejściowych. Możesz dostosować rozmiar bloku HDFS, aby dostosować liczbę mapperów.
- Liczba reduktorów: może być zdefiniowana przez użytkownika (domyślnie jest to 1)
- Przestrzeń kluczowa pośrednich par klucz-wartość jest równomiernie rozdzielana pomiędzy Reducery za pomocą funkcji haszującej (te same klucze w różnych mapperach trafiają do tego samego Reducera)
- Partycja pośrednich par klucz-wartość generowanych przez Mappery może być dostosowana do potrzeb użytkownika

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Przegrody

- Użytkownik mówi systemowi MapReduce, ile będzie zadań Reduce, powiedzmy r.
- Kontroler główny wybiera funkcję haszującą, która ma zastosowanie do kluczy i tworzy numer wiadra od 0 do r - 1.
- Każdy klucz, który jest wyprowadzany przez zadanie Mapy jest haszowany, a jego para klucz-wartość jest umieszczana w jednym z r plików lokalnych
- Pliki lokalne są zorganizowane jako sekwencja par (klucz, lista wartości)
- Każdy plik jest przeznaczony dla jednego z zadań Redukcji.
- Opcjonalnie, użytkownicy mogą określić własną funkcję haszującą lub inną metodę przypisywania kluczy do zadań Redukcji.
- Jednak niezależnie od użytego algorytmu, każdy klucz jest przypisany do jednego i tylko jednego zadania Redukcji.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

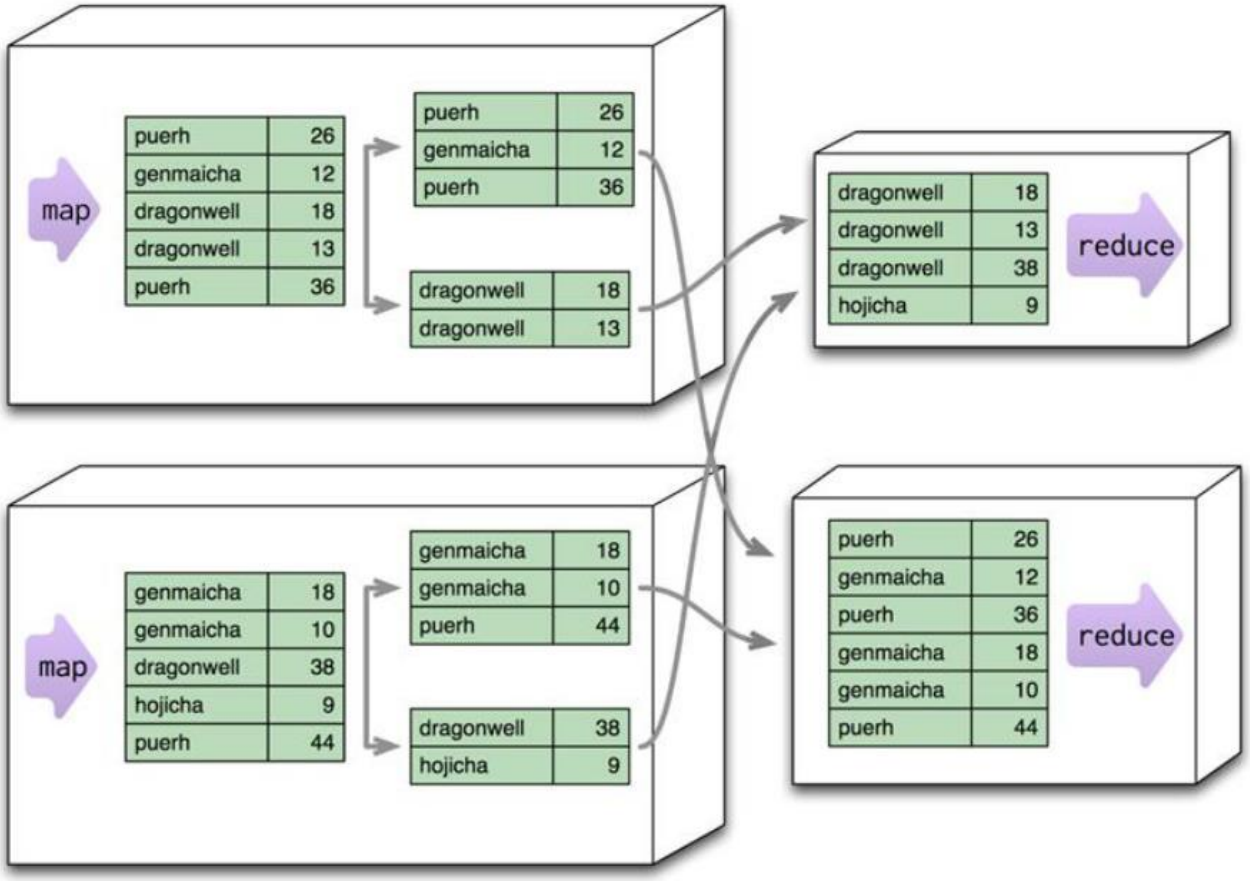
Dostosowanie przegród do potrzeb klienta

- Możemy określić partycjoner, który:
- dzieli pośrednią przestrzeń kluczy
- przypisuje pośrednie pary klucz-wartość do Reduktorów
- n partycji →n Reduktorów
- Domyślny partycjoner przypisuje w przybliżeniu taką samą liczbę kluczy do każdego Reducera.
- Ale partycjoner bierze pod uwagę tylko klucz, a ignoruje wartość
- Brak równowagi w ilości danych związanych z każdym kluczem jest stosunkowo powszechny w aplikacjach przetwarzających tekst
- W tekstach częstotliwość występowania dowolnego słowa jest odwrotnie proporcjonalna do jego rangi w tablicy częstotliwości
- Najczęstsze słowo wystąpi mniej więcej dwa razy częściej niż drugie najczęstsze słowo, trzy razy częściej niż trzecie najczęstsze słowo itd.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Przykładowe przegrody



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

MapReduce: pełny obraz

- Programiści określają dwie funkcje:
- map (k1, v1) → [(k2, v2)]
- reduce (k2, [v2]) → [(k3, v3)]
- Wszystkie wartości o tym samym kluczu są redukowane razem
- Programiści mogą również określić dwie dodatkowe funkcje:
- combine (k2, [v2]) → [(k3, v3)]
- Mini-reduktory, które uruchamiają się po fazie mapy
- Używane jako optymalizacja w celu zmniejszenia ruchu sieciowego
- partition (k2, liczba partycji) → partycja dla k2
- Dzieli przestrzeń kluczową dla równoległych operacji redukcji
- Ramy wykonawcze zajmują się wszystkim innym...

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

MapReduce runtime

- Ważną ideą stojącą za MapReduce jest oddzielenie tego, co z rozproszonego przetwarzania od tego, jak.
- Programista przesyła zadanie do węzła składającego w klastrze
- Ramy wykonawcze ("runtime") zajmują się wszystkim innym:
- w przejrzysty sposób obsługuje wszystkie aspekty wykonywania kodu rozproszonego
- na klastrach od jednego węzła do kilku tysięcy węzłów

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Zadania MapReduce "Runtime"

- Obsługuje planowanie
- Przypisuje pracowników do zadań mapowania i redukcji
- Obsługuje "dystrybucję danych"
- Dostarczanie danych do robotników na szczycie rozproszonego FS
- Obsługuje synchronizację
- Zbiera, sortuje i tasuje dane pośrednie
- Obsługa błędów i usterek
- Wykrywa awarie robotów i restartuje je
- Obsługa lokalizacji danych
- Dane i pracownicy muszą być blisko siebie
- Architektura nic nie dzielonego
- Każdy węzeł jest niezależny i samowystarczalny

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Zasady działania w czasie rzeczywistym w Hadoop

- Lokalizacja danych:
- Nie przenoś danych do pracowników... przenieś pracowników do danych!
- Uruchom robotników na węźle, który ma dane lokalnie.
- Zadania są tworzone z wejściowych podziałów w DFS
- 1 mapa na podział + N redukcji określonych przez konfigurację
- Optymalizacja
- preferuj węzły, które znajdują się na tej samej szafie w datacenter co węzeł posiadający dany blok danych - pasmo międzyszafkowe jest znacznie mniejsze niż pasmo wewnątrzszafkowe!
- Architektura współdzielenia-nic
- Architektura MapReduce oparta na współdzieleniu niczego oznacza, że zadania nie są od siebie zależne
- Programiści nie martwią się o kwestie obliczeń rozproszonych
- Wystarczy, że napiszą funkcje Map i Reduce

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

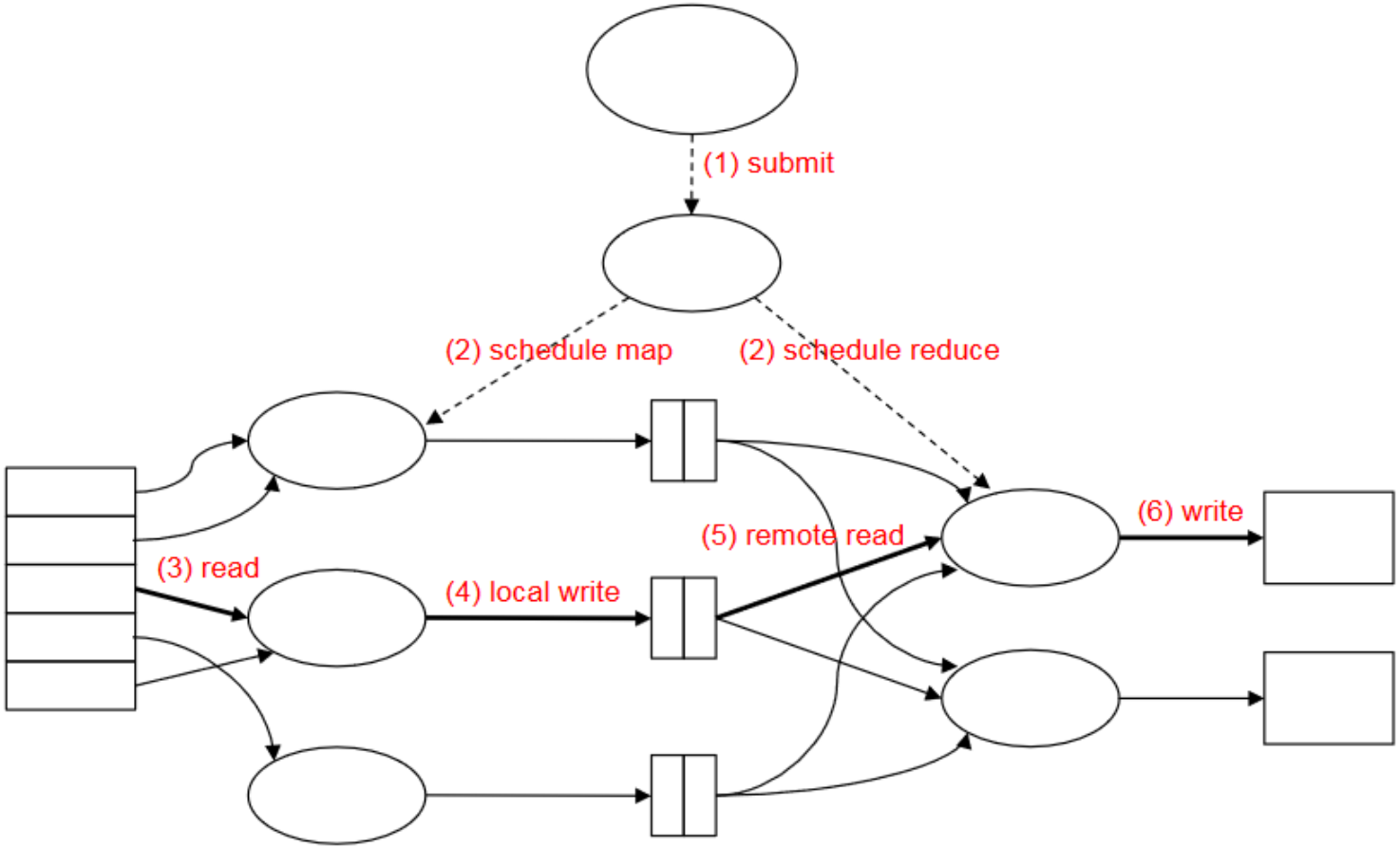
MapReduce w pracy

- Architektura master-slave
- Korzystając z biblioteki dostarczanej przez system map-reduce, taki jak Hadoop, program użytkownika generuje:
- proces kontrolera Master
- pewną liczbę procesów Worker na różnych komputerach.
- Proces Master koordynuje wszystkie zadania uruchamiane w systemie poprzez planowanie zadań
- Procesy Worker wykonują zadania i wysyłają raporty o postępie do procesu Master
- Jeśli zadanie się nie powiedzie, proces Master może je przełożyć na inny proces Workers

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

MapReduce wykonanie



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

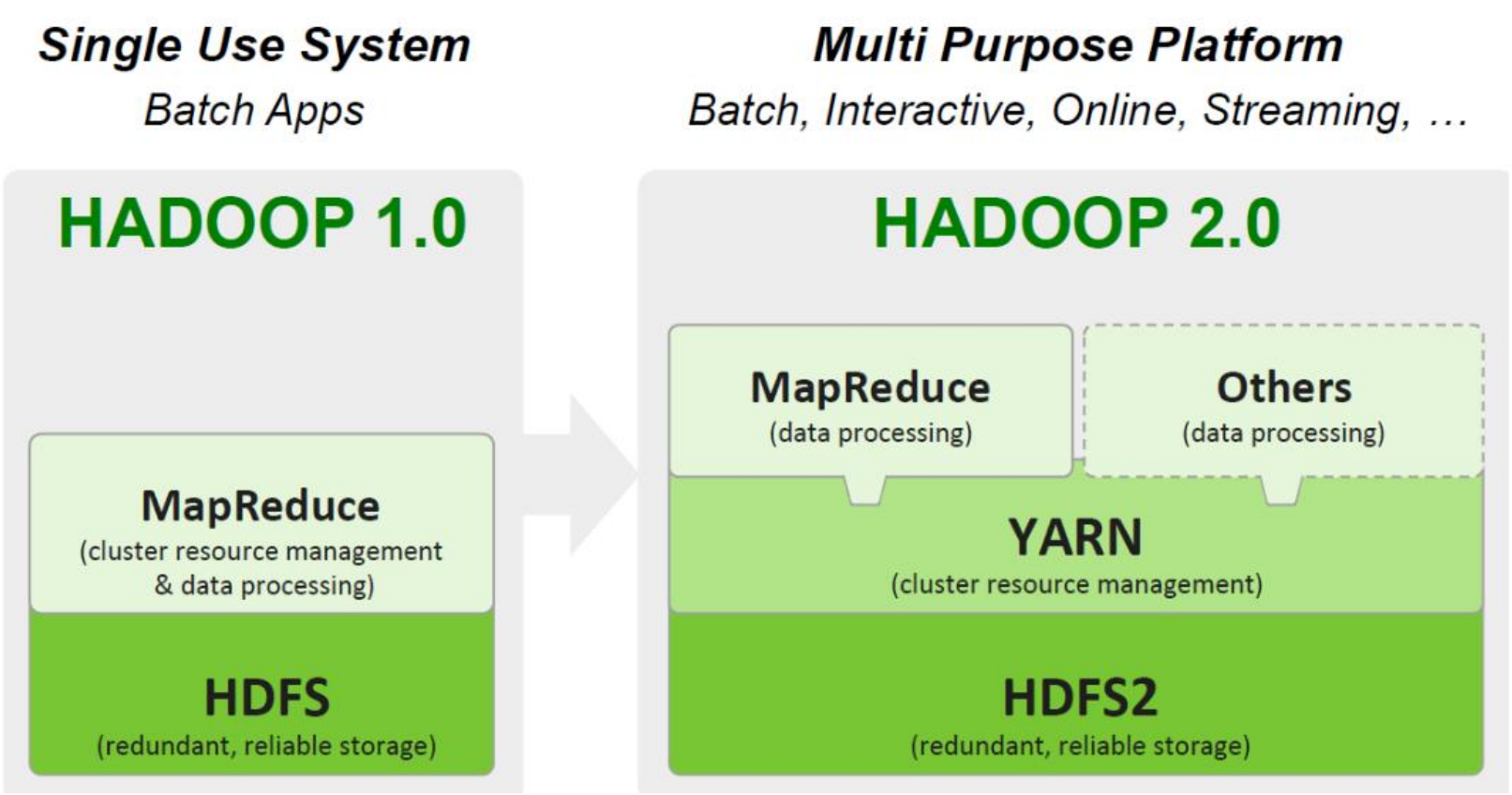
MapReduce wykonanie

- Normalnie Worker obsługuje albo zadania Map (Map worker) albo zadania Reduce (Reduce worker), ale nie oba.
- Master:
- tworzy pewną liczbę zadań Map i pewną liczbę zadań Reduce, zgodnie z wyborem programu użytkownika.
- przydziela zadania Robotnikom
- śledzi status każdego zadania Map i Reduce (bezczynność, wykonywanie w danym Workerze, ukończone).
- Zadanie Map tworzy plik dla każdego zadania Reduce na dysku lokalnym Workera, który wykonuje zadanie Map i informuje Master o lokalizacji i rozmiarach każdego z tych plików.
- Gdy zadanie Redukcji jest przydzielane przez Mistrza do Robotnika, zadanie to otrzymuje wszystkie pliki, które stanowią jego dane wejściowe.
- Zadanie Reduce zapisuje swoje dane wyjściowe do pliku w rozproszonym systemie plików.

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

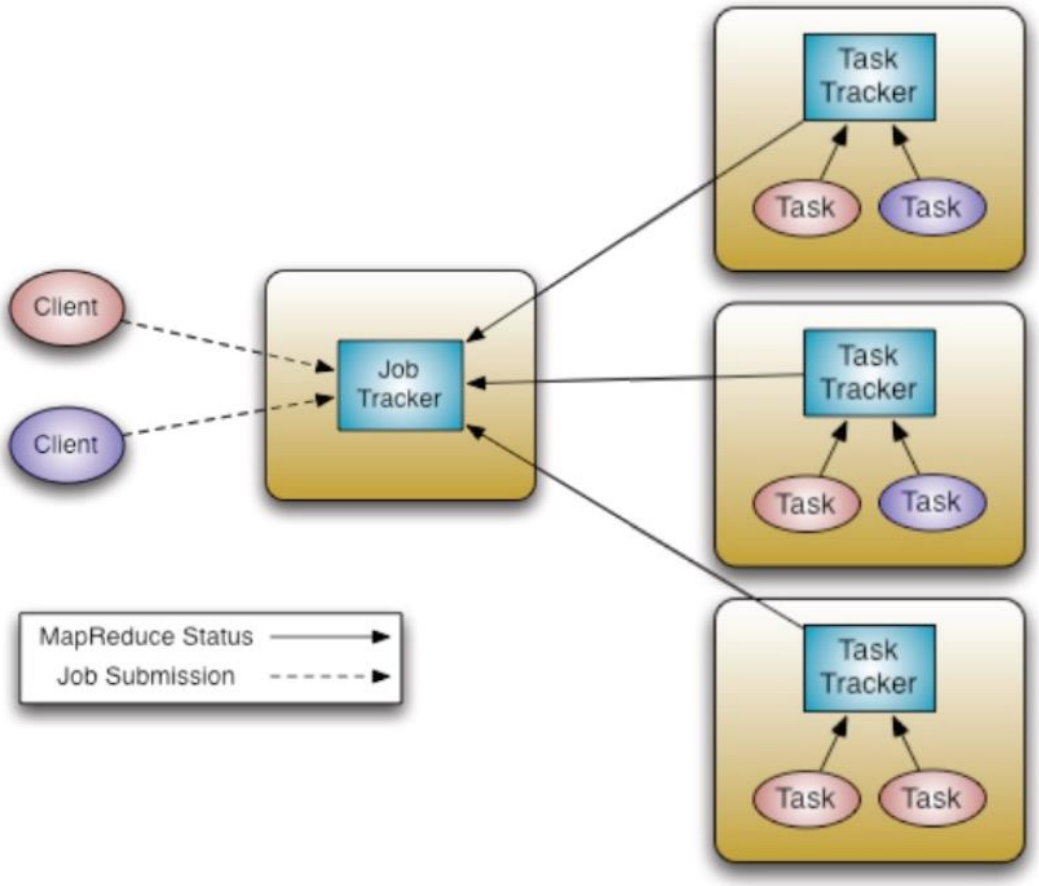
Hadoop 1.0 vs Hadoop 2.0



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Implementation of the work round (1.0)



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

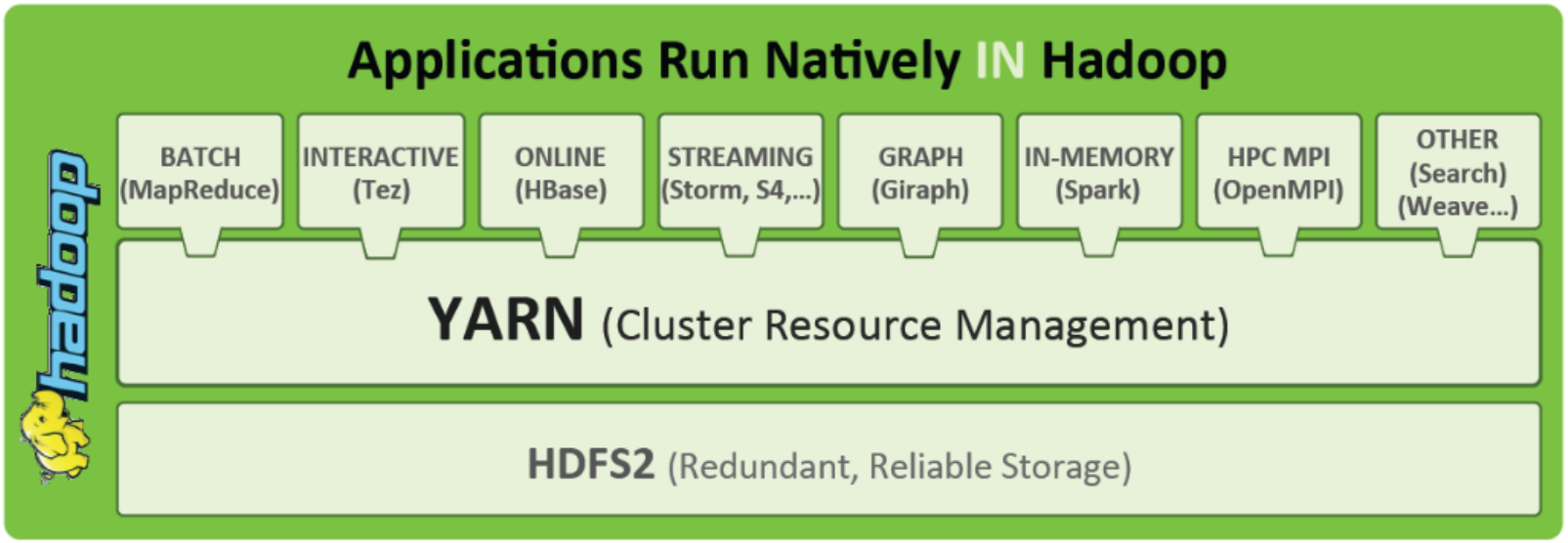
MapReduce/Hadoop 1 ograniczenia

- Skalowalność
- Maksymalny rozmiar klastra - 4 000 węzłów
- Maksymalna ilość współbieżnych zadań - 40,000
- Dokładna synchronizacja w JobTrackerze
- Dostępność
- Awaria zabija wszystkie oczekujące w kolejce i działające zadania
- Twardy podział zasobów na sloty map i redukcji
- Niskie wykorzystanie zasobów
- Brak wsparcia dla alternatywnych paradygmatów i usług
- Iteracyjne aplikacje zaimplementowane przy użyciu MapReduce są 10x wolniejsze

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

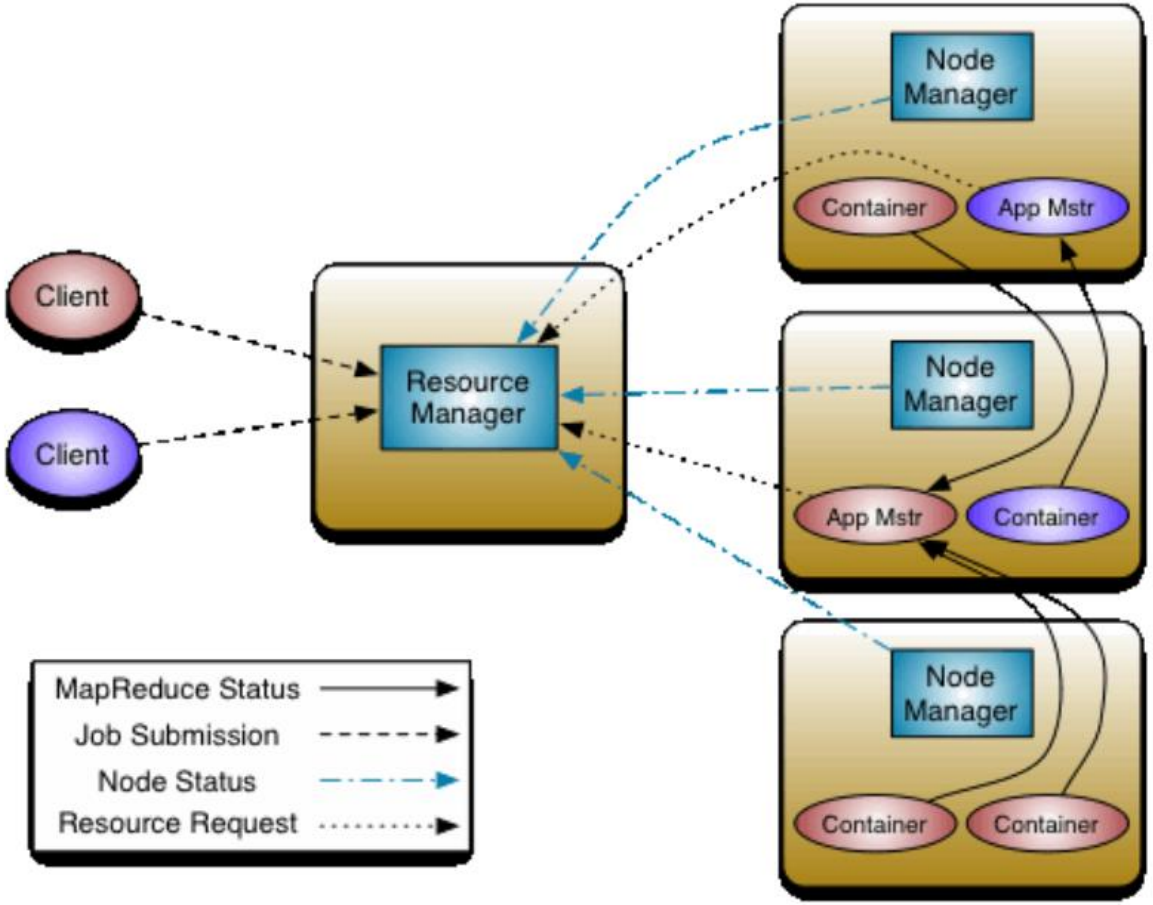
YARN: Hadoop poza partiami



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Hadoop 2.x - Wdrażanie zadań w YARN



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Wdrożenie uruchamiania zadań w YARN (2.x)

W przypadku YARN zaangażowanych jest pięć podmiotów:

Klient(y), który(e) przesyła(ją) zadanie

ResourceManager, który koordynuje wykonanie zadania; posiada dwa główne komponenty:

ApplicationsManager, który przyjmuje zgłoszenia zadań, negocjuje pierwszy węzeł do wykonania aplikacji specyficzny ApplicationMaster i restartuje w innym miejscu ApplicationMaster przy niepowodzeniu;

Scheduler, który jest odpowiedzialny za przydzielanie zasobów różnym uruchomionym aplikacjom zgodnie z ich wymaganiami.

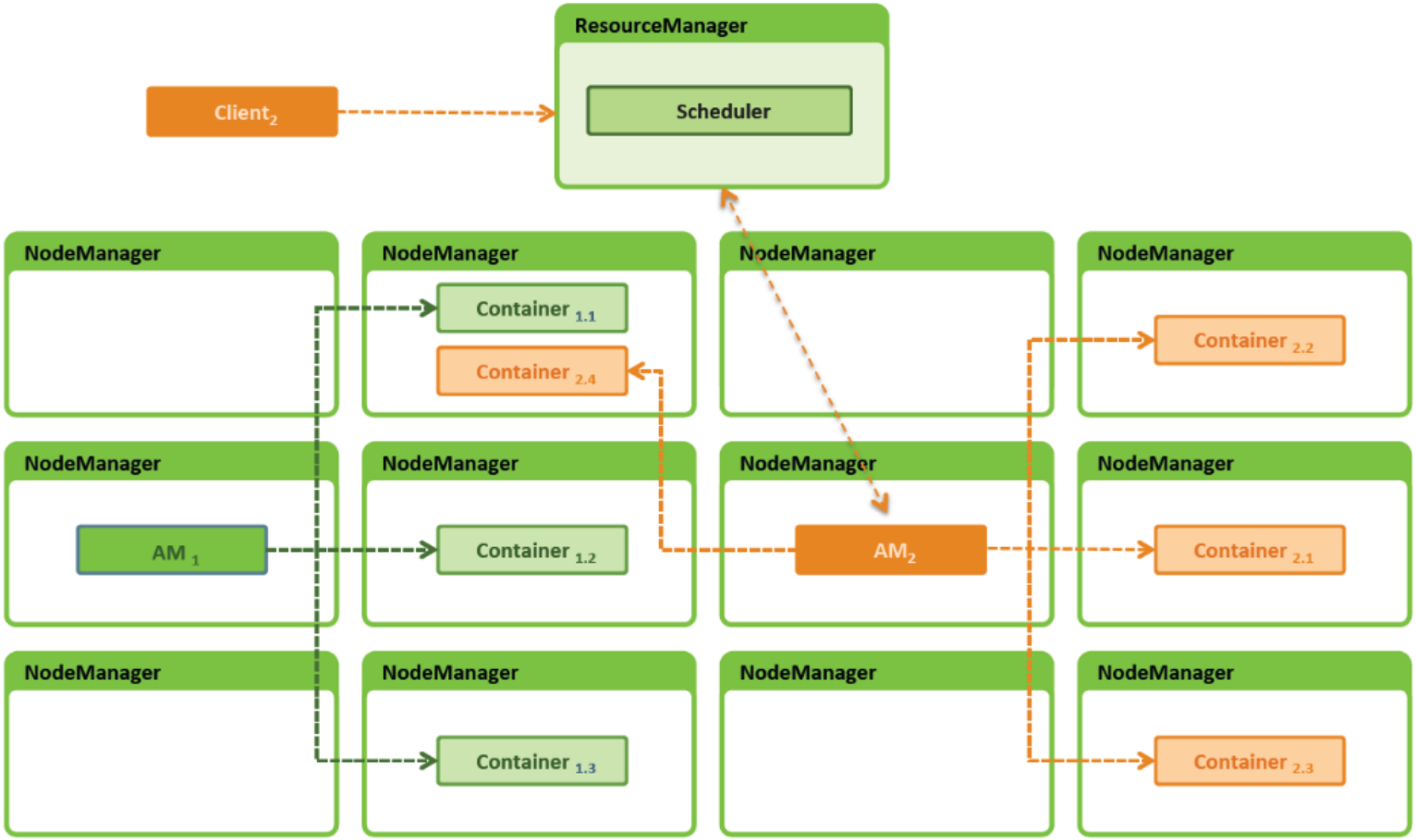
NodeManager(y), który(e) koordynuje(ą) pojedynczy węzeł (jeden NM dla każdego węzła)

ApplicationMaster(s), który negocjuje zasoby z ResourceManager i współpracuje z NodeManager(s) w celu wykonania i monitorowania zadań (jeden AM dla każdej aplikacji)

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

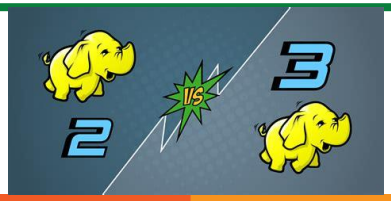
Praca w praktyce



iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the
Erasmus+ Programme
of the European Union

Apache Hadoop 2 vs Hadoop 3



Features	Hadoop 2.x	Hadoop 3.x
Min Java Version Required	Java 7	Java 8
Fault Tolerance	Via replication	Via erasure coding
Storage Scheme	3x replication factor for data reliability, 200% overhead	Erasure coding for data reliability, 50% overhead
Yarn Timeline Service	Scalability issues	Highly scalable and reliable
Standby NN	Supports only 1 SBNN	Supports only 2 or more SBNN
Heap Management	We need to configure HADOOP_HEAPSIZE	Provides auto-tuning of heap

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Apache HBase

- Open-source'owa, rozproszona, skalowalna baza danych NoSQL na Hadoop i HDFS
- Zorientowana na rodzinę kolumn baza danych oparta na Bigtable firmy Google
- Skalowalność liniowa, modułowa i pozioma
- Zapewnia losowy dostęp w czasie rzeczywistym do odczytu/zapisu dużych danych na HDFS
- Ściśle spójne odczyty i zapisy
- https://hbase.apache.org/

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Apache Pig

Apache Pig

- Szkielet przepływu danych na szczycie Hadoop/MapReduce
- Wykorzystuje Pig Latin - język deklaratywny podobny do SQL do manipulacji i transformacji danych
- Używa leniwej oceny,
- Nadaje się do prototypowania
- Znacznie mniej kodu w porównaniu do MapReduce
- Wiele UDF i bibliotek jest łatwo dostępnych
- Deklaracje Pig Latin są konwertowane na kod Hadoop/MapReduce
- Kompilacja jest ukryta przed użytkownikiem
- Wykonywana po stronie serwera
- https://pig.apache.org/

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Apache Hive

- Hurtownia danych na szczycie Hadoop
- Zaprojektowany dla
- łatwe podsumowanie danych
- wyszukiwanie ad-hoc
- analizy dużych ilości danych
- HiveQL obsługuje większość standardów SQL
- Instrukcje HiveQL są automatycznie tłumaczone na zadania MapReduce
- Rozszerzalność -Typów, funkcji, formatów, skryptów
- https://hive.apache.org/

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Wzorce projektowe dla MapReduce

- MapReduce jest ramą, a nie narzędziem
- Musisz dopasować swoje rozwiązanie do ram map i reduce
- W niektórych sytuacjach może to być wyzwaniem
- Trzeba wziąć algorytm i rozbić go na kroki filtrowania/agregacji
- Filtr staje się częścią funkcji mapy
- Agregacja staje się częścią funkcji redukcji
- Czasami możemy potrzebować wielu etapów MapReduce
- MapReduce nie jest rozwiązaniem każdego problemu, nawet nie każdego problemu, który z zyskiem może wykorzystać wiele węzłów obliczeniowych działających równolegle!
- Ma sens tylko wtedy, gdy:
- pliki są bardzo duże i rzadko aktualizowane
- Musimy iterować po wszystkich plikach, aby wygenerować jakąś interesującą właściwość danych w tych plikach

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union

Referencje

- Apache Hadoop
  - https://hadoop.apache.org/
- Nathan Marz, James Warren, Big Data Principles and best practices of scalable realtime data systems. Manning Publications Co., 2015.
- Tom White, Hadoop The Definitive Guide: Storage and Analysis at Internet Scale, 4th Edition, O'Reilly Media, 2015

iBigWorld:2020-1-PL01-KA203-082197

Co-funded by the Erasmus+ Programme of the European Union