# Apache Hadoop

## Wolna implementacja GFS, MapReduce oraz Big Table

Michał Jaszczyk

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki
Seminarium Systemów Rozproszonych

6 listopada 2008

# Apache Hadoop

# Hadoop - części składowe

- HDFS - rozproszony system plików
- MapReduce - framework do obliczeń rozproszonych
- HBase - system zarządzania bazą danych
- Pig, ZooKeeper - dodatki

# Trochę historii

- *The Google File System*
  Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung, 2003
- *MapReduce: Simplified Data Processing on Large Clusters*
  Sanjay Ghemawat, Jeffrey Dean, 2004
- *BigTable: A Distributed Storage System for Structured Data*
  Sanjay Ghemawat, Jeffrey Dean, Fay Chang, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, 2006

# Trochę historii cd.

- 2006: Powstaje projekt Hadoop
- 2006-04-01: Wersja 0.1.0
- 2008-01-23: Hadoop staje się głównym projektem
- 2008-10-30: Wersja 0.18.2

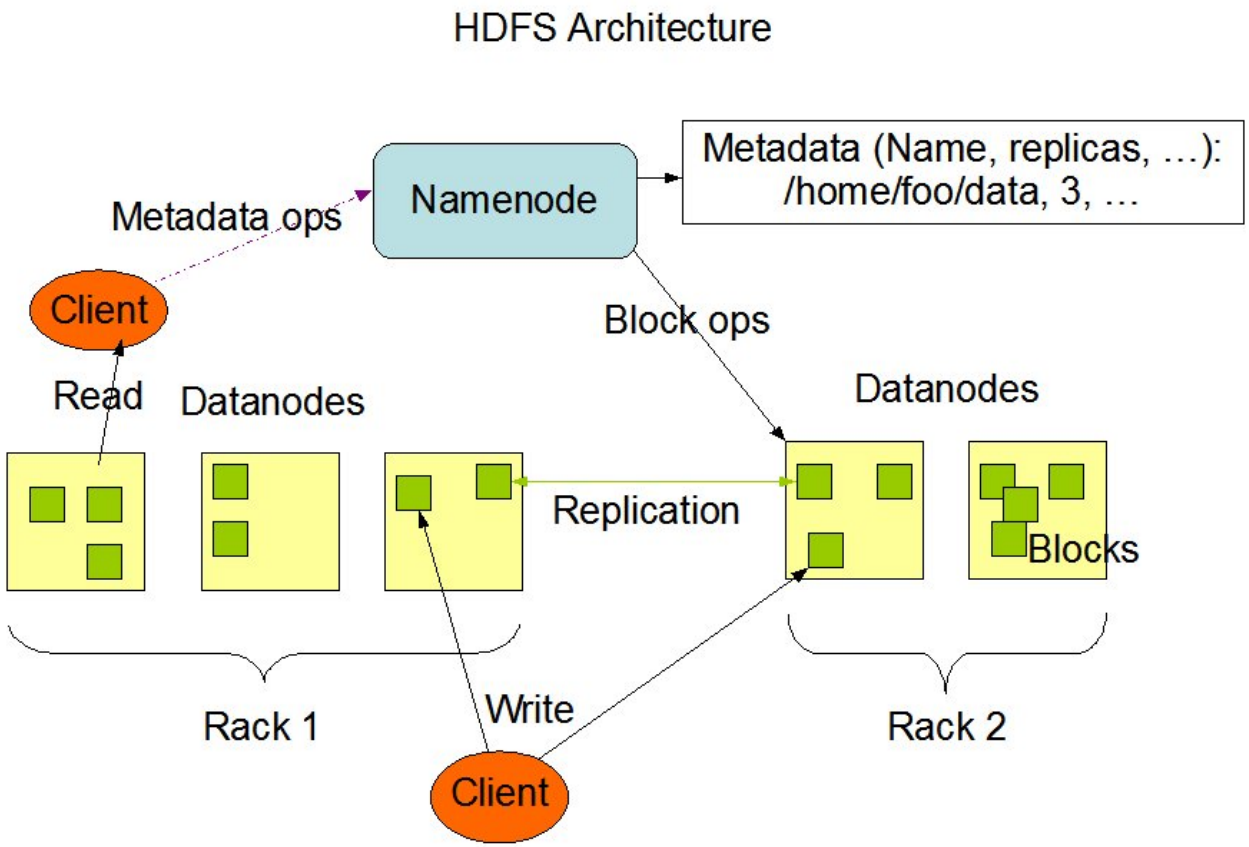
# Założenia

- Odporność na awarie żelastwa
- Bardzo duże pliki
- Sekwencyjny dostęp do plików
- Prosty model dostępu do danych (zapisz raz, czytaj wiele razy)
- "Przenoszenie obliczeń jest prostsze niż przenoszenie danych"
- Przenośność

# Cechy

- Replikacja danych (domyślnie trzykrotna)
- Dostęp przez bibliotekę (trwają prace nad FUSE)
- Nie spełnia specyfikacji POSIX
- Są quota, ale na liczbę plików
- Standardowe drzewo katalogów
- Prawa dostępu typu rwx/ugo
- Nie ma snapshotów (trwają prace)

# Ogólny schemat



# Rodzaje węzłów

- NameNode
  - Punkt dostępowy dla klienta
  - Zarządza metadanymi (przestrzeń nazw, rozmieszczenie replik)
  - Single point of failure (ale może być hot backup)
- DataNode
  - Obsługuje zlecenia odczytu/zapisu bloków od klienta
  - Okresowo wysyła do NameNode listę przechowywanych bloków

# Organizacja i replikacja danych

- Metadane na NameNode, dane na DataNode
- Metadane w formie dziennika transakcji, tzw. EditLog
- Może być zreplikowany
- Pliki podzielone na bloki wielkości 64MiB
- Bloki trzymane jako pliki w lokalnym systemie plików na DataNode'ach

# Organizacja i replikacja danych cd.

- Domyślnie 3 repliki na blok, ale można zmienić
- NameNode decyduje o rozmieszczeniu replik
- Szafoświadomość
- Dwie repliki w jednej szafie, trzecia w innej
- Centroświadomość
- Optymalny wybór repliki do odczytu
- Rebalancing na podstawie ilości wolnego miejsca na dysku
- Staging (cache'owanie po stronie klienta pierwszego bloku)
- Pipelining (DataNode'y przy zapisie ustawiają się w łańcuch)
- SafeMode

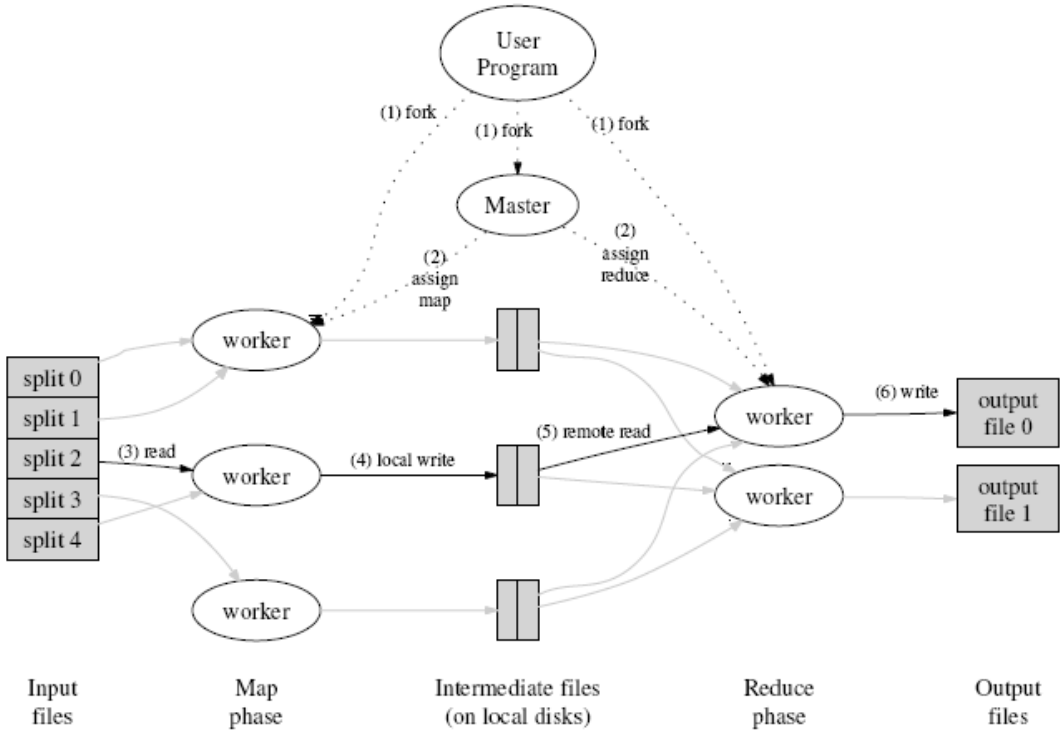
# Użytkowanie

```
hadoop fs -get hdfs://nn.example.com/user/hadoop/fi
hadoop fs -put localfile hdfs://nn.example.com/hado
hadoop fs -ls hdfs://nn.example.com/user/hadoop/dir
hadoop fs -cp /user/hadoop/file1 /user/hadoop/file2
hadoop fs -mv hdfs://nn.example.com/file1 hdfs://nn
hadoop fs -cat hdfs://nn1.example.com/file1
hadoop fs -rm hdfs://nn.example.com/file
```

# Jak to działa

1. Dane wejściowe są dzielone na części
2. Części są przydzielane do Mapperów
3. Mappery mapują
4. Wyniki z Mapperów są przesyłane do Reducerów
5. Reducery redukują
6. Wyniki z Reducerów to wynik całego obliczenia

# Jak to działa - obrazek



# Architektura

- Job Tracker
  - Przyjmuje zlecenia od użytkowników
  - Single Point of Failure
  - Nie ma recovery rozpoczętego obliczenia
  - Rozdziela zadania do Task Trackerów
  - Szafoświadomy
- Task Tracker
  - Wykonuje pojedyńcze obliczenie Map lub Reduce
  - Ma sloty, żeby sterować obciążeniem

# Mapper - interfejs

```
public interface Mapper<K1,V1,K2,V2>
extends JobConfigurable, Closeable {

    public void map(K1 key,
                    V1 val,
                    OutputCollector<K2, V2> output,
                    Reporter reporter
                    ) throws IOException;
}
```

# Mapper - przykład

```
public static class MyMapper
extends MapReduceBase
implements Mapper<LongWritable, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key,
                    Text value,
                    OutputCollector<Text, IntWritable> output,
                    Reporter reporter
                    ) throws IOException {

        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            output.collect(word, one);
        }
    }
}
```

# Reducer - interfejs

```
public interface Reducer<K2,V2,K3,V3>
extends JobConfigurable, Closeable {

    public void reduce(K2 key,
                       Iterator<V2> values,
                       OutputCollector<K3,V3> output,
                       Reporter reporter
                       ) throws IOException;
}
```

# Reducer - przykład

```
public static class MyReducer
extends MapReduceBase
implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key,
                       Iterator<IntWritable> values,
                       OutputCollector<Text, IntWritable> output,
                       Reporter reporter) throws IOException {

        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```

# Konfiguracja

- JobConf
- InputFormat
- OutputKeyComparator
- Combiner
- Partitioner
- CompressionCodec
- OutputValueGroupingComparator
- OutputFormat

# Nie lubię Javy

- Hadoop Streaming
- Hadoop Pipes

# HBase

HBase to dość wierna kopia BigTable.

# Pig

Pig

# ZooKeeper

ZooKeeper

# Powered by Hadoop

- Amazon A9
- AOL
- Facebook
- Google :)
- IBM
- ImageShack
- Joost
- Last.fm
- New York Times
- Yahoo!

# Amazon S3 i EC2

- S3 = Simple Storage Service
  - Każdy może utworzyć swoje wiaderko
  - Dostęp przez HTTP, REST lub SOAP
  - Można hostować statyczną zawartość witryn WWW
  - Opłaty per gigabajtomiesiąc oraz transfer
- EC2 = Elastic Compute Cloud
  - Każdy może zamówić sobie wirtualną maszynę
  - Można robić to dynamicznie przez HTTP, REST lub SOAP
  - Na wirtualnej maszynie można odpalać cokolwiek się chce
  - Opłaty per maszynogodzinę

# Pytania

Pytania?

# Koniec

Koniec!