



Hadoop w Hurtowni Danych nk.pl

wojciech.langiewicz@nasza-klasa.pl, 4 Developers 2012

Plan prezentacji

- Problem
- Rozwiązanie
- Przykład nk.pl

Problem

Problem

- Ogromna ilość danych
- Które chcemy zbierać
- Analizować
- Wyciągać wnioski
- Jak najszybciej
- I najtaniej

Problem

- Coraz więcej firm/portali/stron
- Coraz więcej użytkowników
- Coraz aktywniejsi użytkownicy
- Maszyny generują coraz więcej danych



Rozwiązania

Rozwiązanie 1

- Usuwamy dane po pewnym czasie
- Początkowo logujemy ostatnie 30 dni
- Wraz ze wzrostem ilości danych ten okres maleje

Rozwiązanie 2

- Zbieramy tylko pewną część danych
- np. logujemy aktywności co 10 użytkowników (próbkiowanie)
- Lub zmniejszamy ilość danych informacji jakie logujemy

Rozwiązanie „optymalne”



- Hadoop - open-source software for reliable, scalable, distributed computing
- Nigdy więcej usuwania danych
- Bez próbkowania
- Commodity hardware, reliability, availability, open, free

Hadoop - założenia

- Dane nie mieszczące się na jednej maszynie
- Więcej maszyn → większa szansa awarii
- Wcześniejsze rozwiązania nie skalowały się
- Skanowanie (scan) dysku jest szybsze niż wyszukiwanie (seek)
- Przenoszenie obliczeń zamiast danych
- Przetwarzanie danych nie mieści się na jednej maszynie

Hadoop - właściwości

- Liniowo skalowany na tysiące maszyn
- Oparty na Javie
- Analizy bez agregacji
- Cały ekosystem
- Dane niestrukturalizowane
- Jeden system – wiele zastosowań
- Prosty w użyciu
- Massive parallel processing
- Używany przez największych
- Bezpieczny – replikacja
- Rozproszony
- Wbudowany failover
- Prosty

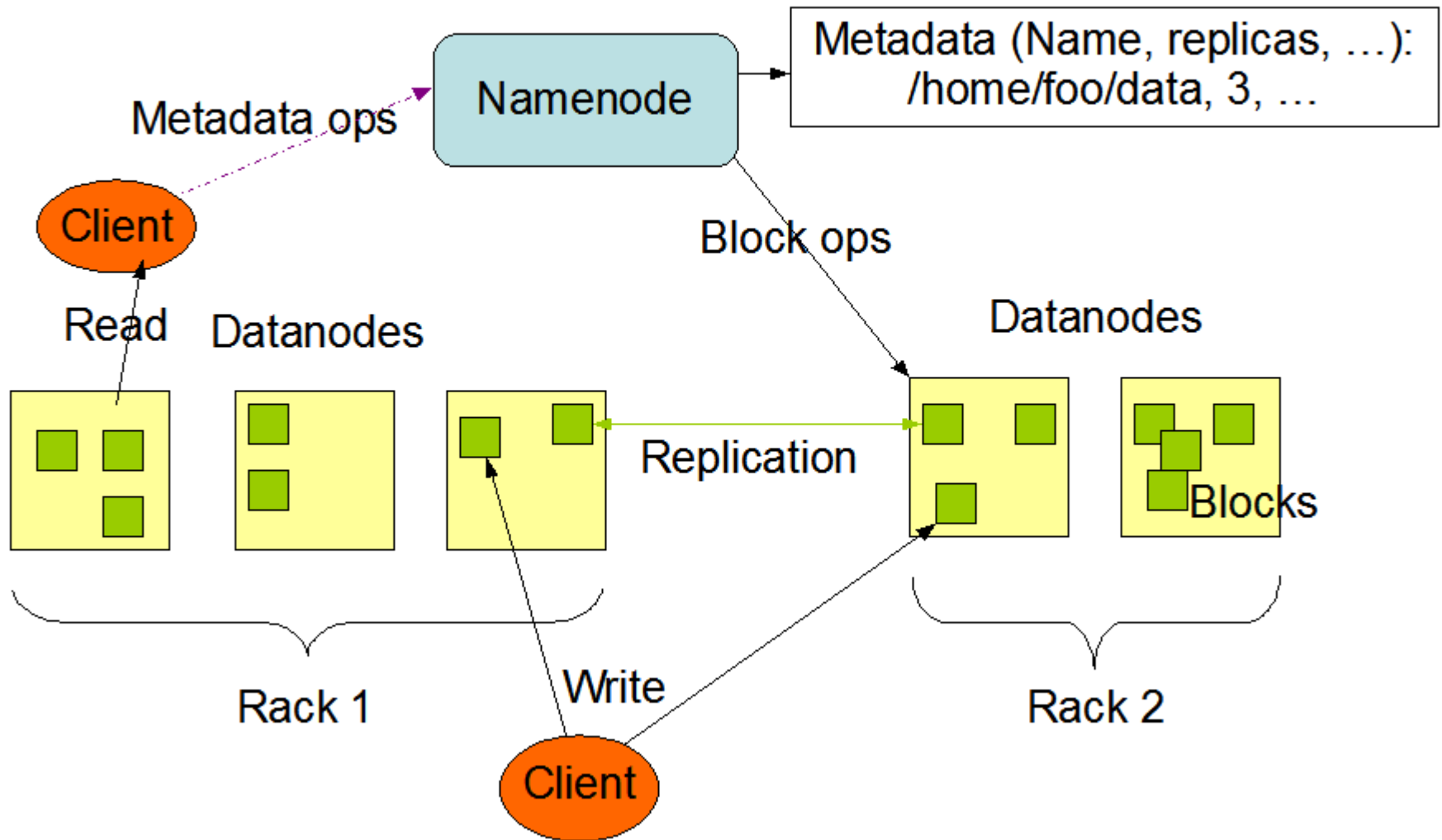
Hadoop - podstawa architektury

- Warstwa „niższa”: przechowywanie danych (HDFS)
- Warstwa „wyższa”: silnik wykonujący zapytania (MapReduce)

HDFS

- Hadoop Distributed File System
- Optymalizacja pod dostęp sekwencyjny
- Wbudowana replikacja
- Metadane przechowywane na serwerze Master (NameNode)

HDFS Architecture



MapReduce

- Paradygmat programowania rozproszonego
- Czyni łatwiejszym rozproszone przetwarzanie danych
- Programista dostarcza implementacje dwóch funkcji, map oraz reduce
- Ukrywa przed programistą większość problemów programowania rozproszonego

MapReduce - pseudokod Wordcount

```
map(String key, String value)
  foreach word in value:
    emit(word, 1);
```

```
reduce(String key, int[] values)
int result = 0;
foreach val in values:
  result += val;
emit(key, result);
```


MapReduce - przykład

```
map (in): abaaabcab
```

```
map (out):
```

```
a 1
```

```
b 1
```

```
a 1
```

```
a 1
```

```
a 1
```

```
b 1
```

```
c 1
```

```
a 1
```

```
b 1
```

```
reduce (in):
```

```
a [1 1 1 1 1]
```

```
b [1 1 1]
```

```
c [1]
```

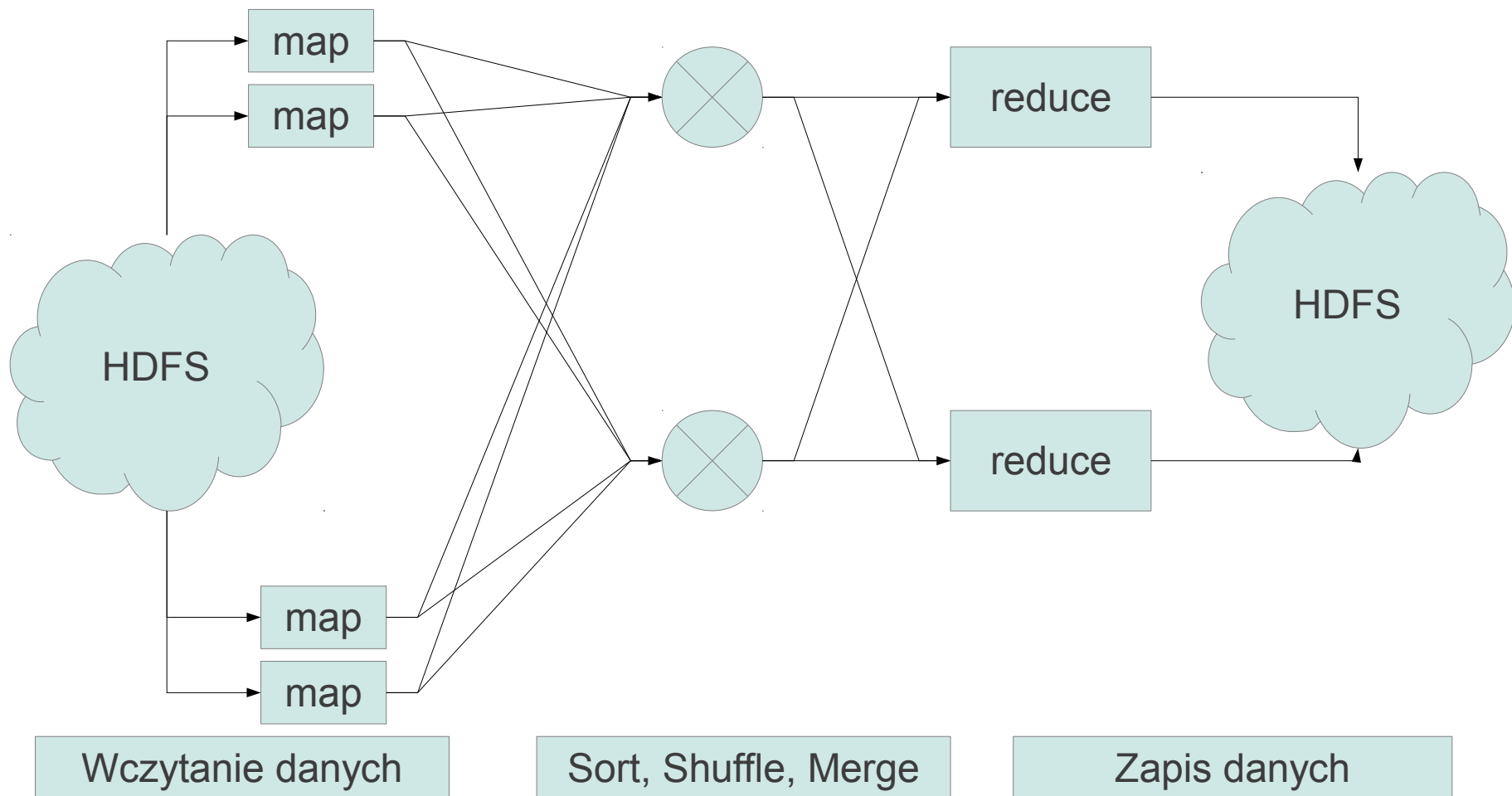
```
reduce (out):
```

```
a 5
```

```
b 3
```

```
c 1
```

MapReduce - diagram



Przykłady zadań

- Szukanie wzorca (grep)
- Zliczenia
- Odwracanie grafu www (inverted index)
- Sortowanie
- Wszystko co możemy zapisać jako MapReduce

Dostęp do danych (interfejsy)

- Java
- Streaming
- Hive
- Pig

Hive

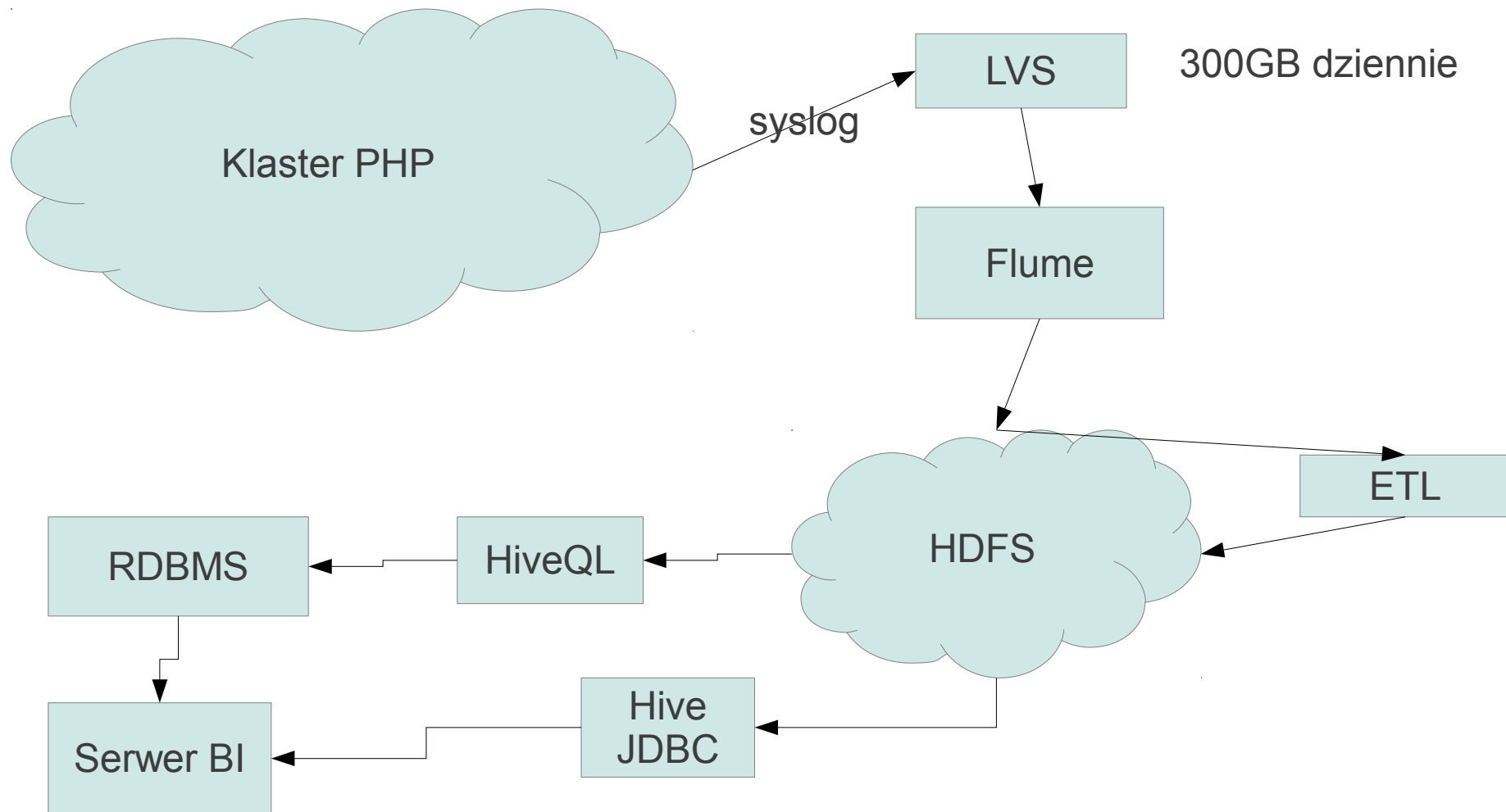
- „Hurtownia danych na Hadoopie”
- Dostęp do danych za pomocą SQL
- Interface JDBC
- Wymaga wcześniejszej konfiguracji bazy z metadanymi

Problemy

- Na start potrzeba kilku serwerów, aby zauważyć wzrost wydajności
- To nie jest OLTP (zapytania mogą trwać minuty, a nawet godziny)
- Projekt we wczesnej wersji, 1.0 dopiero od niedawna (ale i tak popularny)
- Potrzebujesz zarządzania konfiguracją

Przykład nk.pl

Logi aplikacji (PHP)



Co dalej?

- www.cloudera.com
- <http://pig.apache.org>
- <http://hbase.apache.org>



Czas na pytania



Dziękuję za uwagę!
developers.nk.pl