



Wrocław
University
of Science
and Technology

Przetwarzanie danych masowych

Wykład 9 – Platformy zarządzania zasobami obliczeniowymi - wprowadzenie, OpenStack

dr inż. Tomasz Kajdanowicz, Roman Bartusiak, Piotr Bielak,
Krzysztof Rajda

13 grudnia 2021 r.



HR EXCELLENCE IN RESEARCH

Spis treści

Hadoop

YARN

Zookeeper

Swarm

Openstack

Spis treści

Hadoop

YARN

Zookeeper

Swarm

Openstack

Wprowadzenie

Hadoop

- ▶ narzędzie do przetwarzania masowych danych
- ▶ projekt *open source*
- ▶ tryb analityczny i operacyjny

	Operational	Analytical
Latency	1 ms - 100 ms	1 min - 100 min
Concurrency	1000 - 100,000	1 - 10
Access Pattern	Writes and Reads	Reads
Queries	Selective	Unselective
Data Scope	Operational	Retrospective
End User	Customer	Data Scientist
Technology	NoSQL	MapReduce, BSP

Komponenty

Hadoop

Centralnymi komponentami Hadoopa są:

- ▶ HDFS
- ▶ MapReduce
- ▶ YARN
- ▶ Common

HDFS – wprowadzenie

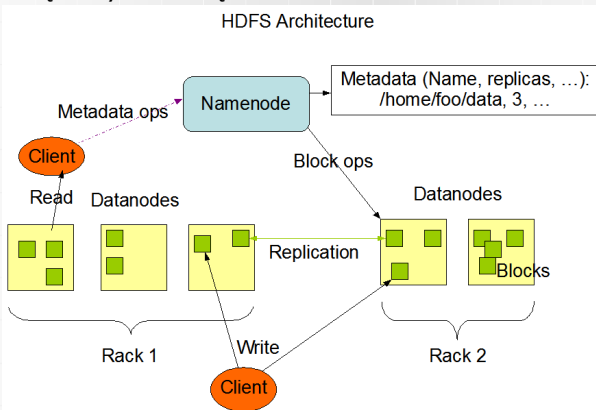
Hadoop

- ▶ ang. *Hadoop Distributed File System*
- ▶ jeden z najważniejszych komponentów Hadoopa
- ▶ zarządza sposobem podziału i zapisu plików w klastrze
- ▶ dane są dzielone w bloki; każdy serwer w klastrze zawiera dane z różnych bloków
- ▶ wbudowana nadmiarowość (ang. *redundancy*)
- ▶ prawa dostępu (ang. *permissions*) do plików i uwierzytelnianie (ang. *authentication*)
- ▶ architektura *master-slave*

HDFS – architektura

Klaster HDFS składa się z:

- ▶ pojedynczej instancji **NameNode** – serwer w roli *master*, który zarządza przestrzenią nazw systemu plików oraz dostępem do plików
- ▶ wiele instancji **DataNodes** (zazwyczaj jedna instancja per węzeł w klastrze) – zarządza wolumenami dyskowymi podłączonymi do węzła



HDFS – zadania komponentów

Hadoop

NameNode:

- ▶ zarządzanie przestrzenią nazw systemu plików (ścieżki)
- ▶ otwieranie, zamykanie oraz zmiana nazw plików i katalogów
- ▶ przypisanie bloków do DataNodes

DataNodes:

- ▶ przetwarzanie zapytań zapisu i odczytu plików od klientów
- ▶ tworzenie i usuwanie bloków (instrukcje NameNode)

HDFS – bloki

Hadoop

- ▶ dane użytkowników są zapisywane w plikach HDFS
- ▶ plik w systemie plików jest dzielony na jeden albo kilka segmentów i zapisywany na indywidualnych DataNodes
- ▶ segmenty (fragmenty) plików są nazywane **blokami**
- ▶ najmniejszą encją, którą HDFS potrafi zapisywać albo odczytywać jest blok
- ▶ domyślny rozmiar bloku wynosi 64MB

MapReduce

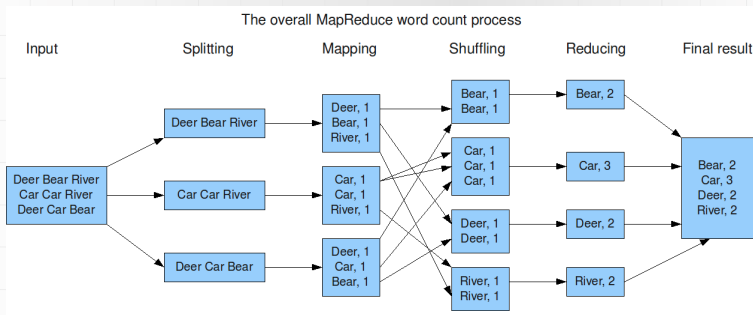
Hadoop

MapReduce – metoda równoległego przetwarzania na rozproszonych serwerach

Etapy w przetwarzaniu MapReduce:

1. **map** – przetwórz dane wejściowe z HDFSa; pliki wejściowe są przekazywane do funkcji mapujących linia po linii; tworzonych jest kilka mniejszych fragmentów danych
2. **shuffle** – przenieś wyniki Mappera to Reducera
3. **reduce** – przyjmuje wyniki Mappera (pośrednie pary klucz-wartość); wyjścia Reducera są końcowymi plikami wyjściowymi w HDFSie; zazwyczaj są to operacje agregacji lub sumowania

MapReduce – przykład



1. Wejście jest dzielone na bloki
2. Mapper zamienia wejście na pary klucz-wartość
3. Pary są porządkowane w fazie *shuffle*
4. Reducer sortuje i grupuje wyniki

Spis treści

Hadoop

YARN

Zookeeper

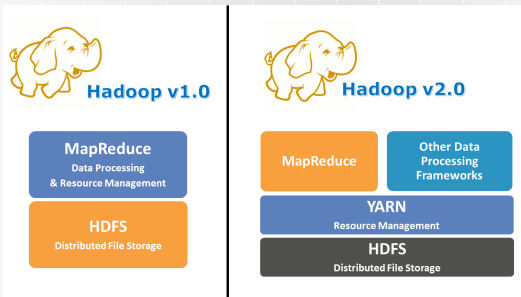
Swarm

Openstack

Hadoop YARN

YARN

- ▶ nie mylić z menadżerem paczek yarn w Javascript
- ▶ odpowiedzialny za część związaną w przetwarzaniem w Hadoopie
- ▶ YARN = Yet Another Resource Negotiator
- ▶ przypisuje zasoby (ang. *resources*) do każdej aplikacji



YARN – komponenty

YARN

Resource Manager

- ▶ komunikacja z klientem
- ▶ monitorowanie zasobów w klastrze
- ▶ zarządzanie *Node Managerami* poprzez przypisywanie zadań (*job*) zgodnie z wymaganiami zasobów (*resource requirements*)
- ▶ alokacja pamięci w postaci kontenerów

Node Manager

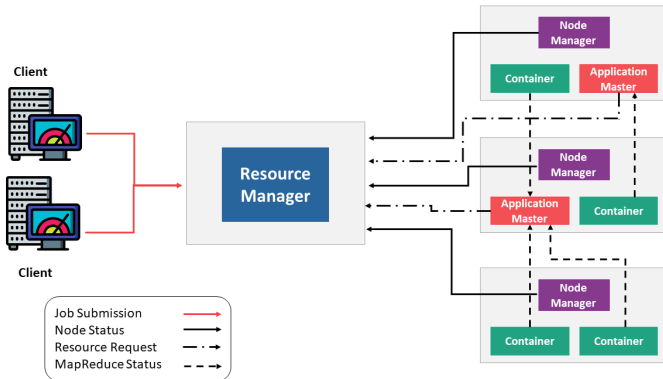
- ▶ monitorowanie uruchomionych aplikacji
- ▶ oferuje zasoby (pamięć, procesory) w postaci kontenerów zasobów (*resource container*)
- ▶ każdy Data Node → jeden Node Manager
- ▶ w kontenerach uruchamiane są zadania (wraz z Application Master)

Application Master

- ▶ każde zadanie jest inicjalizowane przez Application Mastera
- ▶ zarządza globalnym przypisaniem zasobów obliczeniowych to aplikacji (np. pamięć RAM, CPU, przestrzeń dyskowa, dostęp do sieci, itp.)

YARN – diagram architektury

YARN



Spis treści

Hadoop

YARN

Zookeeper

Swarm

Openstack

Zookeeper

Zookeeper

- ▶ nie jest to platforma zarządzania zasobami
- ▶ często używany z Hadoop, HBase, Hive, Kafka itp.
- ▶ zcentralizowany serwis do:
 - ▶ zarządzania konfiguracjami,
 - ▶ naming,
 - ▶ udostępniania rozproszonej synchronizacji,
 - ▶ udostępniania grupowych serwisów.



APACHE
ZooKeeper™

Spis treści

Hadoop

YARN

Zookeeper

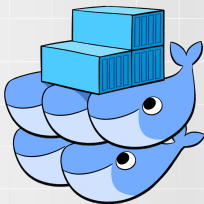
Swarm

Openstack

Docker Swarm – cechy

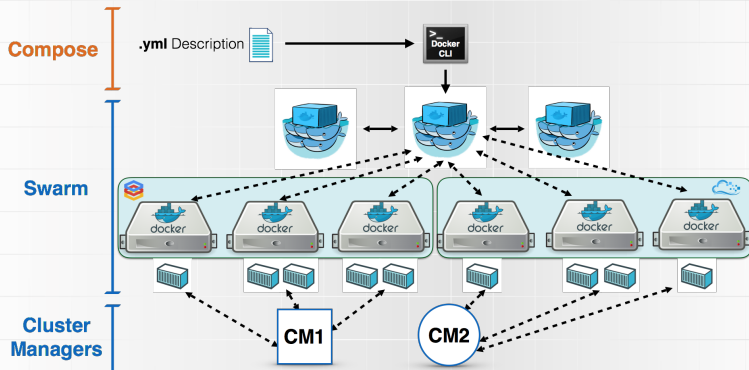
Swarm

- ▶ system zarządzania kontenerami utworzony przez Dockera
- ▶ rozproszenie kontenerów Dockerowych pomiędzy wiele węzłów
- ▶ wprowadzenie pojęć: *stack* oraz *service*;
- ▶ rezerwacja zasobów i limity
- ▶ skalowanie kontenerów
- ▶ rolling updates
- ▶ brak zaawansowanych mechanizmów sieciowych (overlay)
- ▶ bardzo podstawowa funkcjonalność w porównaniu z Kubernetesem (następny wykład!)



Docker Swarm – architektura

Swarm



Spis treści

Hadoop

YARN

Zookeeper

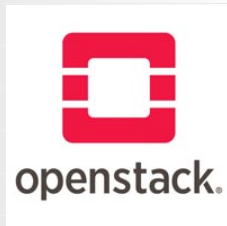
Swarm

Openstack

Openstack

Openstack

- ▶ obecnie w ramach inicjatywy Open Infra
- ▶ najbardziej popularna otwarta (open source) platforma do zarządzania maszynami wirtualnymi
- ▶ działa na poziomie maszyn wirtualnych a nie kontenerów
- ▶ architektura oparta na mikroserwisach
- ▶ wiele wyspecjalizowanych serwisów



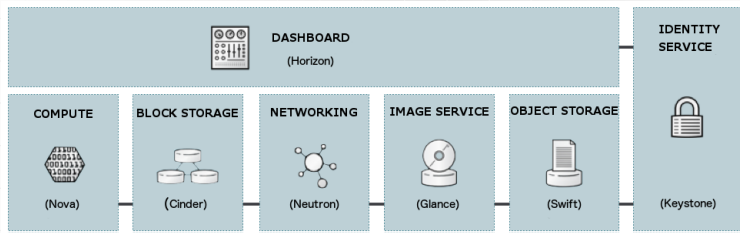
Co jest potrzebne aby utworzyć maszynę wirtualną?

Openstack

- ▶ zasoby obliczeniowe (tzn. fizyczna maszyna na której będzie uruchomiona maszyna wirtualna)
- ▶ obraz systemu operacyjnego
- ▶ dostęp do sieci
- ▶ dostęp do wolumenów dyskowych
- ▶ dostęp do akceleratorów obliczeniowych (GPU, TPU, FPGA)
- ▶ prawa do utworzenia maszyny

Architektura (1)

Openstack





Mikroserwisy

Openstack

- ▶ **NOVA** (Compute)
- ▶ **ZUN** (Containers)
- ▶ **QINLING** (Functions)
- ▶ **IRONIC** (Bare Metal Provisioning)
- ▶ **CYBORG** (Hardware accelerators)
- ▶ **SWIFT** (Object store)
- ▶ **CINDER** (Block Storage)
- ▶ **MANILA** (Shared filesystems)
- ▶ **NEUTRON** (Networking)
- ▶ **OCTAVIA** (Load balancer)
- ▶ **DESIGNATE** (DNS)
- ▶ **KEYSTONE** (Identity)
- ▶ **PLACEMENT** (Placement)
- ▶ **GLANCE** (Image)
- ▶ **HEAT** (Orchestration)
- ▶ **MISTRAL** (Workflow)
- ▶ **AODH** (Alarming)
- ▶ **MAGNUM** (Container Orchestration Engine Provisioning)
- ▶ **SAHARA** (Big Data Processing Framework Provisioning)
- ▶ **TROVE** (Database as a Service)
- ▶ **MASAKARI** (Instances High Availability Service)
- ▶ **MURANO** (Application Catalog)
- ▶ **EC2API** (EC2 API proxy)
- ▶ **HORIZON** (Dashboard)
- ▶ and many more...

Przetwarzanie danych masowych

Wykład 9 – Platformy zarządzania zasobami obliczeniowymi - wprowadzenie, OpenStack

dr inż. Tomasz Kajdanowicz, Roman Bartusiak, Piotr Bielak,
Krzysztof Rajda

13 grudnia 2021 r.