

Тимур Машнин

**Технология хранения
и обработки больших
данных Hadoop**



ЛитРес: Самиздат
Москва

УДК 004.7
ББК 32.973.202
М38

Машнин Т.

Технология хранения и обработки больших данных Hadoop /
Машнин Т. — Москва: ЛитРес: Самиздат, 2021. — 136 с.

ISBN 978-5-532-96881-3

Apache Hadoop - это платформа для распределенной обработки больших наборов данных на кластерах компьютеров с использованием простых моделей программирования. В этой книге вы познакомитесь с общей архитектурой платформы, компонентами стека, такими как HDFS и MapReduce, приложениями Hadoop.

Содержание

Введение	4
Cloudera QuickStart VM	26
Компоненты стека Hadoop	48
HDFS	83
MapReduce	108
Apache Spark	122

Введение



Hadoop

Введение

Hadoop – это программная платформа с открытым исходным кодом Apache для хранения и крупномасштабной обработки больших наборов данных в распределенной среде кластеров компьютеров с использованием простых моделей программирования.


<http://www.apache.org/>

Apache > Hadoop > Apache Hadoop 2.9.2

Wiki | Git | Last Published: 2018-11-13 | Version: 2.9.2

General

Overview

Single Node Setup

Cluster Setup

Commands Reference

Filesystem Shell

Hadoop Compatibility

Interface Classification

Filesystem Specification

Common

CLI Mini Cluster

Native Libraries

Proxy User

Rack Awareness

Secure Mode

Service Level

Authorization

HTTP Authentication

Credential Provider API

Hadoop KMS

Tracing

Architecture

User Guide

Commands Reference

Netmode HA with QM

Netmode HA with NFS

Federation

Viewfs

Snapshot

Apache Hadoop 2.9.2

Apache Hadoop 2.9.2 is a point release in the 2.x.y release line, building upon the previous stable release 2.9.1.

Here is a short overview of the major features and improvements.

- Common
 - » Aliyun OSS Support. See the user documentation for more details.
 - » HADOOP Resource Estimator. See the user documentation for more details.
- HDFS
 - » HDFS Router based federation. See the user documentation for more details.
- YARN
 - » YARN Timeline Service v.2. See the user documentation for more details.
 - » YARN Federation. See the user documentation for more details.
 - » Opportunistic Containers. See the user documentation for more details.
 - » YARN Web UI v.2. See the user documentation for more details.
 - » Changing queue configuration via API (supported only on the Capacity Scheduler). See the user documentation for more details.
 - » Update Resources and Execution Type of an allocated/running container. (supported only on the Capacity Scheduler). See the user documentation for more details.

Getting Started

The Hadoop documentation includes the information you need to get started using Hadoop. Begin with the [Single Node Setup](#) which shows you how to set up a single-node Hadoop installation. Then move on to the [Cluster Setup](#) to learn how to set up a multi-node Hadoop installation.

Hadoop предназначен для масштабирования от отдельных серверов до тысяч машин, каждая из которых обеспечивает локальные вычисления и хранилище.

Фреймворк Hadoop был создан Дагом Каттингом и Майком Кафреллой в 2005 году.

Первоначально этот фреймворк был разработан для поддержки распространения проекта Nutch Search Engine построения поисковых систем.

Даг, который в то время работал в Yahoo, а сейчас является главным архитектором в Cloudera, назвал этот проект в честь слона своего сына.

Его сын назвал своего игрушечного слона Hadoop, и Даг использовал это имя, чтобы так назвать свой проект.

Давайте посмотрим, что делает фреймворк Hadoop таким интересным, масштабируемым и удобным в использовании.

Hadoop начинался как простая среда пакетной обработки.

Идея, лежащая в основе Hadoop, заключается в том, что вместо перемещения данных в вычисления мы переносим вычисления в данные.

И в основе системы Hadoop лежит масштабируемость.

Все модули в Hadoop разработаны с фундаментальным предположением о том, что аппаратное обеспечение рано или поздно выходит из строя.

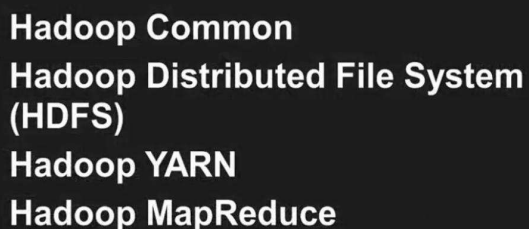
То есть предположением, что отдельная машина или стойка машин, или большой кластер или суперкомпьютер, все они в какой-то момент выйдут из строя, или некоторые их компоненты выйдут из строя.

И компоненты Apache Hadoop – MapReduce и HDFS изначально были созданы на основе Google MapReduce и файловой системы Google.

Еще одна очень интересная вещь, которую приносит Hadoop, – это новый подход к данным.

Новый подход заключается в том, что мы можем сохранить все данные, которые у нас есть, и мы можем взять эти данные и читать данные, создавая схему, во время чтения.

Вместо того, чтобы тратить время на создание схемы, пытаюсь подогнать данные к схеме, которую мы создали заранее, мы сохраняем все данные в приблизительном формате, а затем проецируем их в схему на лету, пока мы эти данные читаем.



Hadoop Common
**Hadoop Distributed File System
(HDFS)**
Hadoop YARN
Hadoop MapReduce

Фреймворк Apache Hadoop содержит четыре основных компонента.

Это Hadoop Common, распределенная файловая система Hadoop или HDFS, Hadoop MapReduce и Hadoop YARN.

Hadoop Common содержит библиотеки и утилиты, необходимые для других модулей Hadoop.

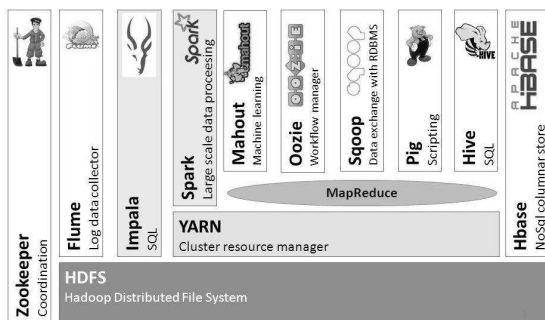
Распределенная файловая система Hadoop хранит данные на обычном компьютере, обеспечивая очень высокую совокупную пропускную способность по всему кластеру компьютеров.

Hadoop YARN – это платформа управления ресурсами, которая отвечает за управление вычислительными ресурсами в кластере и их использование в при планировании пользователей и приложений.

И Hadoop MapReduce – это модель программирования, которая масштабирует данные по множеству процессов.

И все модули фреймворка Hadoop разработаны с фундаментальным предположением, что аппаратное обеспечение выходит из строя.

Hadoop eco system



Если вы посмотрите на HDFS, YARN, MapReduce и всю платформу в целом, она состоит из многочисленных приложений, и каждое из этих приложений создано с учетом этого предположения.

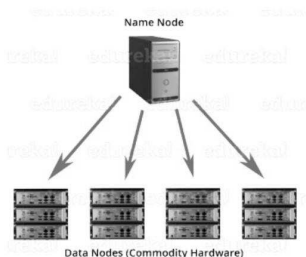
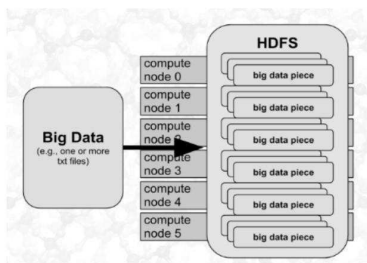
У нас есть различные приложения, такие как Apache PIG, Apache Hive, HBase и другие.

И для конечного пользователя, через Java-код MapReduce, он может получить доступ к любому из этих приложений.

И мы можем строить различного вида системы из этих приложений.

Проекты Apache PIG и Apache Hive предоставляют интерфейсы высокого уровня, обеспечивая доступ к данным через пользовательский интерфейс.

Сам фреймворк Hadoop в основном написан на языке программирования Java и проект также содержит несколько приложений на нативном языке C и утилиты командной строки.



Теперь, давайте немного поговорим о распределенной файловой системе Hadoop.

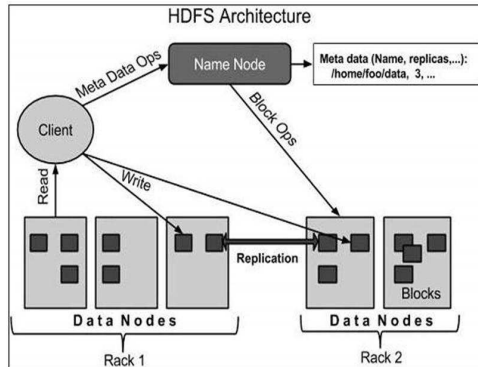
Что такое HDFS по своей сути?

Это распределенная, масштабируемая и переносимая файловая система, написанная на Java для поддержки фреймворка Hadoop.

Каждый Hadoop кластер обычно состоит из одного узла Namenode и кластера узлов Datanode, которые и формируют этот кластер.

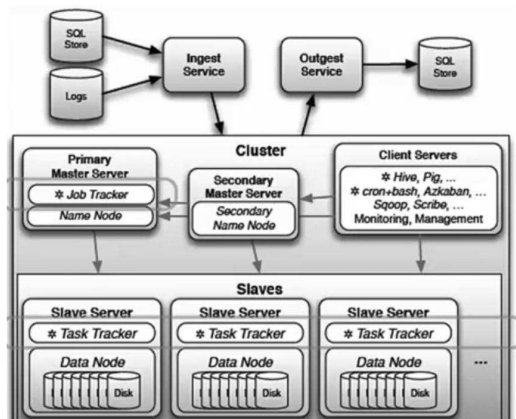
И каждая система HDFS хранит большие файлы, как правило, в диапазоне от гигабайтов до терабайтов.

И надежность системы HDFS достигается путем репликации многочисленных хостов.



Также файловая система HTFS поддерживает так называемый вторичный узел NameNote, который регулярно подключается к первичному узлу NameNote и создает снимки его состояния, запоминая, что система сохраняет в локальных и удаленных каталогах.

В каждой системе, основанной на Hadoop, содержится какая-то версия движка MapReduce.



Типичный движок MapReduce содержит средство отслеживания работы, в которое клиентские приложения могут отправлять задания MapReduce.

И этот трекер работы передает задачи всем доступным трекерам задач, которые есть в кластере.

Таким образом, классический Hadoop MapReduce представляет собой один процесс JobTracker и произвольное количество процессов TaskTracker, или по-другому один мастер узел и множество узлов slave.

MapReduce выполняет работу над огромным набором данных, обрабатывая данные и сохраняя их в HDFS таким образом, что извлечение данных производится проще, чем в традиционном хранилище.

Модель MapReduce следует принципам функционального программирования, вследствие чего пользовательские вычисления выполняются как функции `map` и `reduce`, обрабатывающие данные в виде пар ключ-значение.

Hadoop предоставляет высокоуровневый программный интерфейс для реализации пользовательских функций `map` и `reduce` на различных языках.

Также Hadoop предоставляет инфраструктуру для выполнения заданий MapReduce в виде серий задач `map` и `reduce`.

Задачи `map` вызывают функции `map` для обработки наборов входных данных.

Затем задачи `reduce` вызывают функции `reduce` для обработки промежуточных данных, сгенерированных функциями `map`, формируя окончательные выходные данные.

Задачи `map` и `reduce` выполняются изолированно друг от друга, что обеспечивает параллельность и отказоустойчивость вычислений.