Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Automation and Applied Informatics

Analyze and optimize the performance of Hive

Bachelor's Thesis

Author
Barnabas Maidics

Advisor
Peter Vary
Akos Dudas, PhD

October 5, 2018

Contents

HALLGATÓI NYILATKOZAT

Alulírott *Maidics Barnabas*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2018. október 5.

Maidics Barnabas
hallgató

Kivonat

Jelen dokumentum egy diplomaterv sablon, amely formai keretet ad a BME Villamosmérnöki és Informatikai Karán végző hallgatók által elkészítendő szakdolgozatnak és diplomatervnek. A sablon használata opcionális. Ez a sablon LaTeX alapú, a *TeXLive* TeX-implementációval és a PDF-LaTeX fordítóval működőképes.

Abstract

This document is a LaTeX-based skeleton for BSc/MSc theses of students at the Electrical Engineering and Informatics Faculty, Budapest University of Technology and Economics. The usage of this skeleton is optional. It has been tested with the *TeXLive* TeX implementation, and it requires the PDF-LaTeX compiler.

1 Introduction

The digital era has led to large amounts of data being amassed by companies every day. Data comes from multiple sources: sensors, sales data, communication systems, logging of system events etc.. According to Forbes [1] 2.5 quintillion bytes of data created each day. That means 2.5 million Terabytes per day. Bigger corporations can easily create hundreds of Terabytes a day. So we need a new solution to process this amount of data. The traditional relational databases (RDBMS) can deal only with Gigabytes. Hadoop provides a software framework to scale up our system for storing, processing and analyzing big data.

In this chapter, I will write about the basics of Hadoop architecture, why Hive was created on top if it and the performance issues it faces.

1.1 Hadoop basics

Apache Hadoop is an open source distributed framework for managing, processing and storing huge amount of data in clustered systems built from commodity hardware. All modules in Hadoop was designed with an assumption that hardware failures are common and should be automatically handled by the framework. One of the most important characteristic of Hadoop that it partitions the data and computation across many hosts and executing computation in parallel close to the data it uses. [3]

The base of the Hadoop framework contains the following modules:

- HDFS - Hadoop Distributed File System: designed to store large data sets reliably and stream those at high bandwidth to user applications.
- Hadoop MapReduce: an implementation of the MapReduce programming model for large data processing
- YARN - Yet Another Resource Negotiator: a resource management and job scheduling technology
- Hadoop Common: contains libraries and utilities needed by other Hadoop modules

1.1.1 Hadoop vs. traditional databases

Traditional databases cannot be used when we want to process and store big data. The main differences between Hadoop and traditional RDBMS:

- **Data Volume**: RDBMS works better when the data volume is low (Gigabytes). However when data size is huge (Terabytes-Petabytes) traditional databases fail. On the other hand Hadoop can easily handle this amount of data.
- **Data Variety**: this generally means the type of data to be processed. Hadoop has the ability to store and process data whether it is structured, semi-structured or unstructured. Even though it is mostly used for large amount of unstructured data. In contrast, traditional RDBMS can only be used to manage structured or semi-structured data.
- **Scalability**: RDBMS provides vertical scalability. You can add more resources, memory or CPU to a machine in the cluster. Whereas Hadoop provides horizontal scalability. It means we can add more machines to an existing cluster. As a result of this Hadoop becomes fault tolerant. We can easily recover data in case of a failure of one of the machines.
- **Data Processing**: Apache Hadoop supports OLAP (Online Analytical Processing) that involves very complex queries and aggregations. The database design is de-normalized, having fewer tables. On the other hand, RDBMS supports OLTP (Online Transaction Processing), which involves fast query processing. The database design is normalized having large number of tables. [2]

1.1.2 HDFS - Hadoop Distributed File System

1.1.3 MapReduce

1.1.4 Yarn

Acknowledgements

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Bibliography

[1] Forbes. How much data do we create every day. https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read.

[2] W3Training School. Hadoop vs rdbms. https://www.w3trainingschool.com/difference-big-data-hadoop-traditional-rdbms.

[3] Wikipedia. Apache hadoop. https://en.wikipedia.org/wiki/Apache_Hadoop.