Big Data Integration and Analysis

Requirements Specification

Prepared By:

Gaurav Kumar

Héctor Ugarte

Miguel Mármol

Tina Boroukhian

University of Bonn

5 May 2015

TABLE OF CONTENTS

1. Introduction

1.1 Purpose

This document specifies the Requirements Specification for Big Data Integration and Analysis using Apache Hadoop and Spark. It describes the scope of the system, an overall description and both functional and non-functional requirements for the software.

1.2 Project Scope

The integration will be done with only CSV files as raw data and using Apache MapReduce as framework. The analysis will be done using Apache Spark. Both tasks under Ubuntu operating system.

1.3 Client Definition

- **Client:** is the person or group of people that want to integrate and analyze certain data set.

2. Overall description

2.1 Product perspective

The Big Data Integration and Analysis project is intended using Apache Hadoop 2.6. Following the next general process:

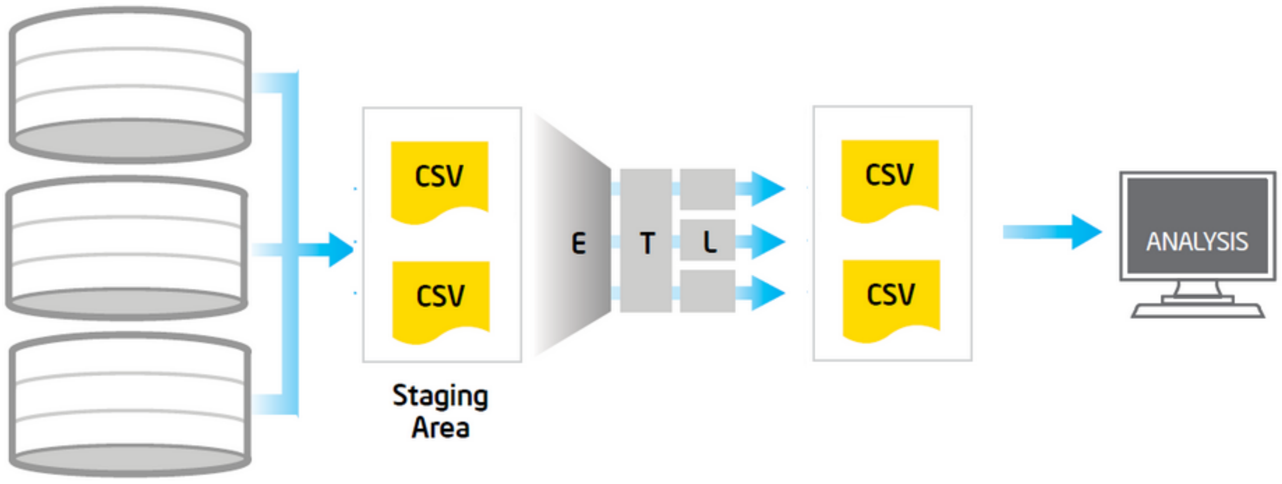

Fig 1. Overall process to be done.

- Extract data from sources. During the extract step, data is collected from different source systems and in CSV file format (flat files with delimiters).
- Transform that data into a common format that fits other data in the warehouse. The transform step may include multiple data manipulations, such as moving, splitting, translating, merging, sorting, pivoting, and more. For example, a customer name might be split into first and last names, or dates might be changed to the standard ISO format (e.g., from 07-24-13 to 2013-07-24). Often this step also involves validating the data against data quality rules.
- Load the data into the data warehouse for analysis. This step can be done in batch processes or row by row, more or less in real time.

2.2. Apache Hadoop

Apache Hadoop is an open-source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware.

The core of Apache Hadoop consists of a storage part (Hadoop Distributed File System (HDFS)) and a processing part (MapReduce). Hadoop splits files into large blocks and distributes them amongst the nodes in the cluster. To process the data, Hadoop MapReduce transfers packaged code for nodes to process in parallel, based on the data each node needs to process.

The base Apache Hadoop framework is composed of the following modules:

- Hadoop Common – contains libraries and utilities needed by other Hadoop modules;
- Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;
- Hadoop YARN – a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications; and
- Hadoop MapReduce – a programming model for large scale data processing.

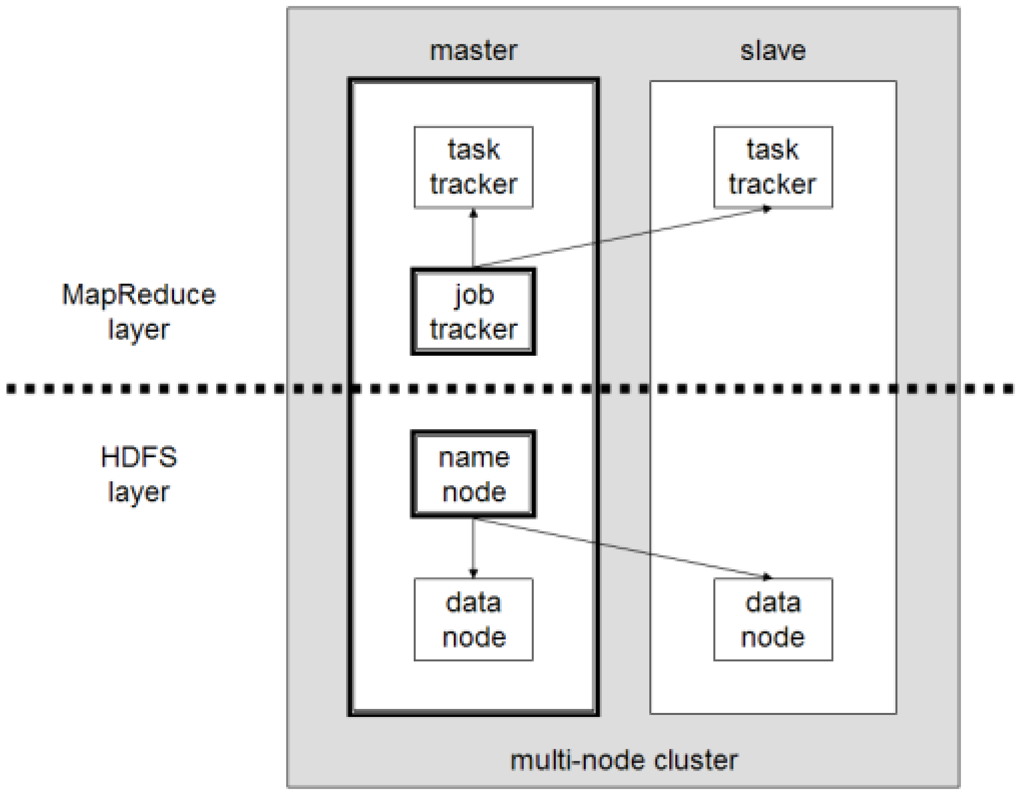

Fig 2. A multi-node Hadoop cluster architecture.

3. Specific Requirements

The following section illustrates the requirements for the project. Those requirements are divided between functional and non-functional requirements.

3.1. Functional Requirements

The functional requirements are grouped according use case model.

A requirement has the following properties:

- Requirement ID Uniquely identifies requirement within all PMS documents.
- Title Defines the functional group the requirement belongs to. Gives the requirement a symbolic name.
- Description The definition of the requirement.
- Priority Defines the order in which requirements should be implemented. Priorities are designated (highest to lowest) “1”, “2”, and “3” … Requirements of priority 1 must be implemented in the first productive system release. The requirements of priority 2 and lower are subject of special release agreement, which is out of scope of this document.
- Risk Specifies risk of not implementing the requirement. It shows how the particular requirement is critical to the system. There are following risk’s levels and

associated impact to the system if the requirement is not implemented or implemented incorrectly:

Critical (C) – will break the main functionality of the system. The system can not be used if this requirement is not implemented.

High (H) – will impact the main functionality of the system. Some function of the system could be inaccessible, but the system can be generally used.

Medium (M) – will impact some system's features, but not the main functionality. System can be used with some limitation.

Low (L) – the system can be used without limitation, but with some workarounds.

Req ID.	R01.
Title	**Get raw sample data in CSV files.**
Description	Get or collect data from our sources. Input data must be in CSV file format.
Priority	1
Risk	C
References	

Req ID.	R02.
Title	**Move data to HDFS.**
Description	Move data to HDFS (hadoop Distributed file system) so that data can be processed with MapReduce programming model.
	1
Priority	C
Risk	R01.
References	

Req ID.	R03.
Title	**Clean data.**
Description	Clean and format data in a propper manner such as standard date and correct number formatting.
Priority	1
Risk	C
References	R01, R02.

Req ID.	R04.
Title	**Reduce columns.**
Description	Reduce the columns so that we keep only needed columns for desired results.
Priority	1
Risk	M
References	R02.

Req ID.	R05.
Title	**Analyze data.**
Description	Use analytic queries on above formatted data. Analytic queries contain aggregation functions such as finding max, min, average or count.
	1
Priority	C
Risk	R04.
References	

Req ID.	R06.
Title	**Get results.**
Description	get desired result and move output from HDFS to local machine.
	1
Priority	C
Risk	R05.
References	

4. Non-functional Requirements

4.1 Scalability

The system will be designed for working with multiple computers and files.

4.2 Performance

The system will perform with little response time even with high volumes of data.

4.3 Fault Tolerant

The system will be able to behave as fault tolerant, in case of a failure of one or more machines, it will continue working.

4.4 Distributed Environment

The system will work under a distributed environment.