Contents

[HDFS Architecture 2](#_Toc482881806)

[1. Introduction 2](#_Toc482881807)

[2. Assumptions and Goals 2](#_Toc482881808)

[2.1 Hardware Failure 2](#_Toc482881809)

[2.2 Streaming Data Access 2](#_Toc482881810)

# HDFS Architecture

# Introduction

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware.  HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

# Assumptions and Goals

## Hardware Failure

Hardware failure is the norm rather than the exception.

## Streaming Data Access

Applications that run on HDFS need streaming access to their data sets. They are not general purpose applications that typically run on general purpose file systems. HDFS is designed more for batch processing rather than interactive use by users. The emphasis is on high throughput of data access rather than low latency of data access.

## Large Data sets

Applications that run on HDFS have large data sets. A typical file in HDFS is gigabytes to terabytes in size. Thus, HDFS is tuned to support large files.

## Simple Coherency Model

HDFS applications need a write-once-read-many access model for files. A file once created, written, and closed need not be changed except for appends and truncates. Appending the content to the end of the files is supported but cannot be updated at arbitrary point.

## Moving Computation is Cheaper than Moving data

A computation requested by an application is much more efficient if it is executed near the data it operates on. This is especially true when the size of the data set is huge. This minimizes network congestion and increases the overall throughput of the system.

## Portability Across Heterogeneous Hardware and Software paltforms