****

**CMP9781 Big Data Analytics & Modelling Assessment 1**

**School of Engineering and Physical Sciences**

**25847208@students.lincoln.ac.uk**

**Tarteel Alkaraan (25847208)**

**University of Lincoln**

Table of Contents

[1. Description of Distributed Big Data Processing Ecosystem 3](#_Toc190105740)

[1.1 5 V’s 3](#_Toc190105741)

[1.1.1 Volume 3](#_Toc190105742)

[**1.1.2 Velocity** 3](#_Toc190105743)

[**1.1.3 Variety** 3](#_Toc190105744)

[**1.1.4 Value** 4](#_Toc190105745)

[**1.1.5 Veracity** 4](#_Toc190105746)

[1.2 Fault Tolerance & Resilience 4](#_Toc190105747)

[**1.2.1 Fault Tolerance** 4](#_Toc190105748)

[**1.2.2 Resilience** 5](#_Toc190105749)

[1.3 Hadoop Distributed File Systems (HDFS) 5](#_Toc190105750)

[1.4 Resource Manager & Scheduler 5](#_Toc190105751)

[**1.4.1 Resource Manager** 5](#_Toc190105752)

[**1.4.2 Scheduler** 6](#_Toc190105753)

[1.5 YARN 6](#_Toc190105754)

[1.6 Apache Spark, Hive, Flume & Kafka 6](#_Toc190105755)

[**1.6.1 Apache Spark** 6](#_Toc190105756)

[**1.6.2 Apache Hive** 7](#_Toc190105757)

[**1.6.3 Apache Flume** 7](#_Toc190105758)

[**1.6.4 Apache Kafka** 7](#_Toc190105759)

[1.7 Resilient Distributed Datasets 8](#_Toc190105760)

[1.8 Data Lakes 8](#_Toc190105761)

[1.9 Spark ML 8](#_Toc190105762)

[2. Describe Real-Life Use Case for Big Data Analytics 9](#_Toc190105763)

[3. Linear Regression Analysis 10](#_Toc190105764)

[4. References 11](#_Toc190105765)

# **1. Description of Distributed Big Data Processing Ecosystem**

## **5 V’s**

## **1.1.1 Volume**

Volume of data that needs to be processed is increasing rapidly more storage capacity, computation and tools and techniques (Yu, 2025b). Think about how many devices there are currently producing data, compared to say 10 or 20 years ago. Think about what amount of data would be produced in 10 or 20 years from now. Think about the rate of growth of data (Al-Khafajiy, 2023a). Refers to amount of big data being generated at a minimum, many terabytes but also as much as petabytes (segment, 2024). It is like base of big data as it is initial size and amount of data that is collected. If volume of data is large enough, it can be considered big data (Robinson, 2023).

### **1.1.2 Velocity**

Big data is being generated fast everyday 900 million photos are uploaded to Facebook, 500 million tweets are uploaded to Twitter, 3.5 billion searches are performed on Google daily, and big data needs to be processed fast in which search results need to be returned immediately (Yu, 2025b). Data is being produced an at alarming rate but think about how we need to analyse it. We need to be timely in our analysis, which means we often must have a real-time data. Funnily enough companies used to run “batch processes” overnight to do some of this. Think about how we efficiently analyse data streams Al-Khafajiy, 2023).

### **1.1.3 Variety**

Various formats, types, and structures like text, numerical, images, audio, video, sequences, time series, social media data, and multi-dimensional arrays. Static versus streaming data, a single application can be generating and or collecting many types of data. To extract knowledge all these types of data need to be linked together (Yu, 2025b). Think about all different sensors out in world, all producing data. Consider what forms of data might be used, things like cameras, pressure sensors, and GPS trackers. Also think about how we might need not just raw data, but also data that draws in associated data from other sources (Al-Khafajiy, 2023a).

### **1.1.4 Value**

Most important V of big data, it is all well and good having access to big data but unless we can turn it into value it is useless. It is important that businesses clearly understand costs and benefits associated with big data (Yu, 2025b). Data does not have any reason to exist if we are not going to use it. But clearly we can use data to gain a variety of insights given some thought. Worth thinking about ethics at this point (Al-Khafajiy, 2023). Refers to actionable insight that can be derived from big data sets (segment, 2024). Also, refers to benefits that big data can provide, and it relates directly to what organisations can do with that collected data (Robinson, 2023).

### **1.1.5 Veracity**

Refers to messiness or trustworthiness of data. With many forms of big data, quality and accuracy are less controllable (think of twitter posts with hashtags, abbreviations, typos, colloquial speech, reliability, and accuracy of content). But big data and analytics technology now allows us to work with these types of data (Yu, 2025b). Refers to inconsistencies and uncertainty in data, that is data which is available can sometimes get messy and quality and accuracy are difficult to control. Think about what would happen if we allowed intrusion into GPS network and flight systems were entirely automated. Think about how we determine valid data from invalid data (Al-Khafajiy, 2023a).

## **1.2** **Fault Tolerance & Resilience**

### **1.2.1 Fault Tolerance**

Fault Tolerance is a process that enables an operating system to respond to a failure in hardware or software. This fault-tolerance definition refers to the system’s ability to continue operating despite failures or malfunctions (Fortinet, 2025). Fault Tolerance refers to the ability of a system (computer, network, and cloud cluster) to continue operating without interruption when one or more of its components fail (Imperva, 2024). Fault Tolerance is the ability for a system or application to continue service without interruption in the event of a hardware or software failure. The goal of fault tolerance is to prevent a fault from manifesting itself as a failure (Lahti, 2005).

### **1.2.2 Fault Resilience**

Fault Resilience is the ability of a software system to recover from failures and continue to function correctly. It does not mean avoiding failures but rather minimising their impact and frequency (Linkedin, 2025). Fault Resilience points towards systems that will respond gracefully to not yet known failure modes, develop emergent resilient behaviour, adapt to changing failure surfaces or even transform based on them (Friedrichsen, 2022). Fault Resilience is the ability of a system to withstand adversities and bounce back to a normal state. In short, it refers to the ability to adapt to difficulties and recover from unexpected events (Kafka, 2024).

## **1.3 Hadoop Distributed File Systems (HDFS)**

HDFS is specially designed for storing huge datasets in commodity hardware. HDFS contains two types of nodes name and data. In HDFS a large file is split into one or more blocks and these blocks are stored in a set of DataNodes (Yu, 2025a). HDFS is a distributed file system, designed to be highly fault tolerant. HDFS splits data into chunks and distributes it across the machines. HDFS is designed to detect and recover from faults quickly, and assumes every node is unreliable (Wingate, 2023). HDFS is a file system that manages large data sets that can run on commodity hardware. HDFS is most popular data storage system for Hadoop and can be used to scale single Apache Hadoop cluster (Holdsworth, 2024).

## **1.4 Resource Manager & Scheduler**

### **1.4.1 Resource Manager**

Resource Manager is defined as the component responsible for managing the interaction with resources across multiple clouds in a federation. It abstracts and maps resources as a unified collection, utilising APIs provided by cloud providers or open standards for management (Assis, 2016). Resource Manager acts as a scheduler that schedules and prioritises processing requests according to individual processing workload requirements (Arcitura, 2025). Resource Manager is responsible for keeping track of the data with the data structures of the database, to allow for efficient retrieval of data from storage via the buffer and storage manager (Noergaard, 2010).

### **1.4.2 Resource Scheduler**

Resource Scheduler is process of identifying when project resources are needed and allocating them based on factors such as capacity planning or resource availability. Main purpose of resource scheduler is to guarantee that there’s no over or under-allocation of resources at any point of project (Landau, 2024). Resource Scheduler uses a set of APIs to create work requests to complete specific tasks such as creating, updating, starting, stopping, enabling, disabling, and deleting schedules (Oracle, 2024). Resource Scheduler helps allocate tasks, manage time, and optimise the use of available resources, ensuring that projects run efficiently and meet deadlines (Świtek, 2025).

## **1.5 YARN**

Apache Hadoop Yet Another Resource Negotiator (YARN) is a resource management system. It provides unified resource management and scheduling for upper-layer applications, remarkably improving cluster resource utilisation, unified resource management, and data sharing (Al-Khafajiy, 2023b). Yarn introduced in Hadoop 2 for better resource allocation and scalability. Yarn splits resource management and job scheduling/monitoring into separate processes. This enables Hadoop to move beyond MapReduce and do all sorts of things (Wingate, 2023). YARN is resource management and job scheduling technology in open source Hadoop distributed processing framework (Stedman, 2020).

## **1.6 Apache Spark, Hive, Flume & Kafka**

### **1.6.1 Apache Spark**

Apache Hadoop/Spark is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. Apache designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures. Apache spark is a unified analytics engine for large-scale data processing (Al-Khafajiy, 2023b).

### **1.6.2 Apache Hive**

Apache Hive data warehouse software facilitates reading, writing, and managing large datasets residing in distributed storage using SQL. Structure can be projected onto data already in storage. A command line tool and JDBC driver are provided to connect users to Hive (Al-Khafajiy, 2023b). Apache Hive is an open-source data warehouse software designed to read, write, and manage large datasets extracted from HDFS (Databricks, 2025). Apache Hive is a distributed, fault-tolerant data warehouse system that enables analytics data at a massive scale. Hive is built on top of Apache Hadoop and allows users to read, write, and manage petabytes of data using SQL (Apache, 2023).

### **1.6.3 Apache Flume**

Apache Flume is a distribute, reliable, available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms (Flume, 2023). Flume is a tool/service/data ingestion mechanism for collecting, aggregating, and transporting large amounts of streaming data from various sources to a centralised data store. Flume is a highly reliable, distributed, and configurable tool. It is designed to copy streaming data from various web servers to HDFS (TutorialsPoint, 2025).

### **1.6.4 Apache Kafka**

Apache Kafka is an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, mission-critical applications. More than 80 percent of all Fortune 100 companies trust and use Kafka (Al-Khafajiy, 2023b). Kafka is an example of what is called a publisher/subscriber messaging system aka pubsub. The point of Apache Kafka is to act as a central event/message queue for real-time applications. Kafka messages aka events are separated into topics. Kafka topics are also partitioned so they can reside across a cluster and be parallelised (Al-Khafajiy, 2023c).

## **1.7 Resilient Distributed Datasets**

## **1.8 Data Lakes**

## **1.9 Spark ML**

Spark ML is the machine learning module of Apache Spark, an open-source distributed computing system designed to process large-scale data. And PySpark ML is the Python library for Spark ML, which makes it simpler to work with Spark using Python (Barjatiya, 2023). Spark ML

# **2. Describe Real-Life Use Case for Big Data Analytics**

# **3. Linear Regression Analysis**

# **4. References**

Al-Khafajiy, D. M. (2023a) Lecture 1 Introduction to Big Data.

Al-Khafajiy, D. M. (2023b) Lecture 2 Frameworks.

Al-Khafajiy, D. M. (2023c) Lecture 7 Data Stream Mining.

Apache (2023) *Apache Hive*. Available from <https://hive.apache.org/>.

Arcitura (2025) *Resource Manager*. Available from <http://patterns.arcitura.com/big-data-patterns/mechanisms/resource_manager#:~:text=A%20resource%20manager%20acts%20as,the%20example%20in%20Figure%201>.

Assis, M. R. M., Bittencourt, L. F. (2016) A survey on cloud federation architectures: Identifying functional and non-functional properties. *Journal of Network and Computer Applications*, 72 51-71. Available from <https://www.sciencedirect.com/science/article/pii/S1084804516301436> [accessed 2016/09/01/].

Barjatiya, P. (2023) *Unleashing the Power of Machine Learning with Spark ML and PySpark ML*. Available from <https://medium.com/data-and-beyond/unleashing-the-power-of-machine-learning-with-spark-ml-and-pyspark-ml-9120697c0745#:~:text=Spark%20ML%20is%20the%20machine,work%20with%20Spark%20using%20Python>.

Databricks (2025) *Apache Hive*. Available from <https://www.databricks.com/glossary/apache-hive>.

Flume (2023) *Apache Flume*. Available from <https://flume.apache.org/>.

Fortinet (2025) *What Is Fault Tolerance?* Available from <https://www.fortinet.com/uk/resources/cyberglossary/fault-tolerance#:~:text=Fault%20tolerance%20is%20a%20process,operating%20despite%20failures%20or%20malfunctions>.

Friedrichsen, U. (2022) *Resilience vs. Fault tolerance*. Available from <https://www.ufried.com/blog/resilience_vs_fault_tolerance/>.

Holdsworth, J. (2024) *What is Hadoop Distributed File System (HDFS)?* Available from <https://www.ibm.com/think/topics/hdfs>.

Imperva (2024) *Fault Tolerance*. Available from <https://www.imperva.com/learn/availability/fault-tolerance/>.

Kafka, A. (2024) *Fault Tolerance and Resiliency in Apache Kafka: Safeguarding Data Streams*. Available from <https://www.ksolves.com/blog/big-data/apache-kafka/fault-tolerance-and-resiliency-in-apache-kafka>.

Lahti, C. B., Peterson, Roderick (2005) Chapter 7 - Domain III: Delivery and Support. *Sarbanes-Oxley IT Compliance Using COBIT and Open Source Tools*. Burlington: Syngress, 175-221.

Landau, P. (2024) *Resource Scheduling in Project Management*. Available from <https://www.projectmanager.com/blog/better-resource-scheduling#:~:text=Resource%20scheduling%20is%20the%20process,any%20point%20of%20the%20project>.

Linkedin (2025) *How do you test and monitor the resilience of your software systems?* Available from <https://www.linkedin.com/advice/0/how-do-you-test-monitor-resilience-your-software#:~:text=Resilience%20does%20not%20mean%20avoiding,to%20changing%20conditions%20and%20requirements>.

Noergaard, T. (2010) Chapter 7 - An Introduction to the Fundamentals of Database Systems. In: T. Noergaard (ed.) *Demystifying Embedded Systems Middleware*. Burlington: Newnes, 305-328.

Oracle (2024) *Resource Scheduler Overview*. Available from <https://docs.oracle.com/en-us/iaas/Content/resource-scheduler/concepts/resourcescheduler-overview.htm#:~:text=Resource%20Scheduler%20uses%20a%20set,%2C%20disabling%2C%20and%20deleting%20schedules>.

Robinson, S. (2023) *5V's of big data*. Available from <https://www.techtarget.com/searchdatamanagement/definition/5-Vs-of-big-data>.

Segment, T. (2024) *Understanding the 5 V's of Big Data*. Available from <https://segment.com/data-hub/big-data/characteristics/#:~:text=Big%20data%20is%20often%20defined,variety%2C%20veracity%2C%20and%20value>.

Stedman, C. (2020) *Apache Hadoop YARN*. Available from <https://www.techtarget.com/searchdatamanagement/definition/Apache-Hadoop-YARN-Yet-Another-Resource-Negotiator>.

Świtek, M. (2025) *Resource Scheduler*. Available from <https://teamdeck.io/resources/resource-scheduler/>.

Tutorialspoint (2025) *Apache Flume - Quick Guide*. Available from <https://www.tutorialspoint.com/apache_flume/apache_flume_quick_guide.htm>.

Wingate, J. (2023) Lecture 8 Apache Hadoop.

Yu, D. M. (2025a) Hadoop Part 1.

Yu, D. M. (2025b) Introduction of Big Data.