Rishabh Johri

[rjohri@deloitte.com](mailto:rjohri@deloitte.com)

Theory Write up : Retail Analytics

**Part 1 – Big Data and Hadoop Basics**

**1. What is Big Data and Why it Matters**

Big Data refers to extremely large and complex datasets that cannot be effectively processed, stored, or analyzed using traditional database systems. It is characterized by the **5 Vs**:

* **Volume** – Massive data sizes (terabytes to petabytes).
* **Velocity** – High-speed generation and streaming (IoT sensors, clickstreams).
* **Variety** – Structured (tables), semi-structured (JSON/XML), and unstructured (video, text).
* **Veracity** – Ensuring data quality, accuracy, and trustworthiness.
* **Value** – Deriving meaningful insights that impact decision-making.

**Importance for enterprises**:

* Enables data-driven decision making and competitive advantage.
* Improves operational efficiency by detecting inefficiencies and bottlenecks.
* Supports personalization and customer engagement through recommendation engines and predictive analytics.
* Helps in fraud detection, compliance monitoring, and risk management.

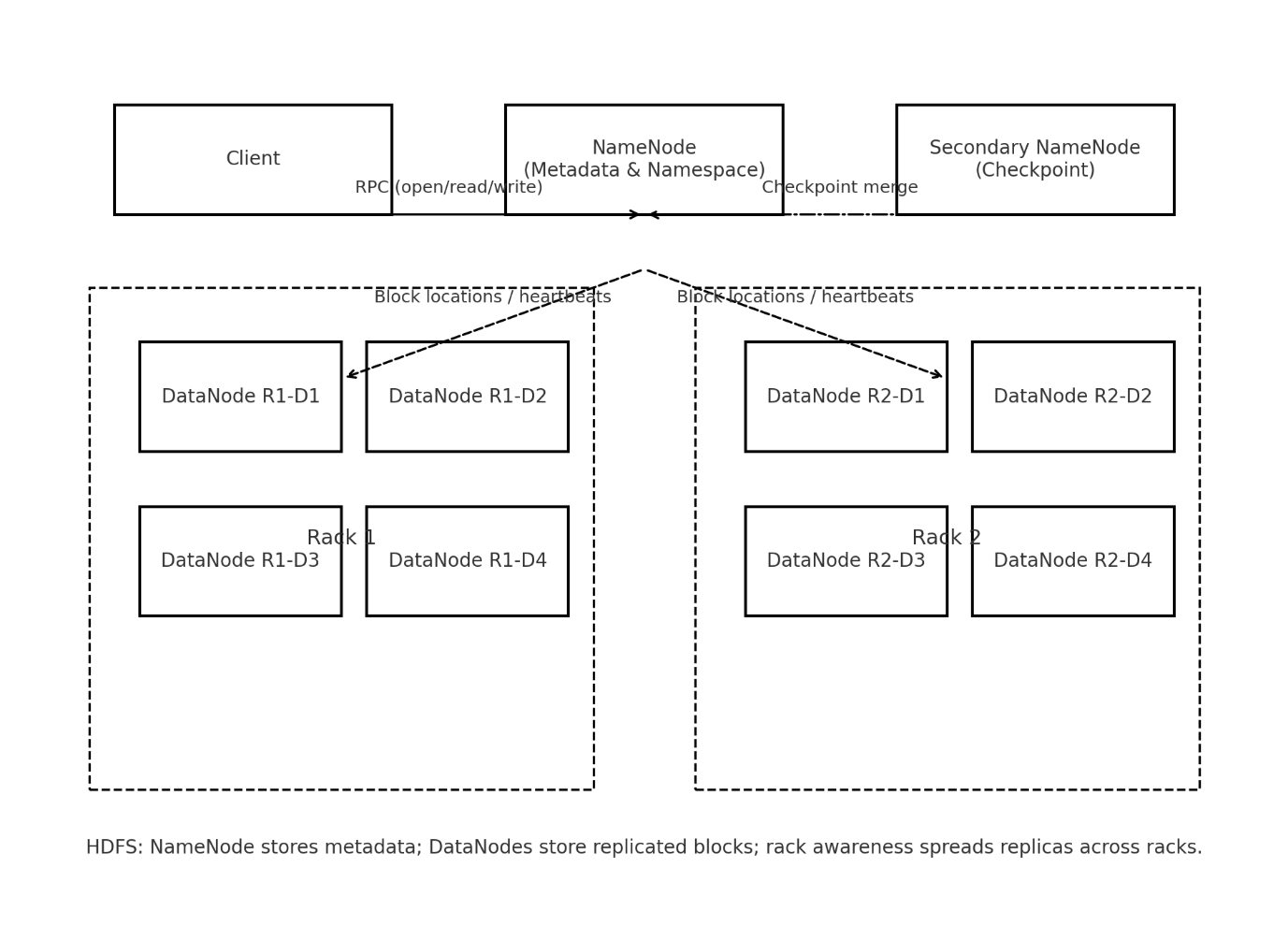
**Real-world use cases**:

* Retail: Dynamic pricing, demand forecasting, recommendation engines.
* Finance: Fraud detection, credit scoring, algorithmic trading.
* Healthcare: Predictive diagnostics, genome sequencing, drug discovery.
* Manufacturing: Predictive maintenance, IoT-driven supply chain analytics.
* Smart Cities: Traffic management, energy optimization, surveillance analytics.
* Social Media & Entertainment: Content recommendations, sentiment analysis, trend detection.

**2. Hadoop Ecosystem Components**

The **Apache Hadoop ecosystem** provides a framework for distributed storage and large-scale processing of Big Data across commodity hardware clusters.

1. **HDFS (Hadoop Distributed File System)**
   * A fault-tolerant, distributed storage system that splits files into fixed-size blocks (typically 128 MB) and replicates them across nodes for reliability.
   * Provides high throughput, optimized for batch processing.
2. **YARN (Yet Another Resource Negotiator)**
   * The cluster resource management layer in Hadoop.
   * **ResourceManager** allocates CPU/memory resources across applications.
   * **NodeManagers** run on worker nodes, monitoring resource usage and reporting to ResourceManager.
   * Allows multiple processing engines (MapReduce, Spark, Hive, etc.) to share the same cluster.
3. **MapReduce**
   * A programming paradigm for parallel data processing.
   * **Map phase**: input data is split into key-value pairs and processed in parallel.
   * **Shuffle phase**: intermediate results are grouped and redistributed.
   * **Reduce phase**: aggregates or computes final results.
   * Example: Counting word frequencies across terabytes of text.

Together, **HDFS** handles storage, **YARN** manages resources, and **MapReduce** executes distributed computation.

**3. HDFS Architecture and Data Storage/Retrieval**

**Core components**:

* **NameNode (Master)**: Maintains metadata (namespace, block locations, permissions).
* **DataNodes (Workers)**: Store actual data blocks on local disks.
* **Secondary NameNode**: Periodically checkpoints NameNode metadata (not a hot standby).

**Data Storage Process**:

1. A client writes a file to HDFS.
2. The file is split into blocks (128 MB default).
3. Each block is replicated (default factor: 3) across multiple DataNodes for fault tolerance.
4. The NameNode records block locations but never stores the data itself.

**Data Retrieval Process**:

1. A client requests a file.
2. The NameNode returns block location metadata.
3. The client directly contacts the respective DataNodes in parallel to fetch blocks.
4. Blocks are assembled back into the original file at the client side.

**Fault Tolerance**:

* If a DataNode fails, HDFS automatically retrieves blocks from replicas on other nodes.
* Rack-aware placement ensures replicas are distributed across racks, minimizing data loss risk.

**Advantages**:

* Scalability: Supports thousands of nodes.
* High Availability: Block replication protects against hardware failures.
* Cost Efficiency: Runs on commodity hardware.

