



BUILDING UP ENTERPRISE BIG DATA LAKE

Bin Jiang
10/29/2017

Introduction to the Course

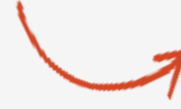
- Myself
- Objective
- Who this Course is for
- Prerequisite Skills
- 10+ Frameworks
- Course Outlines and Schedules
- 600+ Topics
- 100+ Labs
- 80+ Best Practices
- 8 Labs/Mini Projects
- 500+ Interview Questions
- Job Market

About Me

The 15+ years of extensive work experiences of all phases of software development with Invesco, RBC, CIBC, CGI and ING to lead the teams on creating enterprise application architecture: Portal, SOA, ESB, EAI, BPM and ECM/CMS.

- 1 **Principal big data engineer** of one of big financial groups in Toronto.
- 2 **Big data practitioner** in architecting and implementing **real time advanced big data analytics** applications, **cyber security detection** and **recommendation engine**.
- 3 Strong hands-on experiences on building up enterprise big data platform with **Hadoop ecosystem** and delivering the big data solutions.

SELECT ME



Objective

- Explore various approaches to starting and growing a Data Lake, the course shows you methods for setting up different tiers of data, from raw untreated landing areas to carefully managed and summarized data. You'll learn how to enable self-service to find, understand, and provision data; how to provide different interfaces to users with different skill levels; and how to do all of that in compliance with enterprise data governance policies
- Learn to build various tiers of a Data Lake, such as data intake, management, consumption, and governance, with a focus on practical implementation scenarios
- Find out the key considerations to be taken into account while building each tier of the Data Lake

Objective

- Understand Hadoop-oriented data transfer mechanism to ingest data in batch, micro-batch, and real-time modes
- Explore various data integration needs and learn how to perform data enrichment and data transformations using Big Data technologies
- Guide you in developing Data Lake's capabilities. It will focus on architect data governance, security, data quality, data lineage tracking, metadata management and semantic data tagging. By the end of this book, you will have a good understanding of building a Data Lake for Big Data

Who this Course is for

Whoever is looking for the following most in-demand big data roles as:

Big Data ETL Developer



Big Data Engineer



Big Data Security Expert

Big Data Governance Steward



Big Data Platform Support



Big Data Architect



Big Data Administrator



Prerequisite Skills

<hr/>	
Hadoop	Knowledge
Programming	Nice to Have Java, Scala and Python
SQL	Must
ETL	Nice to Have
<hr/>	

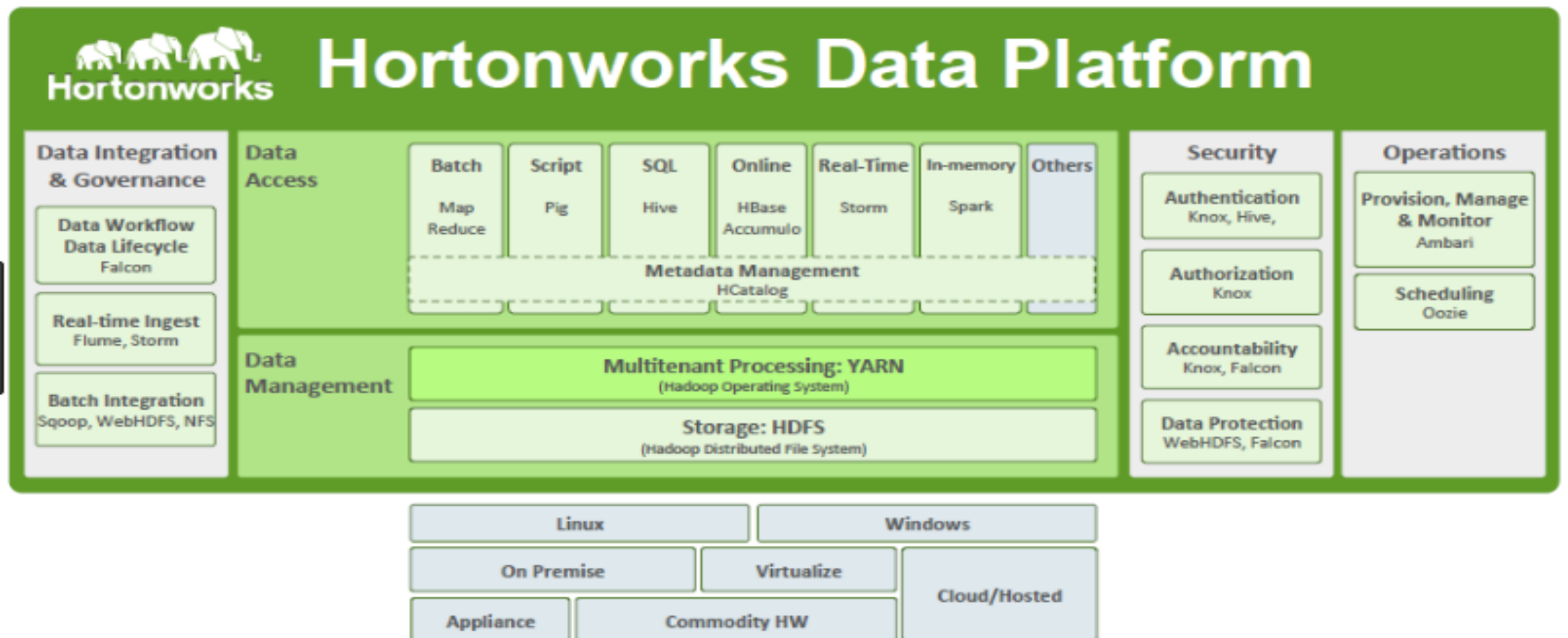
10+ Frameworks

Apache **Atlas**



Course Outlines and Schedules

Class 1 - Big Data Architecture



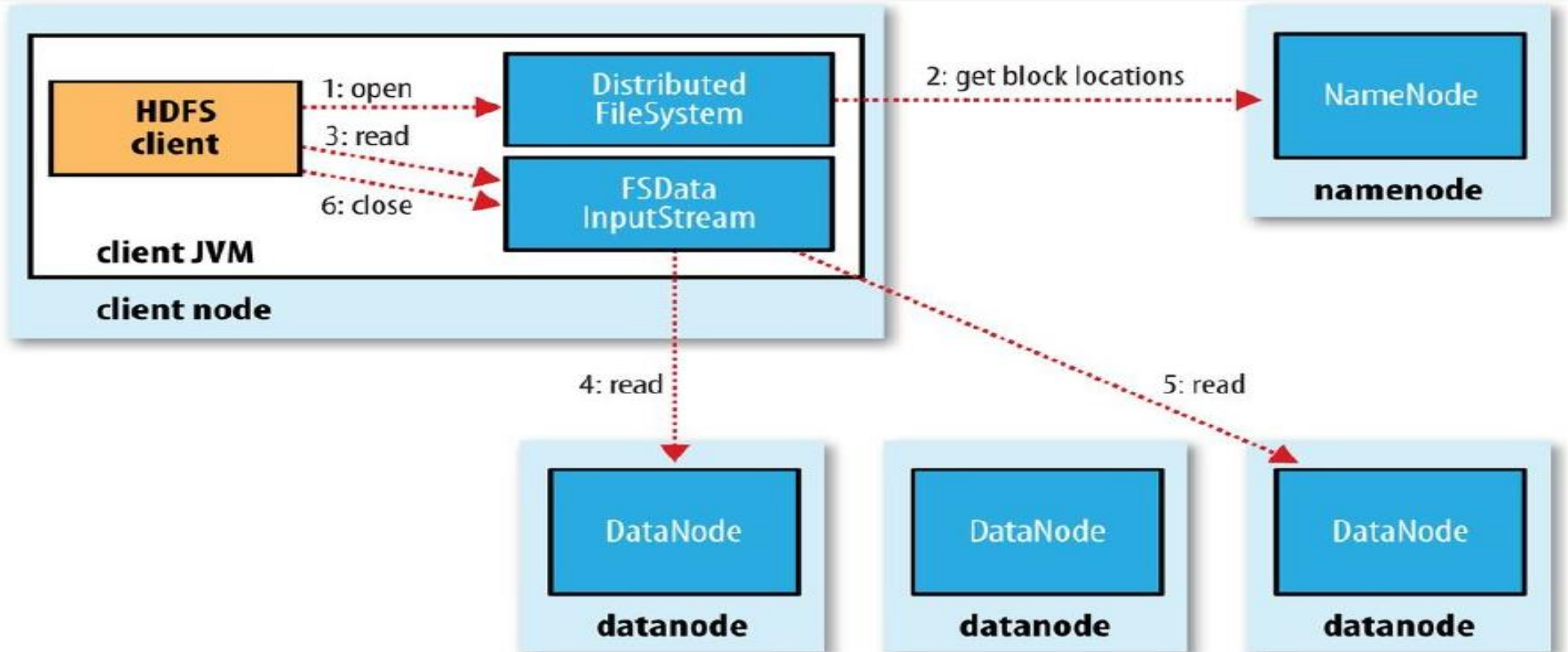
Course Outlines and Schedules

Class 2 - HDFS, MapReduce and YARN

- **HDFS Federation**
- **HDFS HA**
- **HDFS Failover and Fencing**
- **HDFS Dataflow**
- **HDFS Coherency Model**
- **HDFS Permission**
- **How does MapReduce work**
- **YARN Capacity Scheduler**
- **Resource Manager HA**
- **Anatomy of a YARN Application Run**

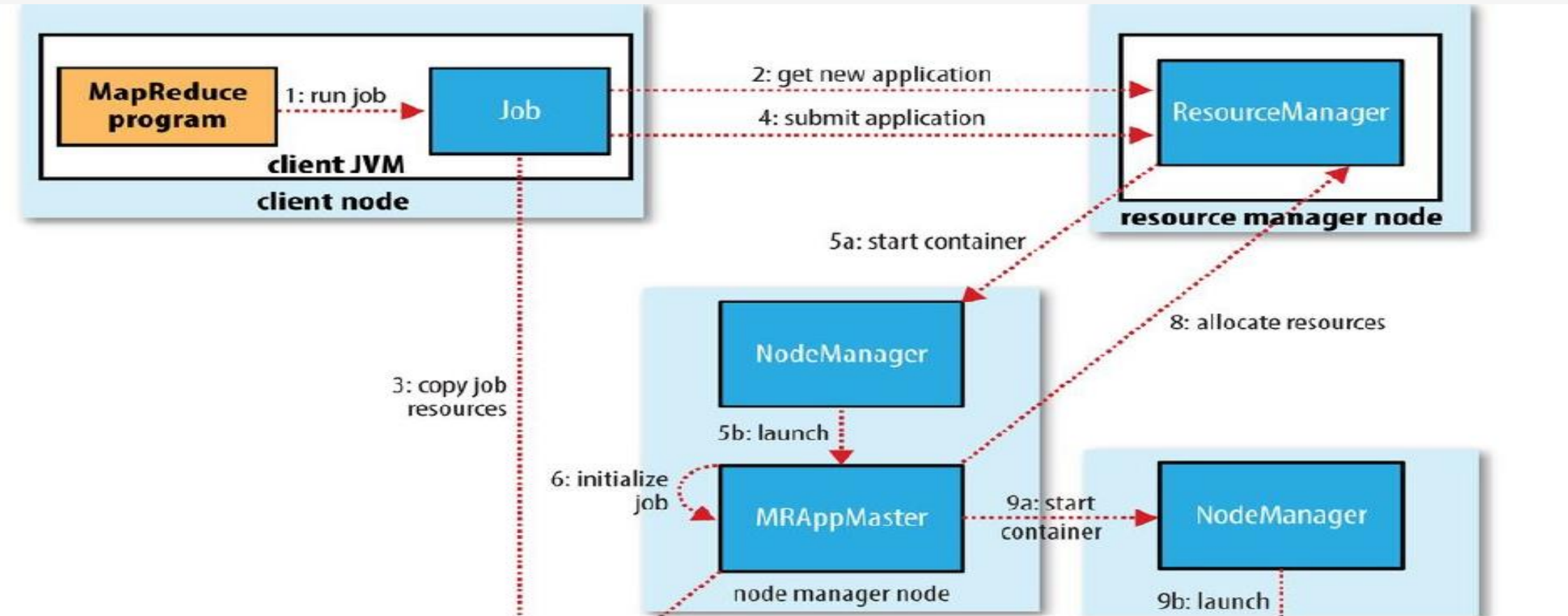
Course Outlines and Schedules

Class 2 - HDFS, MapReduce and YARN



Course Outlines and Schedules

Class 2 - HDFS, MapReduce and YARN



Course Outlines and Schedules

Class 3 - Hive - Big Data Warehouse Framework

- **Hive Architecure**
- **HiveQL**
- **Working with Complex Data Type – Map, Struct, Array and Union**
- **Table Partitioning**
- **Bucket**
- **Data Serializer and Deserizlizer**
- **Data Format Conversion**
- **Multi Table Inserts**
- **Data Exchange**
- **Join**
- **Advanced Aggregation – GROUPING SETS, ROLLUP and CUBE**
- **Analytics Functions**
- **Securing Hive**
- **Extending Hive**

Course Outlines and Schedules

Class 4 - Sqoop - RDBMS to Big Data Migration Tool

- **Sqoop Import and Export**
- **Fine-tuning Data Import**
- **Controlling the Number of Import Processes**
- **Data Splitting**
- **Changing Data Type**
- **File Formats**
- **Incremental Imports**
- **Sqoop Job and Metastore**
- **Controlling Distributed Cache**
- **Protecting Password**
- **Best Practices**

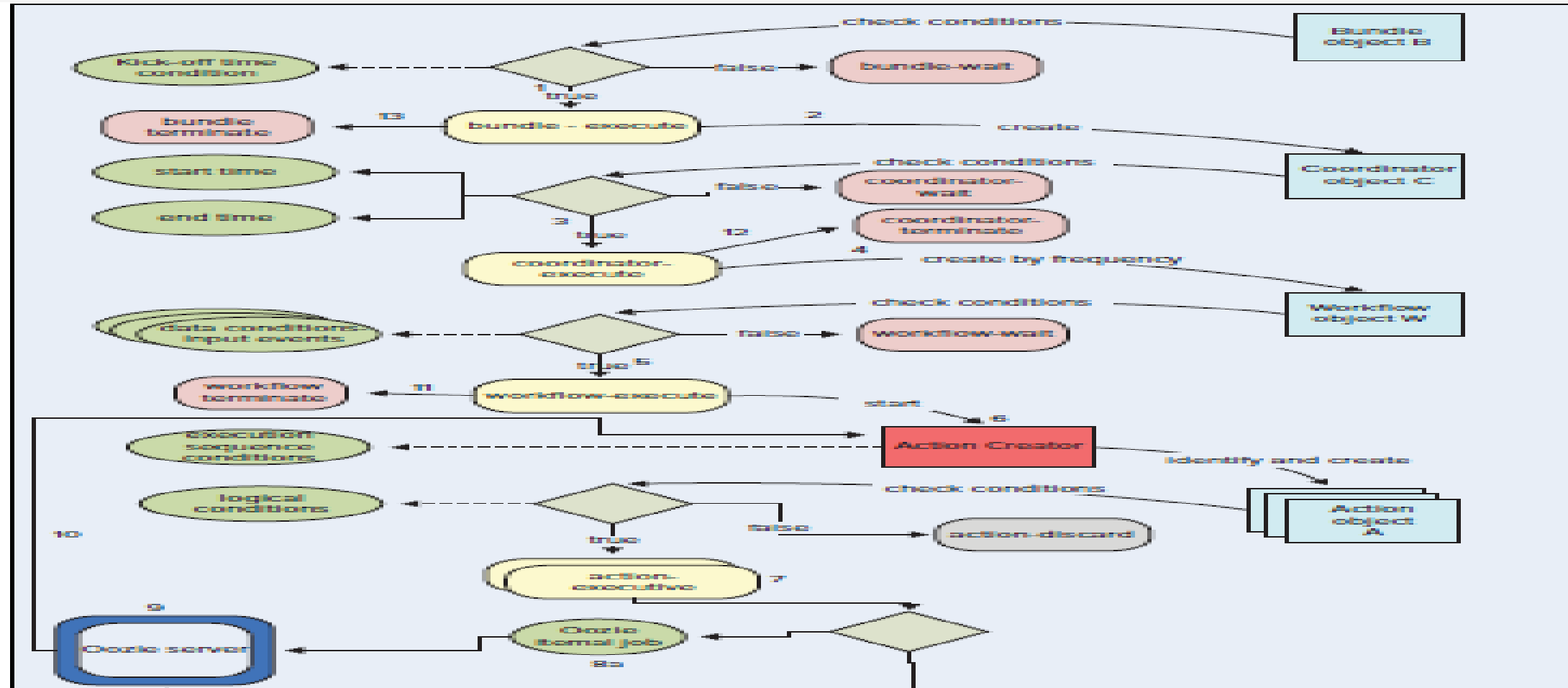
Course Outlines and Schedules

Class 5 - Oozie and Falcon - Big Data Workflow Scheduler and Data Governance

- **Basics of Actions**
- **Oozie Execution Model**
- **Action Sync vs Async**
- **Oozie Subworkflow**
- **Oozie and Hcatalog**
- **EL Functions**
- **Workflow Lifecycle**
- **Configuration and Parameterization**
- **Capture output**
- **Enterprise Oozie Application Architecture**
- **Time-based Trigger and Data-based Trigger**
- **Pass Arguments to Oozie Jobs**
- **Oozie Library**
- **Supporting Uber Jar**
- **Cron Scheduling**
- **Emulating Async Data Processing**
- **Oozie SLA**

Course Outlines and Schedules

Class 5 - Oozie and Falcon - Big Data Workflow Scheduler and Data Governance



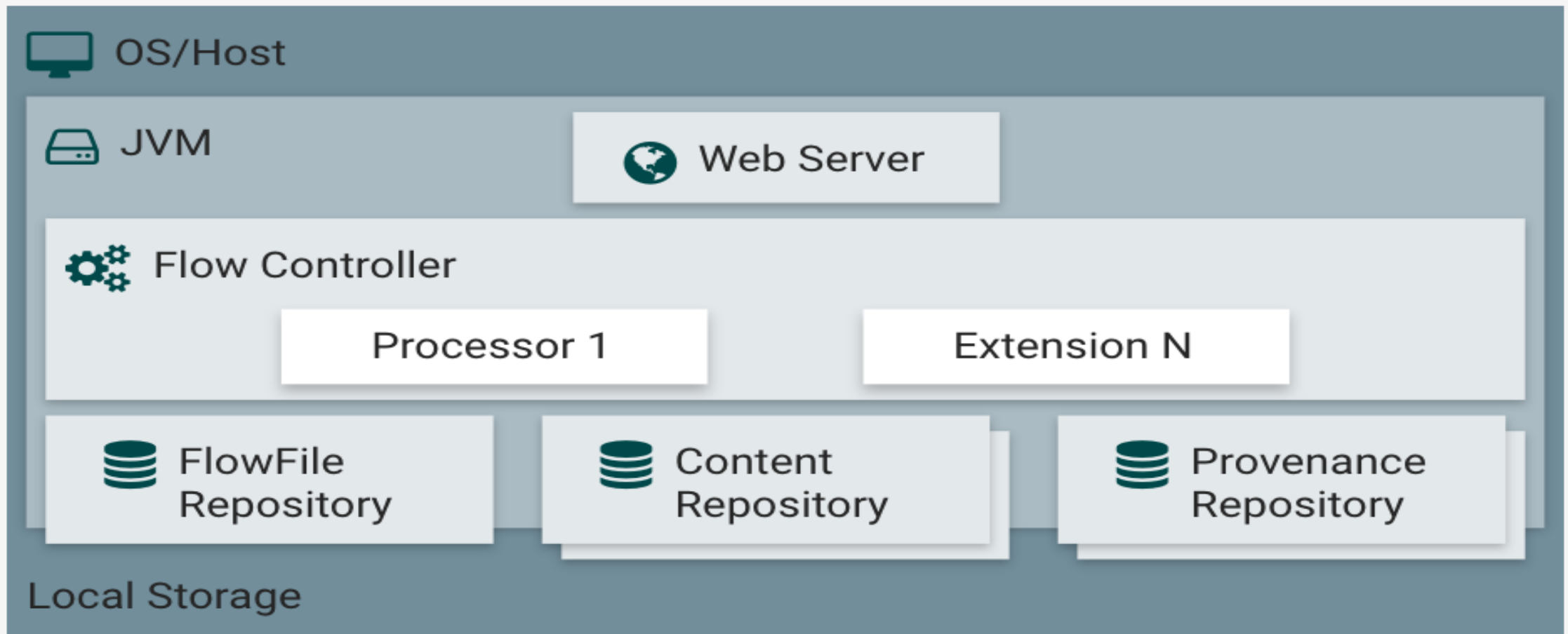
Course Outlines and Schedules

Class 6 - Nifi - Dataflow Management Platform

- **Nifi FlowFile and Processor**
- **Nifi Process Group**
- **Flow Controller**
- **Controller Services**
- **Reporting Task**
- **Site-to-Site**
- **Working with Processor**
- **Extending Nifi**
- **StateManager**
- **Securing Nifi**
- **Data Governance**
- **Clustering**
- **Nifi Repositories**
- **Performance Consideration**
- **Classloader Isolation**
- **Nifi Package**

Course Outlines and Schedules

Class 6 - Nifi - Dataflow Management Platform



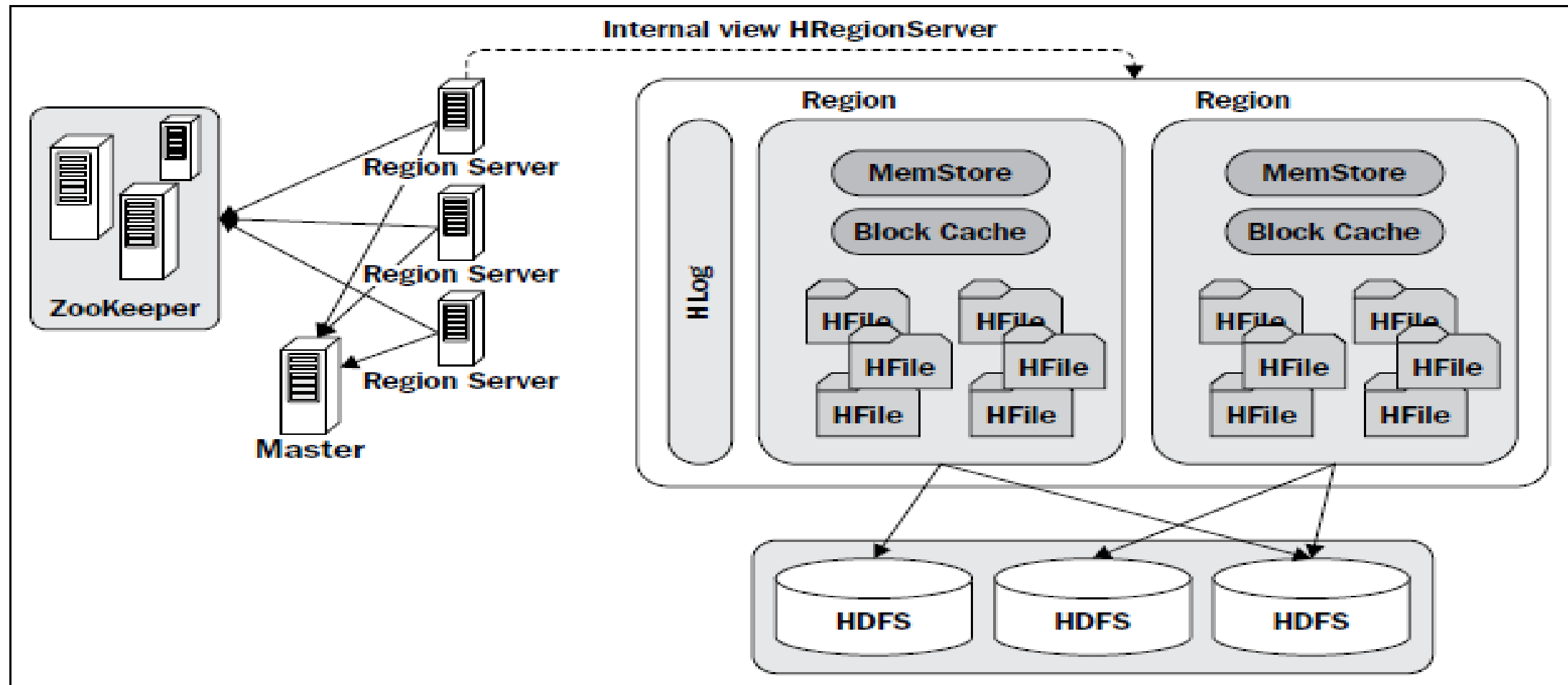
Course Outlines and Schedules

Class 7 - HBase - Hadoop NoSQL Database

- **HBase Architecture and Design**
- **Block Cache, WAL, MemStore and HFile**
- **HBase Replication**
- **HBase HA**
- **HBase Data Modeling**
- **HBase Table Design**
- **HBase Compaction**
- **Accessing HBase**
- **Securing HBase**
- **Advanced HBase APIs**
- **HBase, Spark and Hive Integration**
- **HBase Sizing and Tuning**
- **HBase Trouble Shooting**
- **HBase Design Pattern**
- **HBase Secondary Index and Solr**

Course Outlines and Schedules

Class 7 - HBase - Hadoop NoSQL Database



Course Outlines and Schedules

Class 8 - Kylin - OLAP Engine for Big Data

- **Model**
- **Batch CUBE Design, Build and Deploy**
- **Streaming CUBE Design, Build and Deploy**
- **Optimize CUBE**
- **Connectivity and RESTful APIs**
- **Integrate Kylin with MS Excel and BI tools such as Tableau, Power BI**
- **Enable Security with LDAP**

Course Outlines and Schedules

Class 8 - Kylin - OLAP Engine for Big Data

The screenshot displays the Apache Kylin web interface for editing a cube. The browser address bar shows `localhost:7070/kylin/cubes/edit/streaming_stocks_cube`. The interface includes a top navigation bar with tabs for 'Insight', 'Model' (selected), 'Monitor', and 'System'. A sidebar on the left shows a '+ New' button and a list of models, including 'streaming_stocks_model'. The main area is titled 'Cube Designer' and features a progress bar with seven steps: 1. Cube Info (completed), 2. Dimensions (active), 3. Measures, 4. Refresh Setting, 5. Advanced Setting, 6. Configuration Overwrites, and 7. Overview. Below the progress bar, there are buttons for '+ Add Dimension' and 'Auto Generator', along with a 'Filter ...' search box. A table lists the dimensions for the cube:

ID	Name	Table Name	Type	Column	DAY_START	Actions
1	DAY_START	DEFAULT.STREAMING_STOCKS_TABLE	normal	Column	DAY_START	
2	HOUR_START	DEFAULT.STREAMING_STOCKS_TABLE	normal	Column	HOUR_START	
3	MINUTE_START	DEFAULT.STREAMING_STOCKS_TABLE	normal	Column	MINUTE_START	
4	TICKER	DEFAULT.STREAMING_STOCKS_TABLE	normal	Column	T	
5	EXCHANGE	DEFAULT.STREAMING_STOCKS_TABLE	normal	Column	E	

At the bottom of the table, there are 'Prev' and 'Next' navigation buttons. The footer of the interface shows 'Apache Kylin | Apache Kylin Community' and a Windows taskbar at the very bottom with the time '10:25 PM' and date '2017-03-17'.

Course Outlines and Schedules

Class 9 - Advanced Kafka - Distributed Streaming Platform

- **Master various Kafka components – consumer, producer and brokers**
- **Perform different operations on topic**
- **Integrate Kafka with various consumers**
- **Play with Kafka partitions and distribute data between them**
- **Understand insights of high & low level Kafka APIs**
- **Learn the concepts of latest version of Kafka**
- **Learn Zookeeper**
- **Learn Kafka Best Practices**
- **Master balancing of Kafka Cluster**
- **Understand Replication and its importance in Kafka**
- **Develop Live Kafka Project**

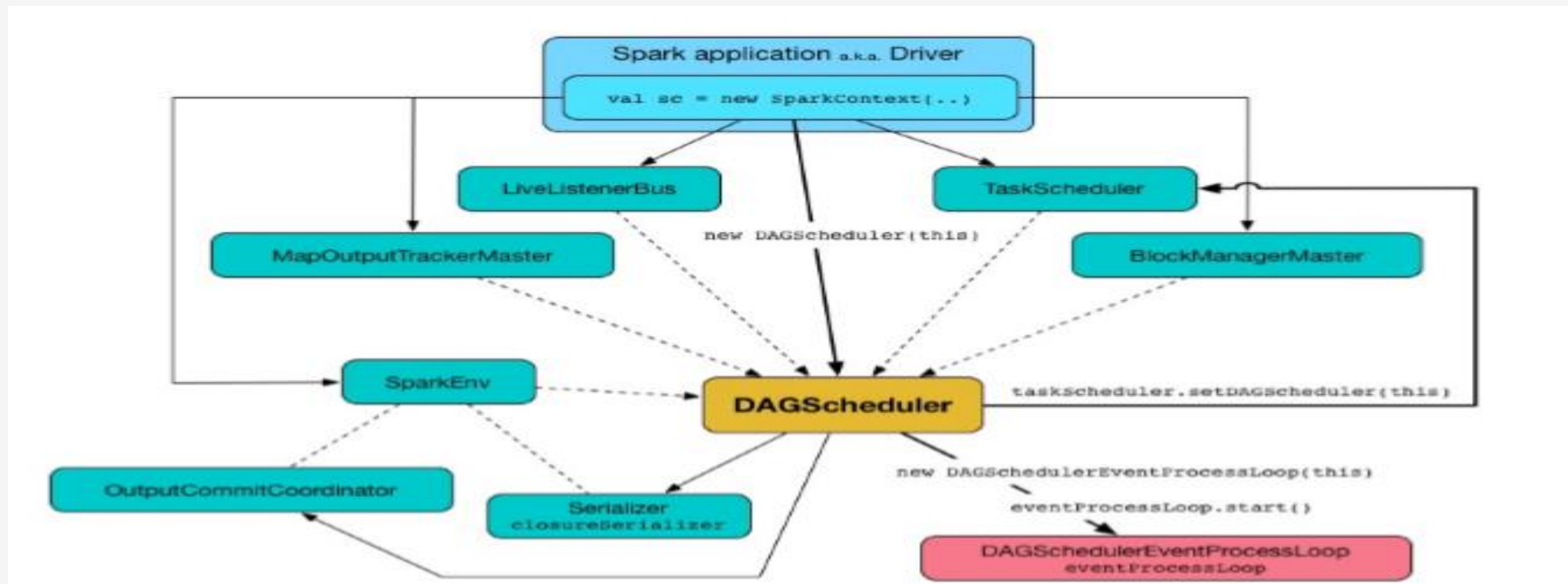
Course Outlines and Schedules

Class 10 - Advanced Spark ETL - Spark Core

- **Spark Architecture**
- **Spark RDD Linage – Login Execution Plan**
- **Spark Execution Model**
- **Spark Memory Management**
- **Spark Runtime Environment**
- **Spark Checkpoint**
- **Spark On YARN**
- **Spark Optimization**
- **Spark Services**
- **Spark Deployment**
- **Spark Tools**

Course Outlines and Schedules

Class 10 - Advanced Spark ETL - Spark Core



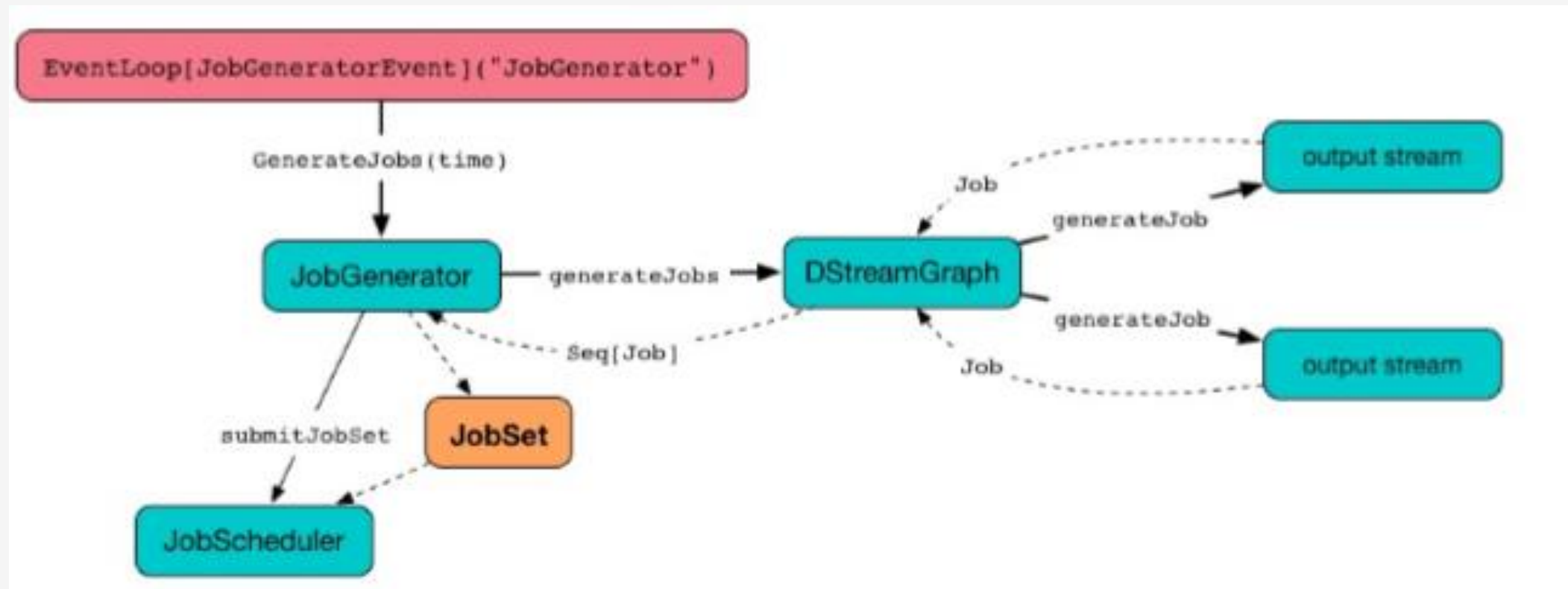
Course Outlines and Schedules

Class 11 - Advanced Spark ETL - Spark Streaming

- **StreamingContext**
- **Streaming Checkpoint**
- **Streaming Listeners**
- **DStream and DStreamGraph**
- **Job Scheduler and Generator**
- **Stateful and Windows Operation**
- **Receiver Tracker and Supervisor**
- **Direct Kafka Stream, Kafka RDD, OffsetRange and LocationStrategy**
- **Advanced File Streaming**
- **Dynamic Allocation**
- **Back Pressure**

Course Outlines and Schedules

Class 11 - Advanced Spark ETL - Spark Streaming



Course Outlines and Schedules

Class 12 - Advanced Spark ETL - Spark SQL

- **SparkSession, Dataset and DataFrame**
- **Dataset Operators and Datasource APIs**
- **Structured Query Plan**
- **Logical Query Plan Optimizer**
- **Tunsten Execution Backend**
- **Hive Integration**
- **Thrift JDBC/ODBC Server**
- **Catalyst**
- **Spark UDF**
- **Spark Window Aggregate Operations**

Course Outlines and Schedules

Class 13 - Advanced Spark ETL - Spark Machine Learning and Deep Learning

- **Classification**
- **Regression**
- **Frequent itemsets (via FP-growth Algorithm)**
- **Feature extraction and selection**
- **Clustering**
- **Statistics**
- **Linear Algebra**
- **Recommendation (Collaborative filtering)**
- **Dimensionality reduction**
- **Model import and export**
- **Lower-level optimization primitives and higher-level pipeline APIs**
- **Deeplearning4j on Spark**
- **Distributed Deep Learning on Apache Spark**

Course Outlines and Schedules

Class 14 – Data Security, Data Governance and Visualization

- **Authentication - Kerberos, LDAP and Apache Knox Gateway**
- **Authorization - Apache Ranger**
- **Data Protection - Encryption at Rest or In Transit and SSL**
- **Data Masking**
- **Apache Atlas - High Level Architecture**
- **Apache Atlas - Type System**
- **Apache Atlas - Metadata Repository**
- **Apache Atlas - Rest API**
- **Apache Atlas - Bridges**
- **Apache Zeppelin**

Course Outlines and Schedules

Class 15 – Big Data Best Practices, Design Patterns and Interview Preparation

- **Big Data Best Practices**
- **Big Data Design Patterns**
- **Big Data Interview Questions**
- **Mini Projects**
- **Big Data Trends**

Course Introduction – 100+ Labs

- **Spark Core - Creating a Pair RDD with Map**
- **Spark Core - Programmatically Specifying the Schema**
- **Spark SQL - Working with a Parquet File**
- **Creating External Hive Tables using Spark SQL**
- **Integrate Spark Streaming with Kafka, Redis, Elastic Search, Logstash and Cassandra**
- **User-to-Item Collaborative Filter (ALS) Recommendation with Spark Streaming and Spark Mllib**
- **Machine Learn Practices (Work2Vec, StandardScaler, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient-boosted trees, K-means, FP-growth)**
- **Export Data from HBase Table into Hive Table**
- **Bulk Load Data From HDFS into HBase**

Course Introduction – 100+ Labs

- **Scheduling the jobs using Oozie**
- **Create Dynamic Partitioned External Hive Table based on Parquet File**
- **Nifi - Connect to External Sources using ListenHTTP**
- **Nifi - Fun with HBase, HDFS, Hive and Kafka**
- **Data Import with Sqoop**
- **Tag based policies with Apache Ranger and Apache Atlas**
- **Define and Process Data Pipelines in Hadoop With Apache Falcon**
- **Securing HDFS with Knox and Ranger**
- **Hive Authorization Using Apache Ranger**

Course Introduction – 80+ Best Practices

- **Choose the best file format for your big data**
- **Efficient storage with compression**
- **Applying MapReduce patterns to big data**
- **Dealing with Large Files using HBase**
- **Advanced Pattern for HBase Data Modeling**
- **Data Ingestion and Egress Patterns**
- **Data Profiling, Validation, Reduction, Transformation and Cleansing Patterns**
- **Best Practices for Hive**
- **What is the best way of doing splitting for Hive**
- **Gracefully Dealing with Bad Input Data using Spark**

Course Introduction – 8 Mini Projects

- **Prediction Airline Delay with Hive and Logistic Regression**
- **Food Mart Sales Data Analytics with Apache Kylin OLAP and CUBE, Hive**
- **End to End Streaming Stock Price CUBE build with Apache Kylin OLAP and streaming CUBE, Kafka, Nifi and HBase**
- **ETL Registered Business Locations from RDBMS into HDFS using Sqoop, Oozie and Hive**
- **Insurance Claim Near Real-Time Event Processing with Nifi, Storm, Kafka, Spark and HBase**
- **Real time Recommendation with Spark Streaming, Spark SQL and Spark MLlib, ELK**
- **Stock Price Ingestion and Visualization with Nifi, Solr, Kibana**
- **Real Time Fraud Transaction Detection with Storm, Nifi, Spark ML**

Course Introduction – 500+ Interview Questions

- **How do you debug a performance issue or a long running Spark job?**
- **How will you assign only 60% of cluster resources to DEV, 40% to QA?**
- **Assume you are doing a join and you notice that all but one reducer is running for a long time how do you address the problem?**
- **In what kind of situation, the following exception happen in Spark and how to avoid it: Job aborted due to stage failure: Task not serializable?**
- **How Many Partitions Does An RDD Have?**
- **Why you see the error "ERROR OneForOneStrategy: ... java.io.NotSerializableException" in Spark?**

Course Introduction – 500+ Interview Questions

- **How do you efficiently join large data set with medium data set in Spark?**
- **How to out put Spark RDD to RDBMS, Cassandra and HDFS in the same action?**
- **What is the difference between DStream updateStateByKey and mapWithState?**
- **How do you troubleshooting such HBase issues as Too Many Regions, Too Many Columns Family, Hotspotting and Timeout**
- **How to connect to hive on remote secure cluster**
- **How does Sqoop to use an encrypted password file to connect to the data warehouse**
- **Explain how does spark execution model and memory management work?**

Programming on Hadoop Ecosystem Frameworks

- **Scala version 2.11.8**
- **OpenJDK 64-Bit Server VM, Java 1.8.0_151**
- **Spark version 2.2.0**
- **Hadoop 2.7.3**
- **Kafka 0.10.1.2**
- **Sqoop 1.4.6**
- **Hive 1.2.1 / Hive 2.1.0**
- **HBase 1.1.2**
- **Phoenix 4.7.0**
- **Apache Atlas 0.8.0**
- **Falcon 0.10.0**
- **Apache Storm 1.1.0**
- **IntelliJ Community 2016.2.5**

Effective Big Data Solutions

- **Data Sources**
- **Data Format**
- **Big Data Ingestion Design Pattern**
- **Big Data Ingestion Approaches**
- **Ad Hoc Queries over Hadoop**
- **OLAP over Big Data**
- **Batch Distributed Data Processing**
- **Real Time Streaming Data Processing**
- **Machine Learning**
- **Graph Processing**

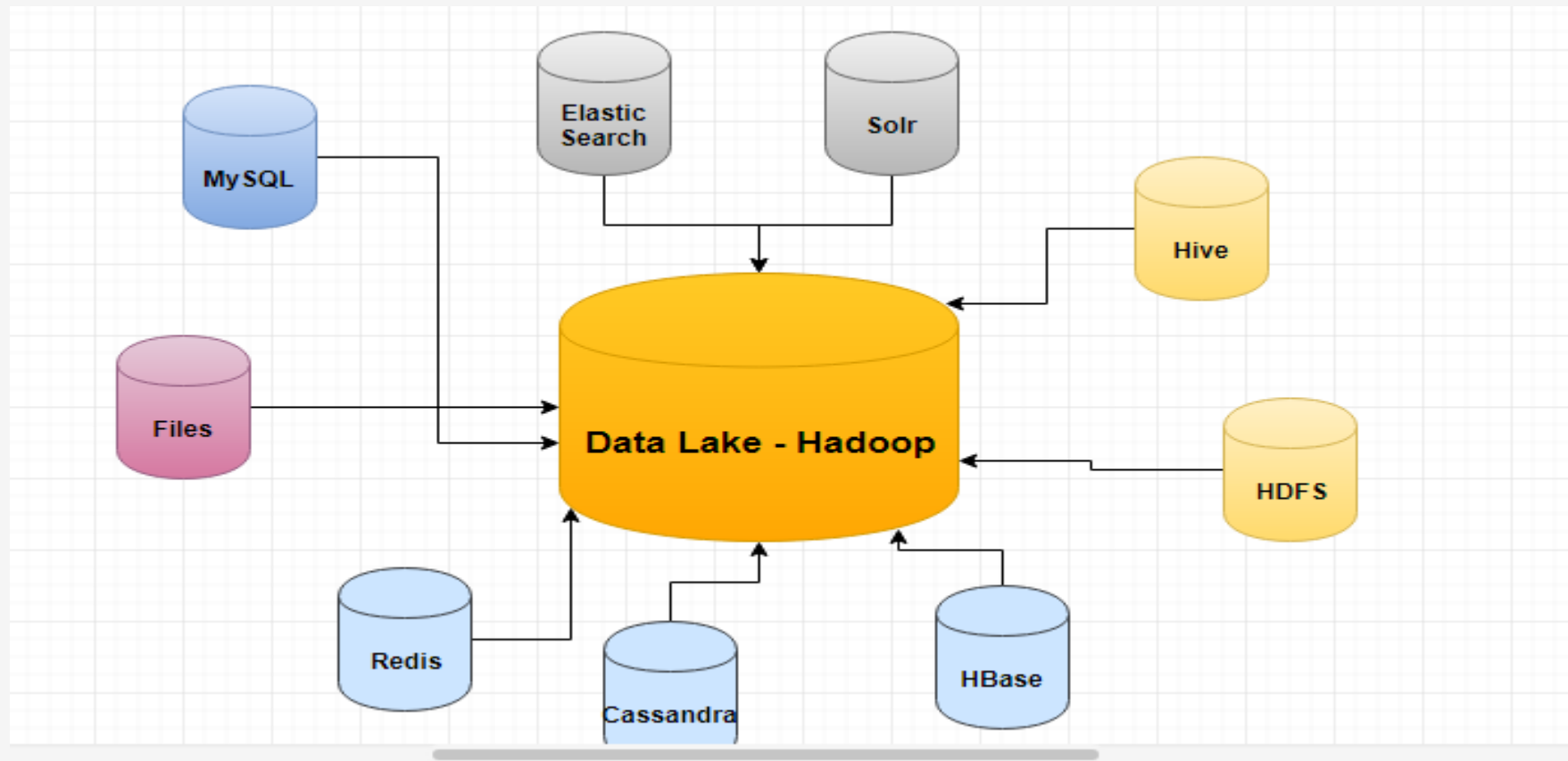
Effective Big Data Solutions

➤ Data Sources

- ❖ **RDBMS - MySQL**
- ❖ **NoSQL - HBase, Cassandra and Redis**
- ❖ **Data Lake – Hive and HDFS**
- ❖ **Search Engine – ElasticSearch and Solr**
- ❖ **Flat File**

Effective Big Data Solutions

➤ Data Sources



Effective Big Data Solutions

➤ Data Format

- ❖ XML, JSON, CSV (One Line or Multiple Lines)
- ❖ RDBMS
- ❖ Parquet
- ❖ Avro
- ❖ ORC
- ❖ Sequence File
- ❖ HFile
- ❖ COBOL Format
- ❖ Application Logs (Log4J, IIS)
- ❖ Image, Audio, Video Format
- ❖ Social Media

Effective Big Data Solutions



Effective Big Data Solutions

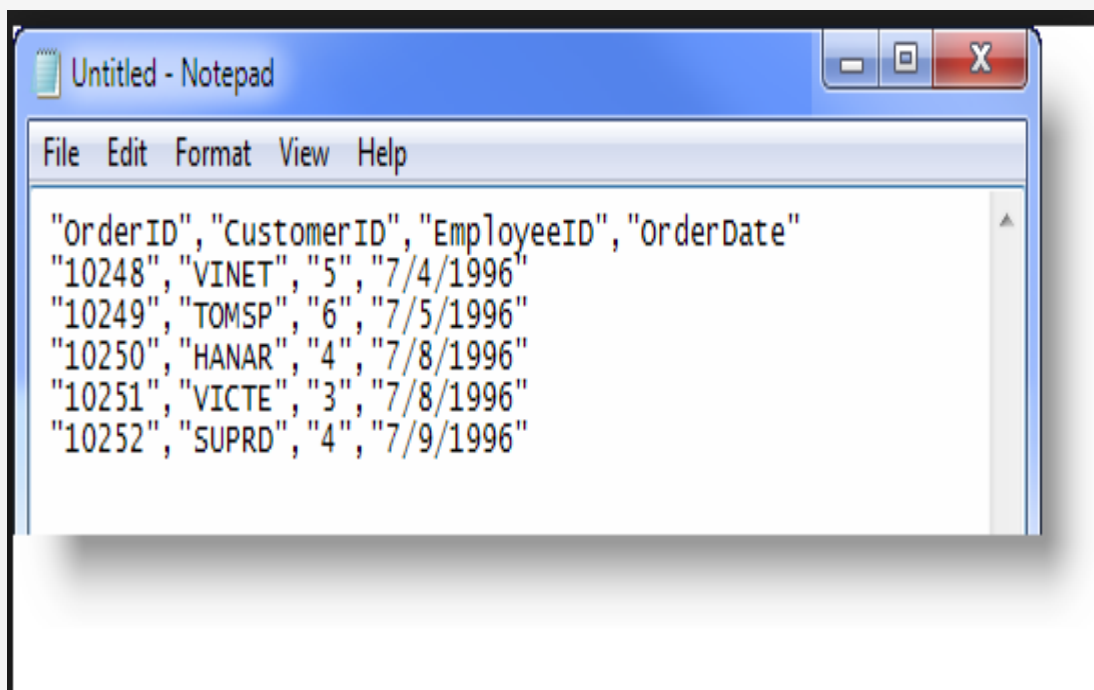
❖ XML, JSON, CSV (One Line or Multiple Lines)

XML

```
<Node>
  <id>10002</id>
  <Name>john</Name>
</Node>
<Node>
  <id>10003</id>
  <Name>Scott</Name>
</Node>
<Node>
  <id>10004</id>
  <Name>Mohan</Name>
</Node>
<Node>
  <id>10001</id>
  <Name>Deepak </Name>
</Node>
```

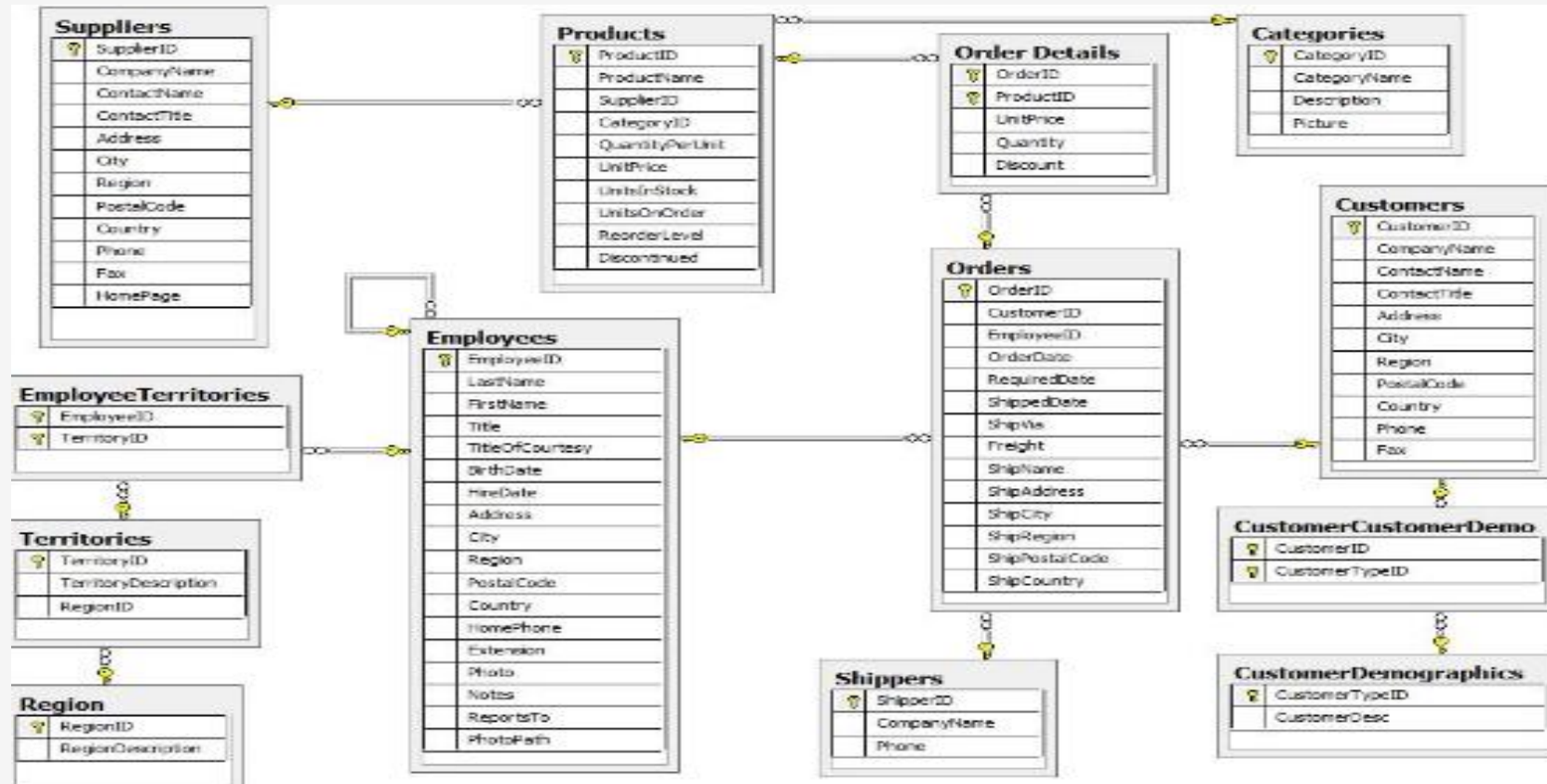
JSON

```
[
  {
    "id":10002,
    "name":"john"
  },
  {
    "id":10003,
    "name":"Scott"
  },
  {
    "id":10004,
    "name":"Mohan"
  },
  {
    "id":10001,
    "name":"Deepak"
  }
]
```



Effective Big Data Solutions

❖ RDBMS



Effective Big Data Solutions

❖ Parquet

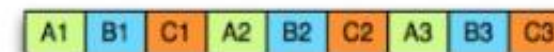
Parquet

- Design based on Google's Dremel paper
- Schema segregated into footer
- Column major format with stripes
- Simpler type-model with logical types
- All data pushed to leaves of the tree
- Integrated compression and indexes

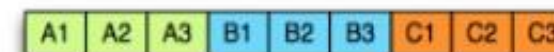
Columnar Storage

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

row-oriented storage

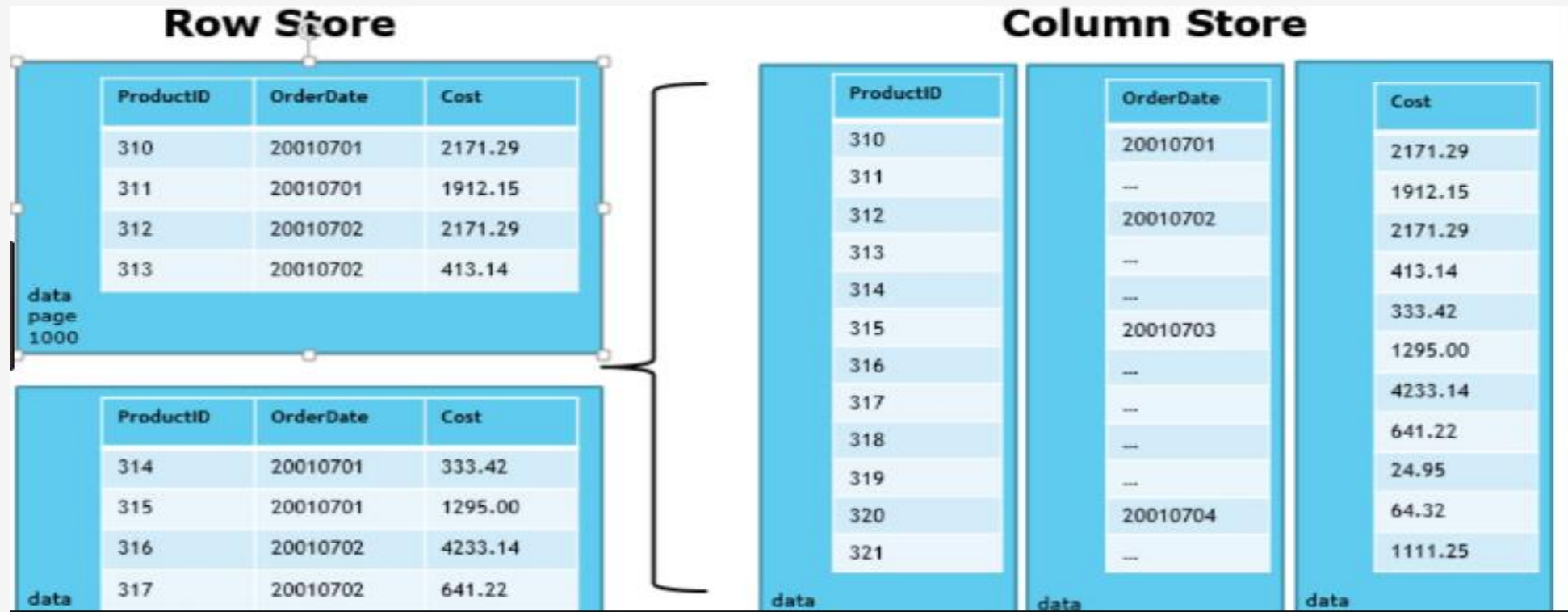


column-oriented storage



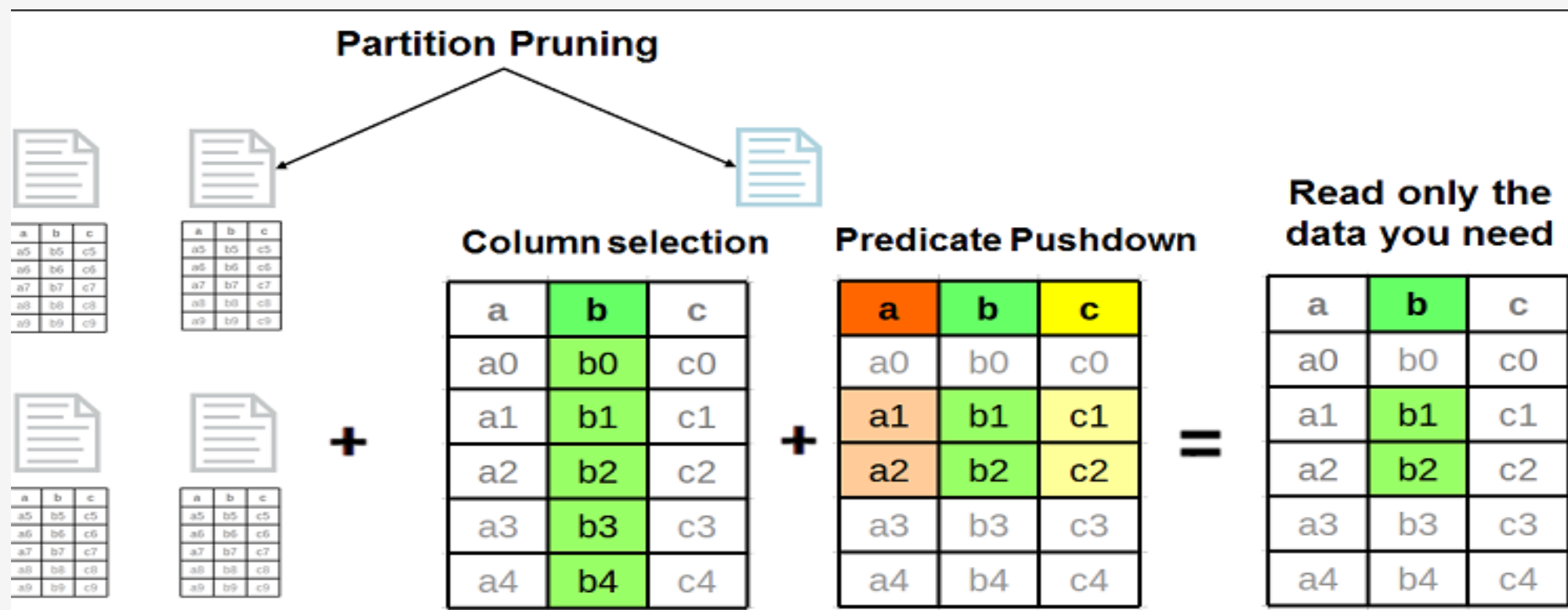
Effective Big Data Solutions

❖ Parquet



Effective Big Data Solutions

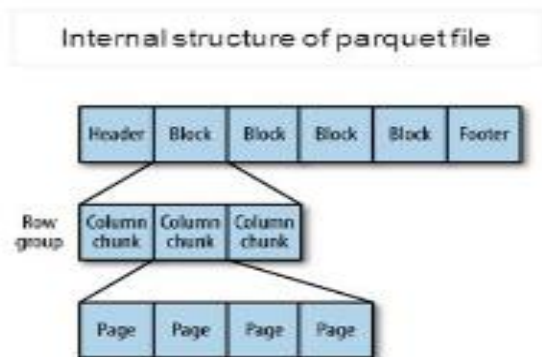
❖ Parquet



Effective Big Data Solutions

❖ Parquet

Parquet file structure & Configuration



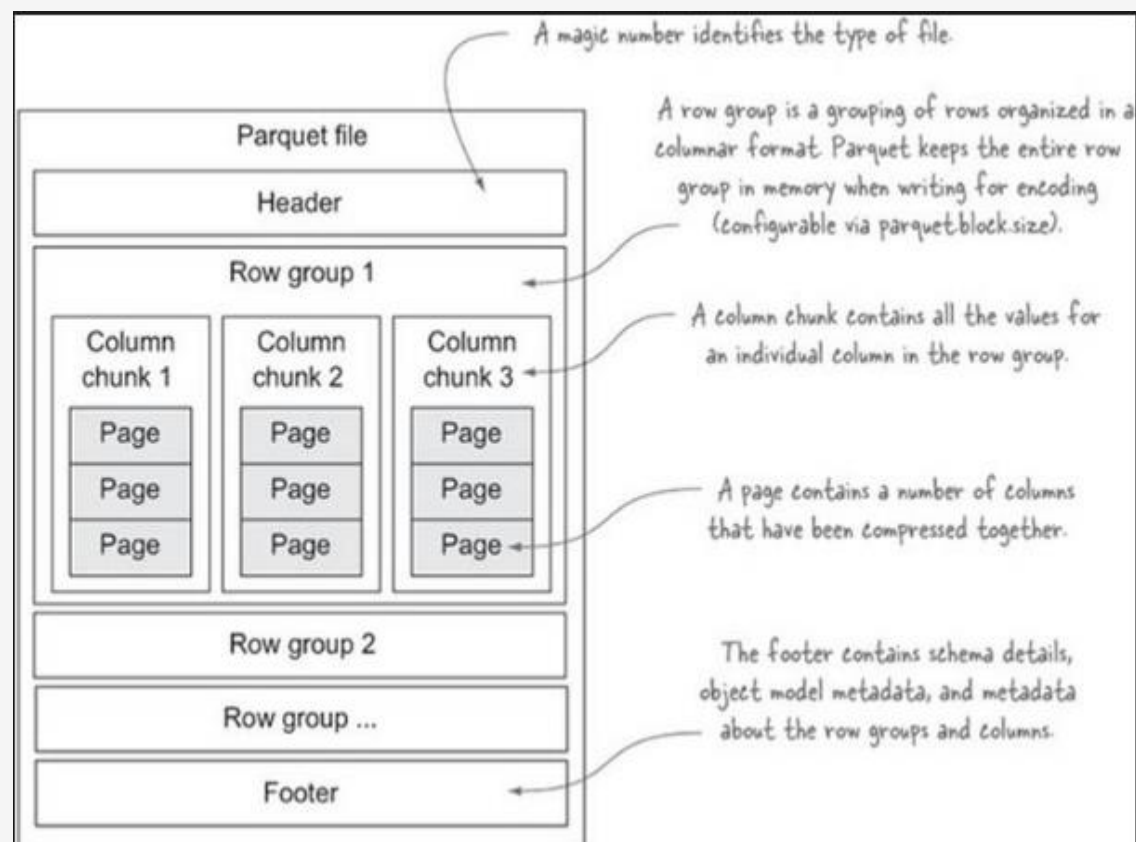
Configurable parquet parameters

Property name	Default value	Description
parquet.block.size	128 MB	The size in bytes of a block (row group).
parquet.page.size	1MB	The size in bytes of a page.
parquet.dictionary.page.size	1MB	The maximum allowed size in bytes of a dictionary before falling back to plain encoding for a page.
parquet.enable.dictionary	true	Whether to use dictionary encoding.
parquet.compression	UNCOMPRESSED	The type of compression: UNCOMPRESSED, SNAPPY, GZIP & LZO

In summation, Parquet is state-of-the-art, open-source columnar format the supports *most* of Hadoop processing frameworks and is optimized for high compression and high scan efficiency

Effective Big Data Solutions

❖ Parquet



```
hive> create table pstock( date    string ,  
    > symbol  string ,  
    > open    float ,  
    > high    float ,  
    > low     float ,  
    > close   float ,  
    > volume  bigint) stored as parquet;
```

OK

Time taken: 0.102 seconds

hive> █

Effective Big Data Solutions

❖ Parquet

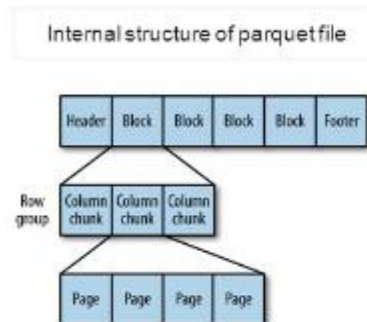
```
hive> create table pstock( date    string ,
> symbol string ,
> open  float ,
> high  float ,
> low   float ,
> close float ,
> volume bigint) stored as parquet;
```

OK

Time taken: 0.102 seconds

hive> █

Parquet file structure & Configuration



Configurable parquet parameters

Property name	Default value	Description
parquet.block.size	128 MB	The size in bytes of a block (row group).
parquet.page.size	1MB	The size in bytes of a page.
parquet.dictionary.page.size	1MB	The maximum allowed size in bytes of a dictionary before falling back to plain encoding for a page.
parquet.enable.dictionary	true	Whether to use dictionary encoding.
parquet.compression	UNCOMPRESSED	The type of compression: UNCOMPRESSED, SNAPPY, GZIP & LZO.

In summation, Parquet is state-of-the-art, open-source columnar format the supports *most* of Hadoop processing frameworks and is optimized for high compression and high scan efficiency

Effective Big Data Solutions

❖ Avro

Avro

Clip slide

- Cross-language file format for Hadoop
- Schema evolution was primary goal
- Schema segregated from data
 - Unlike Protobuf and Thrift
- Row major format

Effective Big Data Solutions

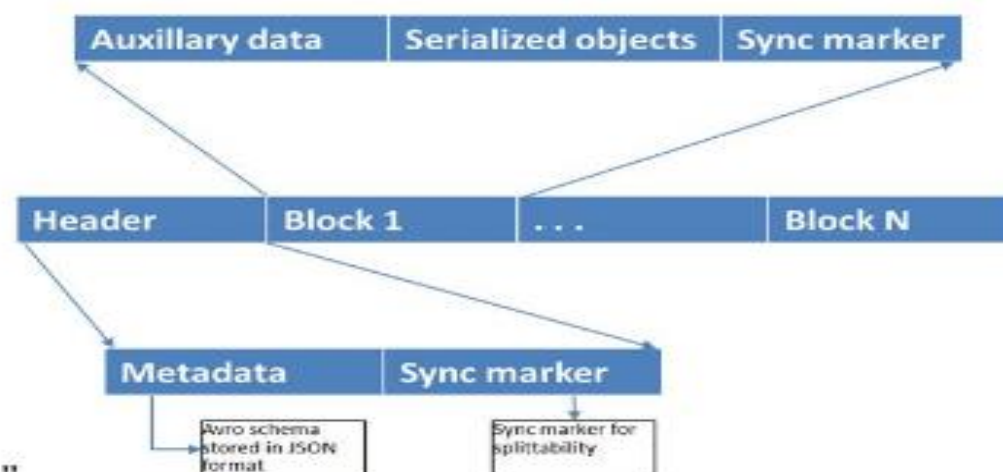
❖ Avro

Avro – File structure and example

Sample AVRO schema in JSON format

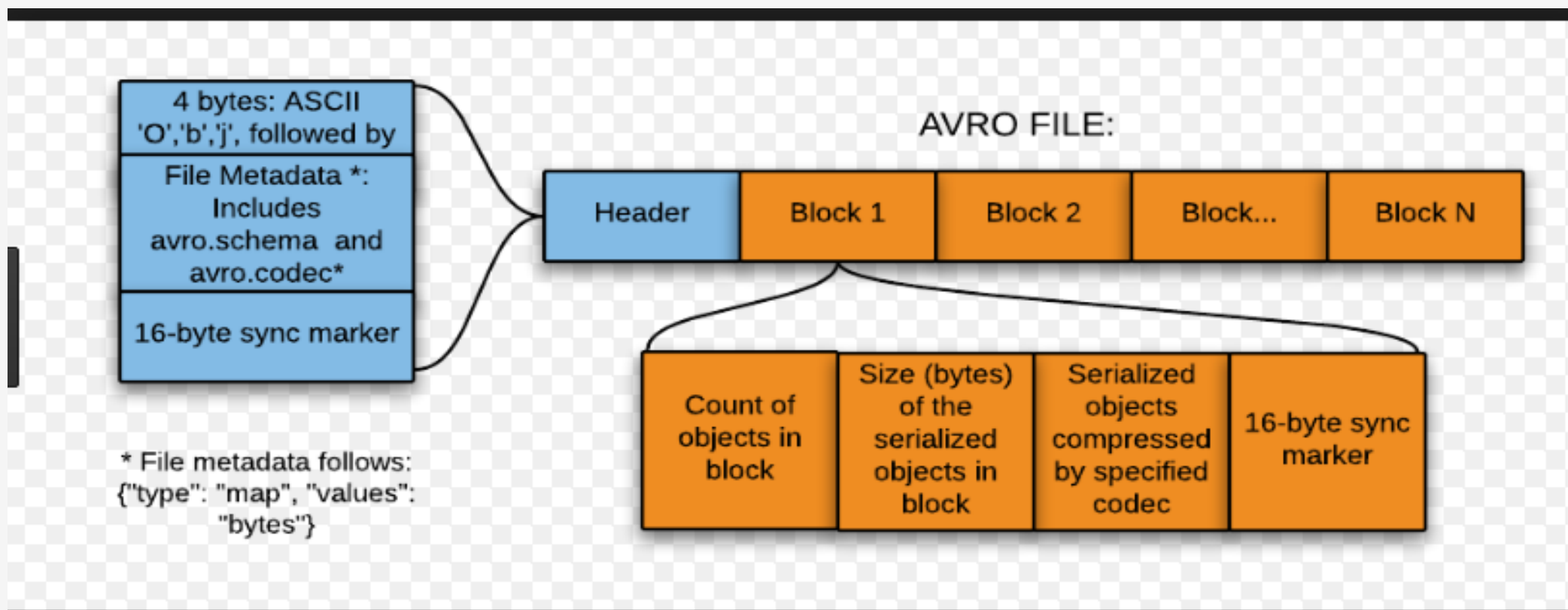
```
{
  "type" : "record",
  "name" : "tweets",
  "fields" : [ {
    "name" : "username",
    "type" : "string",
  }, {
    "name" : "tweet",
    "type" : "string",
  }, {
    "name" : "timestamp",
    "type" : "long",
  } ],
  "doc:" : "schema for storing tweets"
}
```

Avro file structure



Effective Big Data Solutions

❖ Avro



Effective Big Data Solutions

❖ ORC

ORC

Clip slide

- Originally part of Hive to replace RCFile
 - Now top-level project
- Schema segregated into footer
- Column major format with stripes
- Rich type model, stored top-down
- Integrated compression, indexes, & stats

ORC File Basics

- Columnar format
 - Enables user to read & decompress just the bytes they need
- Fast
 - See <https://www.slideshare.net/HadoopSummit/file-format-benchmark-avro-json-orc-parquet>
- Indexed
- Self-describing
 - Includes all of the information about types and encoding
- Rich type system
 - All of Hive's types including timestamp, struct, map, list, and union

Effective Big Data Solutions

❖ Parquet vs Avro vs ORC

FORMAT	COLUMNAR	COMPRESSION	SUPPORT
AVRO	✗	GOOD	HADOOP SPARK ATHENA PRESTO
PARQUET	✓	GREAT	HADOOP SPARK ATHENA PRESTO
ORC	✓	EXCELLENT	HADOOP SPARK ATHENA PRESTO

● ORC & Parquet store metadata

- Stored in file footer
- File schema
- Number of records
- Min, max, count of each column

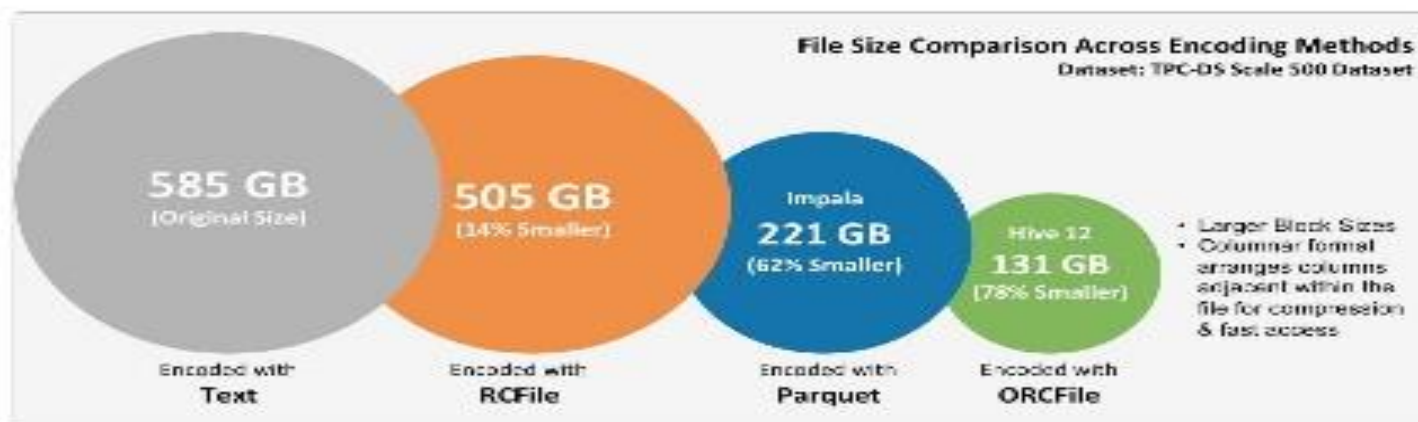
● Provides O(1) Access

Effective Big Data Solutions

❖ Parquet vs Avro vs ORC

File Formats: Avro vs Parquet vs ORC

- **Avro** is row-based storage format, optimized for scans of all fields in a row for each query
- **Parquet** is column-based, best used when dataset has many columns and only a few columns are worked on
- **ORC** is column-based as well

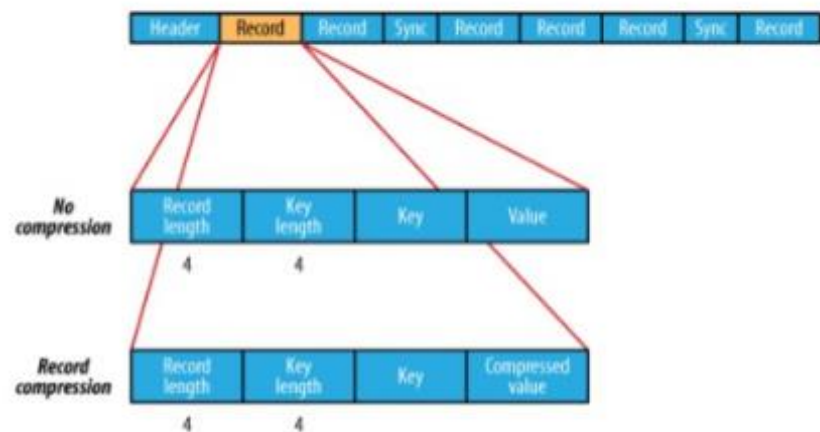


Effective Big Data Solutions

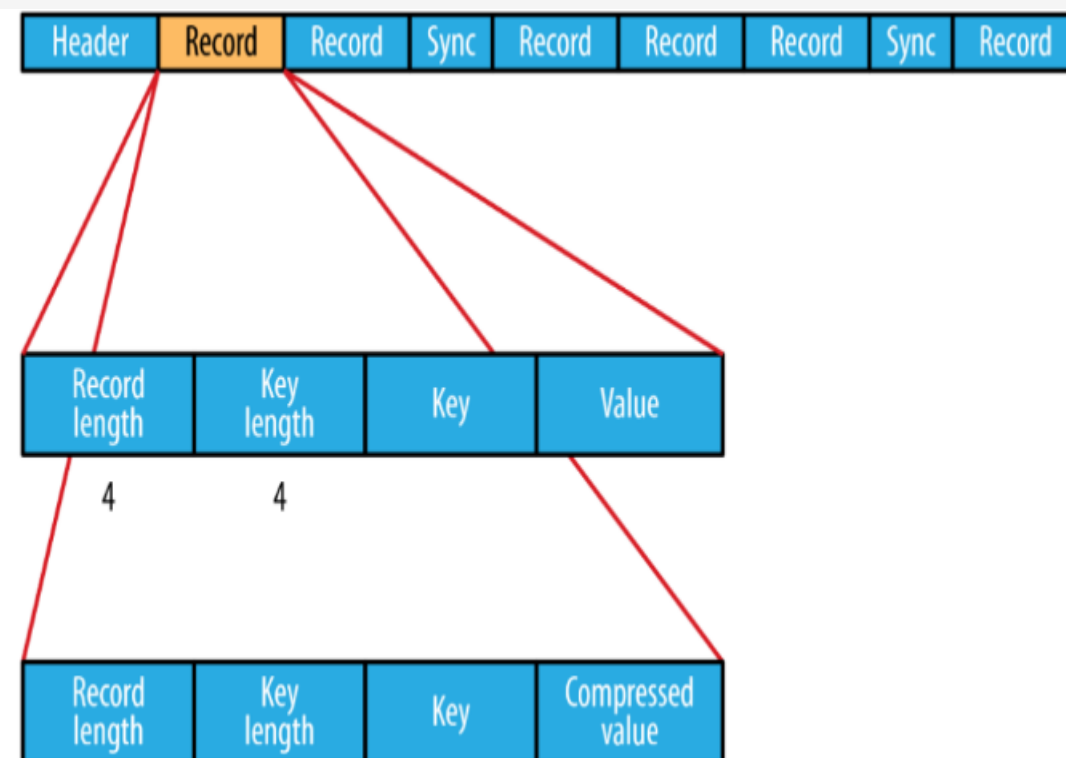
❖ Sequence File

Internals of A sequence file

- A sequence file consists of a header followed by one or more records
- The header contains other fields including the names of the key and value classes, compression details, user defined metadata, and the sync marker.
- A MapFile is a sorted SequenceFile with an index to permit lookups by key.



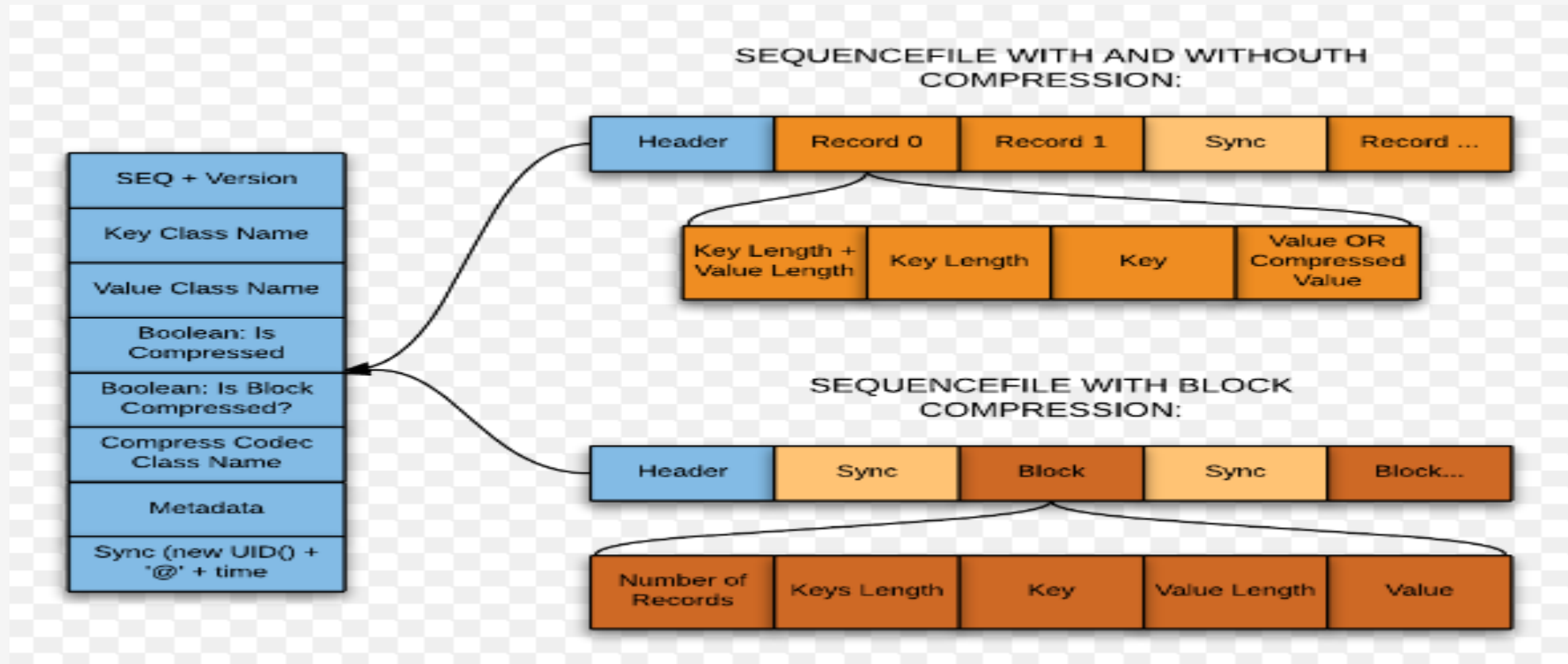
*No
compression*



*Record
compression*

Effective Big Data Solutions

❖ Sequence File



Effective Big Data Solutions

❖ Sequence File

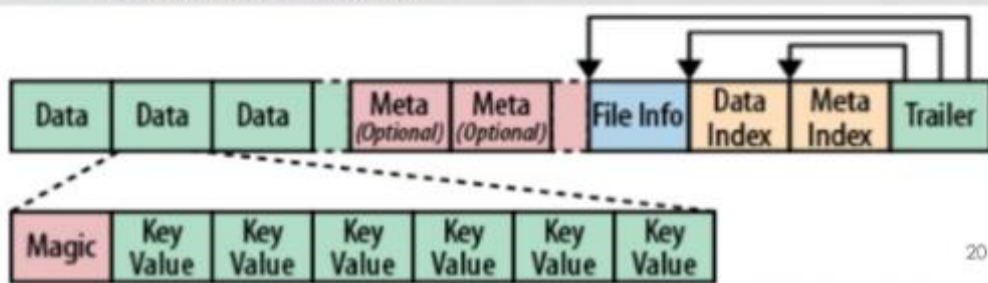
```
hive> CREATE EXTERNAL TABLE wlslog_ext (TIME_STAMP STRING, CATEGORY STRING, TYPE S
STRING, SERVERNAME STRING, CODE STRING, MSG STRING) ROW FORMAT DELIMITED FIELDS TERM
INATED BY ',' STORED AS SEQUENCEFILE LOCATION 'hdfs://localhost:8020/wlslog';
OK
Time taken: 0.224 seconds
hive> SELECT * FROM wlslog_ext;
OK
Apr-8-2014-7:06:16-PM-PDT      Notice  WebLogicServer  AdminServer      BEA-0003
65      Server state changed to STANDBY
Apr-8-2014-7:06:17-PM-PDT      Notice  WebLogicServer  AdminServer      BEA-0003
65      Server state changed to STARTING
Apr-8-2014-7:06:18-PM-PDT      Notice  WebLogicServer  AdminServer      BEA-0003
65      Server state changed to ADMIN
Apr-8-2014-7:06:19-PM-PDT      Notice  WebLogicServer  AdminServer      BEA-0003
65      Server state changed to RESUMING
Apr-8-2014-7:06:20-PM-PDT      Notice  WebLogicServer  AdminServer      BEA-0003
31      Started WebLogic AdminServer
Apr-8-2014-7:06:21-PM-PDT      Notice  WebLogicServer  AdminServer      BEA-0003
65      Server state changed to RUNNING
Apr-8-2014-7:06:22-PM-PDT      Notice  WebLogicServer  AdminServer      BEA-0003
60      Server started in RUNNING mode
Time taken: 0.481 seconds
hive> █
```

Effective Big Data Solutions

❖ HFile

HFILE FORMAT

- The actual storage files are implemented by the *HFile* class
- Store HBase's data efficiently
- Blocks
 - Fixed size
 - Trailer, File Info
 - Others are variable size



HFile Format Information

- All data is stored in a custom (open-source) format, called **HFile**
- Data is stored in **blocks** (64KB default)
 - Trade-off between lookups and I/O throughput
 - Compression, encoding applied after limit check
- Index, filter and meta data is stored in **separate** blocks
- Fixed **trailer** allows traversal of file structure
- Newer versions introduce **multilayered** index and filter structures
 - Only load master index and load partial index blocks on demand
- Reading data requires **deserialization** of block into cells
 - Kind of Amdahl's Law applies

Effective Big Data Solutions

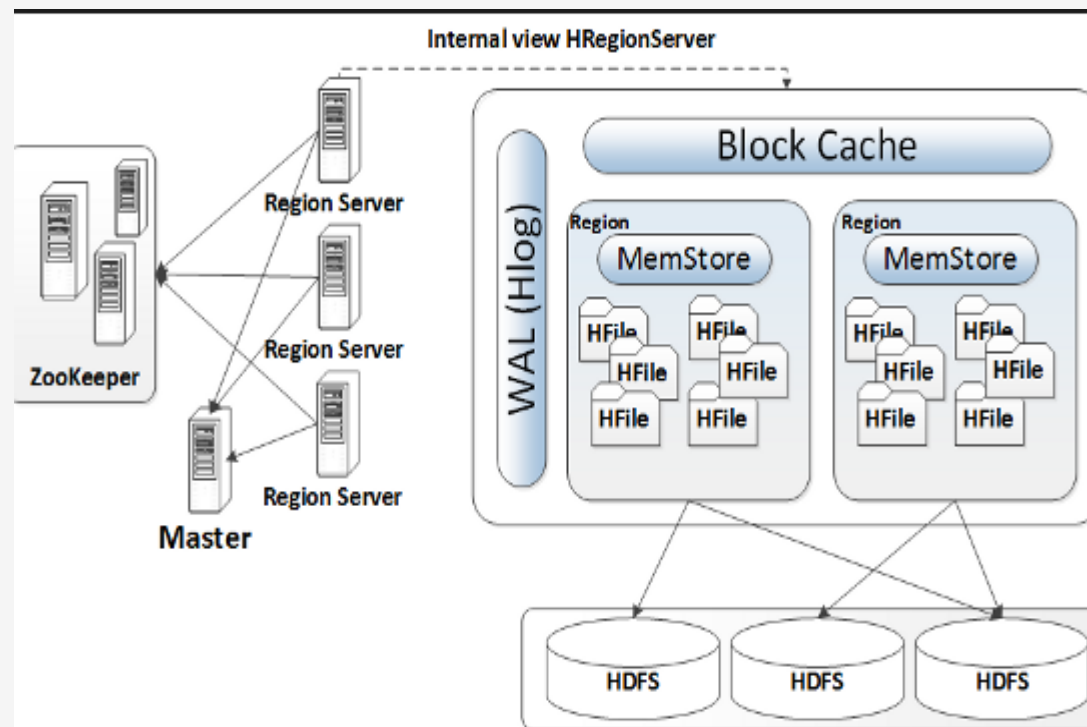
❖ HFile

What is a block?

HFile v2 Format



HFile v2 format figure reproduced from Matteo Bertozzi, "Apache HBase I/O – HFile", <http://blog.cloudera.com/blog/2012/06/hbase-io-hfile-input-output/>



Effective Big Data Solutions

❖ COBOL Format

The OCCURS clause marks an array.

The DEPENDING ON clause marks a counter field for the array, if one exists.

The array ASSIGNMENTS is a nested array within COURSES.

```
01  STUDENT.
   20  ID
   20  FIRST_NAME
   20  LAST_NAME
   *
   20  DATE_OF_BIRTH
   20  NUMOF_COURSES
   20  NUMOF_BOOKS
   20  COURSES.
      25  COURSE OCCURS 8 TIMES DEPENDING ON NUMOF_COURSES.
         30  COURSE_ID
         30  COURSE_TITLE
         30  INSTRUCTOR_ID
         30  NUMOF_ASSIGNMENTS
         30  ASSIGNMENTS OCCURS 4 TIMES DEPENDING ON NUMOF_ASSIGNMENTS.
            40  ASSIGNMENT_TYPE
            40  ASSIGNMENT_TITLE
            *
            40  DUE_DATE
            40  GRADE
      20  BOOKS.
         25  BOOK OCCURS 1 TO 5 TIMES DEPENDING ON NUMOF_BOOKS.
            30  ISBN
            *
            30  RETURN_DATE
```

PIC 9(8).
PIC X(32).
PIC X(32).
YYYYMMDD
PIC S9(8) COMP.
PIC 9(4) COMP.
PIC 9(4) COMP.

PIC 9(8).
PIC X(48).
PIC 9(8).
PIC 9(4) COMP.
PIC X(12).
PIC X(48).
YYYYMMDD
PIC S9(8) COMP.
PIC S9V9.

PIC X(10).
YYYYMMDD
PIC 9(8) COMP.

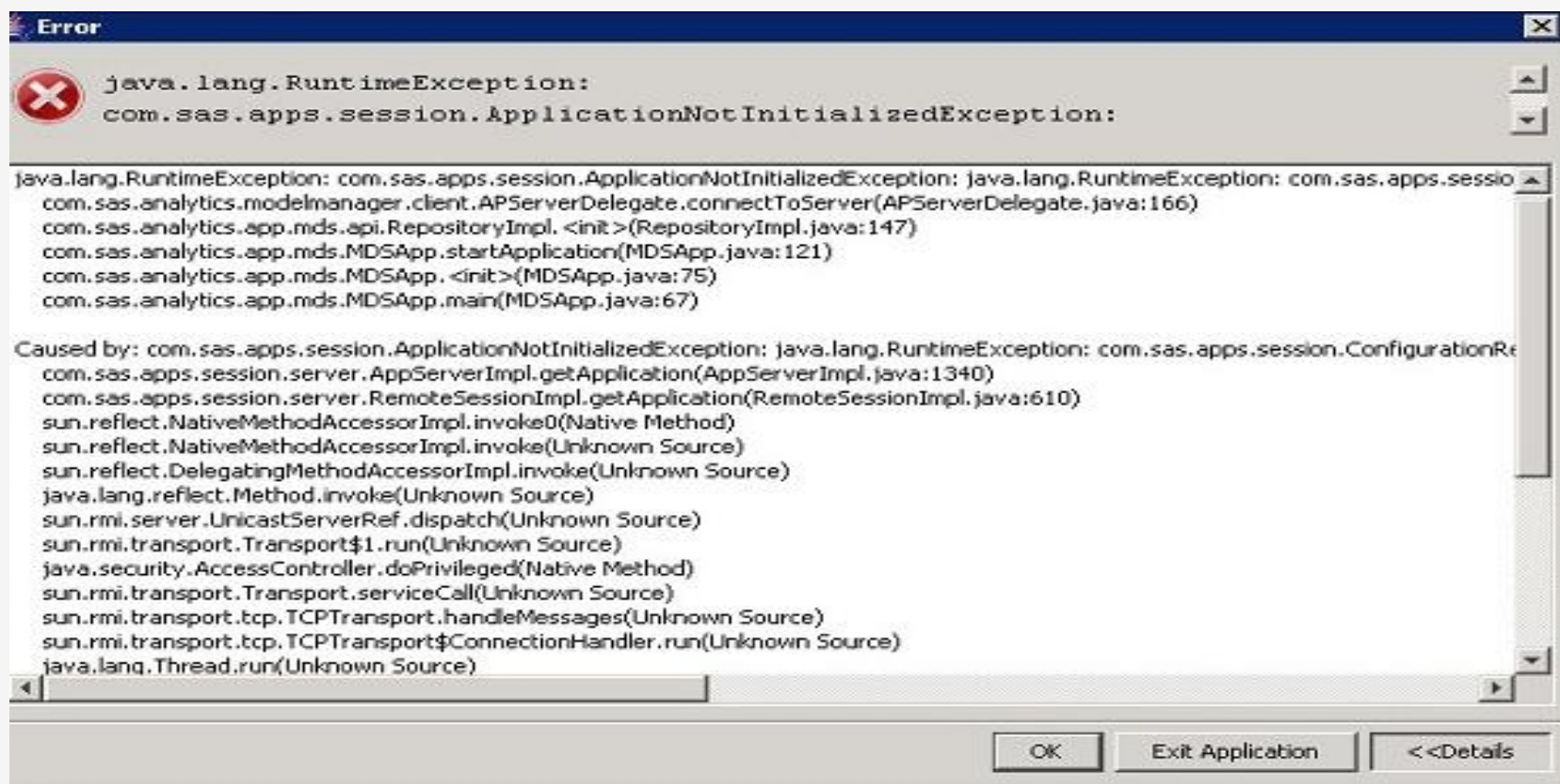
Effective Big Data Solutions

❖ Application Logs (Log4J, IIS)

Attribute	Common log format	Combined log format
IP address	Yes	Yes
User ID	Yes	Yes
Time of request	Yes	Yes
Text of request	Yes	Yes
Status code	Yes	Yes
Size in bytes	Yes	Yes
Referer		Yes
HTTP agent		Yes

Effective Big Data Solutions

❖ Application Logs (Exception)



Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

- ❖ **Data Ingestion**
- ❖ **Data Profiling**
- ❖ **Data Validation**
- ❖ **Data Cleansing**
- ❖ **Data Transformation**
- ❖ **Data Reduction**
- ❖ **Advanced Design Pattern**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Ingestion**

- **The context of data ingest and egress**
- **Types of data in the enterprise**
- **Ingest and egress for multistructured data**
- **The ingress and egress for the NoSQL data**
- **The ingress and egress for structured data**
- **The ingress and egress for semi-structured data**

Effective Big Data Solutions

➤ Big Data Ingestion Design Pattern

❖ Data Profiling

Data profiling is a necessary first step in getting any meaningful insight into the data ingested by Hadoop, by understanding the content, context, structure, and condition of data

- The data type inference
- The basic statistical profiling
mean, median, mode, maximum, minimum, and standard deviation
- The pattern-matching
- The string profiling
- The unstructured text profiling
Stop Words Removal, Stemming, TFIDF
- Mask based profiling

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Validation**

- **Constraint validation**

- ☐ **Null checks**
- ☐ **Range checks**
- ☐ **Data type constraint**
- ☐ **Unique constraints**

- **Regex validation**

- ☐ **Dates, Credit Card, Email or Phone Numbers are prone to be represented in multiple ways**
- ☐ **String length and pattern validation**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Validation**

- **Corrupt data validation**

- ☐ **Data corruption from the perspective of the corrupt data being treated as a noise or as an outlier**
- ☐ **Noise can be defined as a random error in measurement**
- ☐ **Outliers are also a kind of noise but the value of error is too far away from the expected value**
- ☐ **Data corruption in sensor data**
- ☐ **Data corruption in structured data**
- ☐ **The common techniques are binning, regression, and clustering**

Effective Big Data Solutions

➤ Big Data Ingestion Design Pattern

❖ Data Validation

- Corrupt data validation

- ❑ Binning

- ✓ Creating a set of sorted values partitioned into bins
 - ✓ Replaced by the mean or median of that partition

- ❑ Regression

- ✓ It is a technique that fits the data values to a function

- ❑ Clustering

- ✓ Clustering by grouping similar values together to find the values that are outside of the cluster and may be considered as an outlier

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Validation**

- **Unstructured text data validation**

- ❑ **Data pre-processing techniques**

- ❑ **Discover the metadata from the textual data and organize it in a way that facilitates further processing**

- ❑ **Fuzzy Match**

- ❑ **The Bloom filter is a space-optimized data structure specifically used to filter a smaller dataset from a larger dataset by testing whether an element belonging to the smaller dataset is a member of the larger one or not**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Cleansing**

- **Constraint cleansing**

- ☐ **Depending on the business rule, either the invalid records are removed or appropriate cleansing steps are applied to the invalid data**
- ☐ **Filling the missing values with the appropriate values is a complex, could use constant global label or the mean value for data or probabilistic measure, such as Bayesian inference or a decision tree**
- ☐ **The invalid data is cleansed as per the business rules by filtering the invalid data, or by replacing the invalid values with the maximum range value if the invalid data is higher than the range**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Cleansing**

- **Regex cleansing**
 - ☐ **Using regular expressions and pattern-based filtering of records to cleanse invalid data**
 - ☐ **Filtering fields that do not match a specific pattern**
 - ☐ **Splitting string into tokens**
 - ☐ **Extracting data that matches a pattern**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Cleansing**

- **Corrupt data cleansing**

- ❑ **Binning:**

- ✓ **Removal of noisy data by applying a smoothing function**

- ❑ **Regression**

- ✓ **Noise removal can be done using regression by identifying the regression function and removing all the data values that lie far away from the function's predicted value**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Cleansing**

- **Unstructured text data cleansing**
 - ❑ **Performing pre-processing steps, such as lowercase conversion, stop word removal, stemming, punctuation removal, extra spaces removal, identifying numbers, and identifying misspellings**
 - ❑ **Textual data can also be represented using alternative forms of spellings**
 - ❑ **Deduplication**
 - ❑ **Identify the misspelled words and replace them with the correct ones**
 - ❑ **Numerical value identification from within the text enables us to pick all the numbers**
 - ❑ **Extraction of data, which matches certain patterns using regex**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Transformation**

- **Normalization**
- **Aggregation**
- **Generalization**
- **Data integration (Join)**
- **Format Conversion**
- **Split Data Set**

Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Data Reduction**

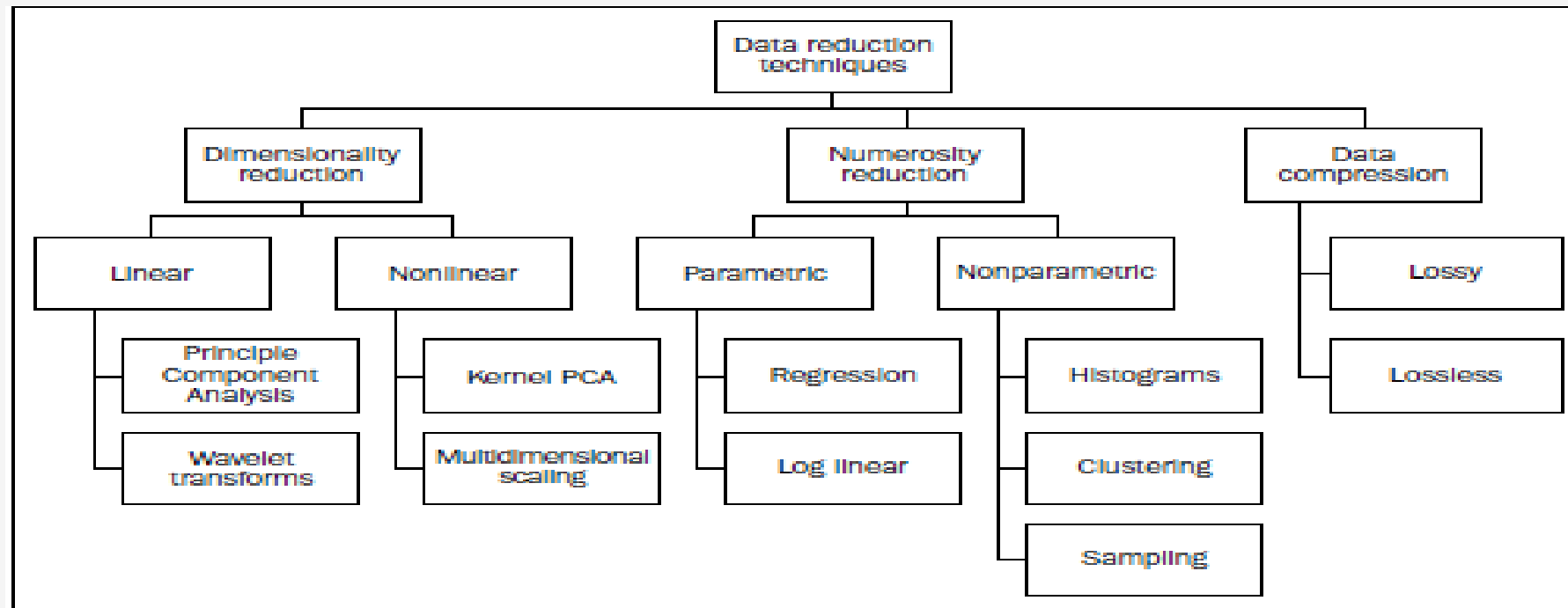
Data reduction aims to obtain a reduced representation of the data. It ensures data integrity, though the obtained dataset after the reduction is much smaller in volume than the original dataset.

- **Dimensionality reduction**
- **Numerosity reduction**
- **Compression**

Effective Big Data Solutions

➤ Big Data Ingestion Design Pattern

❖ Data Reduction



Effective Big Data Solutions

➤ **Big Data Ingestion Design Pattern**

❖ **Advanced Design Pattern**

- **Clustering textual data**
- **Topic discovery**
- **Natural language processing**
- **Classification**

Effective Big Data Solutions

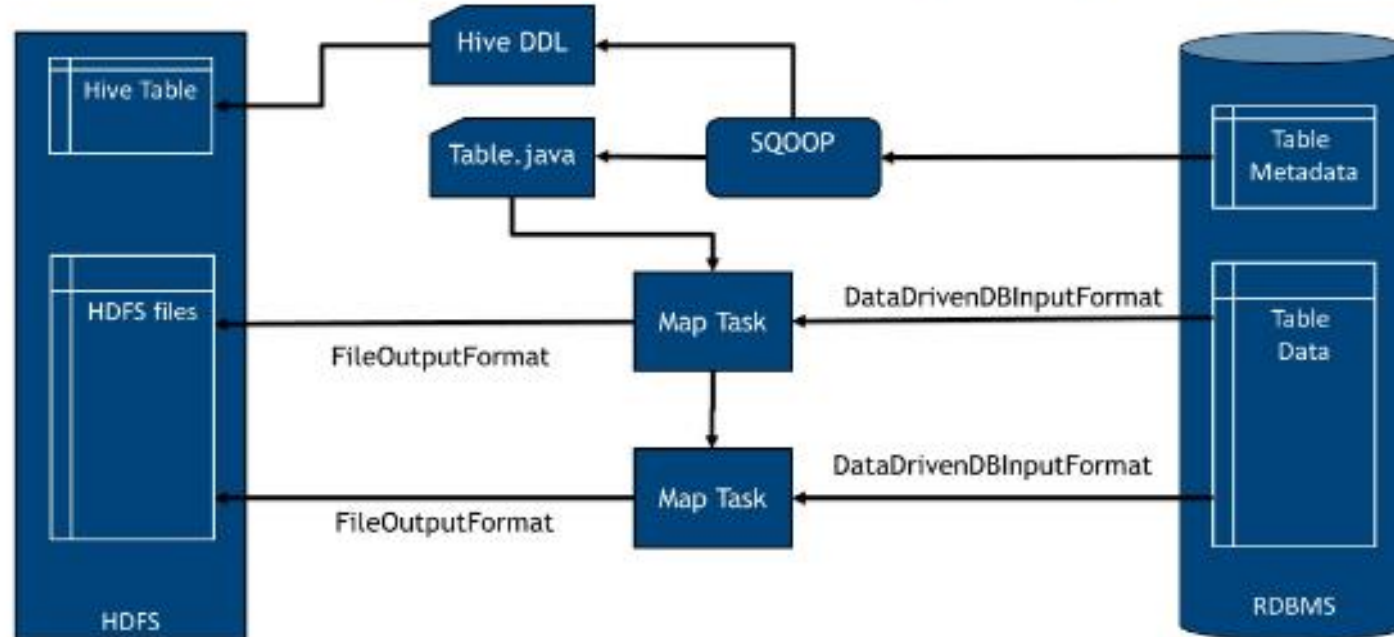
➤ **Big Data Ingestion Approaches**

❑ **Batch Mode Ingestion**

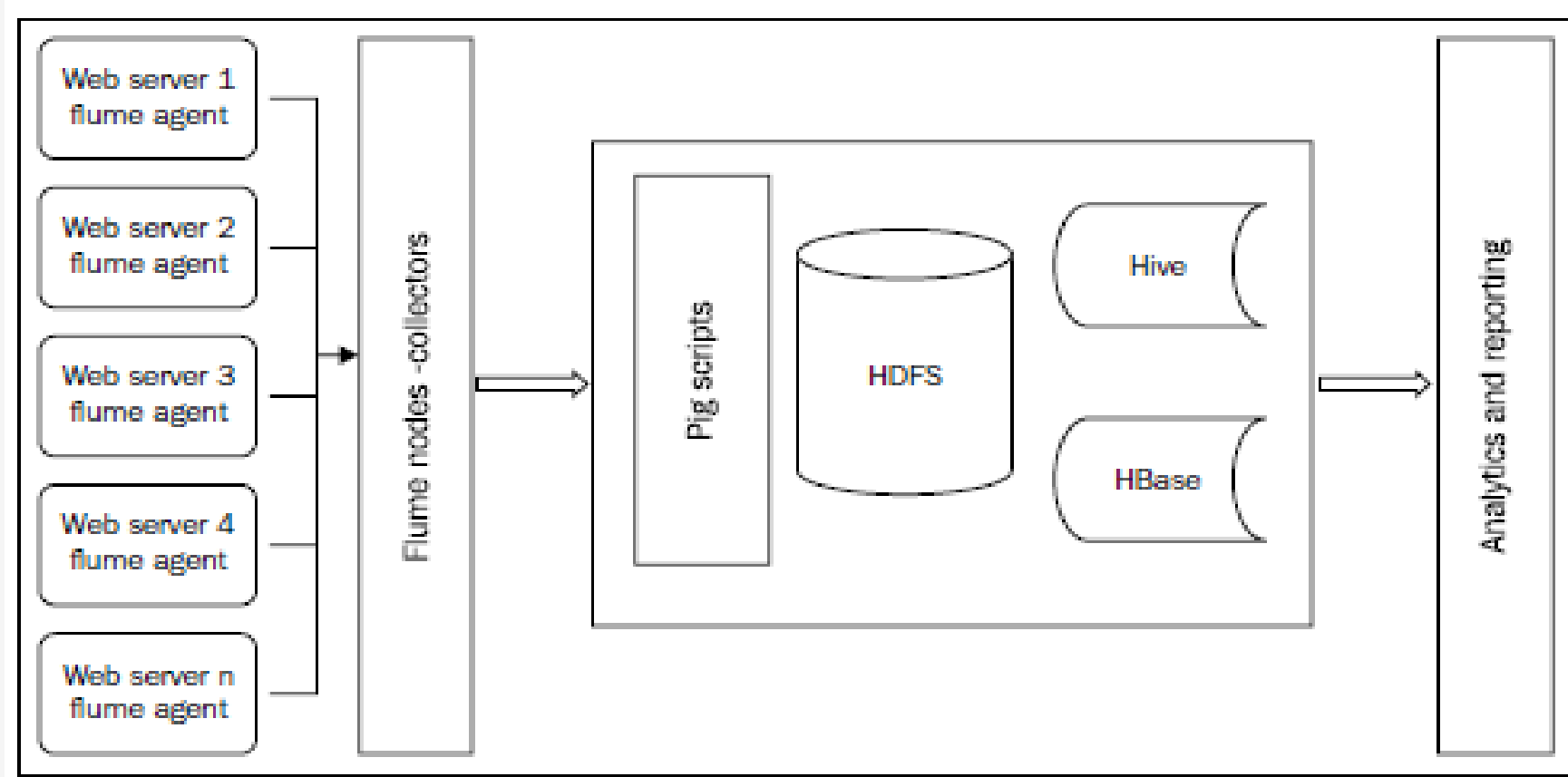
- **RDBMS to Hadoop using Sqoop**
- **Mainframe to Hadoop using Sqoop**
- **Logs to Hadoop using Nifi, Flume and Splunk or ELK**
- **Social Media to Hadoop using Nifi**
- **IoT to Hadoop using Nifi**
- **HDFS to HBase using Spark ETL**
- **CSV to Hive using Spark ETL**
- **RDBMS to HDFS using StreamSet**
- **RDBMS to Hive using Spark ETL**

Effective Big Data Solutions

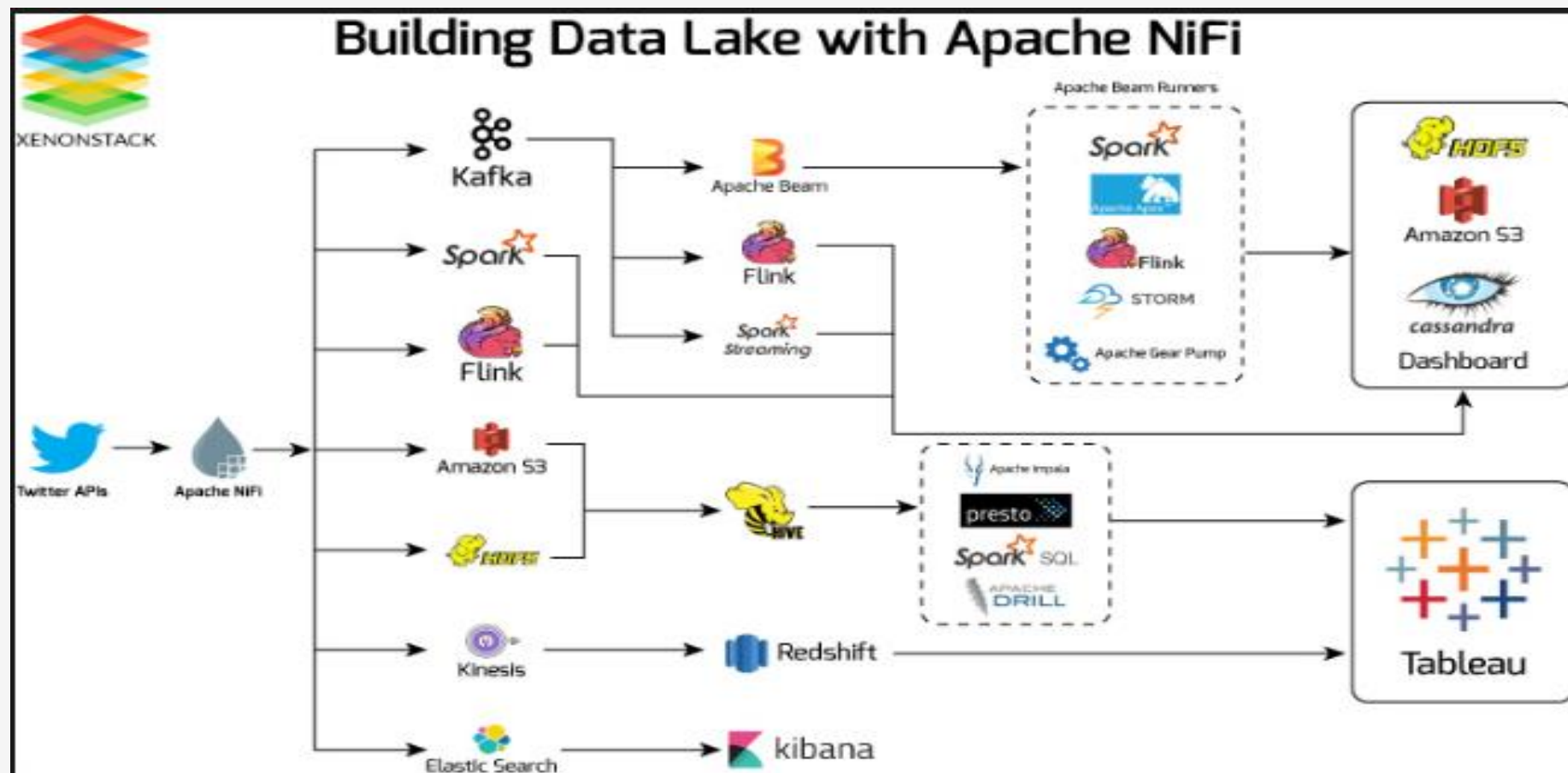
How SQOOP works (import)



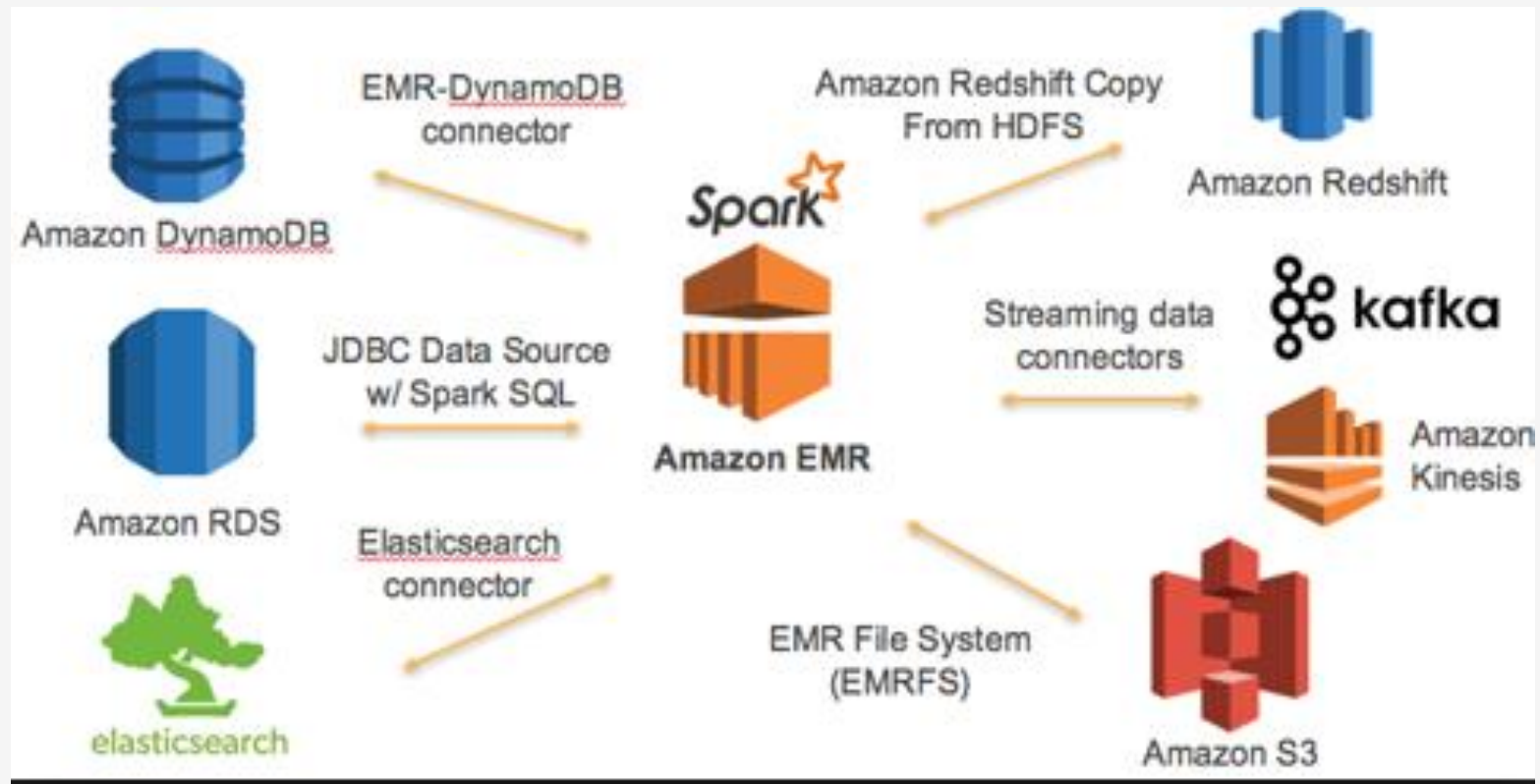
Effective Big Data Solutions



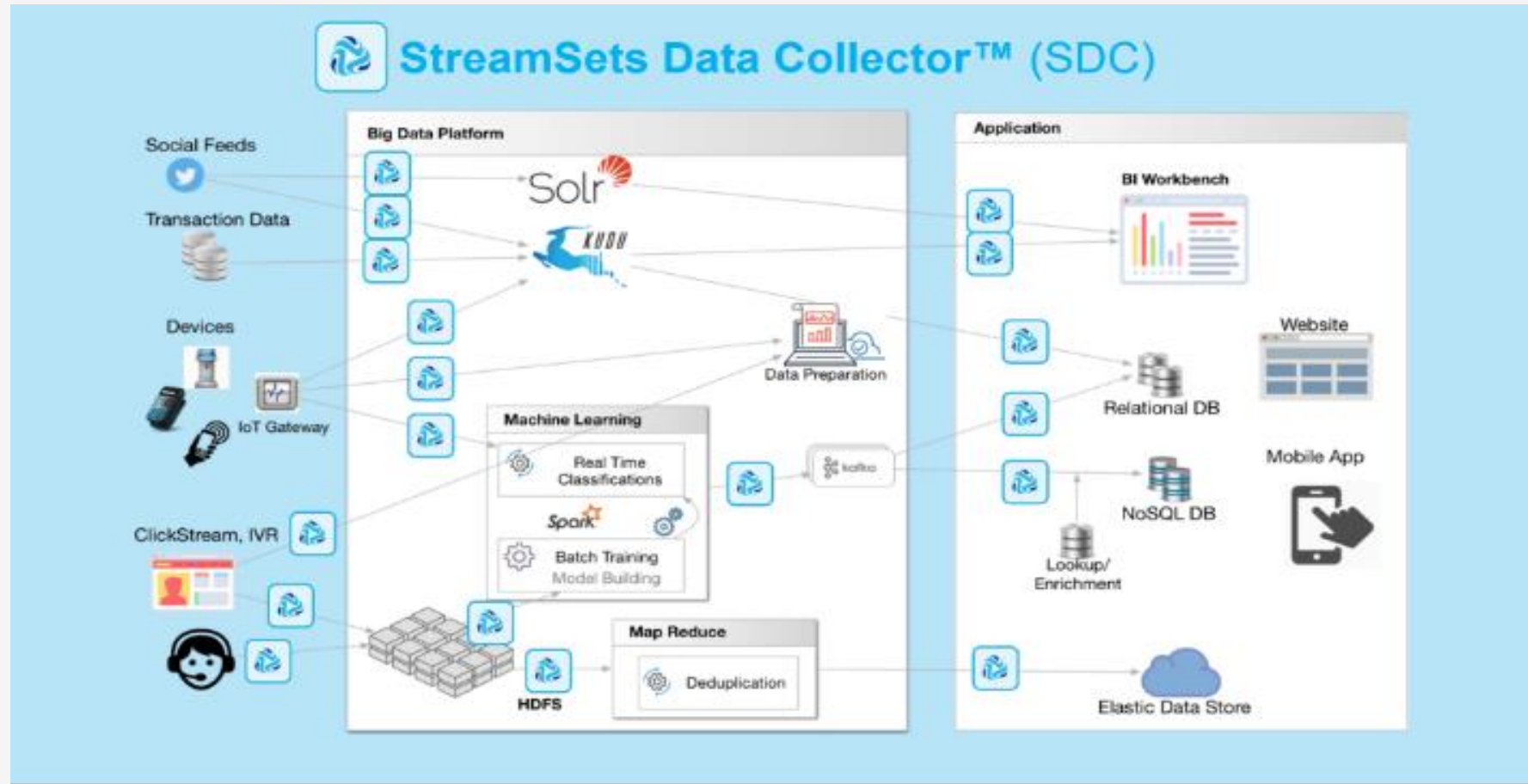
Effective Big Data Solutions



Effective Big Data Solutions



Effective Big Data Solutions



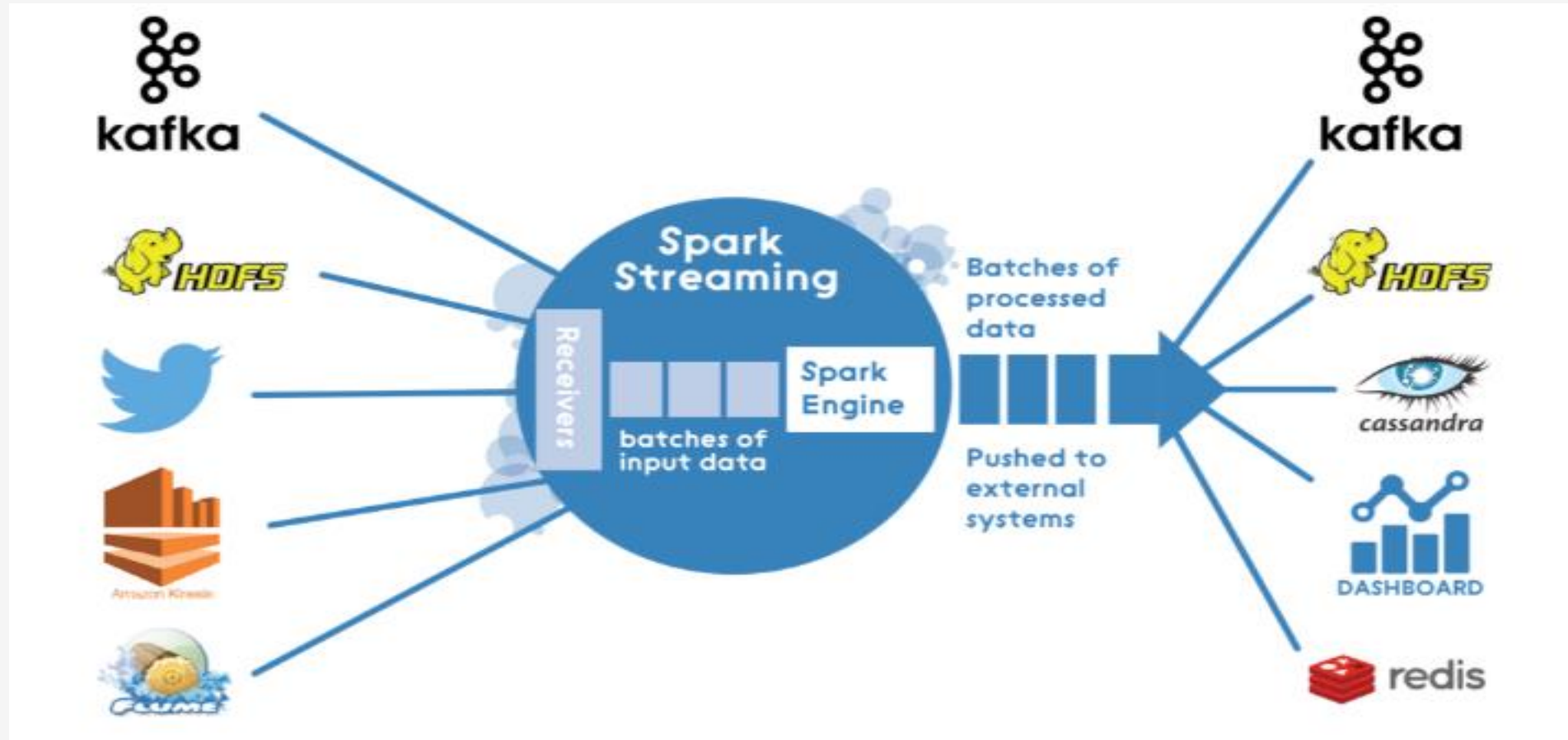
Effective Big Data Solutions

➤ **Big Data Ingestion Approaches**

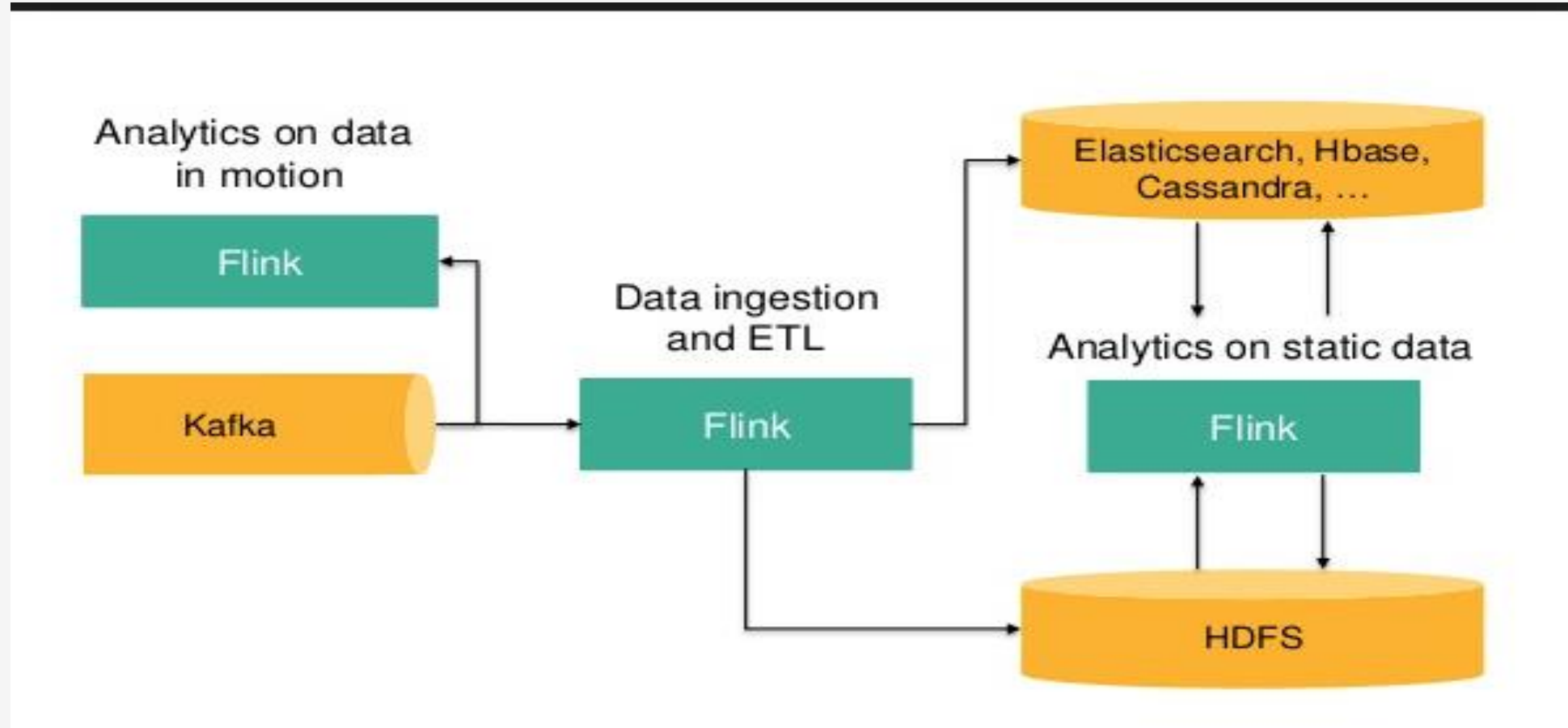
❑ **Streaming Mode Ingestion**

- **Twitter to Hadoop using Nifi, Kafka and Spark Streaming**
- **IoT to HDFS using Nifi, Kafka and Storm**
- **Web Analytics to HBase using Kafka and Storm**
- **DB Change using Kafka Connector and Flink**

Effective Big Data Solutions



Effective Big Data Solutions



Effective Big Data Solutions

➤ Ad Hoc Queries over Hadoop

Ad Hoc Analytics	Relational Databases	Big Data Sets
Data Volume	Megabytes - Gigabytes	Terabytes - Petabytes
Data Velocity	Near real-time updates (Seconds)	Real-time updates (milliseconds)
Data Variety	Structured data	Structured and Unstructured Data
Data Model	10s of tables/variables	100s - 1000s of tables/variables

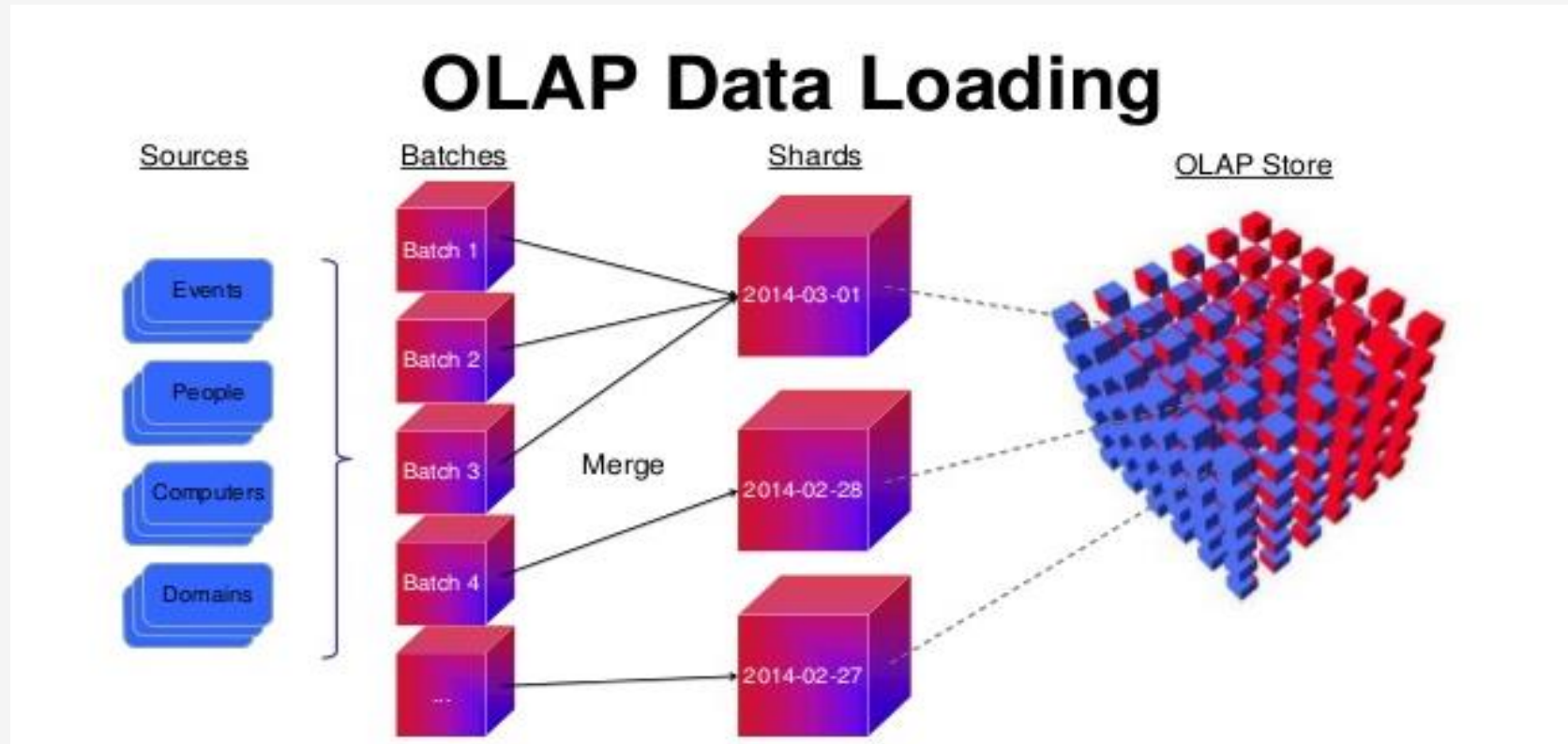
Effective Big Data Solutions

➤ **Ad Hoc Queries over Hadoop**

- ☐ **Presto**
- ☐ **Hive LLAP**
- ☐ **Impala**
- ☐ **Spark SQL**

Effective Big Data Solutions

➤ OLAP over Big Data



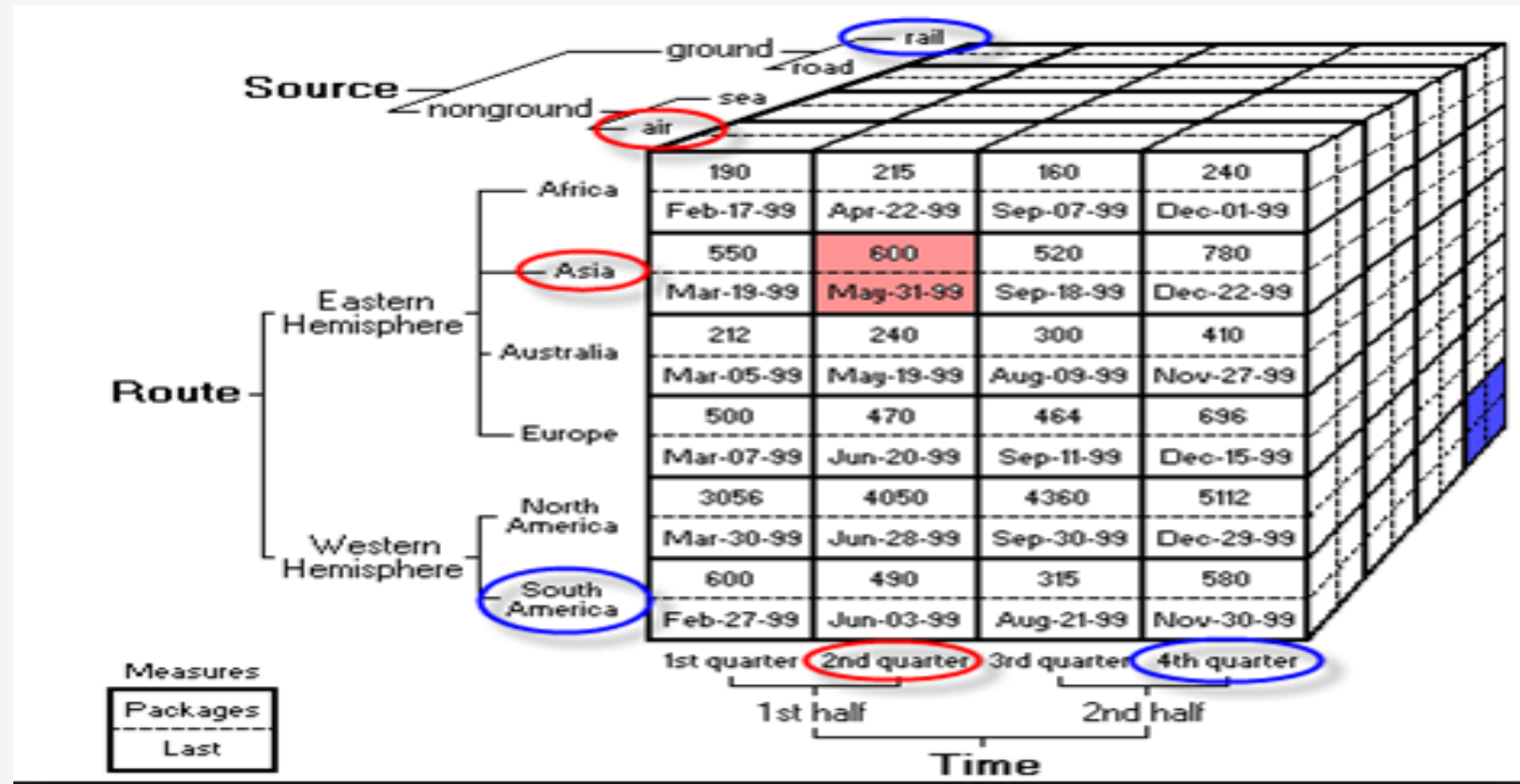
Effective Big Data Solutions

➤ OLAP over Big Data

OLTP	OLAP
-On line transaction processing	on-line analytical processing
-update, insert and delete	select
-normalized	denormalised
-more number of tables	less number of tables
-limited number of index	more number of indexes
-source system	target system
-endusers	business analysis
-single source system	multisource system

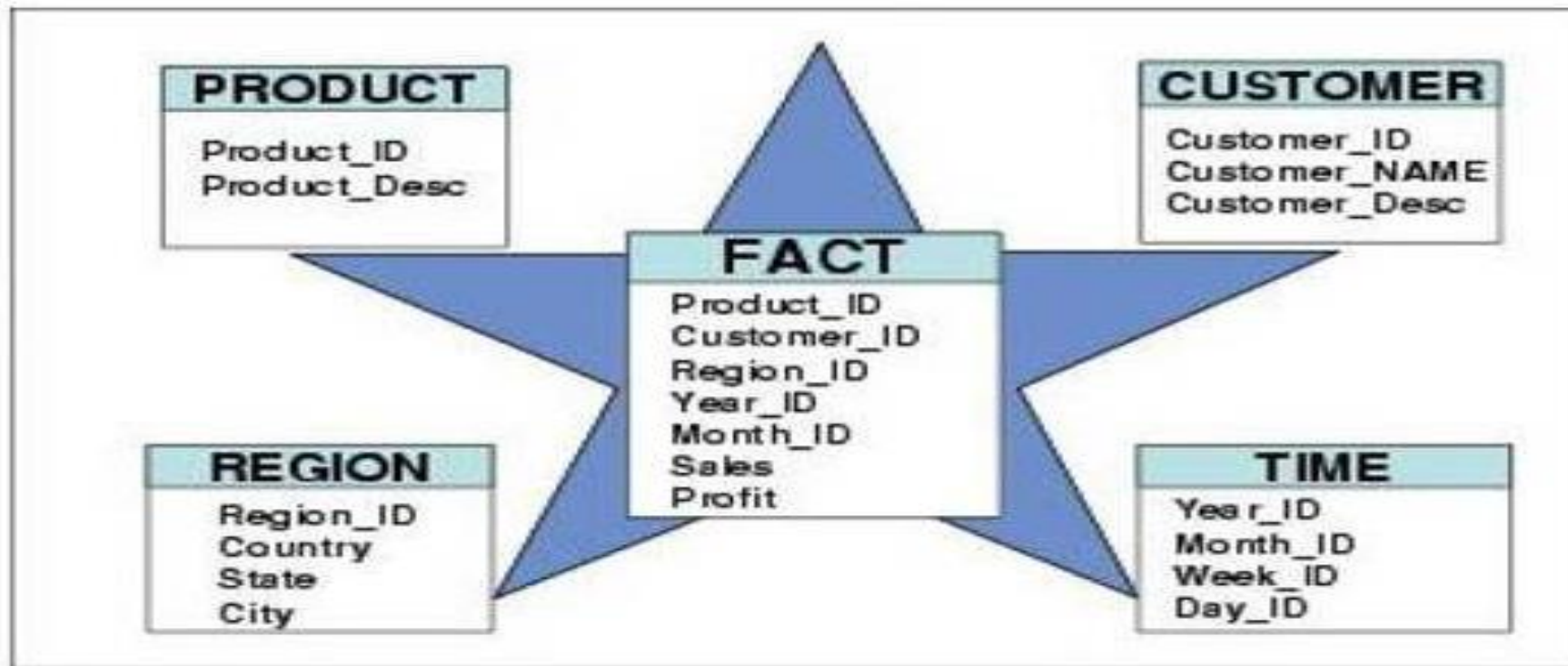
Effective Big Data Solutions

➤ OLAP over Big Data – Dimension and Measure



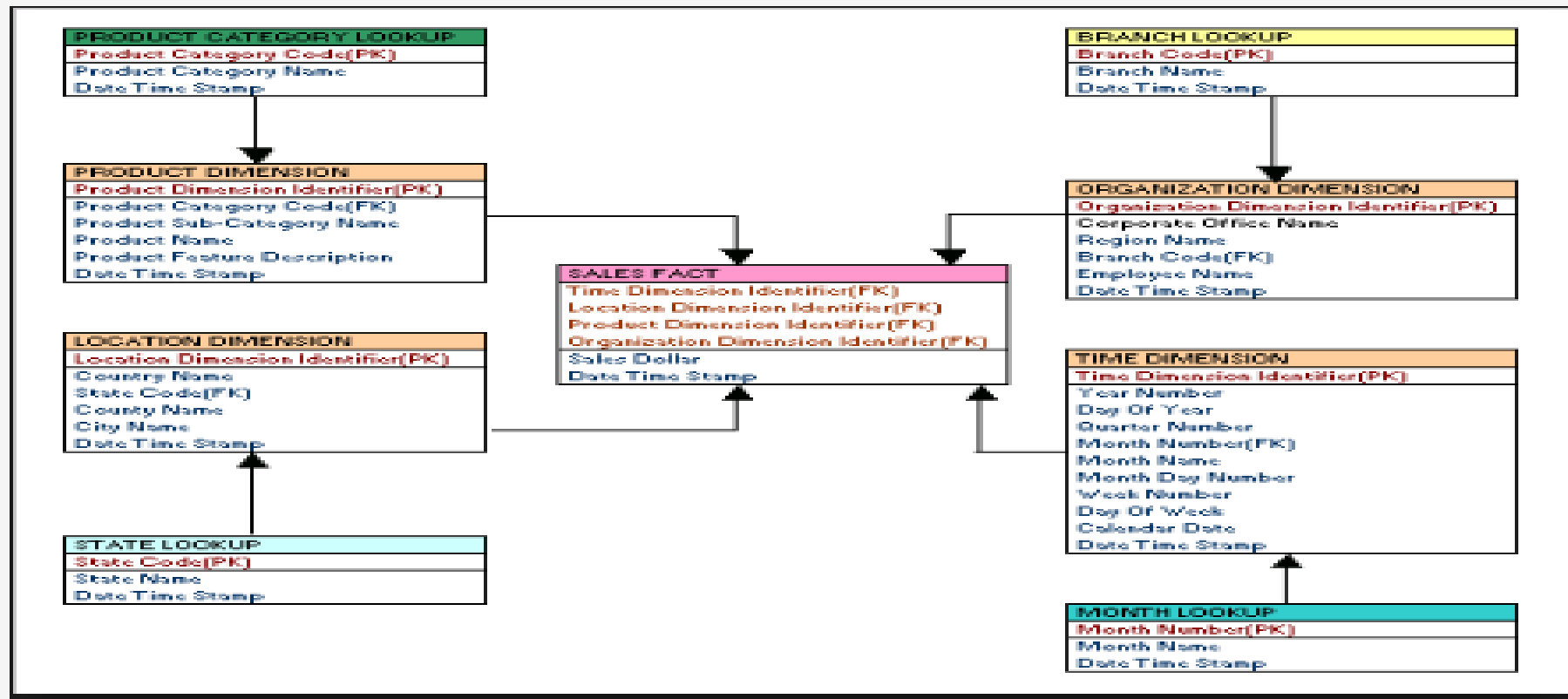
Effective Big Data Solutions

➤ OLAP over Big Data – Star Schema



Effective Big Data Solutions

➤ OLAP over Big Data – Snowflake Schema



Effective Big Data Solutions

➤ OLAP over Big Data – SCD

A **Slowly Changing Dimension** (SCD) is a **dimension** that stores and manages both current and historical data over time in a data warehouse. It is considered and implemented as one of the most critical ETL tasks in tracking the history of **dimension** records.

Effective Big Data Solutions

➤ OLAP over Big Data

	Snowflake Schema	Star Schema
DimTable Normalization:	3 Normal Form	2 Normal Denormalized Form
Joins:	Higher number of Joins	Fewer Joins
Ease of Use:	More complex queries and hence less easy to understand	Less complex queries and easy to understand
Query Performance:	More foreign keys-and hence more query execution time	Less no. of foreign keys and hence lesser query execution time
Ease of maintenance/change:	No redundancy and hence more easy to maintain and change	Has redundant data and hence less easy to maintain/change
Type of Datawarehouse:	Good to use for small datawarehouses/datamarts	Good for large datawarehouses
Dimension table:	It may have more than one dimension table for each dimension	Contains only single dimension table for each dimension

Effective Big Data Solutions

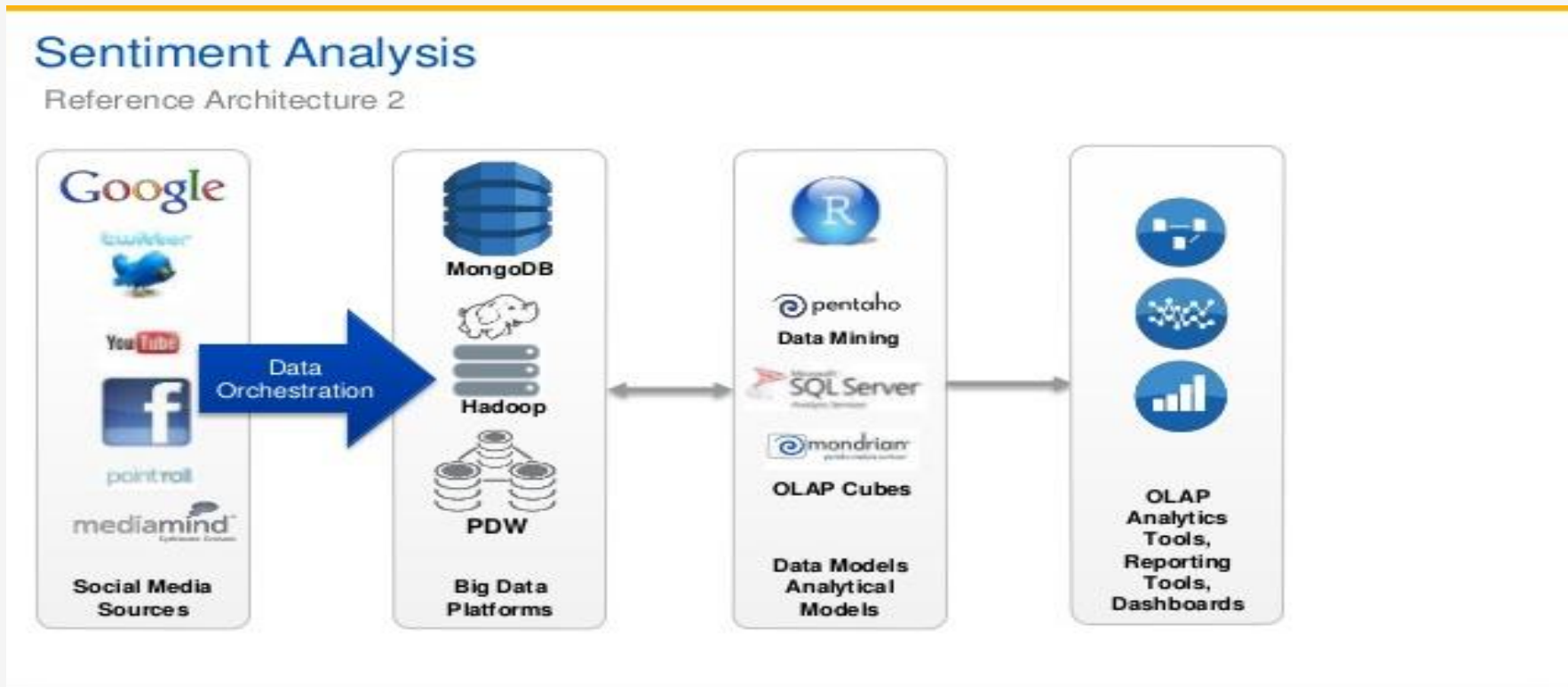
➤ OLAP over Big Data

Big OLAP on Big Data



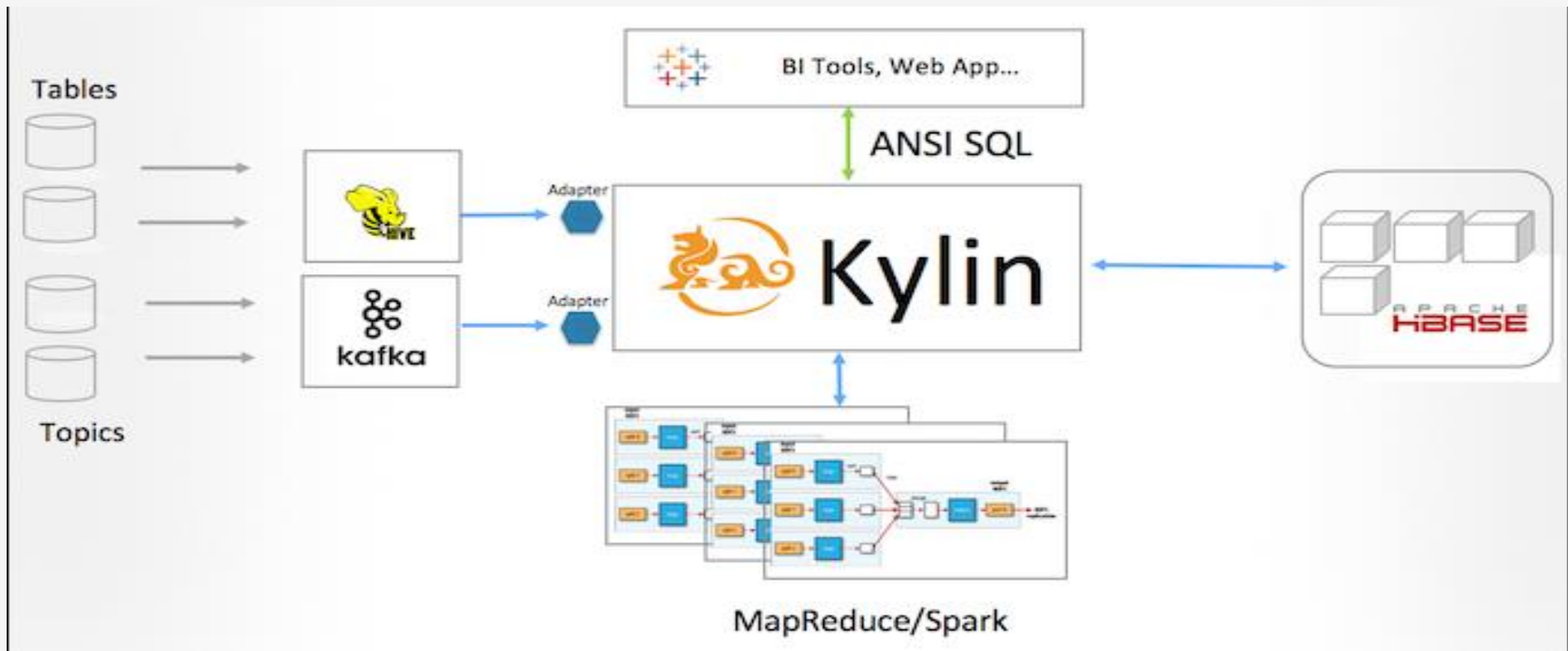
Effective Big Data Solutions

➤ OLAP over Big Data



Effective Big Data Solutions

➤ OLAP over Big Data



Effective Big Data Solutions

➤ Batch Distributed Data Processing

Batch processing is very efficient in processing high volume data. Where data is collected, entered to the system, processed and then results are produced in batches

- ☐ **Credit card companies process billing**
- ☐ **Building Machine Learning Algorithms**
- ☐ **Classifying Text in Money Transfers**
- ☐ **Analyzing and comparing your energy consumption with that of other consumers**
- ☐ **Analyzing IoT data across data centers and cloud**

Effective Big Data Solutions

➤ Real Time Streaming Data Processing

- ☐ **Build a global News Scanner that scrapes news in near real time, and uses sophisticated text analysis, SimHash, Random Indexing and Streaming K-Means to produce a geopolitical monitoring tool that allows users to track major world events as they unfold**
- ☐ **Real-Time Image Recognition**
- ☐ **Approximate computing for stream analytics**
- ☐ **User behavior anomaly detection for information security**
- ☐ **Deduplication and author-disambiguation of streaming records via supervised models based on content encoders**
- ☐ **Large-scale ads CTR prediction**
- ☐ **Real-Time Detection of Anomalies in the Database Infrastructure**

Effective Big Data Solutions

- ☐ Simple Event Processors
- ☐ Stream Processors
- ☐ Complex Event Processors
- ☐ Real time Data Collection and Integration
- ☐ Ad-hoc Data Analytics
- ☐ Real time Anomaly Detection
- ☐ Real time Predictive Analytics
- ☐ Continuous Query

Effective Big Data Solutions

- ☐ Flume
- ☐ NiFi
- ☐ Gearpump
- ☐ Apex
- ☐ Kafka Streams
- ☐ Spark Streaming
- ☐ Storm (and Trident)
- ☐ Flink
- ☐ Samza
- ☐ Ignite
- ☐ Beam

Effective Big Data Solutions

- Real time Streaming
- Near Real time Streaming
- Lambda Architecture
- Kappa Architecture

Effective Big Data Solutions

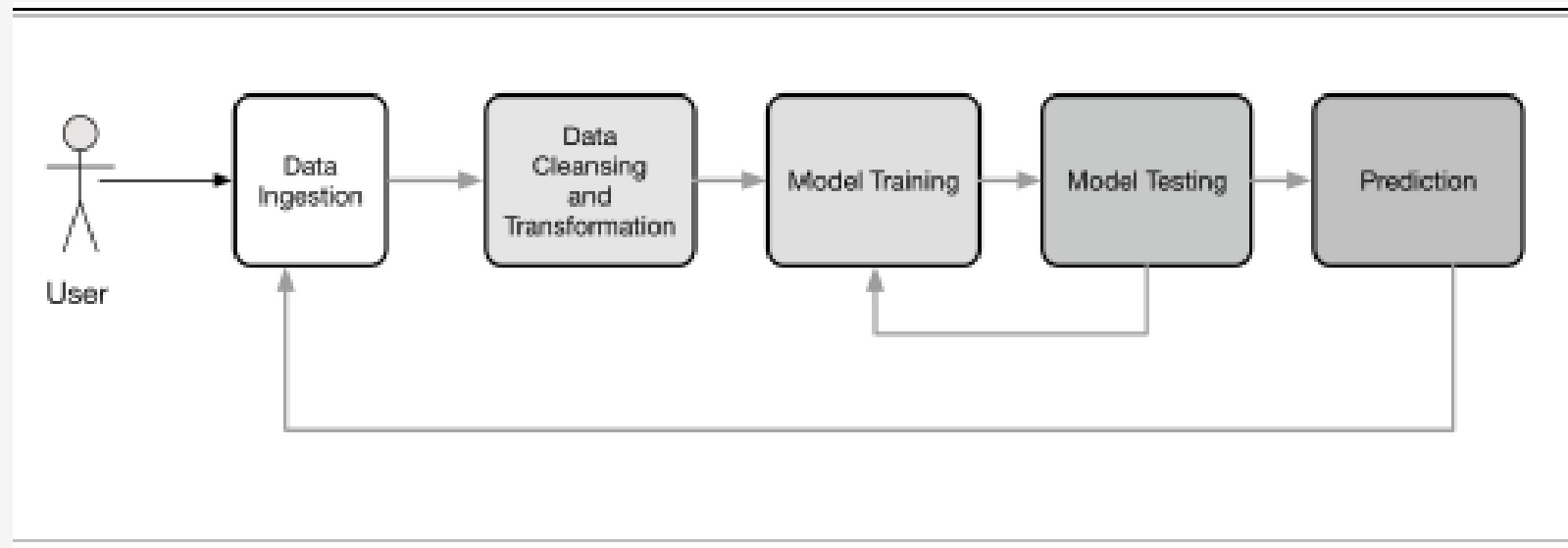
- Stream Joins
- Top N (Trending)
- Rolling Windows

Effective Big Data Solutions

- External Lookup
- Responsive Shuffling (HBase Regions Split)
- Out-of-Sequence Events

Effective Big Data Solutions

➤ Machine Learning



A general machine-learning pipeline

Effective Big Data Solutions

- **Machine Learning – One Pipe Line MovieStream Example**
- **Collecting data about users, their behavior, and our content titles**
- **Transforming this data into features**
- **Training our models, including our training-testing and model-selection phases**
- **Deploying the trained models to both our live model-serving system as well as using these models for offline processes**
- **Feeding back the model results into the MovieStream website through recommendation and targeting pages**
- **Feeding back the model results into MovieStream's personalized marketing channels**
- **Using the offline models to provide tools to MovieStream's various teams to better understand user behavior, characteristics of the content catalogue, and drivers of revenue for the business**

Effective Big Data Solutions

➤ **Machine Learning – Use Cases**

❖ **Recommendation Engine**

- **Personalization**
- **Targeted marketing and customer segmentation**
- **Predictive modeling and analytics**

Effective Big Data Solutions

➤ Machine Learning – Use Cases

❖ Classification Model

- Predicting the probability of Internet users clicking on an online advert; here, the classes are binary in nature (that is, click or no click)
- Detecting fraud; again, in this case, the classes are commonly binary (fraud or no fraud)
- Predicting defaults on loans (binary)
- Classifying images, video, or sounds (most often multiclass, with potentially very many different classes)
- Assigning categories or tags to news articles, web pages, or other content (multiclass)
- Discovering e-mail and web spam, network intrusions, and other malicious behavior (binary or multiclass)
- Detecting failure situations, for example, in computer systems or networks
- Ranking customers or users in order of probability that they might purchase a product or use a service
- Predicting customers or users who might stop using a product, service, or provider (called churn)

Effective Big Data Solutions

➤ Machine Learning – Use Cases

❖ Regression Model

- Predicting stock returns and other economic variables
- Predicting loss amounts for loan defaults (this can be combined with a classification model that predicts the probability of default, while the regression model predicts the amount in the case of a default)
- Recommendations (the Alternating Least Squares factorization model)
- Predicting customer lifetime value (CLTV) in a retail, mobile, or other business, based on user behavior and spending patterns

Effective Big Data Solutions

➤ Machine Learning – Use Cases

❖ Clustering Model

- **Segmenting users or customers into different groups based on behavior characteristics and metadata**
- **Grouping content on a website or products in a retail business**
- **Finding clusters of similar genes**
- **Segmenting communities in ecology**
- **Creating image segments for use in image analysis applications such as object detection**

Effective Big Data Solutions

➤ Machine Learning – Use Cases

❖ Dimensionality Reduction

- Exploratory data analysis
- Extracting features to train other machine learning models
- Reducing storage and computation requirements for very large models in the prediction phase (for example, a production system that makes predictions)
- Reducing a large group of text documents down to a set of hidden topics or concepts
- Making learning and generalization of models easier when our data has a very large number of features (for example, when working with text, sound, images, or video data, which tends to be very high-dimensional)

Effective Big Data Solutions

➤ Machine Learning – Use Cases

❖ Advanced Text Processing

- Text data processing, feature extraction, and the modeling pipeline
- Evaluate the similarity between two documents based on the words in the documents
- Use the extracted text features as inputs for a classification model
- Natural language processing to model words themselves as vectors and Word2Vec model to evaluate the similarity between two words based on their meaning
- Sentiment Analysis

Effective Big Data Solutions

➤ Machine Learning – Use Cases

❖ Real-Time Machine Learning

- **Online learning, where models are trained and updated on new data as it becomes available**

Effective Big Data Solutions

➤ Machine Learning – Use Cases

❖ Deep Learning

[Deep Learning , CNN, RNN] extended version of earlier Neural Network (Machine Learning)

- **Skilled Robotics & Labor Automation**
- **Text Analysis**
- **Cybersecurity**
- **Time Series – Predictive Deep Learning**
- **Prescriptive Deep Learning Systems**

Effective Big Data Solutions

➤ Graph Processing

- ❑ Multi-Label Graph Analysis and Computations Using GraphX
- ❑ PageRanking

Job Market

Looking for Big Data Architect // Scarborough, ON // 12 Months



Dushyant Singh Negi <dsnegi@eteaminc.com>

Fri 05-05, 5:18 PM

You ↕



↩ Reply | ▾

Getting too much email from Dushyant Singh Negi <dsnegi@eteaminc.com>? [You can unsubscribe](#)

Hi,
Greetings!!

My name is Dushyant Singh Negi and I am recruiter at eTeam Inc. eTeam Inc is a global contingency staffing firm servicing fortune 1000 clients globally. We have an excellent job opportunity with one of our client.

Job Title: Big Data Architect

Location: Scarborough, ON

Duration: 12 Months

Job Description:

- 2+ years of hands-on experience with the technologies in the Hadoop ecosystem like Hadoop, HDFS, Spark, MapReduce, Pig, Hive.

Job Market

Hadoop developer



Neha Jaiswal <njaiswal@eteaminc.com>

Mon 05-01, 2:21 PM

You ↕



Reply | v

Getting too much email from Neha Jaiswal <njaiswal@eteaminc.com>? [You can unsubscribe](#)

Greetings!!

My name is Neha Jaiswal and I am recruiter at eTeam Inc. eTeam Inc is a global contingency staffing firm servicing fortune 1000 clients globally. We have an excellent job opportunity with one of our client.

Position: Sr. Big Data Developer

Location : Toronto, ON

Contract: 6+ months

Technical/Functional Skills (in priority order)

We are looking for a Hadoop developer with a very good experience on Big data and Hadoop Ecosystem. The person should have experience on distributed computing, multithreading, java (java 8.0), Kafka, Spark, HBase, Hive and other components to work as part of development team. The person should be familiar with columnar database.

Mandatory Skills –

Job Market



Sam <sam@infinitysts.com>

Fri 05-05, 3:25 PM

You ↵

Inbox

This message was sent with high importance.

This message has been marked as Confidential.

Hi Bin,

I was reviewing your resume today and have a role available which looks to be really good fit for you. Please take a look at the job requirements for consideration:

Position – Big Data Developer

Location – Toronto Downtown, ON

Duration – 6+ months Contract

Job Responsibilities:

- MapReduce(Mandatory),Hadoop(Mandatory),Hive Apache, Pig, Oozie, Flume, Sqoop As a Senior Developer,
- Strong in Java, Maven, Nexus, Jenkins.
- Good Experience on :SQL, CSV, XML, JSON, Copybooks

Job Market

Job Opportunity - Big Data/Cloud Specialist - TD - TDJP00020198



Tyler Fenton <tyler@isgsearch.ca>

Wed 05-03, 5:14 PM

You ↕



↩ Reply | ▾

Hi Bin,

I tried giving you a call. The reason for my message is related to the opportunity available with TD as a Big Data/Cloud Specialist. They need a contract resource for 1 year to help install and integrate a vendor product. I see you are still with Scotia, but if you are interested I would love the chance to speak with you!

Job Description

Position Title: Niche IT- Big Data Hadoop, Grid Computing Cloud Specialist

•# of positions: 1

•Start Date: ASAP

•Duration: 12 months

•Extension possible: Yes, nothing is confirmed at this time.

•Conversion Possible: No

Job Market

Melissa Buenafe

Consultant at Search and Recruitment

...

- Developing and communicating platform strategy and roadmap to IT and business leadership, drive a roadmap that aligns with business and IT priorities
 - Working closely with solution delivery team to provide guidance in industrializing analytical solutions on the Big Data platform
 - Guiding data management standards around data security and quality
 - Collaborating with external platform and application vendors to establish, maintain and augment the big data platform
 - Establishing and ensure adherence to coding and metadata standards and guidelines
 - Defining and report on metrics to assess Big Data platform performance
 - Integrating various applications and tools on common Big Data platform
- Important /key skillset: Expertise in Cloudera, Sqoop, Flume, Spark, data ingestion into Hadoop, Hive

Mon

Hi Bin

thanks for accepting my request to connect on LinkedIn. My name is Francesco Lillo and I am an entrepreneur and experienced IT recruiter running a very specialized search firm. Today I have an important contact in a major financial services and fin-tech company that has a need for a Big Data Dev in Toronto. Your profile looks like a strong fit, so I was wondering whether you would be open to speak. If interested, let's schedule some time on Monday for a quick introductory call. Best regards. Francesco. [438 238 4533](#)

Richa Shukla

Specialise in strategic Digital and Technology recruitment

...

Apr 28

Accenture would like to network with you

My name is Richa Shukla and I work in recruitment for Accenture. I recently came across your LinkedIn profile while searching for talented Digital professionals.

As I read your bio, I thought you might be an excellent person to network with regarding Accenture's current Big Data opportunities based in Toronto.