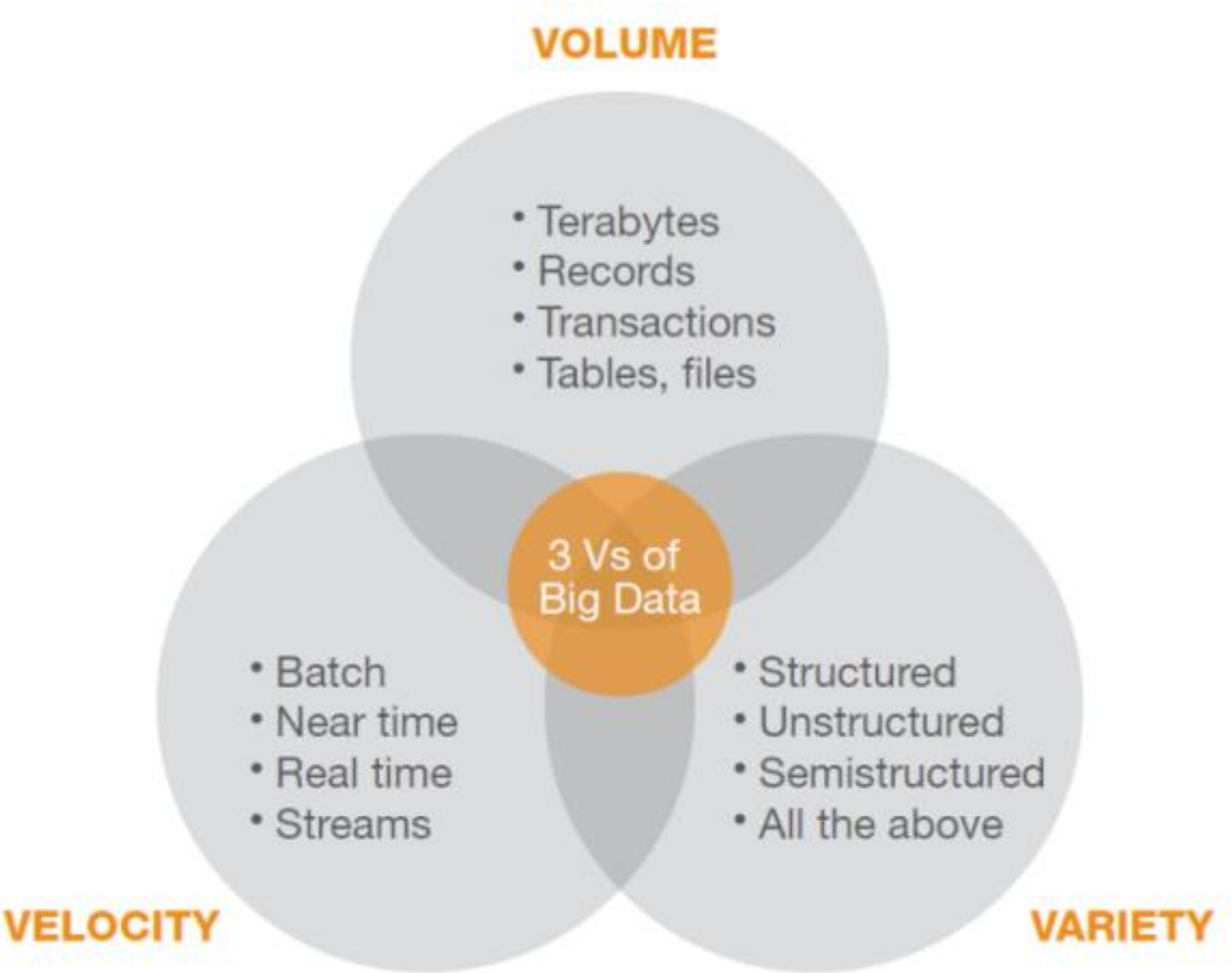
Hadoop Intro and its ecosystem

- Sai Makineni

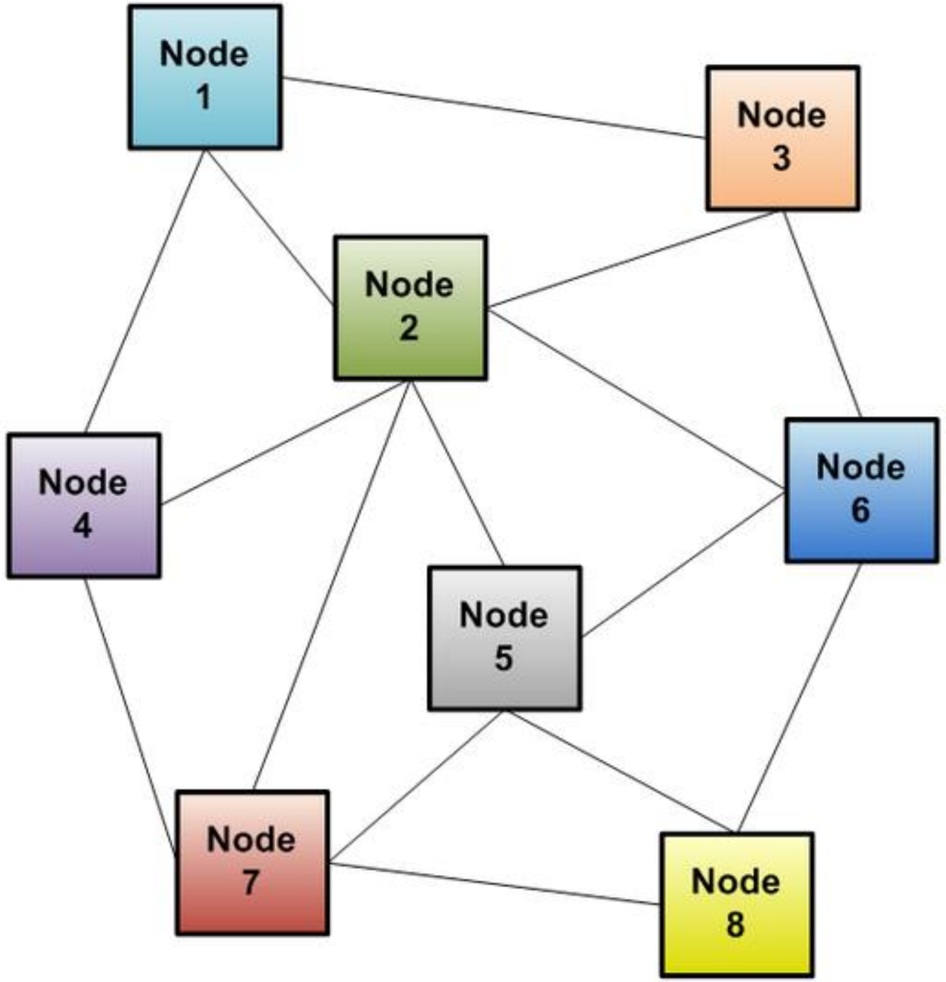
Overview

- Big Data Challenges
- Distributed system and challenges
- Hadoop Introduction
- History
- Who uses Hadoop
- The Hadoop Ecosystem
  - Hadoop core components
    - HDFS
    - YARN
    - Spark
  - Other Hadoop ecosystem components
    - Hbase
    - Hive
    - Sqoop
    - Zookeeper
    - NiFi
    - Kafka

Big Data Challenges



Solution: Distributed system



Distributed System Challenges

- Programming Complexity
- Finite bandwidth
- Partial failure
- The data bottleneck

New Approach to distributed computing

Hadoop:

A scalable fault-tolerant distributed system for data storage and processing

- Distribute data when the data is stored
- Process data where the data is
- Data is replicated

Hadoop Introduction

- Apache Hadoop is an open-source software framework for storage and large-scale processing of data-sets on clusters of commodity hardware.
- Some of the characteristics:
  - Open source
  - Distributed processing
  - Distributed storage
  - Scalable
  - Reliable
  - Fault-tolerant
  - Economical
  - Flexible

History

- Originally built as a Infrastructure for the “Nutch” project.
- Based on Google’s map reduce and google File System.
- Created by Doug Cutting in 2005 at Yahoo
- Named after his son’s toy yellow elephant.

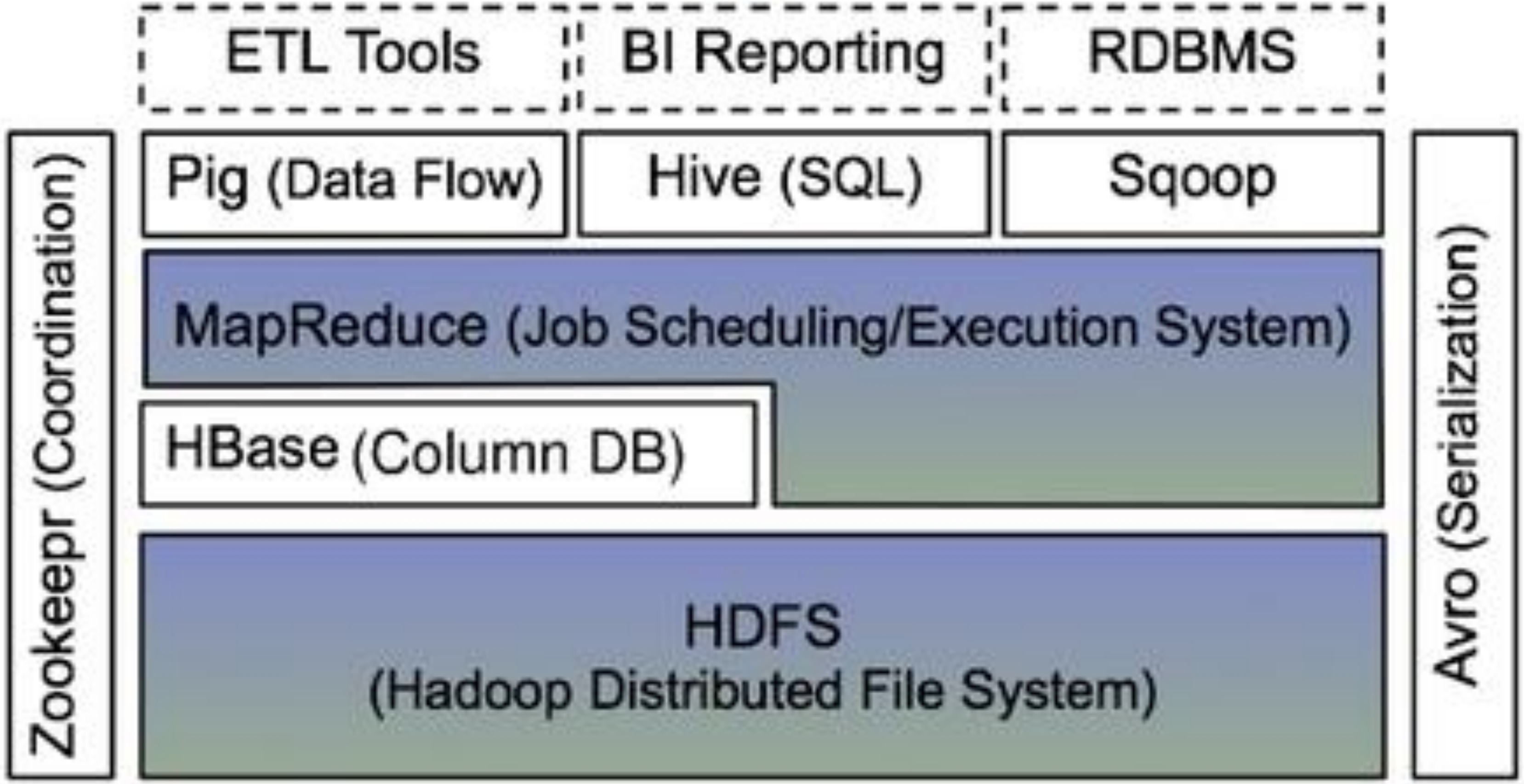
Who uses Hadoop

- http://wiki.apache.org/hadoop/PoweredBy
- http://wiki.apache.org/hadoop/Distributions%20and%20Commercial%20Support

The Hadoop Ecosystem

- http://hadoopecosystemtable.github.io/

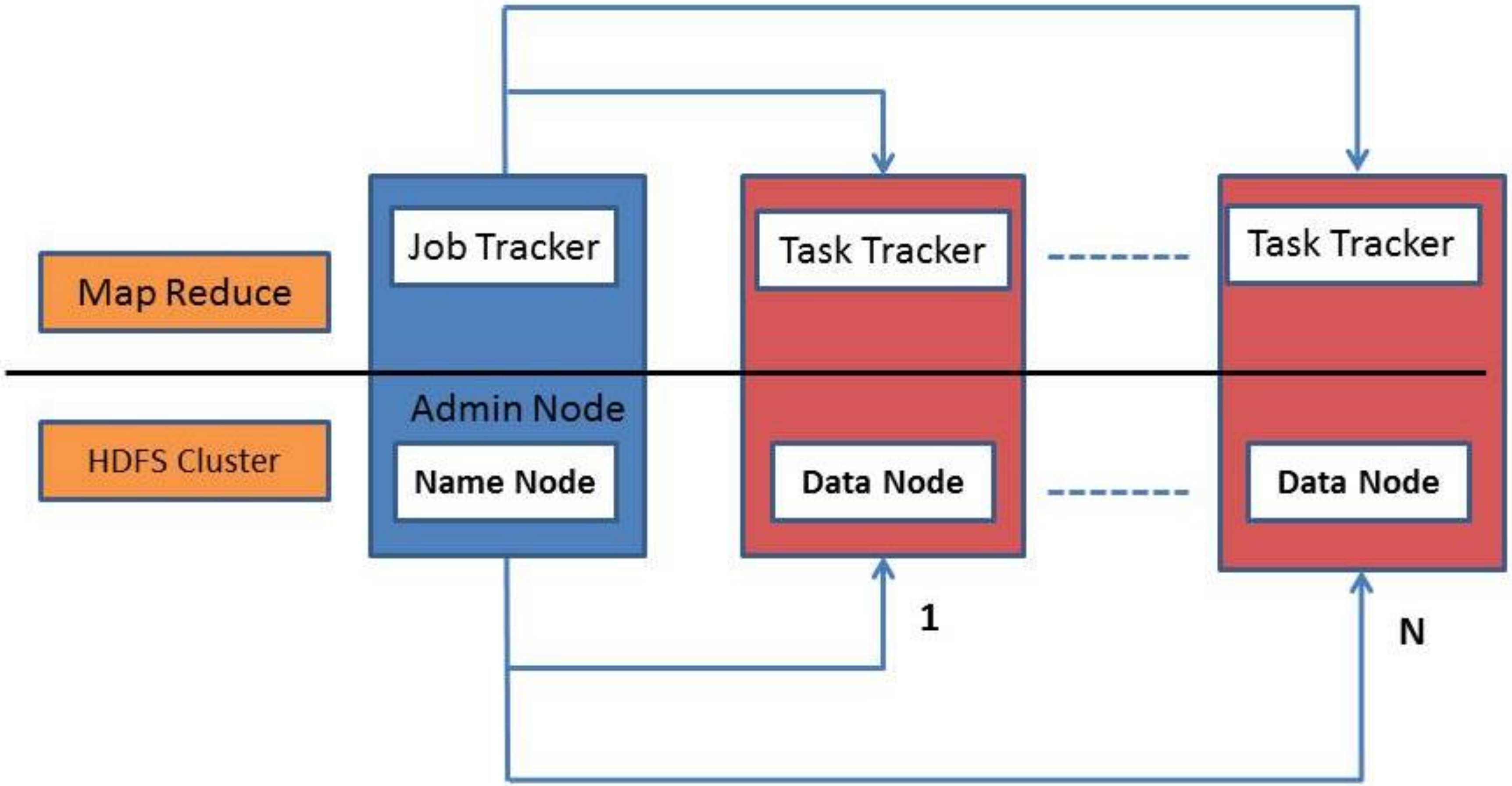
The Hadoop Ecosystem



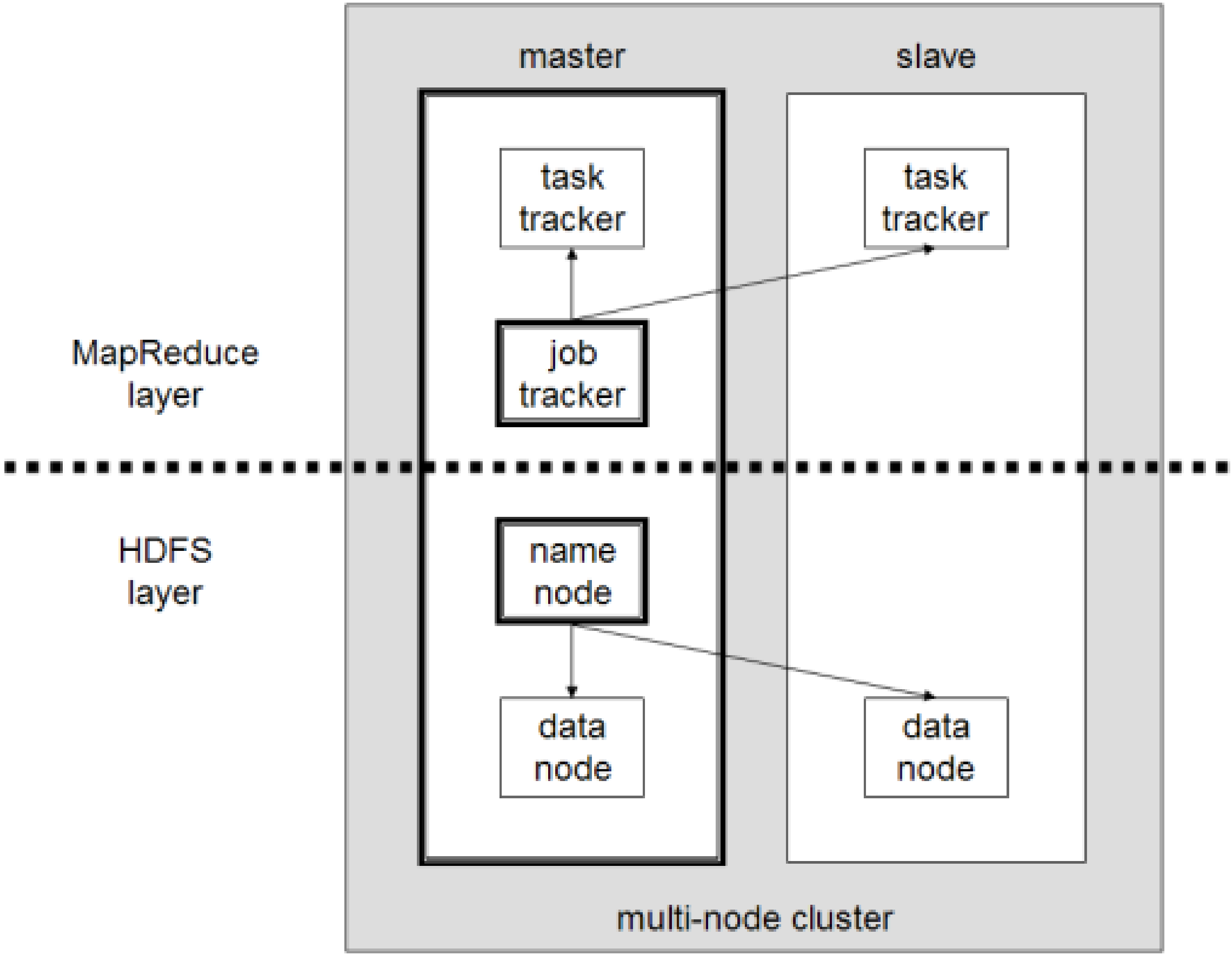
Hadoop Core Components

- HDFS – Hadoop Distributed File System (Storage)
- YARN (Processing)

Hadoop Core Components



A multi-node Hadoop cluster



Nodes

- NameNode:
  - Master of the system
  - Maintains and manages the blocks which are present on the DataNodes
- DataNodes:
  - Slaves which are deployed on each machine and provide the actual storage
  - Responsible for serving read and write requests for the clients
- Jobtracker:
  - takes care of all the job scheduling and assign tasks to Task Trackers.
- TaskTracker:
  - a node in the cluster that accepts tasks - Map, Reduce and Shuffle operations - from a jobtracker

HDFS

- Hadoop Distributed File System (HDFS) is designed to reliably store very large files across machines in a large cluster. It is inspired by the GoogleFileSystem.
- Distribute large data file into blocks
- Blocks are managed by different nodes in the cluster
- Each block is replicated on multiple nodes
- Name node stored metadata information about files and blocks

YARN

- The Mapper:
  - Each block is processed in isolation by a map task called mapper
  - Map task runs on the node where the block is stored

- The Reducer:
  - Consolidate result from different mappers
  - Produce final output

What makes Hadoop unique

- Moving computation to data, instead of moving data to computation.
- Simplified programming model: allows user to quickly write and test
- Automatic distribution of data and work across machines

Other Hadoop components in Ecosystem

HBase	Hadoop database for random read/write access
Hive	SQL-like queries and tables on large datasets
Pig	Data flow language and compiler
Oozie	Workflow for interdependent Hadoop jobs
Sqoop	Integration of databases and data warehouses with Hadoop
Flume	Configurable streaming data collection
ZooKeeper	Coordination service for distributed applications

Hbase

- HBase is an open source, non-relational, distributed database modeled after Google's BigTable.
- It runs on top of Hadoop and HDFS, providing BigTable-like capabilities for Hadoop.

Features of Hbase

- Type of NoSql database
- Strongly consistent read and write
- Automatic sharding
- Automatic RegionServer failover
- Hadoop/HDFS Integration
- HBase supports massively parallelized processing via MapReduce for using HBase as both source and sink.
- HBase supports an easy to use Java API for programmatic access.
- HBase also supports Thrift and REST for non-Java front-ends.

When to use Hbase

- When there is real big data: millions or billions of rows, in other way data can not store in a single node.
- When random read/write access to big data
- When require to do thousands of operations on big data
- When there is no need of extra features of RDMS like typed columns, secondary indexes, transactions, advanced query languages, etc.
- When there is enough hardware.

Difference between Hbase and HDFS

HDFS	Hbase
Good for storing large file	Built on top of HDFS. Good for hosting very large tables like billions of rows X millions of column
Write once. Append to files in some of recent versions but not commonly used	Read/write many
No random read/write	Random read/write
No individual record lookup rather read all data	Fast records lookup(update)

Hive

- An sql like interface to Hadoop.
- Data warehouse infrastructure built on top of Hadoop
- Provide data summarization, query and analysis
- Query execution via MapReduce
- Hive interpreter convert the query to Map reduce format.
- Open source project.
- Developed by Facebook
- Also used by Netflix, Cnet, Digg, eHarmony etc.

Hive

- HiveQL example:

```
SELECT customerId, max(total_cost) from hive_purchases GROUP BY
customerId HAVING count(*) > 3;
```

Pig

- A scripting platform for processing and analyzing large data sets
- Apache Pig allows to write complex MapReduce programs using a simple scripting language.
- High level language: Pig Latin
- Pig Latin is data flow language.
- Pig translate Pig Latin script into MapReduce to execute within Hadoop.
- Open source project
- Developed by Yahoo

Pig

- Pig Latin example:

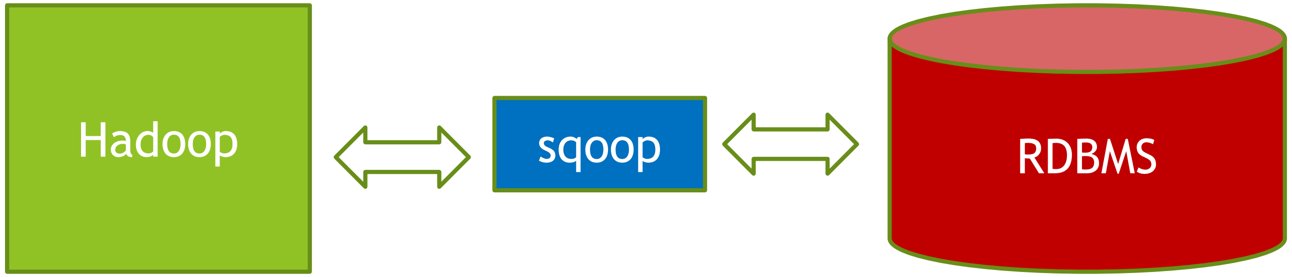
```
A = LOAD 'student' USING PigStorage() AS (name:chararray, age:int, gpa:float);
X = FOREACH A GENERATE name,$2;
DUMP X;
```

Pig and Hive

- Both requires compiler to generate Map reduce jobs
- Hence high latency queries when used for real time responses to ad-hoc queries
- Both are good for batch processing and ETL jobs
- Fault tolerant

Sqoop

- Command-line interface for transforming data between relational database and Hadoop
- Support incremental imports
- Imports use to populate tables in Hadoop
- Exports use to put data from Hadoop into relational database such as SQL server



How Sqoop works

- The dataset being transferred is broken into small blocks.
- Map only job is launched.
- Individual mapper is responsible for transferring a block of the dataset.

How Sqoop works

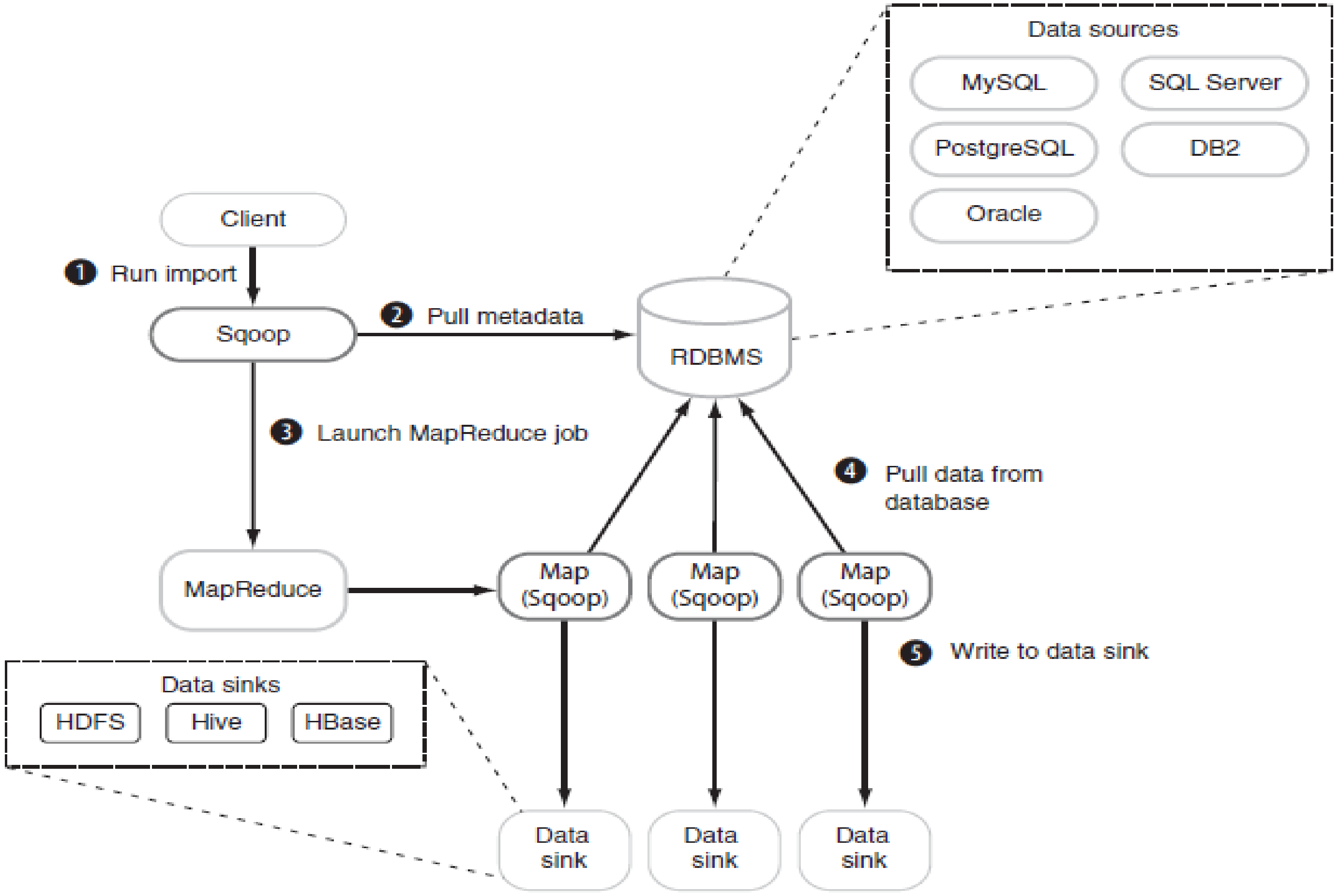


Figure 2.20 **Five-stage Sqoop import overview: connecting to the data source and using MapReduce to write to a data sink**

Flume

- Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS).

How flume works

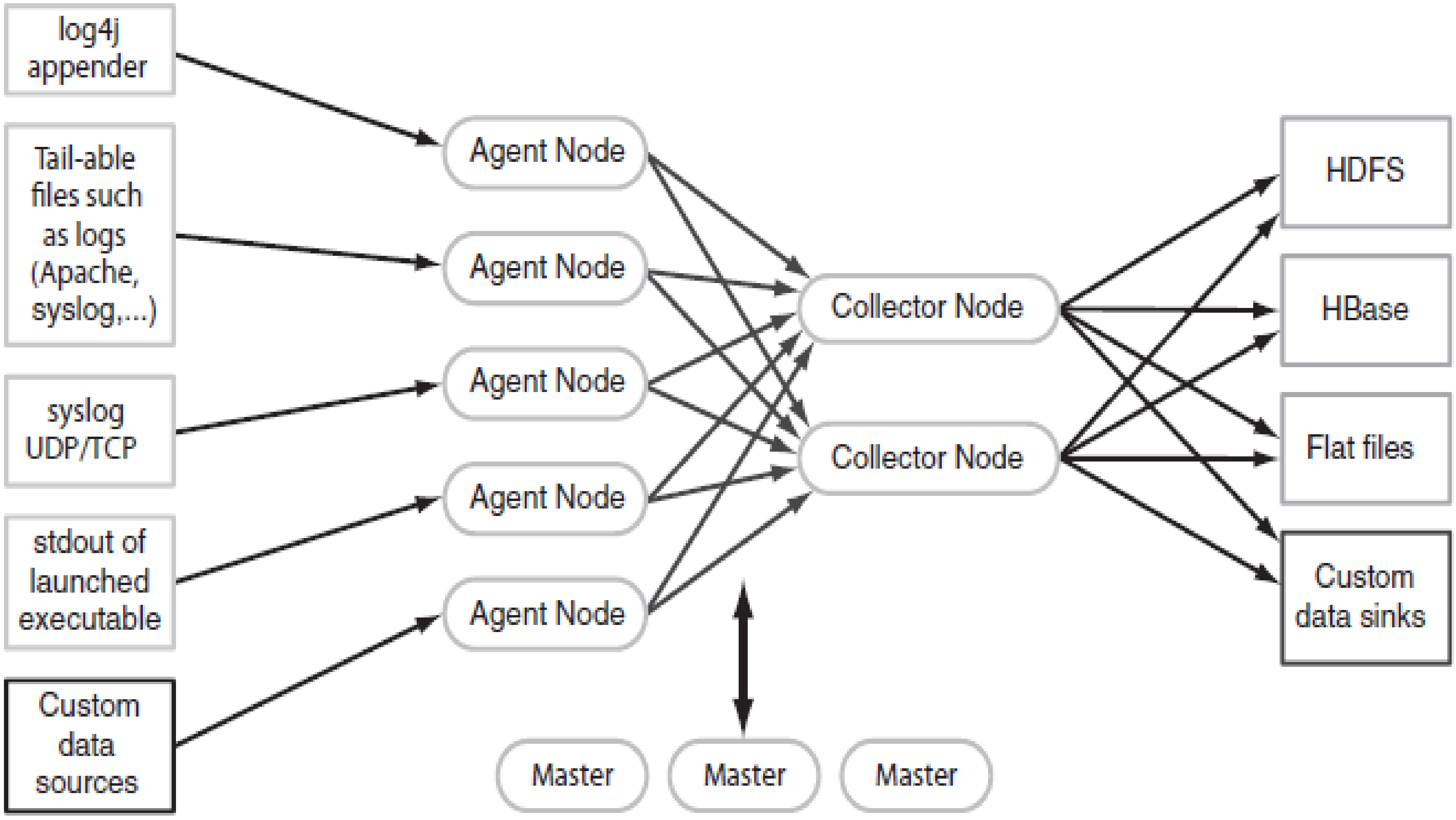


Figure 2.2 Flume architecture for collecting streaming data

How flume works

- Data flows like:

  Agent tier -> Collector tier -> Storage tier

- **Agent nodes** are typically installed on the machines that generate the logs and are data’s initial point of contact with Flume. They forward data to the next tier of ***collector nodes***, which aggregate the separate data flows and forward them to the final ***storage tier***.

Zookeeper

- Because coordinating distributed systems is a Zoo.
- ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services.

Questions?