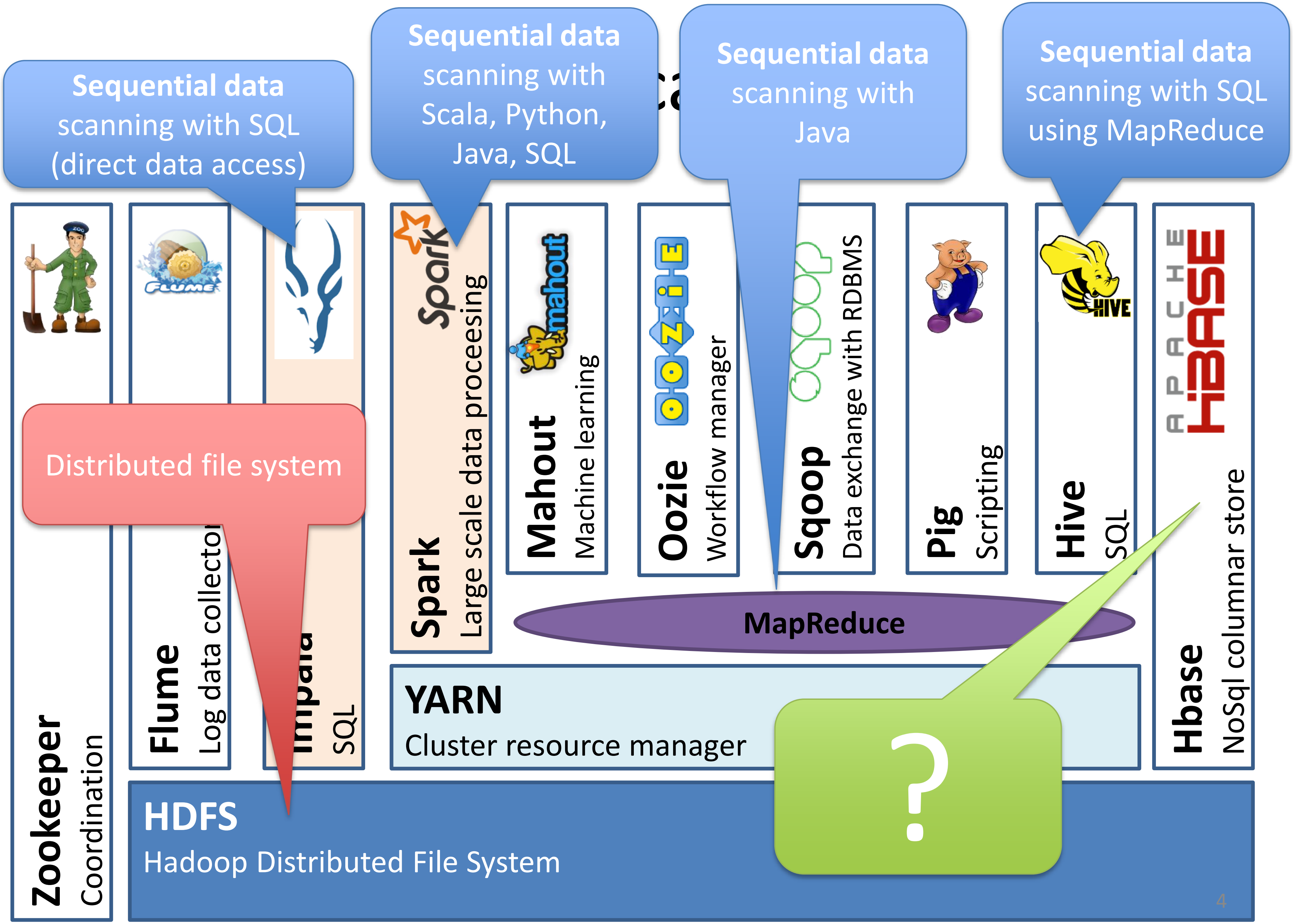
APACHE
HBASE

Cloudera Image for hands-on

- Installation instruction
  - https://cern.ch/zbaranow/CVM.txt

Agenda

- Now
- HBase architecture
- Data operations - hands on
- Summary



Sequential data scanning with SQL (direct data access)

Sequential data scanning with Scala, Python, Java, SQL

Sequential data scanning with Java

Sequential data scanning with SQL using MapReduce

Distributed file system

?

Zookeeper Coordination

Flume Log data collector

Impala SQL

Spark Large scale data processing

Mahout Machine learning

Oozie Workflow manager

Sqoop Data exchange with RDBMS

Pig Scripting

Hive SQL

Hbase NoSql columnar store

MapReduce

YARN Cluster resource manager

HDFS Hadoop Distributed File System

What is HBase?

- NoSQL database on Hadoop
  - Key – value store, schema-less
  - For storing big tables with many rows and columns
  - Consistent inserts, updates and deletes of rows
- Optimized for random reads
  - Data partitioning by row key values
  - Index on row key values
  - Bloom filter
  - Column store
  - Scalable

What HBase is not?

- Not a relational database
- Transactions are not ACID
- Index available only on a row key
- Weak for sequential data scanning

When to use?

- In general:
  - For data too big to store on some central storage
  - For random data access: quick lookups of individual records
  - The data can be represented by key-value sets

- Database of binary records (serialized objects, documents)

- When data set
  - has to be updated
  - is sparse – records have variable number of attributes
  - has custom data types (serialization)

When NOT to use?

- For massive data processing/analytics
  - use MR, Spark, Hive, Impala… instead

- For data sets with very high frequency insertion rates
  - stability concerns - from own experience

- Data schema is complex

- If “I do not know what solution to use”