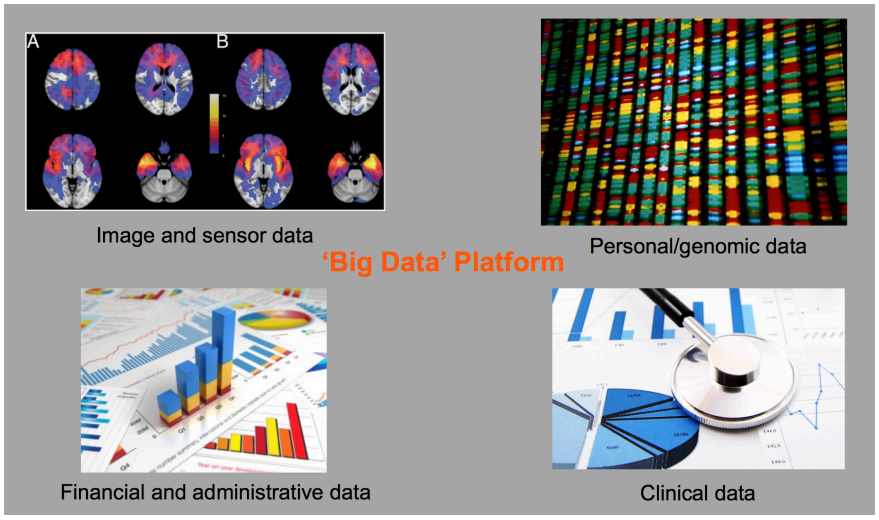
Distributed Data Processing

- Big data processing framework
- Hadoop / Map Reduce
- Spark

- material courtesy of Natl Inst of Computational Sciences/ ORNL / Baer, Begoli et. al

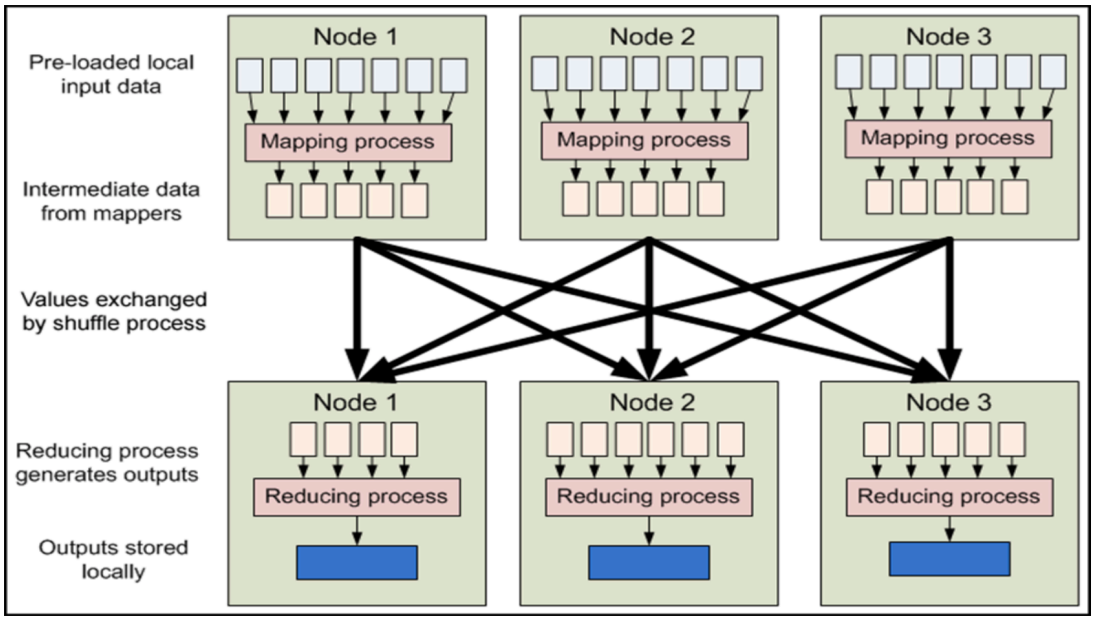
Big Data Applications

- Very large datasets, need to distribute processing of data sets
  - Parallelize data processing



MapReduce Programming Model

- Map Phase and Reduce Phase, connected by a shuffle



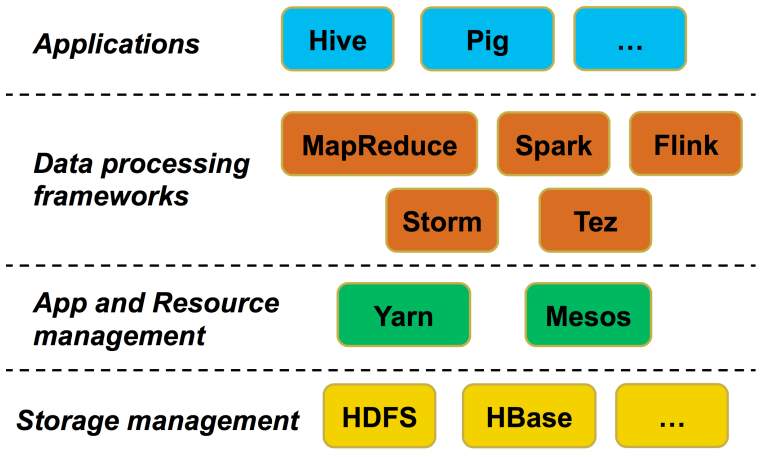
Other Programming Models

- Extend MapReduce to Directed Acyclic Graphs with recovery
  - Apache Tez,
- Microsoft's Dryad and Naiad
- DAG with in-memory resilient distributed data sets
  - Spark
- Extend DAG model to cyclic graphs: Flink
- Allow streaming data: Spark Streaming, Naiad, Kafka, Flink

Hadoop Big Data Platform

- Popular platform for processing large amounts of data
- EcoSystem:
- Storage managers : HDFS, HBASE, Kafka, etc.
- Processing framework: MapReduce, Spark, etc.
- Resource managers: Yarn, Mesos, etc.

Ecosystem

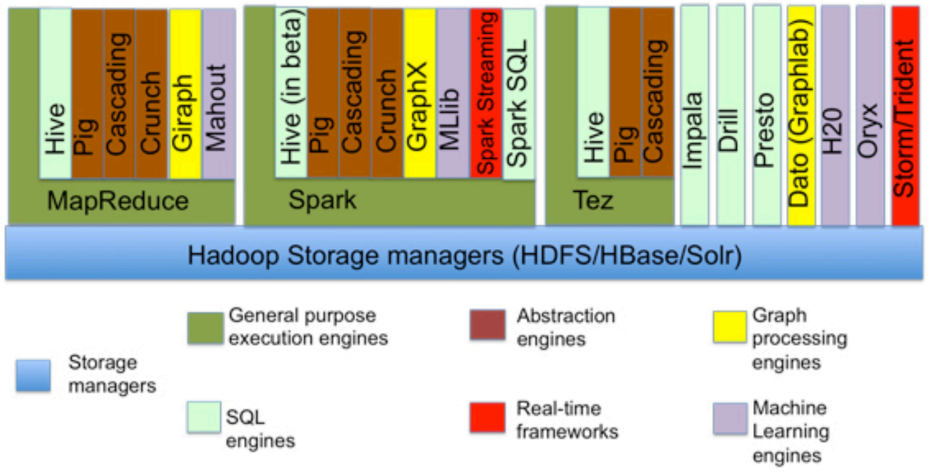


Ecosystem overview

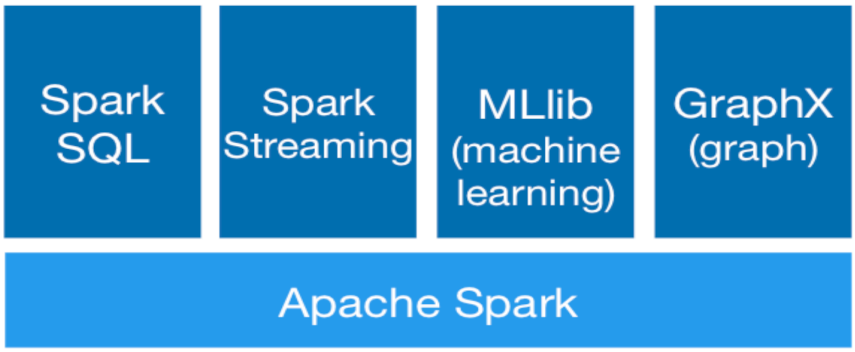
- General purpose framework: low level processing APIs
  - MapReduce, Spark, Flink
- Abstraction frameworks: higher level abstractions for processing
  - Pig
- SQL frameworks: allow data querying : Hive
- Graph processing frameworks: Giraph
- Machine learning frameworks: MLlib, Oyyx (standalone: TensorFlow)
- Real-time/stream processing: Spark Streaming, Storm, Kafka

- Cluster managers: YARN, Mesos (allocate machines to separate frameworks).

Ecosystem Overview



Spark Platform



- Ease of use: supports Java, Scala or Python
- General: combines SQL, streaming, ML, graph processing
- Faster due to in-memory RDDs
- Compatibility: runds on Hadoop, standalone, etc

Spark Architecture

- Resilient Distributed Datasets: **distributed memory**
  - objects cached in RAM across a cluster
- DAG execution engine : eliminates MapReduce multi-stage model
- RDD Narrow transform: Map, Filter, Sample
- RDD Wide transform: SortBy, ReduceBy, GroupBy, Join
- Action: Collect, Reduce