BITS Pilani
Pilani | Dubai | Goa | Hyderabad

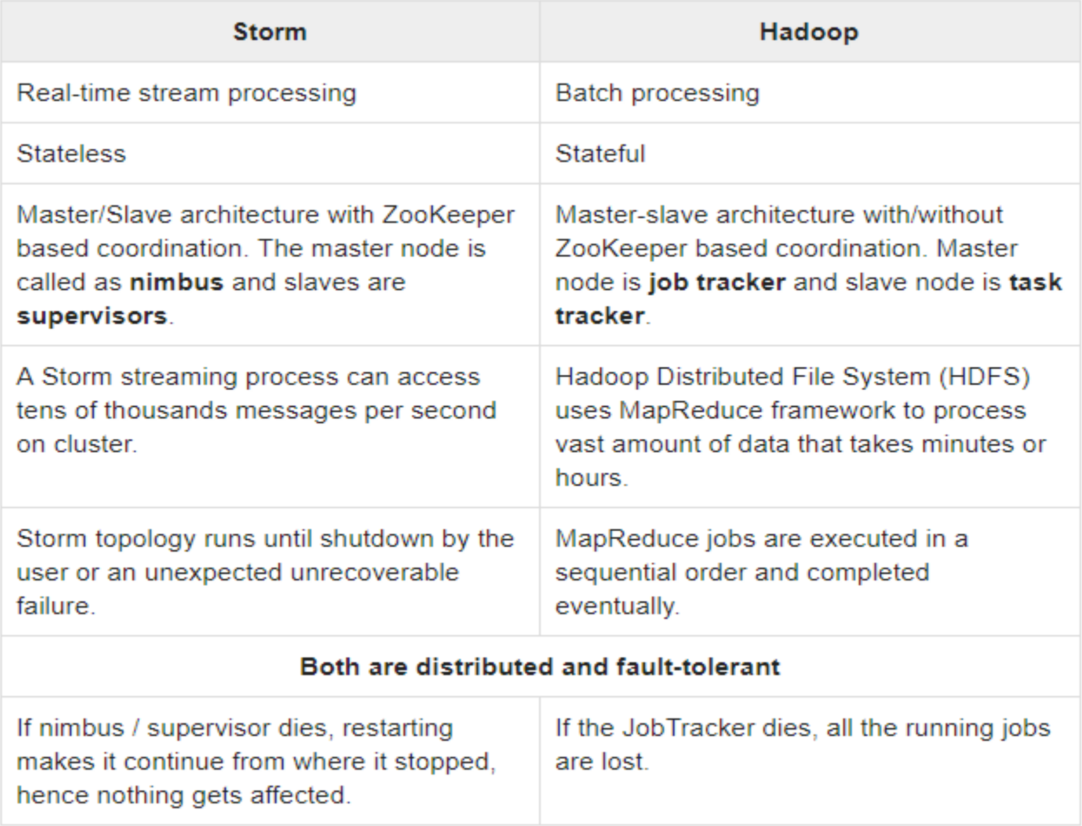
Apache Storm Architecture

Pravin Y Pawar

Apache Storm

- created by Nathan Marz and team
    - ✓ at BackType (then acquired by Twitter)
- standard for distributed real-time processing system
    - ✓ allows you to process large amount of data
- is written in Java and Clojure
- is stateless, it manages distributed environment and cluster state via Apache ZooKeeper
- Is leader in real-time analytics
- easy to setup, operate and it guarantees that every message will be processed through the topology at least once

Storm Vs Hadoop



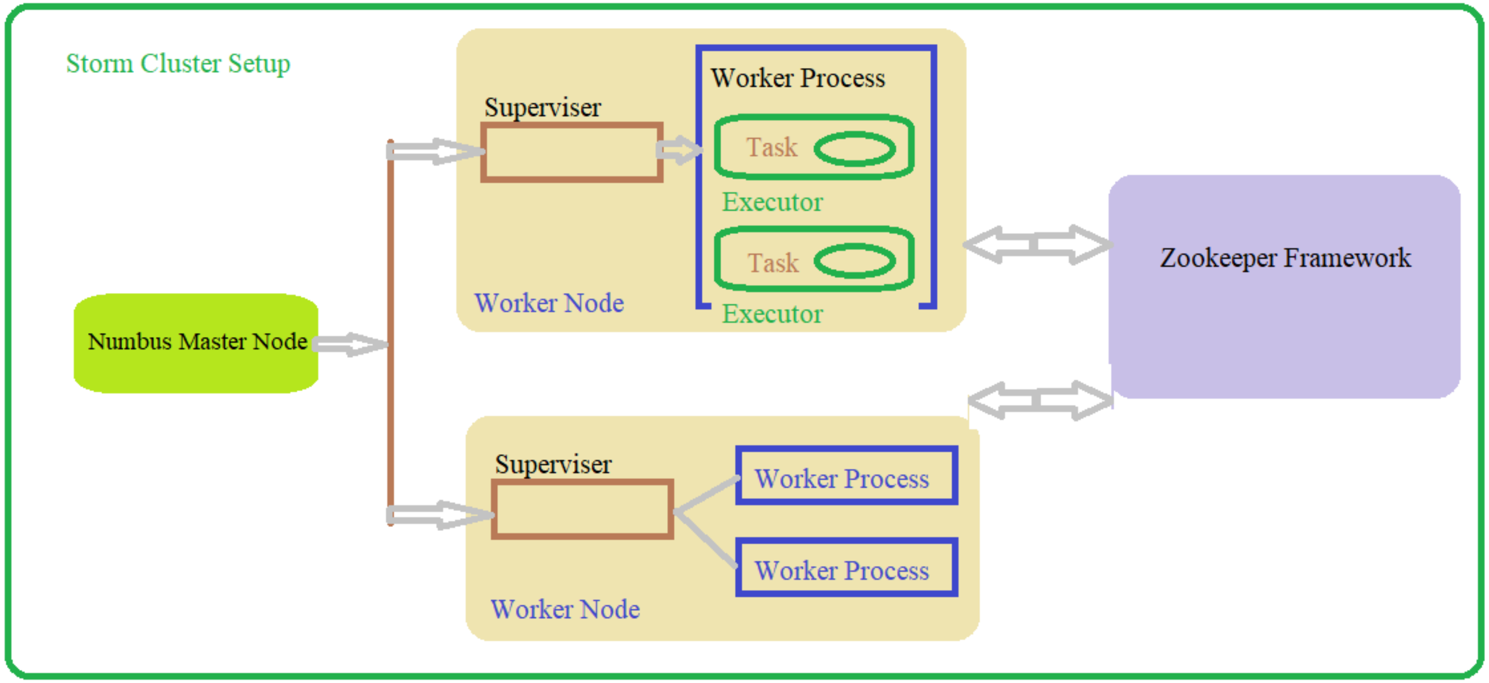
Storm	Hadoop
Real-time stream processing	Batch processing
Stateless	Stateful
Master/Slave architecture with ZooKeeper based coordination. The master node is called as **nimbus** and slaves are **supervisors**.	Master-slave architecture with/without ZooKeeper based coordination. Master node is **job tracker** and slave node is **task tracker**.
A Storm streaming process can access tens of thousands messages per second on cluster.	Hadoop Distributed File System (HDFS) uses MapReduce framework to process vast amount of data that takes minutes or hours.
Storm topology runs until shutdown by the user or an unexpected unrecoverable failure.	MapReduce jobs are executed in a sequential order and completed eventually.
Both are distributed and fault-tolerant	
If nimbus / supervisor dies, restarting makes it continue from where it stopped, hence nothing gets affected.	If the JobTracker dies, all the running jobs are lost.

Source : Tutorial Point

Storm Benefits

- open source, robust, and user friendly
- fault tolerant, flexible, reliable
- allows real-time stream processing
- keep up the performance even under increasing load by adding resources linearly, highly scalable.
- provides guaranteed data processing even if any of the connected nodes in the cluster die or messages are lost

Apache Storm Cluster



Apache Storm Cluster(2)

- Nimbus
  - ✓ Master node of Storm cluster
  - ✓ All other nodes in the cluster are called as worker nodes
  - ✓ is responsible for distributing data among all the worker nodes, assign tasks to worker nodes and monitoring failures.

- Supervisor
  - ✓ The nodes that follow instructions given by the nimbus are called as Supervisors.
  - ✓ A supervisor has multiple worker processes and it governs worker processes to complete the tasks assigned by the nimbus.

Apache Storm Cluster(3)

- Worker process
  - ✓ executes tasks related to a specific topology
  - ✓ A worker process will not run a task by itself, instead it creates executors and asks them to perform a particular task.
  - ✓ A worker process will have multiple executors.

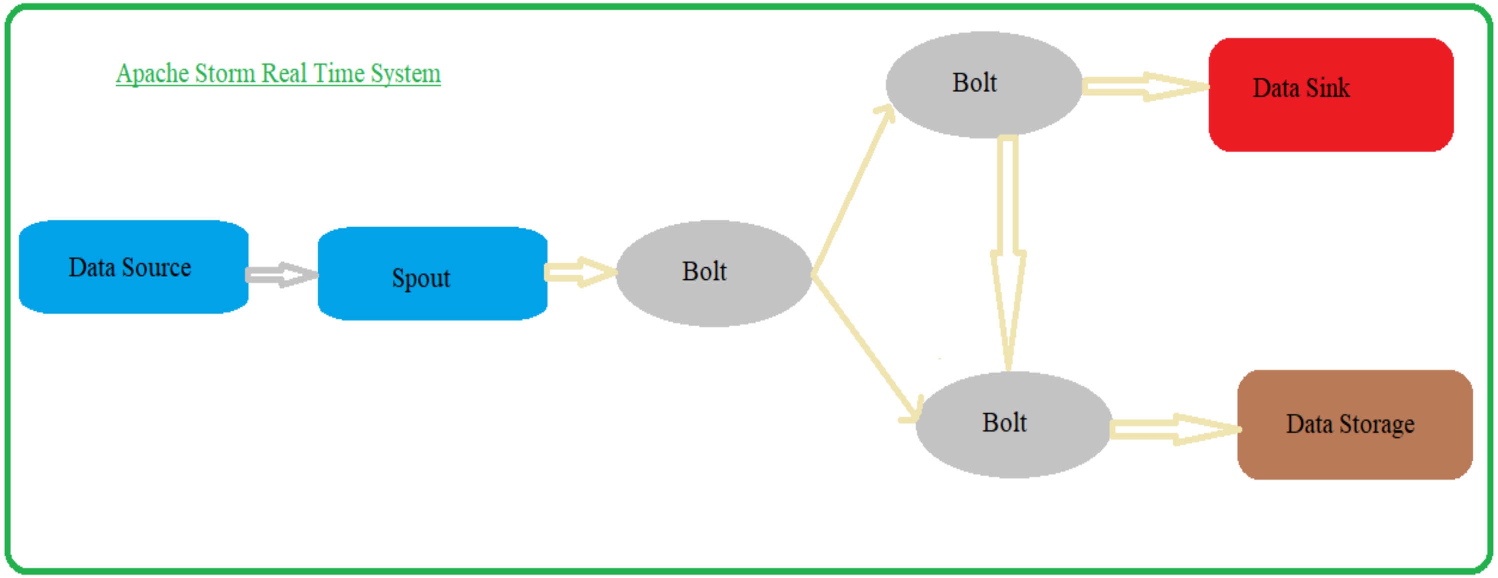
- Executor
  - ✓ nothing but a single thread spawn by a worker process
  - ✓ An executor runs one or more tasks but only for a specific spout or bolt.

- Task
  - ✓ A task performs actual data processing. So, it is either a spout or a bolt.

Storm Workflow

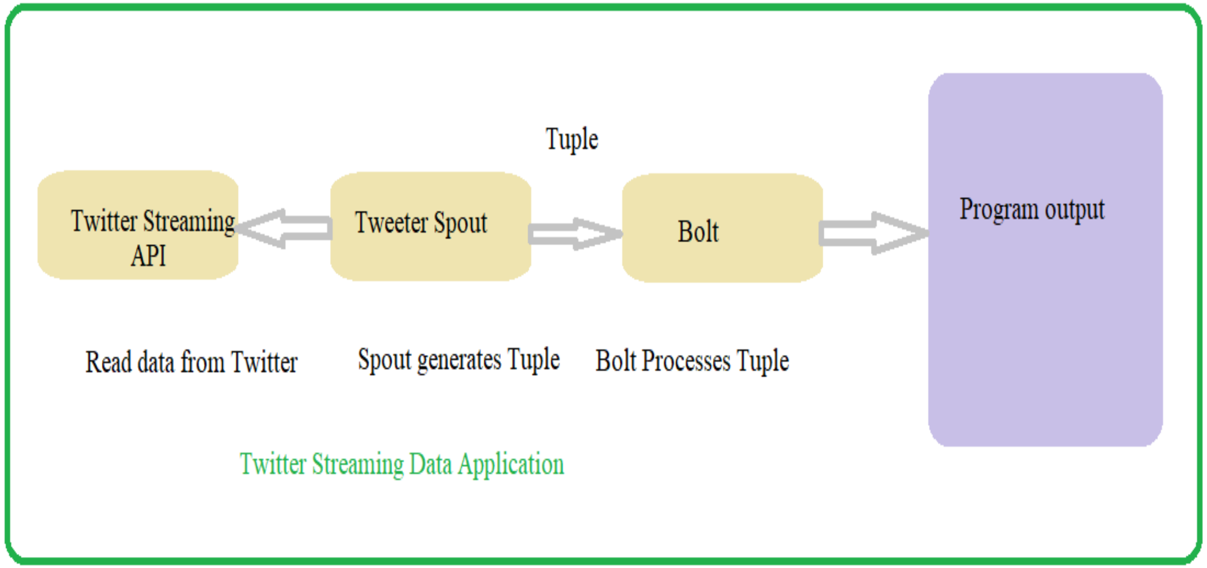
- Initially, the nimbus will wait for the “Storm Topology” to be submitted to it.
- Once a topology is submitted, it will process the topology and gather all the tasks that are to be carried out and the order in which the task is to be executed.
- The nimbus will evenly distribute the tasks to all the available supervisors.
- At a particular time interval, all supervisors will send heartbeats to the nimbus to inform that they are still alive.
- When a supervisor dies and doesn’t send a heartbeat to the nimbus, then the nimbus assigns the tasks to another supervisor.
- When the nimbus itself dies, supervisors will work on the already assigned task without any issue.
- Once all the tasks are completed, the supervisor will wait for a new task to come in.
- In the meantime, the dead nimbus will be restarted automatically by service monitoring tools.
- The restarted nimbus will continue from where it stopped.
- Once all the topologies are processed, the nimbus waits for a new topology to arrive and similarly the supervisor waits for new tasks.

Apache Storm System



Apache Storm System(2)

- Components
  - ✓ Stream
  - ✓ Tuple
  - ✓ Spout
  - ✓ Bolt



Apache Storm System Components

- Stream
  - ✓ sequence of tuples

- Tuple
  - ✓ main data structure in Storm
  - ✓ a list of ordered elements
  - ✓ modelled as a set of comma separated values and passed to a Storm cluster

Apache Storm System Components(2)

- Spouts
  - ✓ Source of stream
  - ✓ accepts input data from raw data sources like Twitter Streaming API, Apache Kafka etc.
  - ✓ can read data from other data sources
  - ✓ “ISpout" is the core interface for implementing spouts
  - ✓ Specific interfaces are IRichSpout, BaseRichSpout, KafkaSpout etc.

Apache Storm System Components(3)

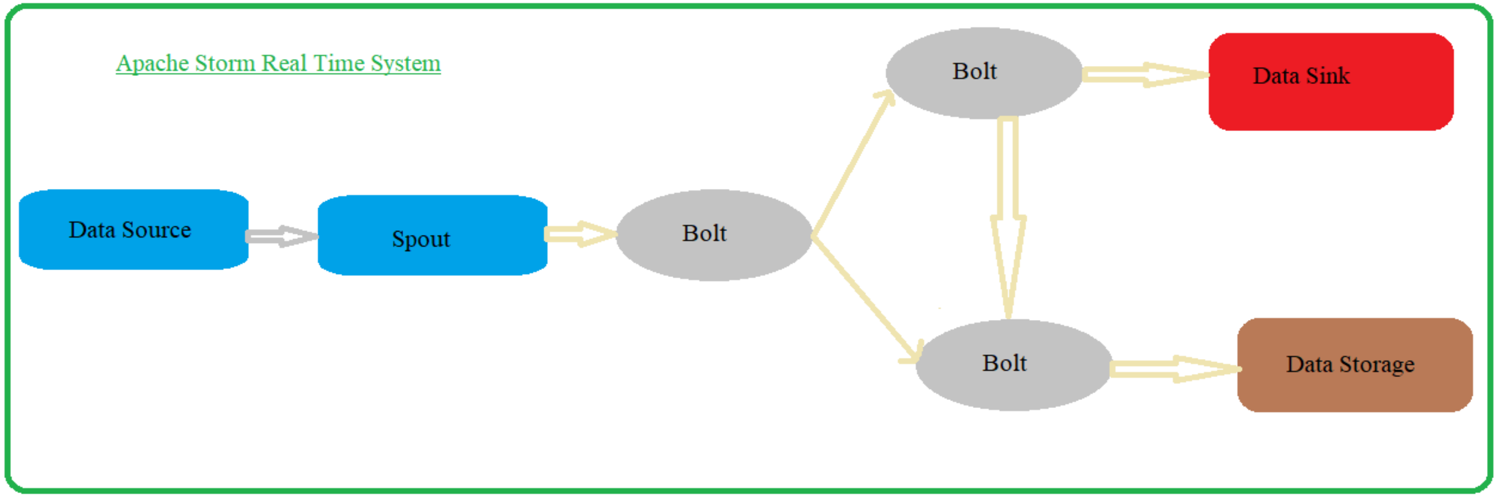
- Bolts
  - ✓ logical processing units
  - ✓ Spouts pass data to bolts and bolts process and produce a new output stream.
  - ✓ perform the operations of filtering, aggregation, joining, interacting with data sources and databases
  - ✓ receives data and emits to one or more bolts
  - ✓ “IBolt” is the core interface for implementing bolts
  - ✓ Common interfaces are IRichBolt, IBasicBolt, etc.

Apache Storm System Components(4)

- Topology
  - ✓ Spouts and bolts are connected together and they form a topology
  - ✓ Application logic is specified inside Storm topology
  - ✓ is a directed graph where vertices are computation and edges are stream of data
  - ✓ Storm keeps the topology always running, until you kill the topology.
  - ✓ Apache Storm's main job is to run the topology and will run any number of topology at a given time.

Apache Storm System Components(5)

- A simple topology starts with spouts
- Spout emits the data to one or more bolts
- Bolt represents a node in the topology having the smallest processing logic
- Output of a bolt can be emitted into another bolt as input



Thank You!