Apache Storm Overview

Upendra Sabnis

About Storm :

- A free and open source distributed, fault tolerant and highly scalable realtime computation system.
- Hadoop → Batch processing

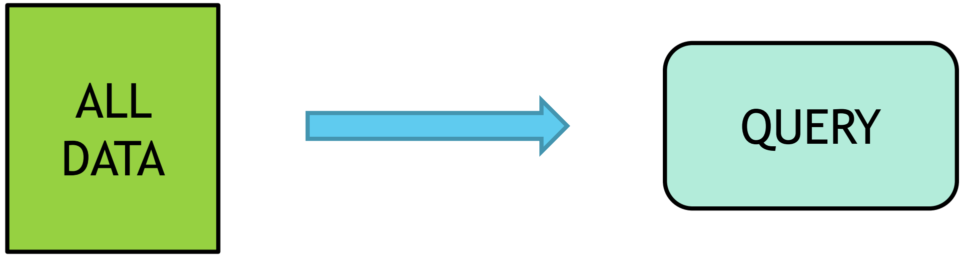
  Storm → Realtime processing
- History :

  Apache Storm was originally created by Nathan Marz and his team at BackType.

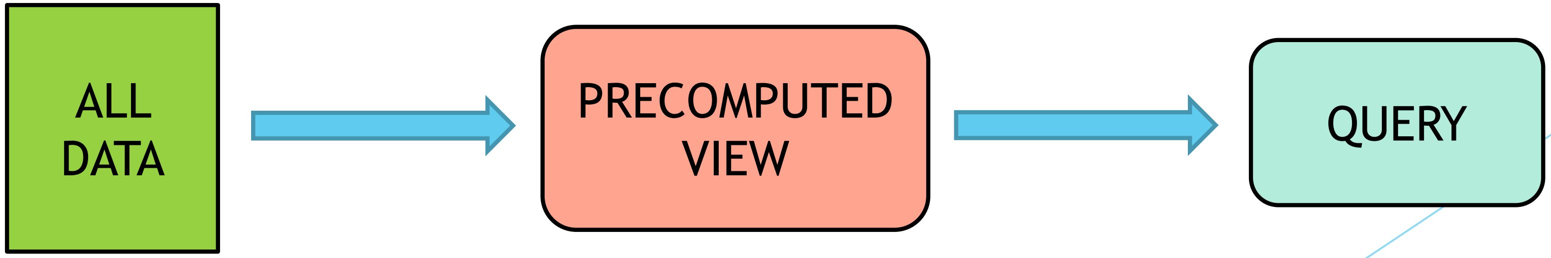
  Later it was made open source and was acquired by Twitter.
- Use-cases: realtime analytics, online machine learning, continuous computation and many more.

Need for computation system

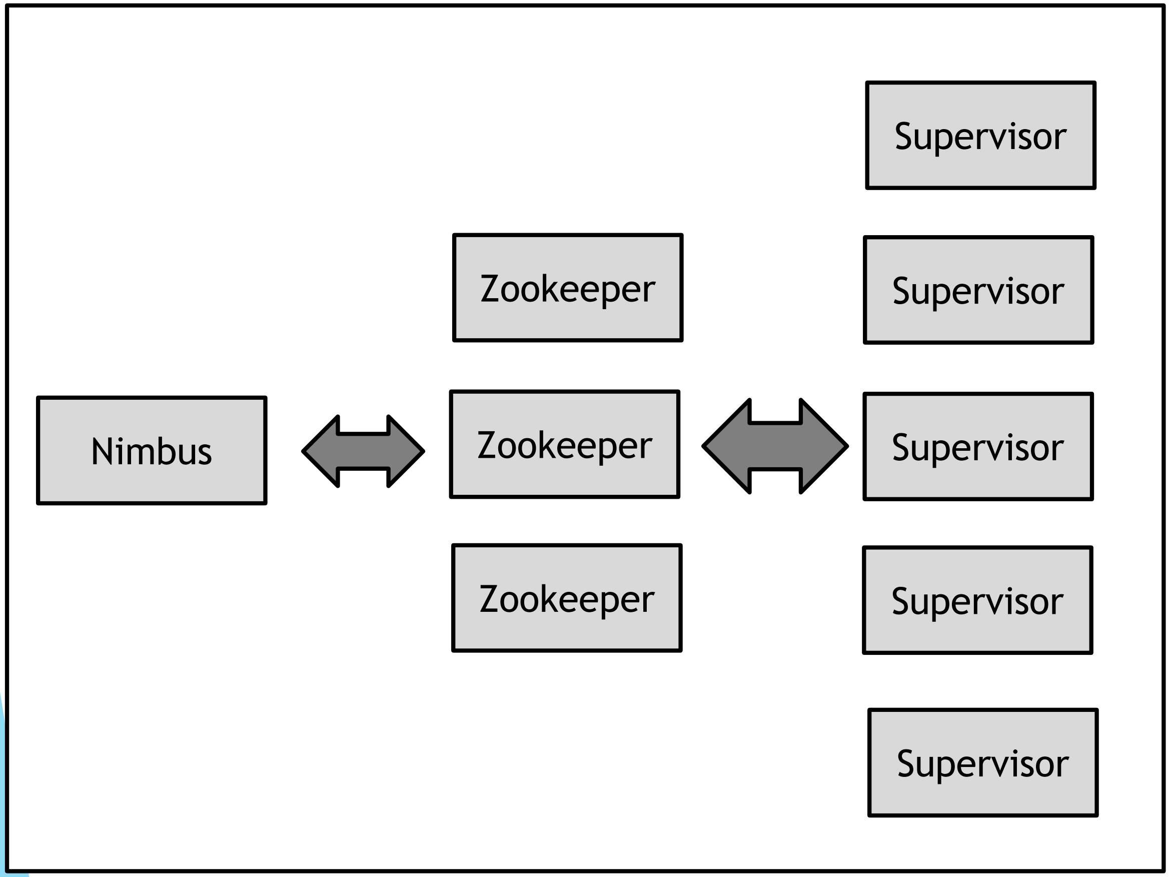
- Old model : Problem with scaling



- With computation system :



Components of a Storm Cluster



- Hadoop → MapReduce jobs

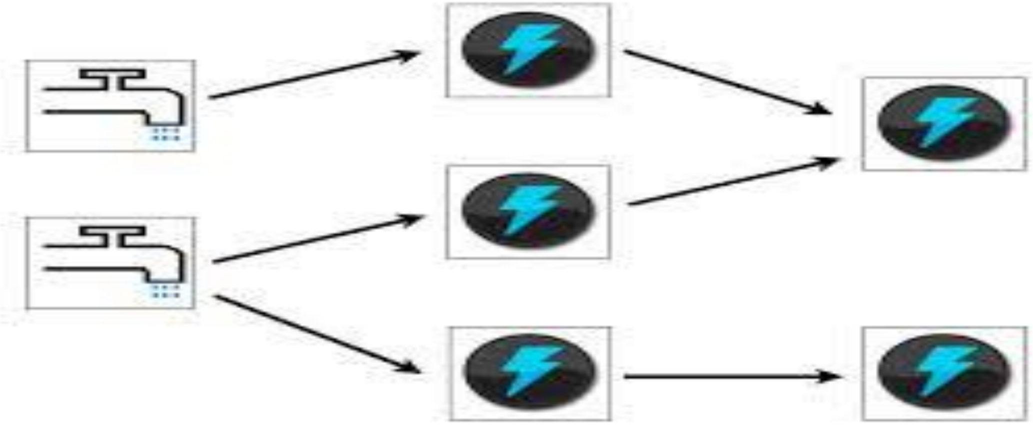
  Storm → Topologies
- Two kinds of nodes : Master node and Worker node

  Master node runs a daemon called “Nimbus”. It is similar to Hadoop’s Job Tracker

  Worker node runs a daemon called “Supervisor”
- Coordination between Nimbus and Supervisor is done through a zookeeper.

Topologies and Streams

- A topology is a graph of computation. Each node in a topology contains processing logic, and links between nodes indicate how data should be passed around between nodes.
- A stream is an unbounded sequence of tuples (named list of values and field in tuple can be object of any type). Storm provides the primitives for transforming a stream into a new stream in a distributed and reliable way.



Spouts and bolts

- Primitives for doing stream transformations are "spouts" and "bolts". Spouts and bolts have interfaces that we implement to run our application-specific logic.

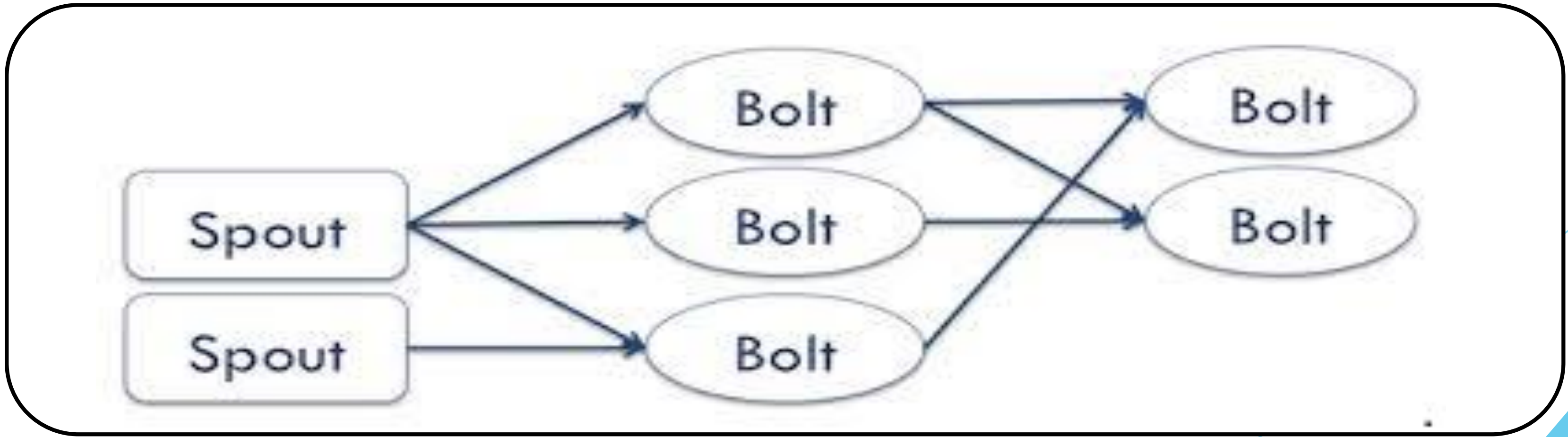
- A spout is a source of streams.

  For example, a spout may read tuples off of a queue or it may connect to the Twitter API and emit a stream of tweets.

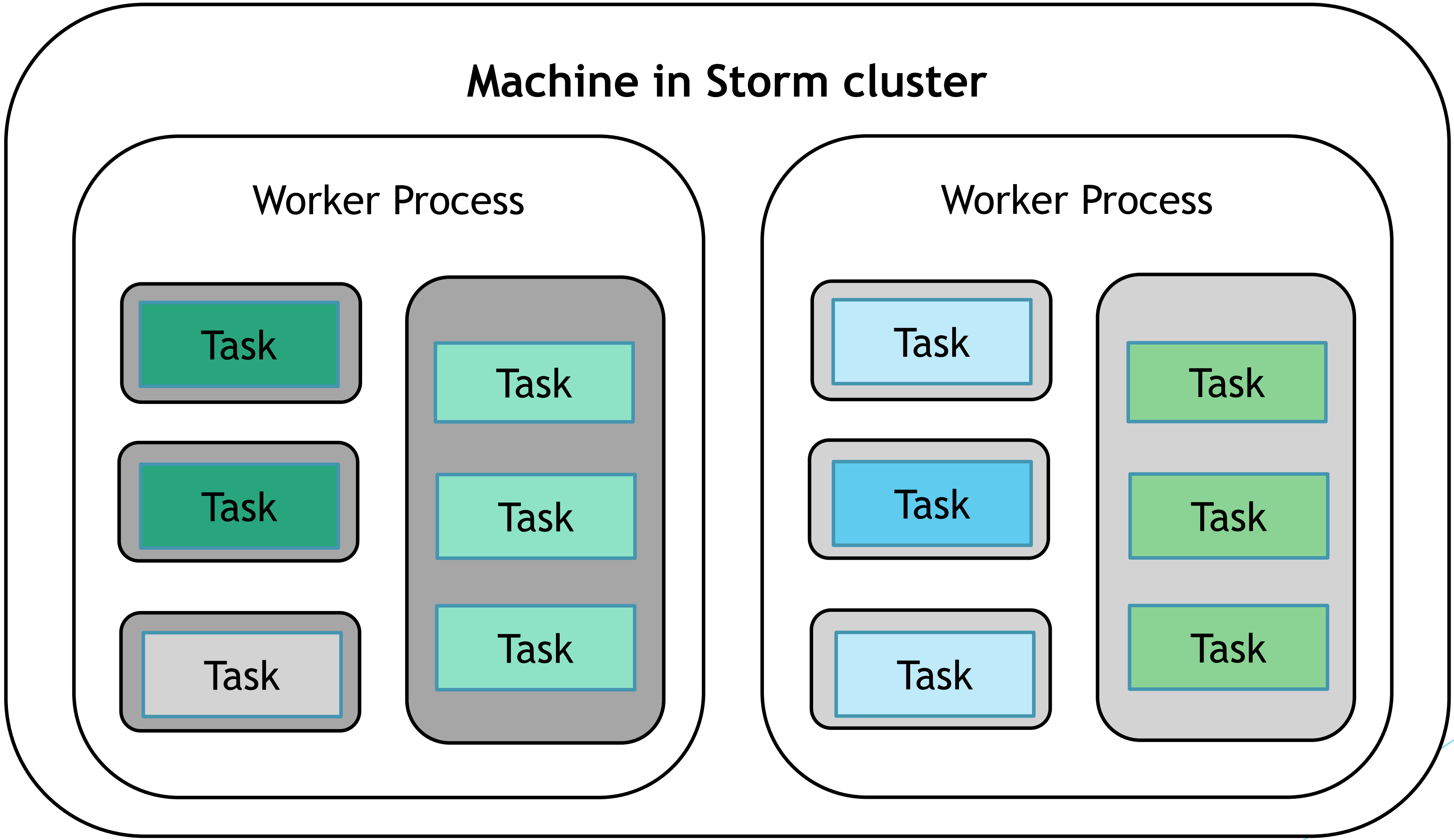
- A bolt consumes any number of input streams, does some processing, and possibly emits new streams. Complex stream transformations, like computing a stream of trending topics from a stream of tweets, require multiple steps and thus multiple bolts. Bolts can do anything from run functions, filter tuples, do streaming aggregations, do streaming joins, talk to databases, and more.

Spouts and bolts continued….

- Networks of spouts and bolts are packaged into a "topology". A topology is a graph of stream transformations where each node is a spout or bolt.
- Edges in the graph indicate which bolts are subscribing to which streams.
- When a spout or bolt emits a tuple to a stream, it sends the tuple to every bolt that subscribed to that stream.



Running a topology : Worker processes, executors and tasks



Code example

```
TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("words", new TestWordSpout(), 10);

builder.setBolt("exclaim1", new ExclamationBolt(), 3)
.shuffleGrouping("words");

builder.setBolt("exclaim2", new ExclamationBolt(), 2)
.shuffleGrouping("exclaim1");
```

Fault tolerance and Guaranteeing message processing

- Nimbus and Supervisors are stateless and fail-fast
- Failures at different level in cluster :

  If worker dies : Supervisor will restart it. If it continuously fails on startup and unable to heartbeat to nimbus, Nimbus will reassign the worker to other machine.

  If node dies : Tasks assigned to that machine will timeout and Nimbus will reassign those tasks to other machines.

  If Nimbus or Supervisor dies : They are fail-fast(halts if some unexpected situation is encountered) and stateless (state is kept on zookeeper or on disk). Better to use tools like deamontools or monit which can restart th daemons like nothing happened

- Is Nimbus a single point of failure ? – Yes
- Guarantee of message processing : The message is always fully processed. Actually API provided by storm takes care of this.

Demo

Resources :

- Apache Storm homepage :
- ❑ https://storm.apache.org/
- Apache Storm documentation :
- ❑ https://storm.apache.org/documentation/Home.html
- Hortonworks Storm :
- ❑ http://hortonworks.com/hadoop-tutorial/processing-streaming- data-near-real-time-apache-storm/
- Storm tutorial :
- ❑ http://hortonworks.com/hadoop-tutorial/simulating-transporting-realtime-events-stream-apache-kafka/
- ❑ http://hortonworks.com/hadoop-tutorial/ingesting-processing-real-time-events-apache-storm/

THANK YOU !!!

APACHE
STORM™