

Introduction to Apache Storm

Miss. Manu¹, Mr. Rakesh V S²

¹Student of 8th semester, Department of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, Karnataka, India

²Assistant professor, Department of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, Karnataka, India

Email: manu.13cs054@citech.edu.in,

Abstract-Big data is a term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Challenges include capture, storage, analysis, data curation, search, sharing, transfer, visualization, querying, updating and information privacy. The term "big data" often refers simply to the use of predictive analytics, user behaviour analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. Apache Storm is a distributed real-time big data-processing system designed to process vast amount of data in a fault-tolerant and horizontal scalable method.

Index Terms – Introduction; Literature Survey; Storm Architecture; Development; Peer Platforms; Conclusion; Reference.

1. INTRODUCTION

In the past few years the digital data has exploded the internet. Some of the major contributors being facebook, twitter, YouTube etc... The data is growing at a very rapid pace and the data volumes are now in terms of millions of Giga Bytes. The technology has evolved to handle this amount of data. The technology to store and process this data, to analyse and make decisions based on such large amount of data have all evolved. The data as well as the technology to process this data is known as Big Data.

The Big Data is generally categorized by the three v's; Volume, Velocity, and Variety. Volume refers to the size of data which generally begins from Kilo Bytes (2¹⁰ Bytes) to Yotta Bytes (2⁸⁰ Bytes). Data Velocity refers to the rate of data ingestion. A large number of pages are added every day from the sources such as laptop, PCs, mobile, tablet etc., Data Variety refers to the types of data. The data could be in the form of text, audio, video, multimedia, HTML files etc.,

Some industrial examples of Big Data are; Retail affinity detection (market basket analysis), Credit card fraud detection, Bank loan risk minimisation, etc.,

1.1 What is Apache Storm?

Apache Storm is a distributed real-time big data-processing system. Storm is designed to process vast amount of data in a fault-tolerant and horizontal scalable method. It is a streaming data framework that has the capability of highest ingestion rates. Though Storm is stateless, it manages distributed environment and cluster state via Apache ZooKeeper. It is simple and you can execute all kinds of manipulations on real-time data in parallel.

Apache Storm is continuing to be a leader in real-time data analytics. Storm is easy to setup, operate and it guarantees that every message will be processed through the topology at least once.

2. LITERATURE SURVEY

Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.

Apache Storm is a **free and open sourced** distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing.

Some benefits of storm are:

- Storm is open source, robust, and user friendly.
- It could be utilized in small companies as well as large corporations.
- It is fault tolerant, flexible, reliable, and supports any programming language.
- It allows real-time stream processing and is unbelievably fast since it has enormous processing power.
- It can keep up the performance even under increasing load by adding resources linearly.
- It is highly scalable.
- Storm performs data refresh and end-to-end delivery response in seconds or minutes depends upon the problem.
- It has very low latency.
- It has operational intelligence.
- It provides guaranteed data processing even if any of the connected nodes in the cluster die or messages are lost.

2.1 Difference between HADOOP and STORM

Hadoop and Storm frameworks are used for analyzing big data. Both of them complement each other and differ in some aspects. Apache Storm does all the operations except persistency, while Hadoop is good at everything but lags in real-time computation. The following table compares the attributes of Storm and Hadoop.

Following are few of the differences listed;

- STORM is real-time stream processing system, whereas HADOOP is a batch processing system.
- STORM is stateless, but HADOOP is stateful.
- The STORM has master/slave architecture with a ZooKeeper based coordination. The master node is Nimbus and the slave node being the Supervisors. HADOOP on the other hand is also a master/slave architecture but it maybe with/without ZooKeeper based coordination. The 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 and unrecoverable failure. In HADOOP, MapReduce jobs are executed in a sequential order and completed eventually.
- In STORM, if Nimbus/Supervisor dies, restarting makes it continue from where it stopped, hence nothing gets affected. Whereas, in HADOOP, if the JobTracker dies all the jobs are lost.
- One similarity in the two is that they are both distributed and fault-tolerant

2.2 Use Cases of Apache Storm

Apache Storm is very famous for real-time big data stream processing. For this reason, most of the companies are using Storm as an integral part of their system. Some notable examples are as follows:

- **Twitter** – Twitter is using Apache Storm for its range of “Publisher Analytics products”. “Publisher Analytics Products” process each and every tweets and clicks in the Twitter Platform. Apache Storm is deeply integrated with Twitter infrastructure.
- **NaviSite** – NaviSite is using Storm for Event log monitoring/auditing system. Every logs generated in the system will go through the Storm. Storm will check the message against the configured set of regular expression and if there is a match, then that particular message will be saved to the database.
- **Wego** – Wego is a travel metasearch engine located in Singapore. Travel related data comes from many sources all over the world with different timing. Storm helps Wego to search realtime data, resolves concurrency issues and find the best match for the end-user.

2.3 Core Concepts of APACHE STORM

Apache Storm reads raw stream of real-time data from one end and passes it through a sequence of small processing units and output the processed/useful information at the other end.

The following diagram depicts the core concept of Apache Storm.

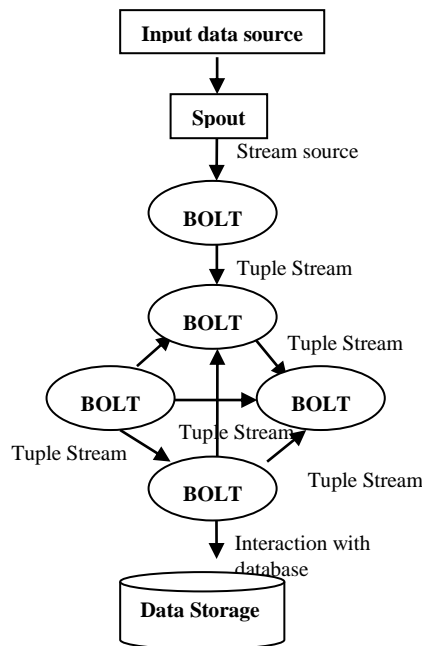


Fig 2.1: Conceptual Components of STORM

The above figure shows the working of components in STORM and the interaction among them. Now, let's understand these components individually. The following table (Table 2.2) provides a brief description of each component.

Following is the description of each conceptual component of STORM:

- **Tuple:** Tuple is the main data structure in Storm. It is a list of ordered elements. By default, a Tuple supports all data types. Generally, it is modelled as a set of comma separated values and passed to a Storm cluster.
- **Stream:** It is an unordered sequence of tuples.
- **Spouts:** Source of stream. Generally, Storm accepts input data from raw data sources like Twitter Streaming API, Apache Kafka queue, Kestrel queue, etc. Otherwise you can write spouts to read data from data sources. "ISpout" is the core interface for implementing spouts. Some of the specific

interfaces are IRichSpout, BaseRichSpout, KafkaSpout, etc.

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

2.4 STREAM GROUPING

Stream of data flows from spouts to bolts or from one bolt to another bolt. Stream grouping controls how the tuples are routed in the topology and helps us to understand the tuples flow in the topology. There are four in-built groupings as explained below.

- **Shuffle grouping:** In shuffle grouping, an equal number of tuples is distributed randomly across all of the workers executing the bolts.
- **Field grouping:** The fields with same values in tuples are grouped together and the remaining tuples kept outside. Then, the tuples with the same field values are sent forward to the same worker executing the bolts. For example, if the stream is grouped by the field "word", then the tuples with the same string, "Hello" will move to the same worker.
- **Global grouping:** All the streams can be grouped and forward to one bolt. This grouping sends tuples generated by all instances of the source to a single target instance (specifically, pick the worker with lowest ID).
- **All grouping:** All Grouping sends a single copy of each tuple to all instances of the receiving bolt. This kind of grouping is used to send signals to bolts. All grouping is useful for join operations.

3. STORM Architecture

3.1 STORM – Cluster Architecture

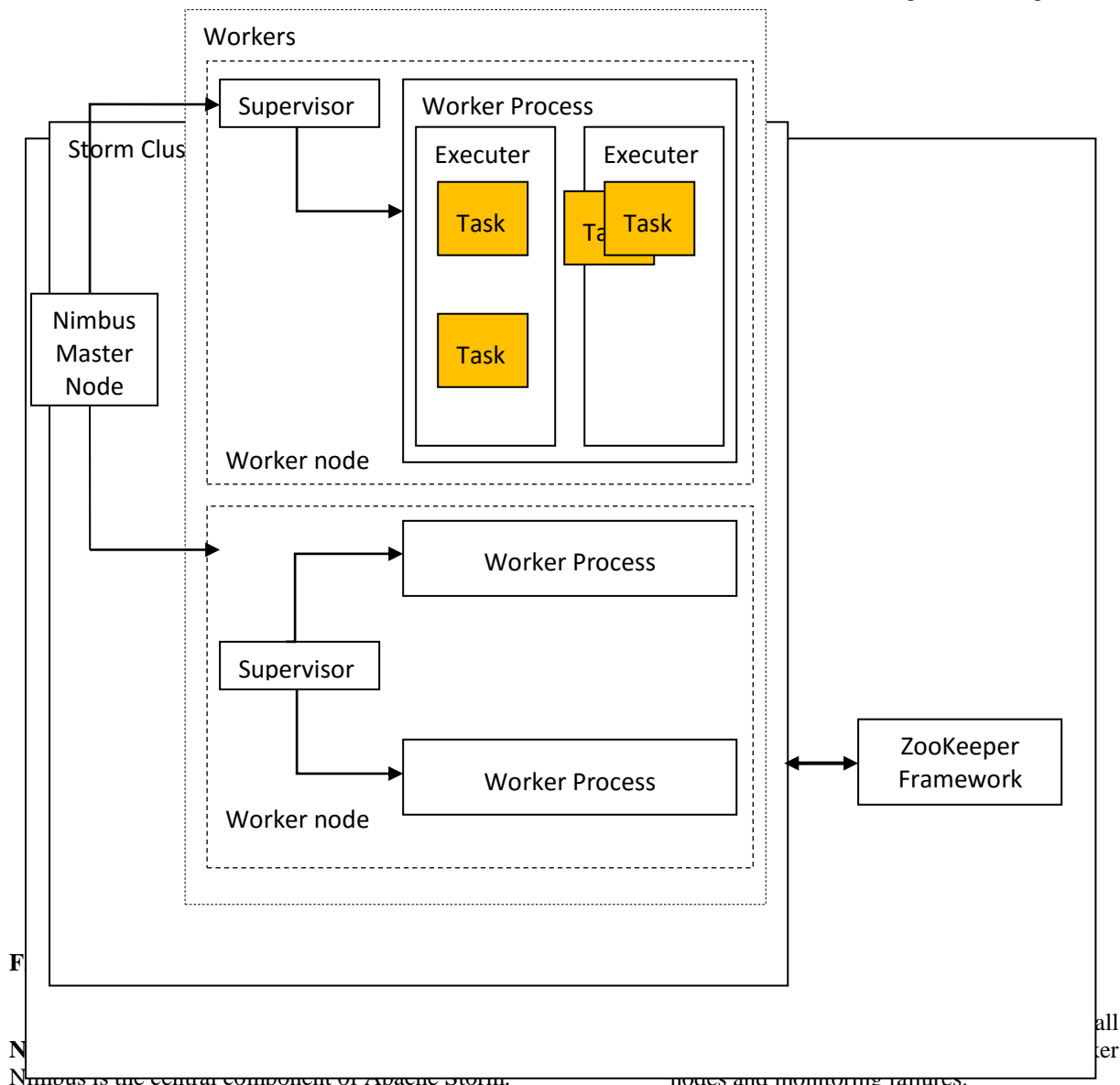
One of the main highlight of the Apache Storm is that it is a fault-tolerant, fast with no "Single Point of Failure" (SPOF) distributed application. We can install Apache Storm in as many systems as needed to increase the capacity of the application.

The following figure (Fig 3.1) shows the architectural diagram of STORM.

A supervisor will have one or more worker process. Supervisor will delegate the tasks to worker processes. Worker process will spawn as many executors as needed and run the task. Apache Storm uses an internal distributed messaging system for the communication between nimbus and supervisors.

Following is the description of each component of STORM:

Nimbus: Nimbus is a master node of Storm cluster. All other nodes in the cluster are called as worker nodes. Master node is responsible for distributing data among all t



Nimbus is the central component of Apache Storm. The main job of Nimbus is to run the Storm topology. Nimbus analyzes the topology and gathers the task to be executed. Then, it will distribute the task to an available supervisor.

- 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.

- **Worker Process:** A worker process will execute 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:** An executor is 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.
- **ZooKeeper Framework:** ZooKeeper is a service used by a cluster (group of nodes) to coordinate between themselves and maintaining shared data with robust synchronization techniques. Nimbus is stateless, so it depends on ZooKeeper to monitor the working node status. ZooKeeper helps the supervisor to interact with the nimbus. It is responsible to maintain the state of nimbus and supervisor.

Storm is stateless in nature. Even though stateless nature has its own disadvantages, it actually helps Storm to process real-time data in the best possible and quickest way.

Storm is not entirely stateless though. It stores its state in Apache ZooKeeper. Since the state is available in Apache ZooKeeper, a failed nimbus can be restarted and made to work from where it left. Usually, service monitoring tools like monit will monitor Nimbus and restart it if there is any failure.

3.3 Work flow of APACHE STORM

A working Storm cluster should have one nimbus and one or more supervisors. Another important node is Apache ZooKeeper, which will be used for the coordination between the nimbus and the supervisors. Let us now take a close look at the workflow of Apache Storm:

- Initially, the nimbus will wait for the “Storm Topology” to be submitted to it. The
- 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.
- Then, 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. Similarly, the dead supervisor can also be restarted automatically. Since both the nimbus and the supervisor can be restarted automatically and both will continue as before, Storm is guaranteed to process all the tasks at least once.
- Once all the topologies are processed, the nimbus waits for a new topology to arrive and similarly the supervisor waits for new tasks.

By default, there are two modes in a Storm cluster:

- **Local mode:** This mode is used for development, testing, and debugging because it is the easiest way to see all the topology components working together. In this mode, we can adjust parameters that enable us to see how our topology runs in different Storm configuration environments. In Local mode, storm topologies run on the local machine in a single JVM.
- **Production mode:** In this mode, we submit our topology to the working storm cluster, which is composed of many processes, usually running on different machines. As discussed in the workflow of storm, a working cluster will run indefinitely until it is shutdown.

4. DEVELOPMENT

Apache Storm is developed under the Apache License, making it available to most companies to use.^[1] Git is used for version control and Atlassian JIRA for issue tracking, under the Apache Incubator program.

The latest version of STORM being 1.1.0 was released on 29th Mar 2017.

5. Peer Platforms

Storm is but one of dozens of streaming engines. Twitter announced Heron (event processor) on June 2nd, 2015^[2] which is API compatible with Storm. There are other comparable streaming data engines such as Spark Streaming and Flink.^[3]

6. CONCLUSION

Apache STORM is an open source distributed data processing tool written in JAVA. It allows real time processing instead of batch processing and it works with a Master/Slave architecture with Zookeeper based coordination. Storm uses a Spout and multiple BOLTs to carry out its working. Spout generates a stream of input data which are forwarded to BOLT for processing. The input stream may go through several BOLTs before the final output is generated and stored in data storage. Tuple being the main data structure in Storm is a list of ordered elements that by default supports all data types. The unordered list of tuples is called a stream.

Stream grouping is one of the important aspects of storm. Stream of data flows from spouts to bolts or from one bolt to another bolt. Stream grouping controls the routing of tuples in the topology and helps understand the tuples flow in the topology. There are four in-build groupings: Shuffle grouping, Field grouping, Global grouping, and All grouping.

Among the various components of Storm cluster, Nimbus is a master node responsible for distributing data among all the worker nodes. A worker node has multiple supervisors that governs worker processes to complete the tasks assigned by Nimbus. The tasks are executed by a single thread spawn by a worker process called as Executer. The executer runs one or more tasks but only for a specific user. ZooKeeper is a service used by clusters to coordinate within themselves and maintaining shared data with robust synchronization techniques.

References

- [1] "Powered By Storm", Documentation. Apache Storm. Retrieved 29th July 2015
- [2] "Flying faster with Twitter Heron", Engineering Blog. Twitter Inc. Retrieved 3rd June 2015
- [3] "Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming", IEEE. May 2016
- [4] "Getting Started with Storm: Continuous Streaming Computation with Twitter's Cluster Technology", Jonathan Leibiusky, Gabriel Eisbruch, Dario Simonassi
- [5] "Storm Real-Time Processing Cookbook", Quinton Anderson