## Top Answers to Storm Interview Questions

### **1. Compare Spark & Storm**

Let’s compare [spark](https://intellipaat.com/blog/what-is-apache-spark/) and [storm](https://intellipaat.com/blog/what-is-apache-storm/) based on different parameters.

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Spark** | **Storm** |
| Data operation | Data at rest | Data in motion |
| Parallel computation | Task parallel | Data parallel |
| Latency | Few seconds | Sub-second |
| Deploying the application | Using Scala, Java, Python language | Using Java API |

### **2. Which components are used for stream flow of data?**

For streaming of data flow, three components are used

* **Bolt:-** Bolts represent the processing logic unit in Storm. One can utilize bolts to do any kind of processing such as filtering, aggregating, joining, interacting with data stores, talking to external systems, etc. Bolts can also emit tuples (data messages) for the subsequent bolts to process. Additionally, bolts are responsible to acknowledge the processing of tuples after they are done processing.
* **Spout:-** Spouts represent the source of data in Storm. You can write spouts to read data from data sources such as[**databases**](https://intellipaat.com/blog/what-is-database/), distributed file systems, messaging frameworks, etc. Spouts can broadly be classified into the following –  
  **Reliable –** These spouts have the capability to replay the tuples (a unit of data in the data stream). This helps applications achieve ‘at least once message processing’ semantic as in case of failures, tuples can be replayed and processed again. Spouts for fetching the data from messaging frameworks are generally reliable as these frameworks provide the mechanism to replay the messages.  
  **Unreliable –** These spouts don’t have the capability to replay the tuples. Once a tuple is emitted, it cannot be replayed irrespective of whether it was processed successfully or not. This type of spouts follows at most once message processing’ semantic.
* **Tuple:-** The tuple is the main data structure in Storm. A tuple is a named list of values, where each value can be any type. Tuples are dynamically typed — the types of the fields do not need to be declared. Tuples have helper methods like getInteger and getString to get field values without having to cast the result. Storm needs to know how to serialize all the values in a tuple. By default, Storm knows how to serialize the primitive types, strings, and byte arrays. If you want to use another type, you’ll need to implement and register a serializer for that type.

### **3. What are the key benefits of using Storm for Real Time Processing?**

* **Easy to operate:** Operating a storm is quite easy.
* **Real fast:** It can process 100 messages per second per node.
* **Fault-Tolerant:** It detects the fault automatically and re-starts the functional attributes.
* **Reliable:** It guarantees that each unit of data will be executed at least once or exactly once.
* **Scalable:** It runs across a cluster of machine

### Check out this video on Apache Storm Tutorial

**×**

### **4. Does Apache act as a Proxy server?**

Yes, It acts as a proxy also by using the mod\_proxy module. This module implements a proxy, gateway, or cache for Apache. It implements proxying capability for AJP13 (Apache JServ Protocol version 1.3), FTP, CONNECT (for SSL),HTTP/0.9, HTTP/1.0, and (since Apache 1.3.23) HTTP/1.1. The module can be configured to connect to other proxy modules for these and other protocols.

### **5. What is the use of Zookeeper in Storm?**

Storm uses [**Zookeeper**](https://intellipaat.com/blog/what-is-apache-zookeeper/) for coordinating the cluster. Zookeeper is not used for message passing, so the load that Storm places on Zookeeper is quite low. Single node Zookeeper clusters should be sufficient for most cases, but if you want failover or are deploying large Storm clusters you may want larger Zookeeper clusters. Instructions for deploying Zookeeper are here. A few notes about Zookeeper deployment :

1. It’s critical that you run Zookeeper under supervision since Zookeeper is fail-fast and will exit the process if it encounters any error case. See here for more details.
2. It’s critical that you set up a cron to compact Zookeeper’s data and transaction logs. The Zookeeper daemon does not do this on its own, and if you don’t set up a cron, Zookeeper will quickly run out of disk space.

### **6. What is ZeroMQ?**

ZeroMQ is “a library which extends the standard socket interfaces with features traditionally provided by specialized messaging middleware products”. Storm relies on ZeroMQ primarily for task-to-task communication in running Storm topologies.

### **7. How many distinct layers are of Storm’s Codebase?**

There are three distinct layers to Storm’s codebase.

**First:** Storm was designed from the very beginning to be compatible with multiple languages. Nimbus is a Thrift service and topologies are defined as Thrift structures. The usage of Thrift allows Storm to be used from any language.

**Second:** all of Storm’s interfaces are specified as Java interfaces. So even though there’s a lot of Clojure in Storm’s implementation, all usage must go through the **Java API**. This means that every feature of Storm is always available via Java.

**Third:** Storm’s implementation is largely in Clojure. Line-wise, Storm is about half Java code, half Clojure code. But Clojure is much more expressive, so in reality, the great majority of the implementation logic is in Clojure.

### **8. What does it mean for a message to be**

A tuple coming off a spout can trigger thousands of tuples to be created based on it. Consider, for example,

the streaming word count topology:TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("sentences", new KestrelSpout("kestrel.backtype.com",

22133,

"sentence\_queue",

new StringScheme()));

builder.setBolt("split", new SplitSentence(), 10)

.shuffleGrouping("sentences");

builder.setBolt("count", new WordCount(), 20)

.fieldsGrouping("split", new Fields("word"));

This topology reads sentences off a Kestrel queue, splits the sentences into their constituent words, and then emits for each word the number of times it has seen that word before. A tuple coming off the spout triggers many tuples being created based on it: a tuple for each word in the sentence and a tuple for the updated count for each word.  
Storm considers a tuple coming off a spout “fully processed” when the tuple tree has been exhausted and every message in the tree has been processed. A tuple is considered failed when its tree of messages fails to be fully processed within a specified timeout. This timeout can be configured on a topology-specific basis using the Config.TOPOLOGY\_MESSAGE\_TIMEOUT\_SECS configuration and defaults to 30 seconds.

***Learn more about***[***Apache Storm Tutorial***](https://intellipaat.com/blog/storm-tutorial/)***in this insightful blog now!***

### **9. When do you call the cleanup method?**

The cleanup method is called when a Bolt is being shut down and should clean up any resources that were opened. There’s no guarantee that this method will be called on the cluster: For instance, if the machine the task is running on blows up, there’s no way to invoke the method.

The cleanup method is intended when you run topologies in local mode (where a Storm cluster is simulated in the process), and you want to be able to run and kill many topologies without suffering any resource leaks.

### **10. How can we kill a topology?**

To kill a topology, simply run:  
**storm kill {stormname}**  
Give the same name to storm kill as you used when submitting the topology.  
The storm won’t kill the topology immediately. Instead, it deactivates all the spouts so that they don’t emit any more tuples, and then Storm waits for Config.TOPOLOGY\_MESSAGE\_TIMEOUT\_SECS seconds before destroying all the workers. This gives the topology enough time to complete any tuples it was processing when it got killed.

### **11. What is combinerAggregator?**

A CombinerAggregator is used to combine a set of tuples into a single field. It has the following signature:

public interface CombinerAggregator {

T init (TridentTuple tuple);

T combine(T val1, T val2);

T zero();

}

Storm calls the init() method with each tuple, and then repeatedly calls the combine()method until the partition is processed. The values passed into the combine() method are partial aggregations, the result of combining the values returned by calls to init().

### **12. What are the common configurations in Apache Storm?**

There are a variety of configurations you can set per topology. A list of all the configurations you can set can be found here. The ones prefixed with “TOPOLOGY” can be overridden on a topology-specific basis (the other ones are cluster configurations and cannot be overridden). Here are some common ones that are set for a topology:

1. **Config.TOPOLOGY\_WORKERS:** This sets the number of worker processes to use to execute the topology. For example, if you set this to 25, there will be 25 Java processes across the cluster executing all the tasks. If you had a combined 150 parallelism across all components in the topology, each worker process will have 6 tasks running within it as threads.
2. **Config.TOPOLOGY\_ACKER\_EXECUTORS:** This sets the number of executors that will track tuple trees and detect when a spout tuple has been fully processed By not setting this variable or setting it as null, Storm will set the number of acker executors to be equal to the number of workers configured for this topology. If this variable is set to 0, then Storm will immediately ack tuples as soon as they come off the spout, effectively disabling reliability.
3. **Config.TOPOLOGY\_MAX\_SPOUT\_PENDING:** This sets the maximum number of spout tuples that can be pending on a single spout task at once (pending means the tuple has not been acked or failed yet). It is highly recommended you set this config to prevent queue explosion.
4. **Config.TOPOLOGY\_MESSAGE\_TIMEOUT\_SECS :** This is the maximum amount of time a spout tuple has to be fully completed before it is considered failed. This value defaults to 30 seconds, which is sufficient for most topologies.
5. **Config.TOPOLOGY\_SERIALIZATIONS:** You can register more serializers to Storm using this config so that you can use custom types within tuples.

### **13. Is it necessary to kill the topology while updating the running topology?**

Yes, to update a running topology, the only option currently is to kill the current topology and resubmit a new one. A planned feature is to implement a Storm swap command that swaps a running topology with a new one, ensuring minimal downtime and no chance of both topologies processing tuples at the same time.

***Interested in learning Apache Storm? Click here to learn more in this***[***Apache Storm Training***](https://intellipaat.com/apache-storm-training/)***!***

### **14. How Storm UI can be used in topology?**

Storm UI is used in monitoring the topology. The Storm UI provides information about errors happening in tasks and fine-grained stats on the throughput and latency performance of each component of each running topology.

### **15. Why does not Apache include SSL?**

SSL (Secure Socket Layer) data transport requires encryption, and many governments have restrictions upon the import, export, and use of encryption technology. If Apache included SSL in the base package, its distribution would involve all sorts of legal and bureaucratic issues, and it would no longer be freely available. Also, some of the technology required to talk to current clients using SSL is patented by RSA Data Security, which restricts its use without a license.

### **16. Does Apache include any sort of database integration?**

Apache is a Web (HTTP) server, not an application server. The base package does not include any such functionality. PHP project and the mod\_perl project allow you to work with databases from within the Apache environment.

### **17. While installing, why does Apache have three config files – srm.conf, access.conf and httpd.conf?**

The first two are remnants from the NCSA times, and generally, you should be fine if you delete the first two, and stick with httpd.conf.

* **srm.conf :-** This is the default file for the ResourceConfig directive in httpd.conf. It is processed after httpd.conf but before access.conf.
* **access.conf :-** This is the default file for the AccessConfig directive in httpd.conf.It is processed after httpd.conf and srm.conf.
* **httpd.conf :-**The httpd.conf file is well-commented and mostly self-explanatory.

### **18. How to check for the httpd.conf consistency and any errors in it?**

We can check syntax for httpd configuration file by using  
following command.

httpd –S

This command will dump out a description of how Apache parsed the configuration file. Careful examination of the IP addresses and server names may help uncover configuration mistakes.

### **19. Explain when to use field grouping in Storm? Is there any time-out or limit to known field values?**

Field grouping in storm uses a mod hash function to decide which task to send a tuple, ensuring which task will be processed in the correct order. For that, you don’t require any cache. So, there is no time-out or limit to known field values.

The stream is partitioned by the fields specified in the grouping. For example, if the stream is grouped by the “user-id” field, tuples with the same “user-id” will always go to the same task, but tuples with different “user-id”‘s may go to different tasks.

***These are described in detail on***[***Big data Hadoop and Spark Community***](https://intellipaat.com/community/)***.***

### **20. What is mod\_vhost\_alias?**

This module creates dynamically configured virtual hosts, by allowing the IP address and/or the Host: header of the HTTP request to be used as part of the pathname to determine what files to serve. This allows for easy use of a huge number of virtual hosts with similar configurations.

### **21. Tell me Is running apache as a root is a security risk?**

No. the Root process opens port 80, but never listens to it, so no user will actually enter the site with root rights. If you kill the root process, you will see the other roots disappear as well.

### **22. What is Multiviews?**

MultiViews search is enabled by the MultiViews Options. It is the general name given to the Apache server’s ability to provide language-specific document variants in response to a request. This is documented quite thoroughly in the content negotiation description page. In addition, Apache Week carried an article on this subject entitled It then chooses the best match to the client’s requirements, and returns that document.

### **23. Does Apache include a search engine?**

Yes, Apache contains a Search engine. You can search a report name in Apache by using the “Search title”.

### **24. Explain how you can streamline log files using Apache storm?**

To read from the log files, you can configure your spout and emit per line as it read the log. The output then can be assigned to a bolt for analysis.

### **25. Mention how storm application can be beneficial in financial services?**

In financial services, Storm can be helpful in preventing

* **Securities fraud :**

1. Perform real-time anomaly detection on known patterns of activities and use learned patterns from prior modeling and simulations.
2. Correlate transaction data with other streams (chat, email, etc.) in a cost-effective parallel processing environment.
3. Reduce query time from hours to minutes on large volumes of data.
4. Build a single platform for operational applications and analytics that reduces the total cost of ownership (TCO)

* **Order routing:** Order routing is the process by which an order goes from the end-user to an exchange. An order may go directly to the exchange from the customer, or it may go first to a broker who then routes the order to the exchange.
* **Pricing:** Pricing is the process whereby a business sets the price at which it will sell its products and services, and maybe part of the business’s marketing plan.
* **Compliance Violations:** compliance means conforming to a rule, such as a specification, policy, standard, or law. Regulatory compliance describes the goal that organizations aspire to achieve in their efforts to ensure that they are aware of and take steps to comply with relevant laws and regulations. And any disturbance regarding compliance is a violation of compliance.

### **26. Can we use Active server pages(ASP) with Apache?**

Apache Web Server package does not include ASP support.  
However, a number of projects provide ASP or ASP-like functionality for Apache. Some of these are:

* **Apache: ASP:-** Apache ASP provides Active Server Pages port to the Apache Web Server with Perl scripting only, and enables developing of dynamic web applications with session management and embedded Perl code. There are also many powerful extensions, including XML taglibs, XSLT rendering, and new events not originally part of the ASP AP.
* **mod\_mono :-** It is an Apache 2.0/2.2/2.4.3 module that provides ASP.NET support for the web’s favorite server, Apache. It is hosted inside Apache. Depending on your configuration, the Apache box could be one or a dozen of separate processes, all of these processes will send their ASP.NET requests to the mod-mono-server process. The mod-mono-server process in turn can host multiple independent applications. It does this by using Application Domains to isolate the applications from each other, while using a single Mono virtual machine.

### **27. Explain what is Toplogy\_Message\_Timeout\_secs in Apache storm?**

It is the maximum amount of time allotted to the topology to fully process a message released by a spout. If the message is not acknowledged in a given time frame, Apache Storm will fail the message on the spout.

### **28. Mention the difference between Apache Kafka and Apache Storm?**

* **Apache Kafka:** [**Apache Kafka**](https://intellipaat.com/blog/what-is-apache-kafka/) is a distributed and robust messaging system that can handle a huge amount of data and allows the passage of messages from one end-point to another. Kafka is designed to allow a single cluster to serve as the central data backbone for a large organization. It can be elastically and transparently expanded without downtime. Data streams are partitioned and spread over a cluster of machines to allow data streams larger than the capability of any single machine and to allow clusters of coordinated consumers.  
  Whereas.
* **Apache Storm:** Apache Storm is a real-time message processing system, and you can edit or manipulate data in real-time. Storm pulls the data from Kafka and applies some required manipulation. It makes it easy to reliably process unbounded streams of data, doing real-time processing what Hadoop did for batch processing. A storm is simple, can be used with any **programming language**, and is a lot of fun to use.

### **29. What is ServerType directive in Apache Server?**

It defines whether Apache should spawn itself as a child process (standalone) or keep everything in a single process (inetd). Keeping it inetd conserves resources.  
The ServerType directive is included in Apache 1.3 for background compatibility with older UNIX-based version of Apache. By default, Apache is set to a standalone server which means Apache will run as a separate application on the server. The ServerType directive isn’t available in Apache 2.0.

### **30. In which folder are Java Application stored in Apache?**

Java applications are not stored in Apache, it can be only connected to another Java webapp hosting webserver using the mod\_jk connector. mod\_jk is a replacement for the elderly mod\_jserv. It is a completely new Tomcat-Apache plug-in that handles the communication between Tomcat and Apache. Several reasons:

* **mod\_jserv was too complex.** Because it was ported from Apache/JServ, it brought with it lots of JServ specific bits that aren’t needed by Apache.
* **mod\_jserv supported only Apache.** Tomcat supports many web servers through a compatibility layer named the jk library. Supporting two different modes of work became problematic in terms of support, documentation, and bug fixes. mod\_jk should fix that.
* **The layered approach** provided by the jk library makes it easier to support both **Apache1.3.x and Apache2.xx.**
* **Better support for SSL.** mod\_jserv couldn’t reliably identify whether a request was made via HTTP or HTTPS. mod\_jk can, using the newer Ajpv13 protocol.

**Top 30 Apache Storm Question and Answers**

**1. What is Apache Strom System?**

Apache Strom is an open-source, distributed, fault-tolerant, and real-time computing system that is used to process the stream of data in real-time. It was invented at BackType/Twitter and in 2011, it was open-sourced. Apache Strom has a very strong community (around 12) and 70 plus contributions.

**2. What the benefits of Apache Storm?**

The following are some core benefits of Apache Storm.

* Apache Storm is very easy to operate.
* It is capable of processing 100 messages per second per node.
* Apache Storm can easily fault and restart the process automatically.
* It provides guarantees to execute each data unit at least once.
* Apache Strom can scale horizontally.

**3. What is the difference between Hadoop and Apache Storm?**

The following are some of the differences between Hadoop and Storm.

| **Hadoop** | **Apache Storm** |
| --- | --- |
| Hadoop is a distributed, batch processing system that uses the MapReduce framework. | The storm is a distributed, real-time data processing system that uses DAGs. |
| The latency in Hadoop is very high. | The latency in Strom is low compared with Hadoop. |
| The framework of Hadoop is written in Java. | The framework of Strom is written in Clojure and Java. |
| Hadoop provides State-full stream processing. | Storm provides stateless stream processing. |
| It is very easy to setup Hadoop but difficult to handle the Hadoop cluster. | The storm is very easy to handle compared to Hadoop. |
| Hadoop is used by companies like Navisite, Twitter, and so on. | The storm is used by companies such as Search Engine Data and so on. |

**4. What are the major components of Apache Storm?**

Apache Strom has the following major components to perform the streaming of data flow.

* **Bolt:**It is a processing unit in the Storm Cluster.
* **Spout:**It is the source of data in the Strom Cluster.
* **Tuple:**It is a named list of values also represented as the main data structure in Storm Cluster.

**5. What is the difference between Apache Strom and Apache Kafka?**

The following are some of the differences between Apache Strom and Apache Kafka.

| **Apache Storm** | **Apache Kafka** |
| --- | --- |
| Apache Strom is used to processing messages in a real-time manner. | Apache Kafka is a distributed messaging system. |
| Apache Storm is invested by Twitter. | Apache Kafka is invested by LinkedIn. |
| Apache Strom supports all languages. | Apache Kafka also supports all languages but majorly Java is suggested. |
| The latency in Apache Storm is very low. | The latency in Apache Kafka is dependent upon the data source (usually 1-2 seconds). |
| Apache Storm is basically used for Stream Processing. | Apache Kafka is basically used for the Message Broker service. |
| It can process micro-batches. | It can process small batches. |
| Apache Strom has no such dependency. | Apache Kafka has a dependency on Zookeeper. |

**6. What are Streams in Apache Strom?**

The stream is Strom’s core abstraction which is a limitless sequence of Tuples. It provides "spouts" and "bolts" to perform basic Stream transformation into a new Stream.

**7. What are the different types of Apache Strom stream grouping?**

The following are the different types of Apache Strom stream grouping.

* Local grouping
* Global grouping
* Shuffle grouping
* Fields grouping
* None grouping
* Direct grouping
* All grouping

**8. What are the benefits of using Apache Strom in financial services?**

The following are the benefits provided by Apache Strom in financial service.

* It helps to detect securities fraud.
* It maintains order routing.
* It helps to maintain compliance.

**9. What is the use of “Topology\_Message\_Timeout\_secs” in Apache Storm?**

"Topology\_Message\_Timeout\_secs” represents the maximum time that is allocated to process a message. In case the message is not admitted in stipulated time then the message will be failed on Spout.

**10. What is Nimbus Node in Apache Storm?**

Apache Strom Nimbus node is the master node in the Apache Strom cluster which is responsible to keep track of workers. It allocates the resource to workers per requirement. Apache Strom Nimbus node acts as the Namenode of Hadoop.

**11. What are Worker or Supervisor Nodes in Apache Storm?**

Nimbus node assign tasks to Worker nodes and worker nodes are used to perform the actual operations. All worker nodes run the Supervisor daemon which is responsible for starting and stopping the worker process.

**12. What is the use of Zookeeper in Apache Storm Cluster?**

A zookeeper is used to maintain the coordination between the Storm Cluster. It maintains the Storm Cluster state. Zookeeper has very less load because it doesn’t use for message passing.

**13. What is Topology in Apache Storm?**

Topology is used to perform real-time computation in Apache Strom Cluster. It is a graphical representation of computation and implemented as DAG (directed acyclic graph). Every node in a topology comprises processing logic that is also called bolts and the relation between nodes shows the data processing in between nodes.

**14. What are Bolts in Apache Strom?**

Spouts are used to read tuples data from the source and release it to the topology. It is an entry point in Storm topology also called the source of data.

Spouts are divided into two types.

* **Reliable:**This type of Spout can replay in case of failure and call as “at least once message processing”.
* **Unreliable:**This type of Spout does not replay the tuples in case of failure and is called “at most once message processing”.

**15. What is Bolts in Apache Strom?**

Bolts are used to perform all processing in a topology by using functions such as joins, filters, aggregations, and so on. Bout can perform easy transformation as well as complex transformation such as the transformation of tweets would require at least two Bolts.

**16. What are the three main components that are used to run topology?**

The following are the major components that are used to run topology.

* **Worker Process:**It is used to execute a sub-portion of a topology. It belongs to topology and may trigger one or more executors for one or more components of topology.
* **Executor:**It can trigger one or more tasks from the same component. It is a thread that is spawned by a worker process.
* **Task:**It is used to perform actual data processing.

**17. What is the use of a combiner aggregator?**

A combiner aggregator is used to performing the grouping of tuples in a unified field.

**18. What is the command to kill an Apache Storm topology?**

The following is the command to kill Apache Storm topology.

Storm kill {cloudduggu\_topology} here "cloudduggu\_topology" is the name of topology.

**19. What is the use of ZeroMQ in Apache Storm?**

ZeroMQ is used to maintain communication between tasks in the Apache Strom cluster.

**20. Can we use Apache Storm as a Proxy server?**

We can use the mod-proxy module to do that.

**21. What is the command to check tpd.Conf Consistency?**

The following command is used to check tpd.Conf Consistency.

Httpd –S

**22. What are the companies which are using Apache Strom?**

The following is a list of few companies which are using Apache Strom.

* Yahoo!
* Twitter
* Spotify
* Yelp
* Flipboard

**23. What are the different modes of Apache Storm?**

Apache Strom provides the following operational modes.

* **Local Mode:**This mode is used for testing and debugging of topology.
* **Production Mode:**This mode is used for Production Operation.

**24. Can we update a running topology in Apache Strom?**

We can update a running topology by first kill it and then resubmit a new one.

**25. How to monitor a running Apache Storm topology?**

We can use the Apache Strom User interface (UI) tool to monitor the running topology. It provides information such as task error, latency, performance, and so on.

**26. What are the built-in schedulers provided by Apache Storm?**

Apache Storm provides the following list of built-in schedulers.

* Multitenant Scheduler
* Isolation Scheduler
* ResourceAware Scheduler
* Default Scheduler

**27. What will happen if a worker will not work and dead?**

If a worker node is dead then the Nimbus node will restart it. If it will keep getting fail during startup then the Nimbus node will reschedule the worker.

**28. What will happen if a node in the Apache Storm cluster is dead?**

In the case of a Node failure, the task which is getting executed by that node will be timed out and the Nimbus node will reassign that task to another node.

**29. Is there a search engine present in Apache Strom?**

Apache Storm has a search engine present that can be searched by “Search title” in Apache to provide relevant information.

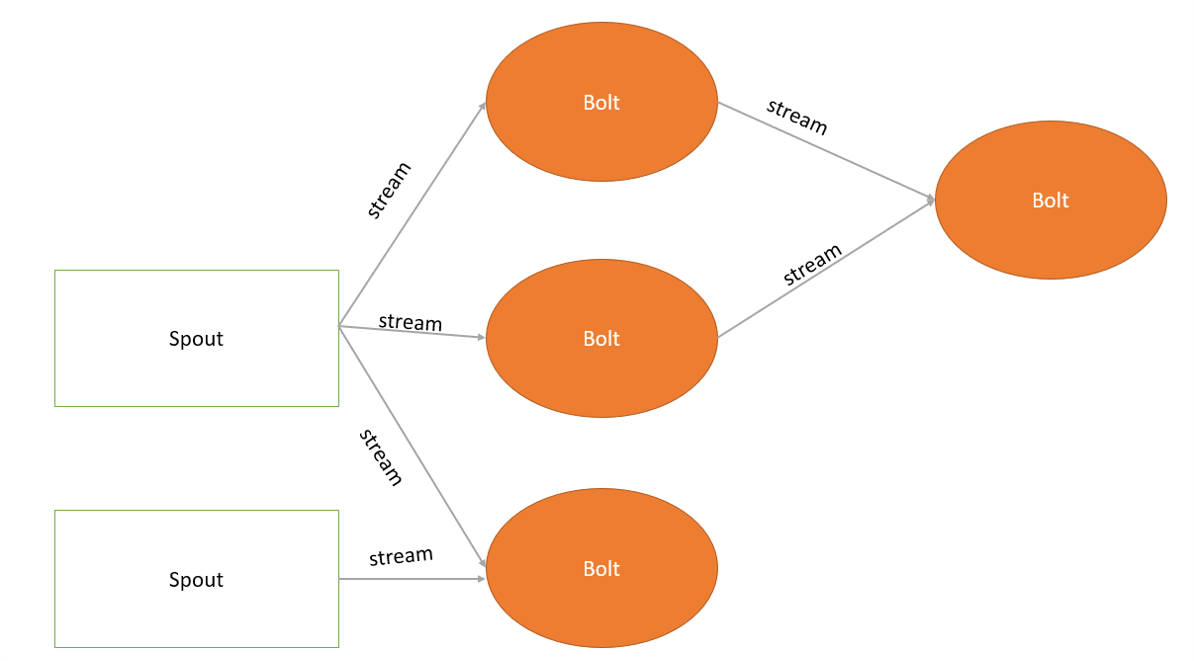
**30. Please specify when and why the cleanup method is started in Apache Storm?**

A cleanup method is called when a Bolt is shut down and clean up is required to clear all open resources.

## What is Apache Storm?

Apache Storm is a free and open-source distributed stream processing framework composed predominantly in Clojure. Founded by Nathan Marz and the unit at BackType, the project open-sourced following its acquisition by Twitter. Storm makes it simple to dependably process unbounded streams of information, producing real-time processing in place of what Hadoop did for batch processing. Storm is uncomplicated, can be utilized with several programming languages.

## What do you mean by "spouts" and "bolts"?

Apache Storm utilizes custom-created "spouts" and "bolts" to describe information origins and manipulations to provide batch, distributed processing of streaming data.  


## Where would you use Apache Storm?

Storm is used for: **Stream processing-**Apache Storm is adopted to the processing of a stream of data in real-time and update numerous databases. The processing rate must balance that of the input data. **Distributed RPC-**Apache Storm can parallelize a complicated query, enabling its computation in real-time. **Continuous computation-**Data streams are continuously processed, and Storm presents the results to customers in real-time. This might need the processing of every message when it reaches or building it in tiny batches over a brief period. Streaming trending themes from Twitter into web browsers is an illustration of continuous computation. **Real-time analytics-**Apache Storm will interpret and respond to data as it arrives from multiple data origins in real-time.

## What are the characteristics of Apache Storm?

1. It is a speedy and secure processing system.
2. It can manage huge volumes of data at tremendous speeds.
3. It is open-source and a component of Apache projects.
4. It aids in processing big data.
5. Apache Storm is horizontally expandable and fault-tolerant.

## How would one split a stream in Apache Storm?

One can use multiple streams if one's case requires that, which is not really splitting, but we will have a lot of flexibility, we can use it for content-based routing from a bolt for example: Declaring the stream in the bolt:

@Override

public void declareOutputFields(final OutputFieldsDeclarer outputFieldsDeclarer) {

outputFieldsDeclarer.declareStream("stream1", new Fields("field1"));

outputFieldsDeclarer.declareStream("stream2", new Fields("field1"));

}

Emitting from the bolt on the stream:

collector.emit("stream1", new Values("field1Value"));

You listen to the correct stream through the topology

builder.setBolt("myBolt1", new MyBolt1()).shuffleGrouping("boltWithStreams", "stream1");

builder.setBolt("myBolt2", new MyBolt2()).shuffleGrouping("boltWithStreams", "stream2");

## Is there an effortless approach to deploy Apache storm on a local machine (say, Ubuntu) for evaluation?

You use the below code, the topology is submitted to the cluster through the active nimbus node.

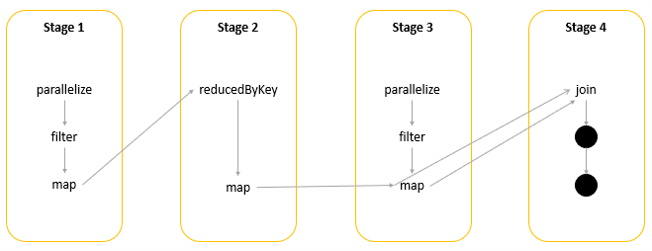
StormSubmitter.submitTopology("Topology\_Name", conf, Topology\_Object);

But if you use the below code, the topology is submitted locally in the same machine. In this case, a new local cluster is created with nimbus, zookeepers, and supervisors in the same machine.

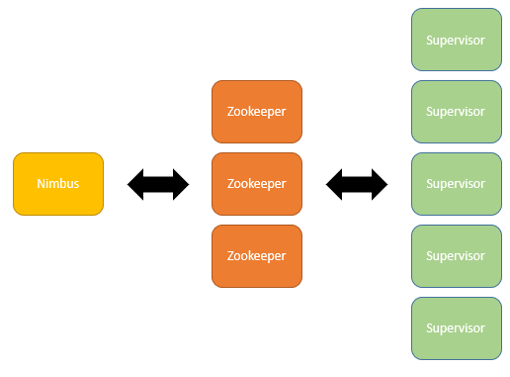
LocalCluster cluster = new LocalCluster();

cluster.submitTopology("Topology\_Name", conf, Topology\_Object);

## What is a directed acyclic graph in Storm?

Storm is a "topology" in the form of a directed acyclic graph (DAG) with spouts and bolts serving as the graph vertices. Edges on the graph are called streams and forward data from one node to the next. Collectively, the topology operates as a data alteration pipeline.  


## What do you mean by Nodes?

The two classes of nodes are the Master Node and Worker Node. The Master Node administers a daemon Nimbus which allocates jobs to devices and administers their performance. The Worker Node operates a daemon known as Supervisor, which distributes the responsibilities to other worker nodes and manages them as per requirement.  


## What are the Elements of Storm?

Storm has three crucial elements, viz., Topology, Stream, and Spout. Topology is a network composed of Stream and Spout. The Stream is a boundless pipeline of tuples, and Spout is the origin of the data streams which transforms the data into the tuple of streams and forwards it to the bolts to be processed.

## What are Storm Topologies?

The philosophy for a real-time application is inside a Storm topology. A Storm topology is comparable to MapReduce. One fundamental distinction is that a MapReduce job ultimately concludes, whereas a topology continues endlessly (or until you kill it, of course). A topology is a graph of spouts and bolts combined with stream groupings.

## What is the TopologyBuilder class?

java.lang.Object -> org.apache.storm.topology.TopologyBuilder

public class TopologyBuilder

extends Object

TopologyBuilder displays the Java API for defining a topology for Storm to administer. Topologies are Thrift formations in the conclusion, but as the Thrift API is so repetitive, TopologyBuilder facilitates generating topologies. Template for generating and submitting a topology:

TopologyBuilder builder = new TopologyBuilder();

builder.setSpout("1", new TestWordSpout(true), 5);

builder.setSpout("2", new TestWordSpout(true), 3);

builder.setBolt("3", new TestWordCounter(), 3)

.fieldsGrouping("1", new Fields("word"))

.fieldsGrouping("2", new Fields("word"));

builder.setBolt("4", new TestGlobalCount())

.globalGrouping("1");

Map conf = new HashMap();

conf.put(Config.TOPOLOGY\_WORKERS, 4);

StormSubmitter.submitTopology("mytopology", conf, builder.createTopology());

## How do you Kill a topology in Storm?

storm kill topology-name [-w wait-time-secs]

Kills the topology with the name: topology-name. Storm will initially deactivate the topology's spouts for the span of the topology's message timeout to let all messages currently processing finish processing. Storm will then shut down the workers and clean up their state. You can annul the measure of time Storm pauses between deactivation and shutdown with the -w flag.

## What transpires when Storm kills a topology?

Storm does not kill the topology instantly. Instead, it deactivates all the spouts so they don't release any more tuples, and then Storm pauses for Config.TOPOLOGY\_MESSAGE\_TIMEOUT\_SECS moments before destroying all workers. This provides the topology sufficient time to finish the tuples it was processing while it got destroyed.

## What is the suggested approach for writing integration tests for an Apache Storm topology in Java?

You can utilize [LocalCluster](https://github.com/apache/storm/blob/822a4685c0278aba9d4e0f43104bc4f86a462222/storm-server/src/main/java/org/apache/storm/LocalCluster.java" \l "L119) for integration testing. You can look at some of Storm's own integration tests for inspiration [here.](https://github.com/apache/storm/blob/e0feb6cf04107029d4b24420f265495cdfbdb098/storm-server/src/test/java/org/apache/storm/TestingTest.java#L38.) Tools you want to use are the FeederSpout and FixedTupleSpout. A topology where all spouts implement the CompletableSpout interface can be run until fulfillment using the tools in the Testing class. Storm tests can also decide to "simulate time" which implies the Storm topology will idle till you call LocalCluster.advanceClusterTime. This can allow you to do asserts in between bolt emits, for example.

## What does the swap command do?

A proposed feature is to achieve a storm swap command that interchanges a working topology with a brand-new one, assuring minimum downtime and no risk of both topologies working on tuples simultaneously.

## How do you monitor topologies?

The most suitable place to monitor a topology is utilizing the Storm UI. The Storm UI gives data about errors occurring in tasks, fine-grained statistics on the throughput, and latency performance of every element of each operating topology.

## How do you rebalance the number of executors for a bolt in a running Apache Storm topology?

You continually need to have larger (or equal number of) jobs than executors. As the quantity of tasks is fixed, you need to define a larger initial number than initial executors to be able to scale up the number of executors throughout the runtime. You can see the number of tasks, like a maximum number of executors:

#executors <= #numTasks

See [here](https://storm.apache.org/documentation/Understanding-the-parallelism-of-a-Storm-topology.html) for details.

## What are Streams?

A Stream is the core concept in Storm. A stream is a boundless series of tuples that are processed and produced in parallel in a distributed manner. We define Streams by a schema that represents the fields in the stream's records.

## What can tuples hold in Storm?

By default, tuples can include integers, longs, shorts, bytes, strings, doubles, floats, booleans, and byte arrays. You can further specify your serializers so that custom varieties can be utilized natively.

## How do we check for the httpd.conf consistency and the errors in it?

We check the configuration file by using:

httpd -S

The command gives a description of how Storm parsed the configuration file. A careful examination of the IP addresses and servers might help in uncovering configuration errors.

## What is Kryo?

Storm utilizes Kryo for serialization. Kryo is a resilient and quick serialization library that provides minute serializations.

## What are Spouts?

A spout is the origin of streams in a topology. Generally, spouts will scan tuples from an outside source and release them into the topology. Spouts can be reliable or unreliable. A reliable spout is able to replay a tuple if it was not processed by Storm, while an unreliable spout overlooks the tuple as soon as it is emitted. Spouts can emit more than one stream. To do so, declare multiple streams utilizing the declareStream method of [OutputFieldsDeclarer](https://storm.apache.org/releases/current/javadocs/org/apache/storm/topology/OutputFieldsDeclarer.html) and define the stream to emit to when applying the emit method on [SpoutOutputCollector.](https://storm.apache.org/releases/current/javadocs/org/apache/storm/spout/SpoutOutputCollector.html) The chief method on spouts is nextTuple. nextTuple either emits a distinct tuple into the topology or just returns if there are no new tuples to emit. It is important that nextTuple does not block any spout implementation as Storm calls all the spout methods on the corresponding thread. Other chief methods on spouts are ack and fail. These are called when Storm identifies that a tuple emitted from the spout either successfully made it through the topology or failed to be achieved. ack and fail are only called for reliable spouts. See [the Javadoc](https://storm.apache.org/releases/current/javadocs/org/apache/storm/spout/ISpout.html) for more information.

## What are Bolts?

All processing in topologies is done in bolts. Bolts can do everything from filtering, aggregations, functions, talking to schemas, joins, and more. Bolts can perform simplistic stream transmutations. Doing complicated stream transformations usually demands multiple actions and hence added bolts.

## Compare Apache Storm with Kafka.

