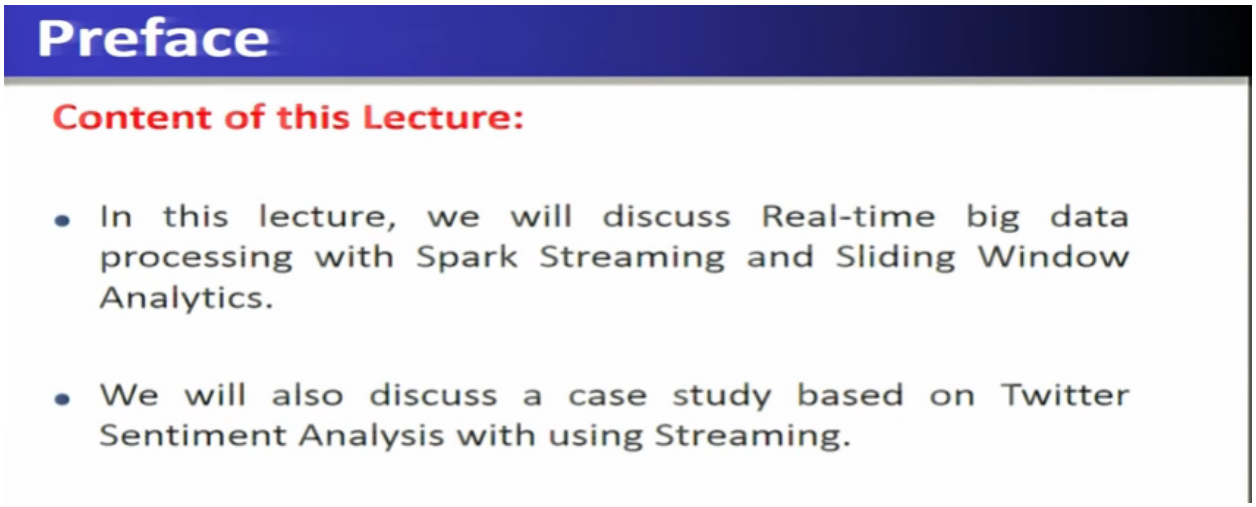
Lecture-19

Spark Streaming and Sliding Window Analytics (Part-I)

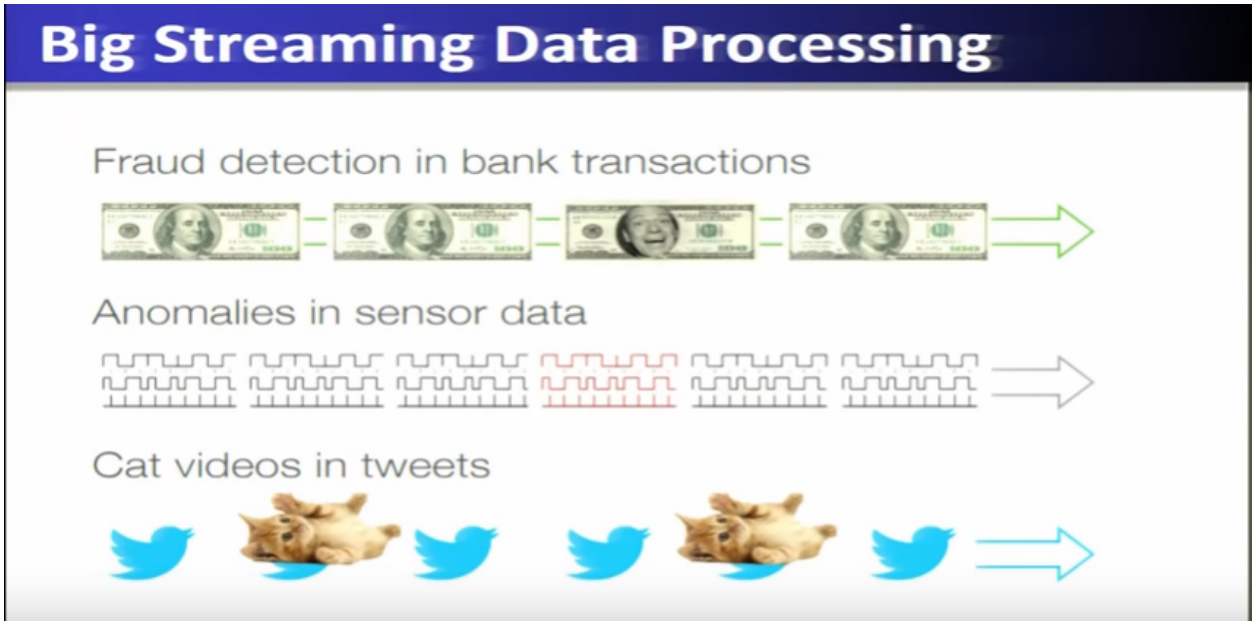
Spark Streaming and Sliding Window Analytics.

Refer slide time: (0:17)



Preface, content of this lecture: in this lecture, we will discuss real time big data processing, with Spark Streaming and sliding window analytics. We will also discuss, a case study, based on Twitter sentiment analysis, using Streaming.

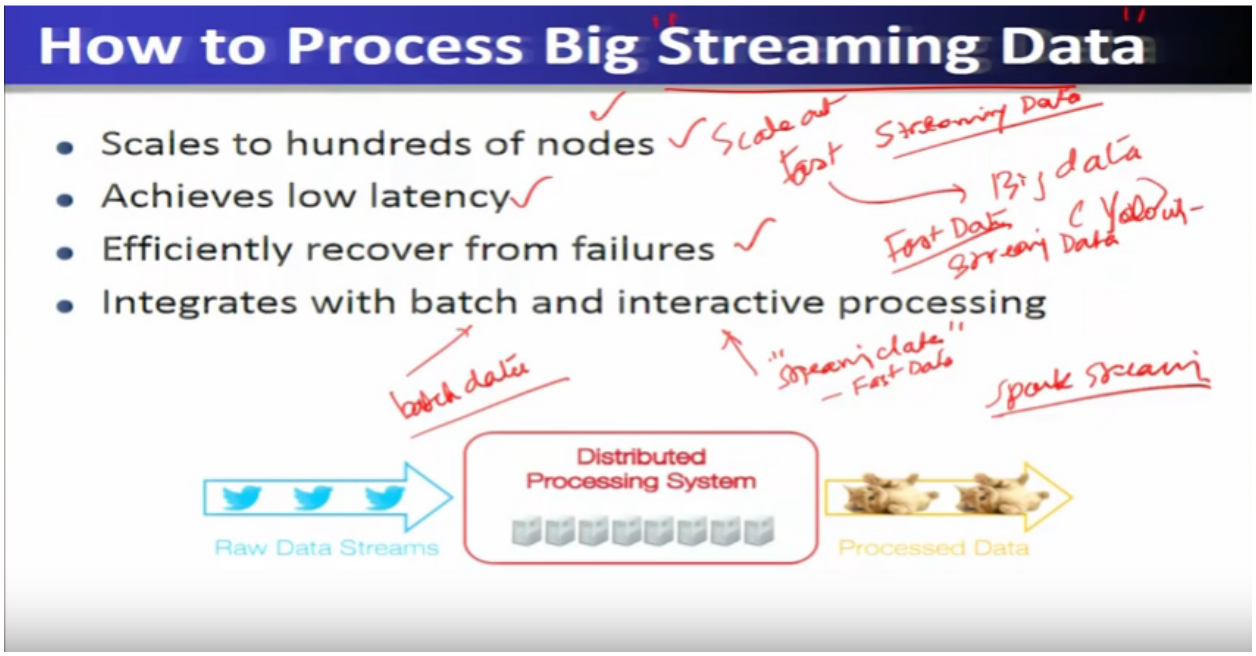
Refer slide time: (0:35)



Big Streaming data processing. the motivation of going for Streaming data processing is, deep-rooted and is based on, the motivation is based on the application such as, fraud detection indifferent banking transactions, which are happening online and, and the real time, how this particular fraud detection can be done? That becomes a challenge. So therefore, this motivates, Streaming data processing system and how using that? We can do this online or in a real time, the fraud detection, in the banking transactions. Another such application which has motivated, the use of Streaming data processing is, detecting the anomalies in the sensor data. Now, you have seen that, lot of IOT deployments are, happening these days,

which basically collects, the sensor data and then, based on these sensing data, the anomalies are to be detected in real time. This also has given, the new way of solving these problems, using the Streaming data processing in real time. similarly we want to find out, we want to do the analysis, of tweets and based on these analysis, we can find out the sentiments or the mood of the people, at a particular instance, so this online in a real-time, this tweet analysis also requires, the Streaming data processing. So these applications are, a newer applications are, now given a reason to think of providing Streaming data processing, for analysis, for analytics in the real time. And therefore, we are going to understand, the new concepts and the very new theory that is, Spark Streaming, how that is being used in today's workload and then ewer applications.

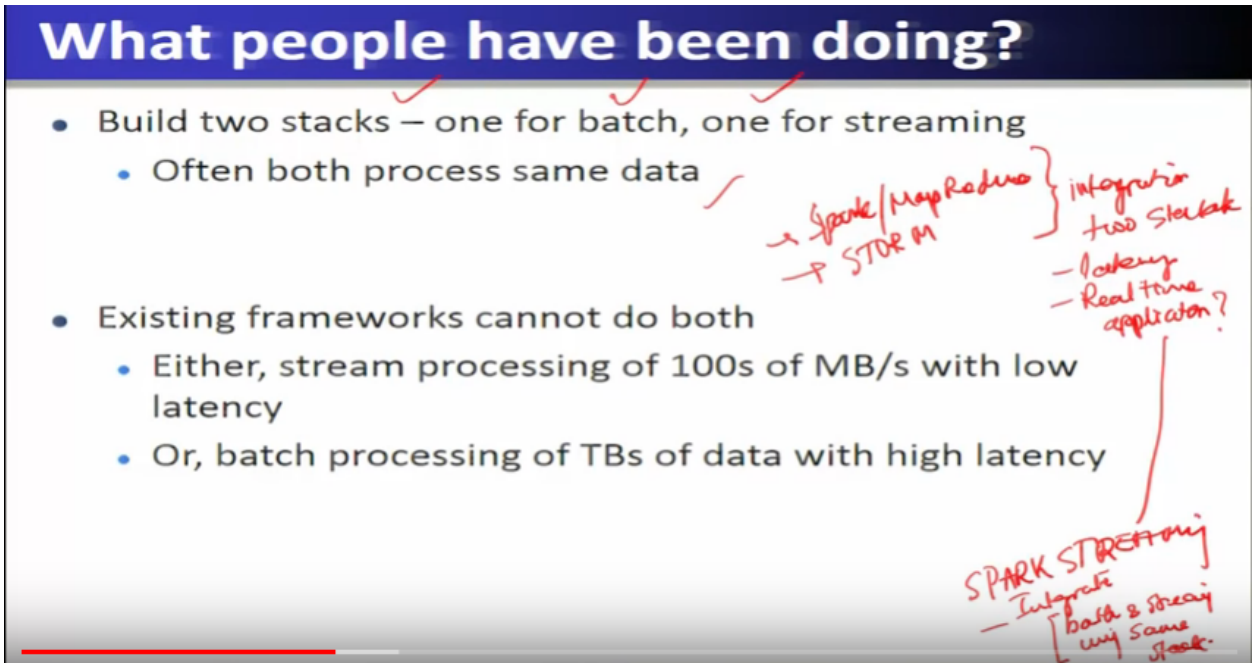
Refer slide time: (3:03)



Now, the question is how to process the big data big Streaming data? So, now here, we are going to deal in this part of the lecture, the data which is called a, 'Streaming Data'. So, if the, the stream of the data is very fast and also a continuous stream, then it is categorized under the Big Data, scenario. Where in this particular, characteristic which is called a, 'Volume', sorry, 'Velocity', requires the infrastructure of a big data to handle it, so that means, we require hundreds, to the thousands of these nodes to scale, that means, to scale out, to deal with this first data, which is also called as the, 'Streaming Data'. So, for the scaling and scalability and for, we require thousands of, hundreds of, nodes to process this big Streaming data, another aspect is about, achieving the low latency, obviously this requires the insight into the technology, we will see how, we can achieve the low latency, in this Streaming data processing, framework. Now another, requirement is about, how to deal with the failures? And how to deal not only with the failures? But, how to efficiently recover from the failures? So that, it can be tolerated by the applications, which are monitoring in the real time, the events and for different applications. So, we have to deal with, the failure recovery which is to be done. So that, it can be, it can cater to the application in a real-time applications, which are based on Streaming data. Now, another thing is that, now there are various interactive processing is, happening with the Streaming data. And this particular Streaming data is also

called as a, ‘Fast Data’. Now, besides this fast data, sometimes you we also require to be, to integrate with another stream of batch data. So, if there are two different modes of data that, means, one is through the batch data, the other is called, ‘Streaming data’, together there are some applications which requires, the integration of both these different type of data and they are different and they are having a different, stack to be processed. So, how are we going to integrate them, in the traditional, in the previous the technologies, they are having a different stacks and therefore, the integration is basically time taking and may not be useful for real-time applications. So here, we are going to see, the new technology which is called the, ‘Spark Streaming’. Which will combine or integrate, this requirement of processing, simultaneously the batch and the interactive data.

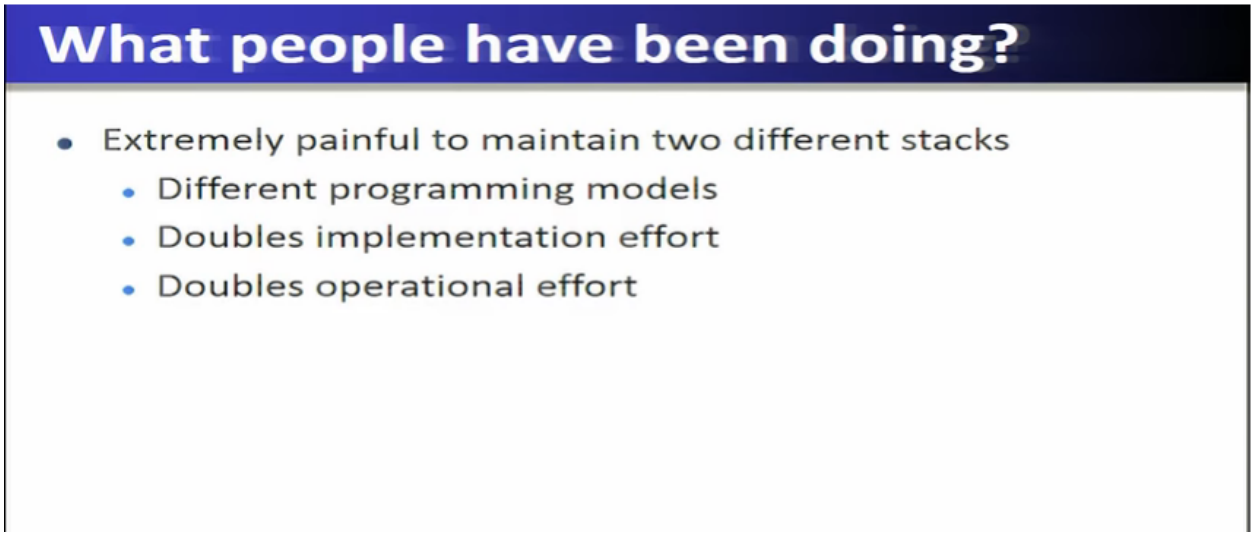
Refer slide time: (7:16)



Now, let us see, how what, what people have been doing? So far that means, what are the other systems, before this Spark is Streaming. And how it was being done. So, actually as we see that, for batch processing and for stream processing there, we requirement of different stacks and often different stacks will are, optimized for different type of data, so integration is not very common, in the previous generation of systems. and for example, for batch processing we have the Spark system and we have the Map Reduce, framework for doing the batch processing and for, the Streaming in the earlier, systems were, such as storm. So, now integration requires two different stacks, to be combined together. So hence, it requires it has lot of latency, involved within it and may not be require, may not be sufficient for different real-time applications. That is why, this particular framework which we are going to discuss, in today's lecture that is called, ‘Spark Streaming’. Which will integrate, both batch and Streaming applications using, the same stack. Therefore, it will be the most efficient way, of dealing with multiple, type of data, that is the batch data and the stream data, when they are required to be processed, at the same time. So, the existing framework, such as, storm and Map Reduce, cannot do both of the, the processing that is for the batch and the Streaming data at the same point of time. So, either the swimming stream processing of hundreds, of megabytes, with a low latency or the batch processing of terabytes of

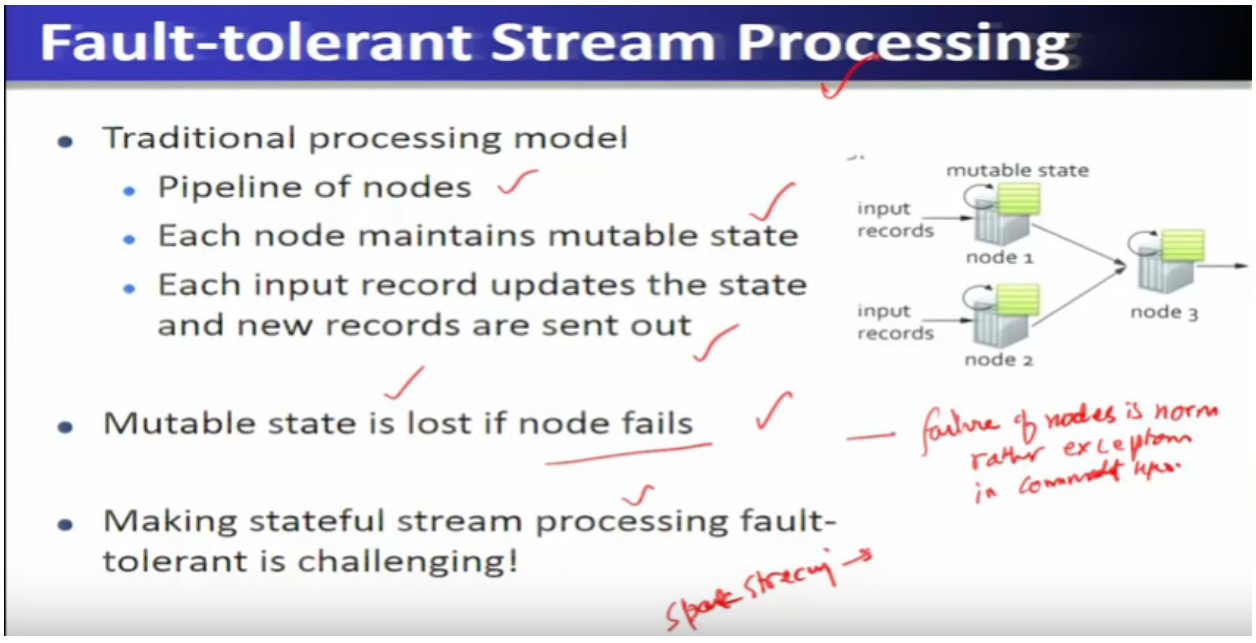
data, with high latency, are to be dealt separately and the combined or integrated viewpoint is not available, as of date before this Spark is Streaming. So, Sparky Streaming is the new technology, which we are going to discuss and we will be also seeing, different use cases where this kind of, batch processing and stream processing together are, required in many applications.

Refer slide time: (10:22)



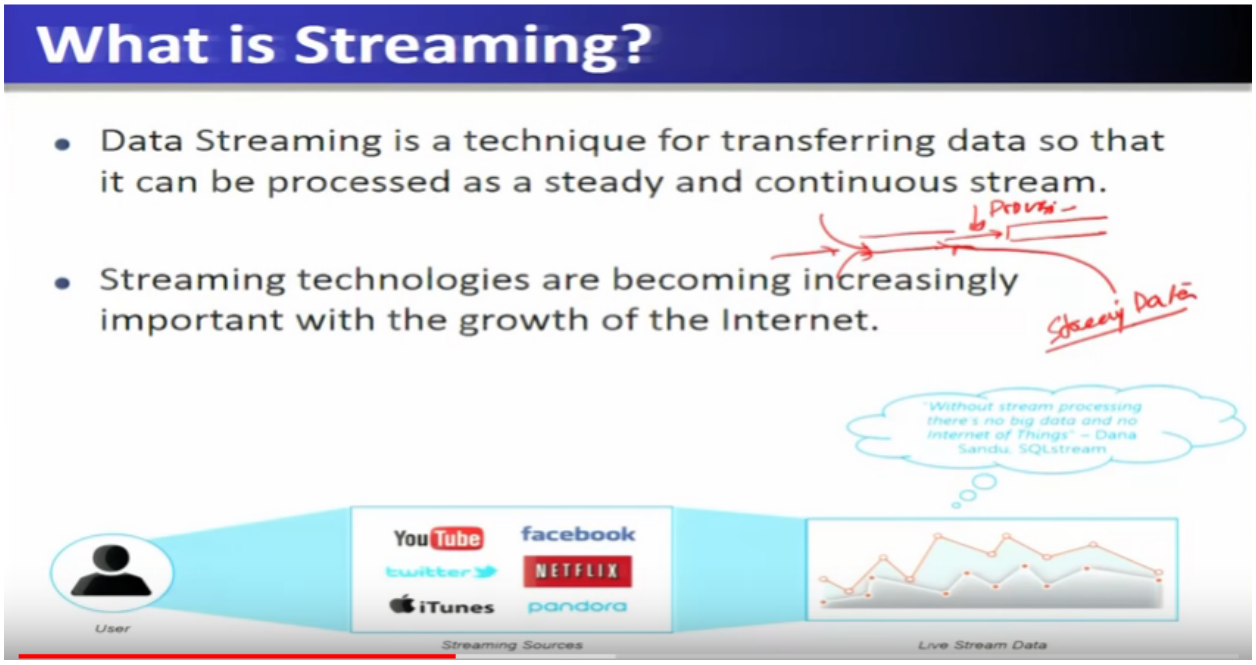
So therefore, there are, it is required to maintain two different stacks, if the, the integration is not supported and also, they may require different programming models and also, different efforts are required and also requires, an operational cost. So that is not feasible.

Refer slide time: (10:45)



Therefore the Spark is Streaming, we are going to discuss, today how that is all integrated and is being useful, for various applications. Now, let us see, the before going in more detail about, the Spark Streaming. So, how we will see, let us see that, how the fault-tolerance is achieved, in the stream processing systems. So, traditional processing models use, the pipeline of different nodes and each node maintains, the mutable States and each input record updates, the state and new records are sent out. Now, the problem with the previous systems, were that, the mutable States, were lost if the node are filled. And this is a normal failure of the nodes is, a norm rather than exception, in the commodity hardware. So therefore, when the node fields, the mutable states which are maintaining, this fault tolerance, of for the mutable states, will be lost. So some things are, basically lost which are, in the traditional the previous systems. Therefore, the stream therefore, making the stateful stream processing fault tolerant is, also very much, needed and in, in Spark Streaming system. Will, we will see that how this stateful, in stream processing, is done and in a fault tolerant manner.

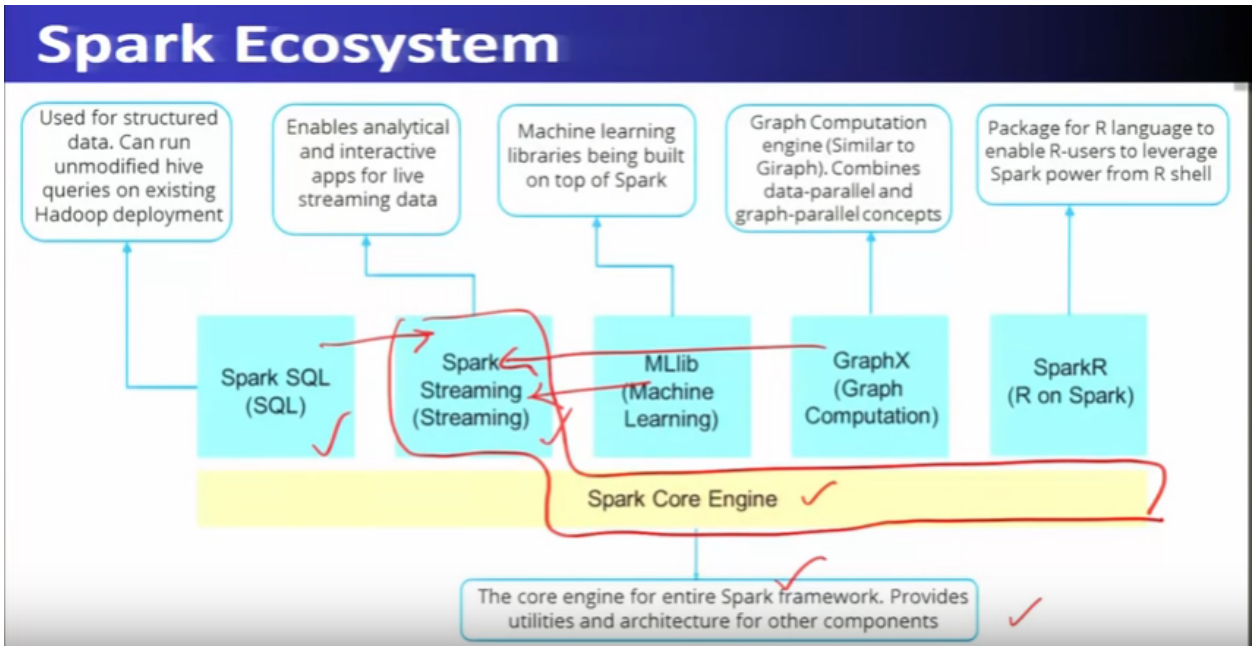
Refer slide time: (12:44)



Now, let us see, what is a Streaming? So Data Streaming is, a technique, for transferring data. So that, it can be processed, as a steady and continuous stream. Which is incoming to the system, in a stream of data, which is incoming to the system. So, you can visualize, as if, the data is flowing continuously on the pipes and as it process, as it passes through, these pipes it has required to be processing, in a real-time. And if that is cause, if that is there then, it is called the, 'Data Streaming', or a, 'Streaming Data'. Now, the sources which generates, the Streaming data are many, for example, the internet traffic, which is flowing also, if it is seen then, it will also be a network Streaming data, similarly the Twitter, Streaming data also can be taken up, in some of the applications. Similarly the Netflix which is, in real-time online, movie watching, that also generates the Streaming data, similarly YouTube data also, a kind of the Streaming data. and there are many other many, many other ways, this Streaming data can be generated,

Streaming data can also be generated from the database, so data is read and is being transmitted, in the form of the Streaming heater, that is ETL. So, companies normally, does this for the analysis, so Streaming data technologies are, becoming increasingly important, I with the growth of the Internet and the Internet enabled, different services, which are available in the form of, Netflix, Facebook, Twitter, YouTube, Pandora, iTunes and so on. There are tons of such, different nowadays, services available through the internet.

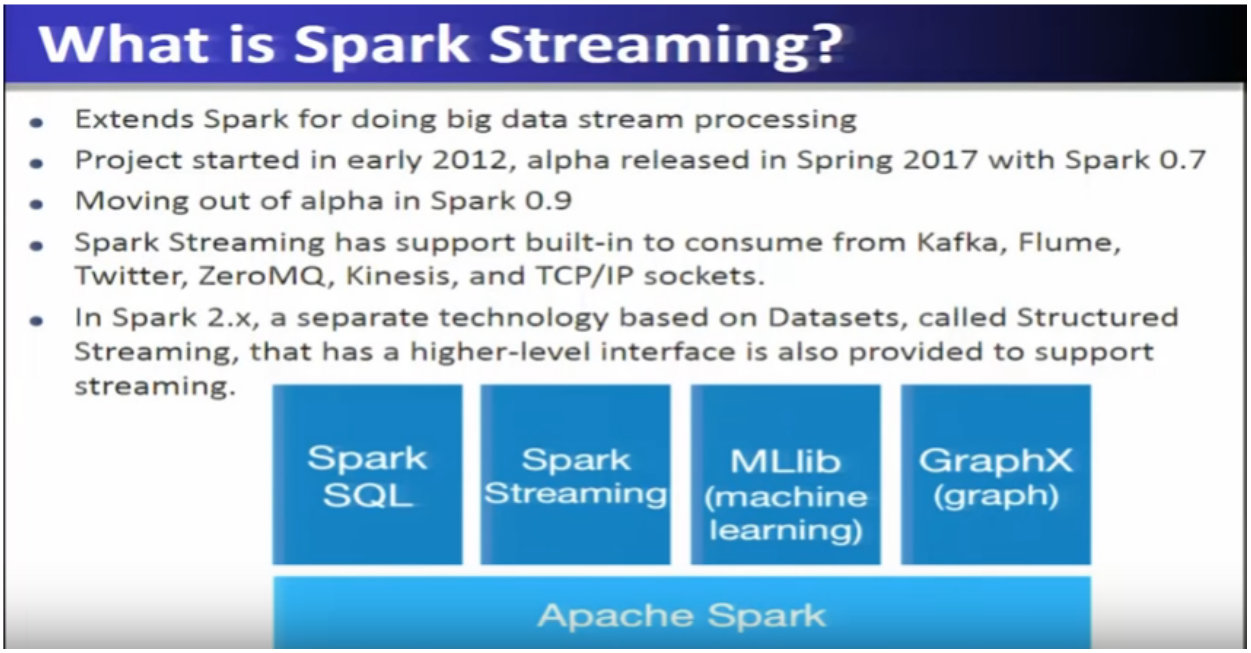
Refer slide time: (14:49)



Now, let us see, the Spark ecosystem. And we will see the positioning of, the Spark streaming system, where it lies. So from in this Spark ecosystem, you will see that, on the core there is a Spark engine and on top of it, is you can see that, Spark Streaming system is running. Now, the Spark is Streaming system, runs over the Spark core and this, enables the analytical and interactive applications for, the live Streaming data. So therefore, these core components of the Spark framework, provides the utilities and architecture, for the other components also. Therefore dealing with this, Spark Streaming or Spark, gives many other advantages. for example, the analytics which is required to be performed, on the Streaming data, may sometimes need the machine learning, on that to be applied on the own Streaming data. Now, with one single common stack, that is a Spark or engine, it is now possible that, we can apply or it is possible that the machine learning libraries, which are built on top of the Spark, can also be used for analytics of the Streaming data. similarly the graph processing, applications such as graph computation engines, which combines data parallel and graph parallel concepts, can also be useful for the analysis, of our analytics of the Streaming data. And they are all integrated, similarly the Spark SQL which is also the structured, way of accessing the key value store, can also be useful, to be used in the Spark Streaming for storage and retrieval purposes. so therefore, having the same in stack, this Spark Streaming will gain a lot of advantages, not only for doing the batch processing and the Streaming data together, there can be integrated and a lot of other, that means, libraries can be used, such as, machine learning can be applied, for the Streaming analysis and graph also can be used, as for the Streaming data analysis. Which can be

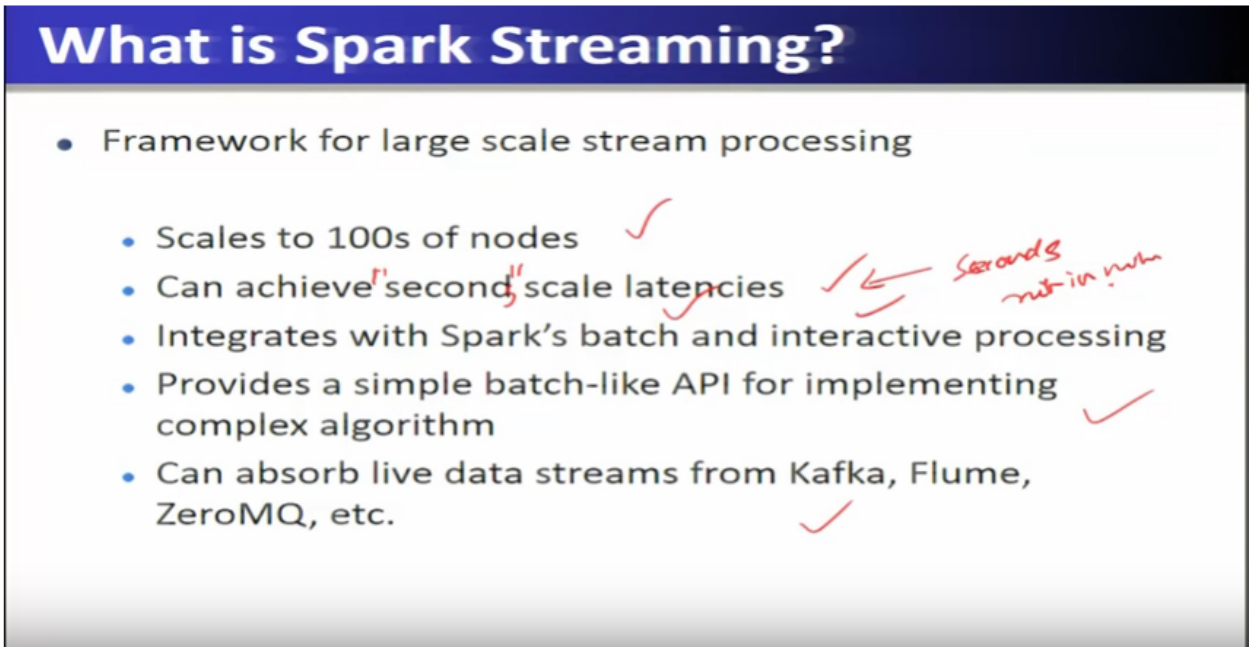
represented in a graph and can be done, an analysis. We will see this Spark ecosystem. so the Spark Streaming, the positioning on top of the Spark core, will gain not the advantages of other such utilities, like machine learning, that is Emilie, graphics, Spark ,SQL and soon. We will see, in further slides, how we are going to utilize the integration of all these together, in solving today's workload problems.

Refer slide time: (17:54)



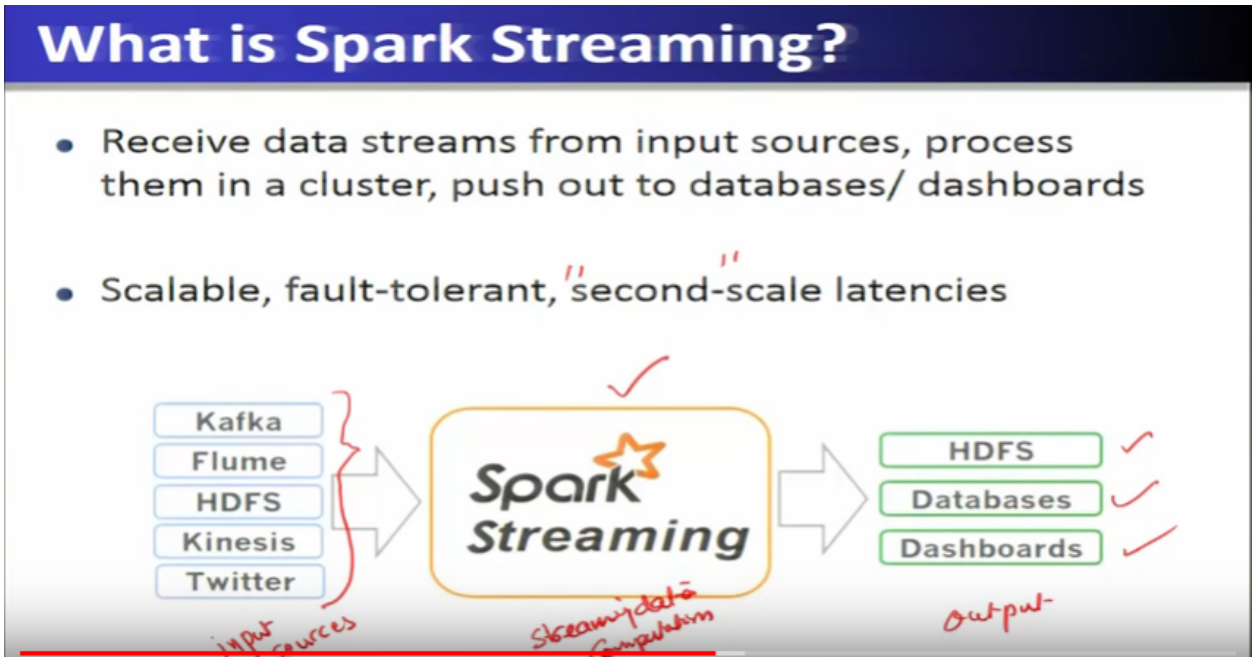
So, what is this Spark Streaming? So is it Spark it extends, the Spark for doing the data, big data Streaming processing. So, big data stream processing, can be now done, with the help of Spark Streaming, which is an extension, which extends the Spark core, to perform this Spark Streaming. Now, this Spark Streaming project was started in, 2012 and it was released in, the form of this Spark 0.9. And Spark Streaming highest, lot of built-in, support to consume the data from different sources, such as, from Kafka is also one of the receiver, which can feed the live stream data, to the Spark stream, similarly flume, Twitter, 0 MQ, kinesis and TCP/IP sockets. Are some of the inbuilt, receiver system, through API is, they can be plugged into, the Spark Streaming system. Now, to get the stream data, for computations in this, scenario. So, in as park to point acts, a separate technology based on, the data set called, 'Structured Streaming', has been designed and that has a higher level interface, provided to provide the, the, the support of Streaming.

Refer slide time: (19:26)



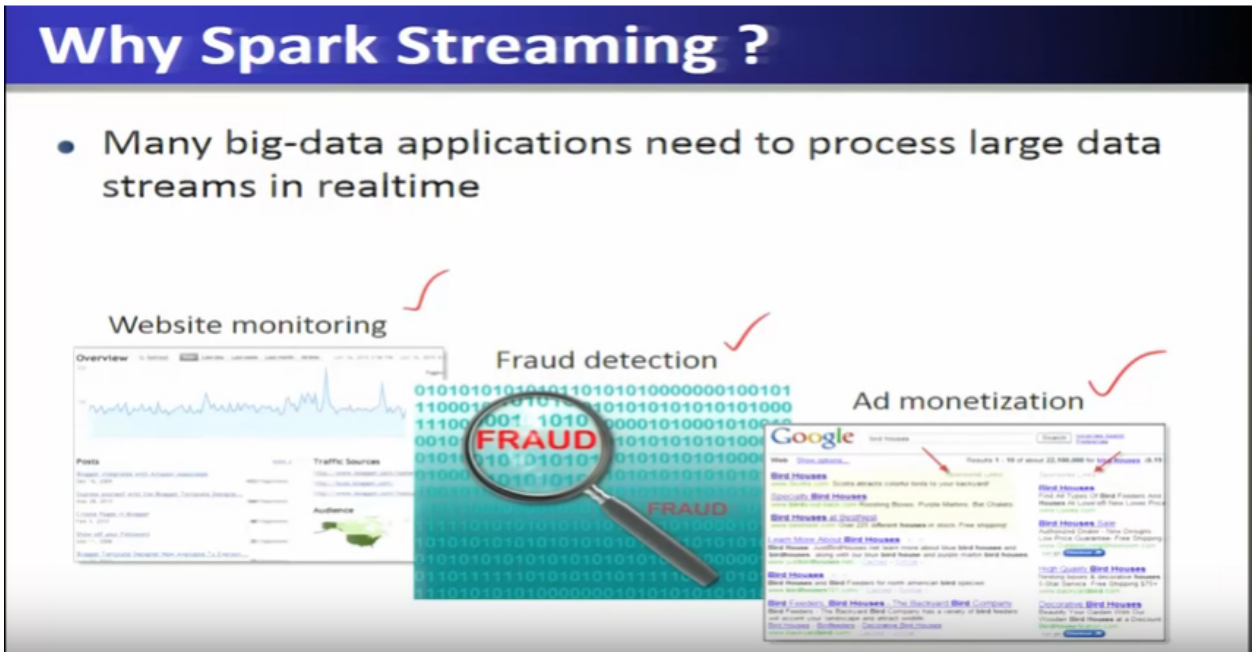
So, let us see, the Streaming, Spark Streaming framework, for a large scale processing of Streaming data. And we will again, let us review that, it should be scale to the hundreds and thousands of the node, it can achieve, can achieve second scaled, the latencies. So, that means, latencies are to be in the form of seconds. So, latencies are in the, in the scale of seconds, not in the minutes are, not latency should be in the minutes or hours and so on. Now, achieving this is the scale, of seconds, latency is not going to be easy task and it requires, the new design. Therefore, we are going to understand the Spark Streaming system. Now, this is Spark Streaming system also integrates, the batch and the interactive processing together, with a unified view with the same stack, therefore, with unification on this batch and interactive application, it is now possible to achieve, the latencies in the, in the scale of seconds. Similarly, it will provide a simple batch like APL is, for implementing the complex algorithms. And it can also, integrate it with the live data streams from, different other life, other tools such as Kafka, flume, 0 MQ and so on. So, that is why this Spark Streaming is now, a day's required very much.

Refer slide time: (21:22)



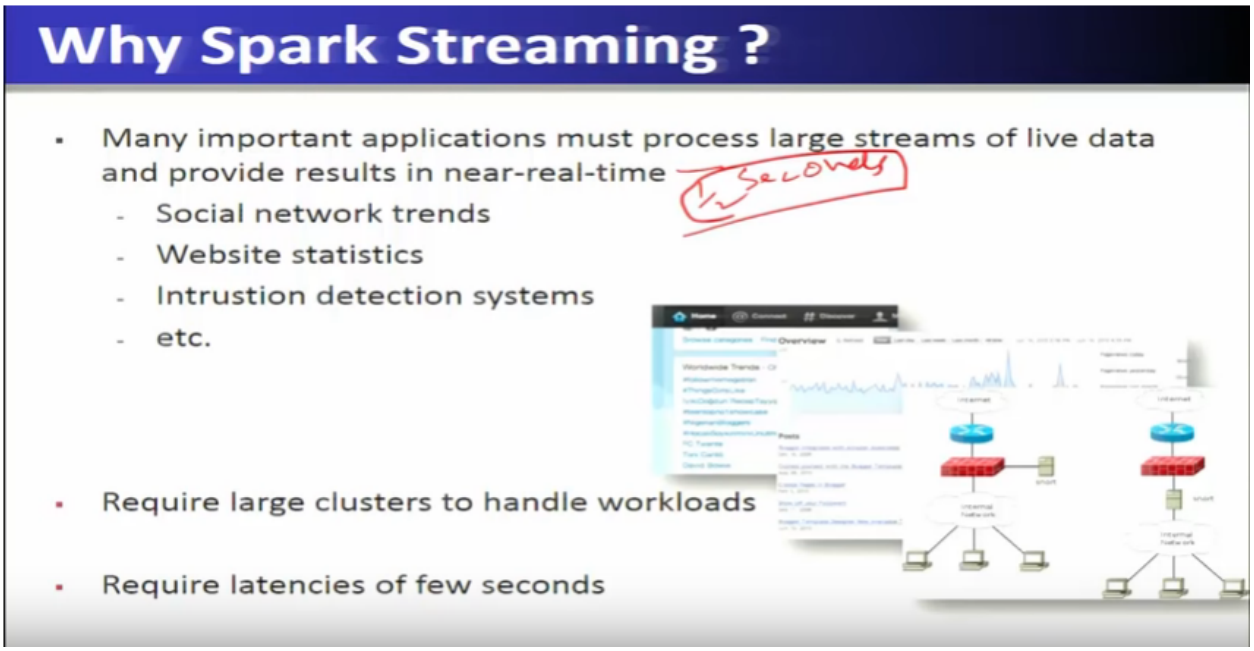
Now, Sparky Streaming receives the data stream from different input sources, process them in the cluster and push it, out to the databases, dashboard for its output. Therefore, Sparky Streaming is a, scale able fault tolerant and having the, the, the latencies of, the scale of, the second's time. So, let us see, through this particular diagram that, the input, the data can be received through different input sources. So, it can be received either from Kafka, flume or HDFS, kinesis or Twitter, this live stream of data is, now fed into the SPARK Streaming system, for the computation of Streaming data, computation. And after that, the output will be either stored on the database our, it will be pushed on the, dashboard for the output. So, we will now see, in this part of the discussion. What is the SPARK Streaming? How it is handling this kind of Streaming of Streaming data of different sources together? And for which is useful for various applications?

Refer slide time: (22:54)



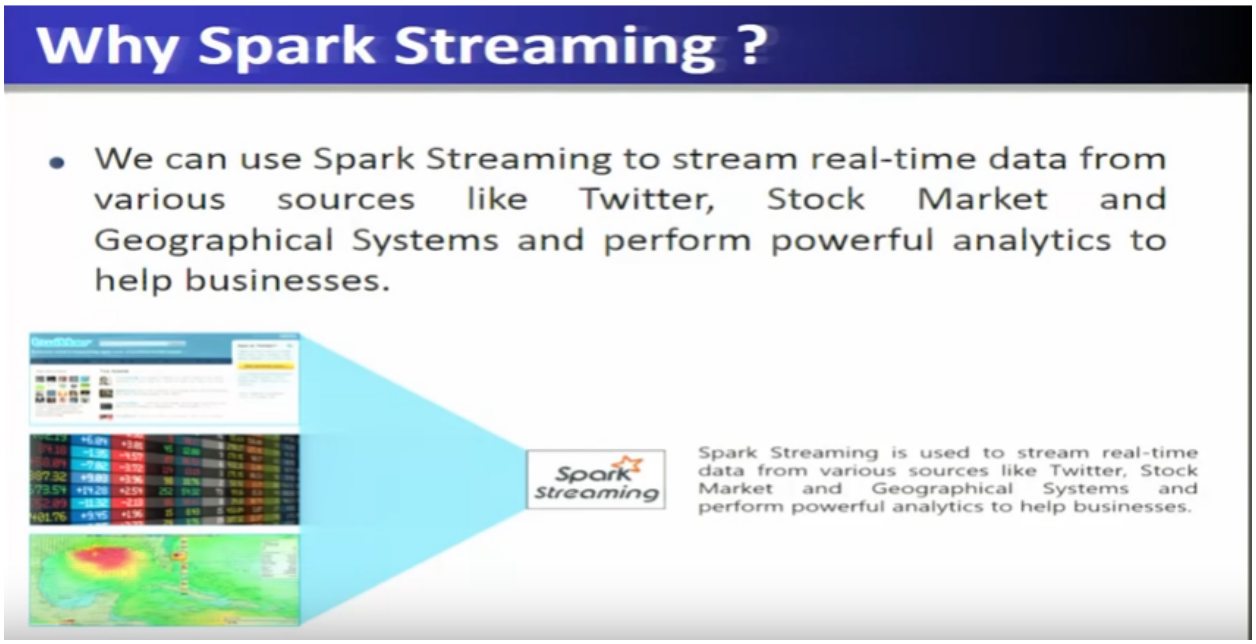
So therefore, again are going to understand about, why the Spark is Streaming? So many big data applications, need to process the large data streams in the real time, such as website monitoring sometimes, require is, required to see the the loads, the hits, which are basically coming on, the website and whether the performance of the website is, going well or not, for has to be monitored. Similarly the online transactions, for the bank or from for the credit card ,also really needs to be analyzed, in real time. So that becomes, Streaming data applications. Similarly various ad monetization also, require to be processed in real time. So, whenever a user clicks on a particular ad and so those ads, are to be so, when the user will click. So all that, data the click-through, data has to be analyzed in real time and therefore, different ads are to be pushed, in that manner.

Refer slide time: (24:13)



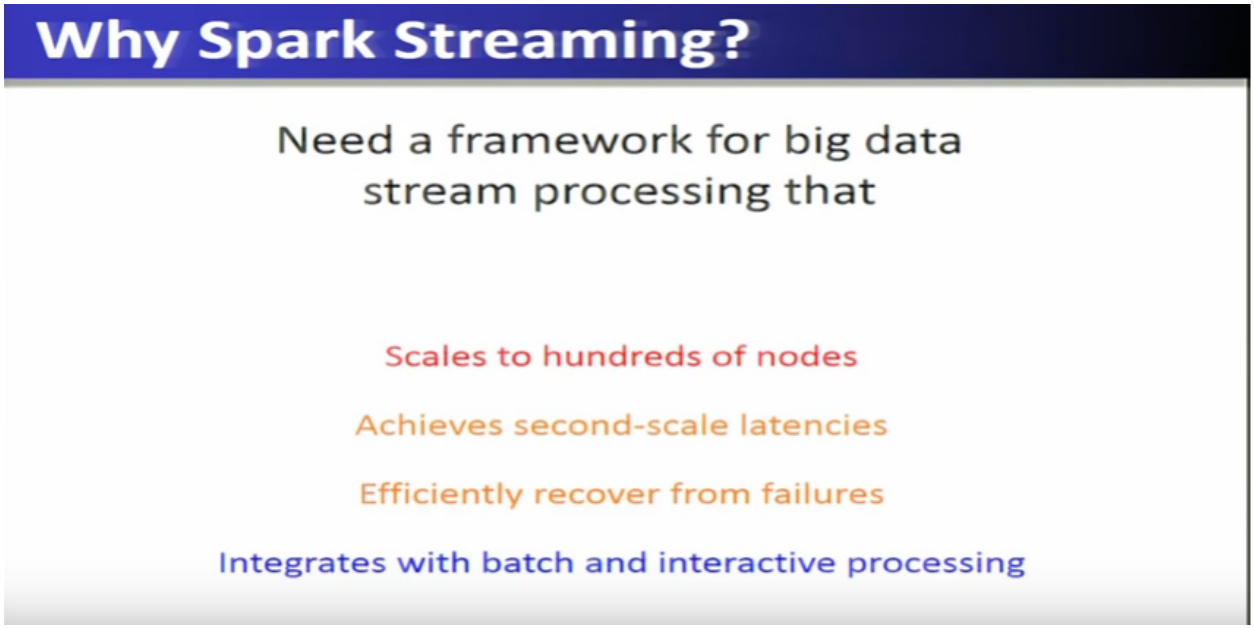
So, the Spark Streaming is very much, needed and many important application, must process the large stream of live data and provides, the results in a near real-time, near real-time means, this particular system, which we are going to discuss that is Spark Streaming, has the latency is up to the half of the second. So, those latencies which are tolerable up to half of the seconds, can be used here, by with the help of Spark Streaming, even lesser than this, can be used in a separate Streaming systems, which are known as, the storm and so on. Which has very less, latency already available. So, let us see that, if the, the different application such as, social network trends, where statistics intrusion detection system and so on. They, they need this kind of Streaming system and require, the large cluster to handle these workloads and also require the, the, the, the latencies of the order seconds.

Refer slide time: (25:29)



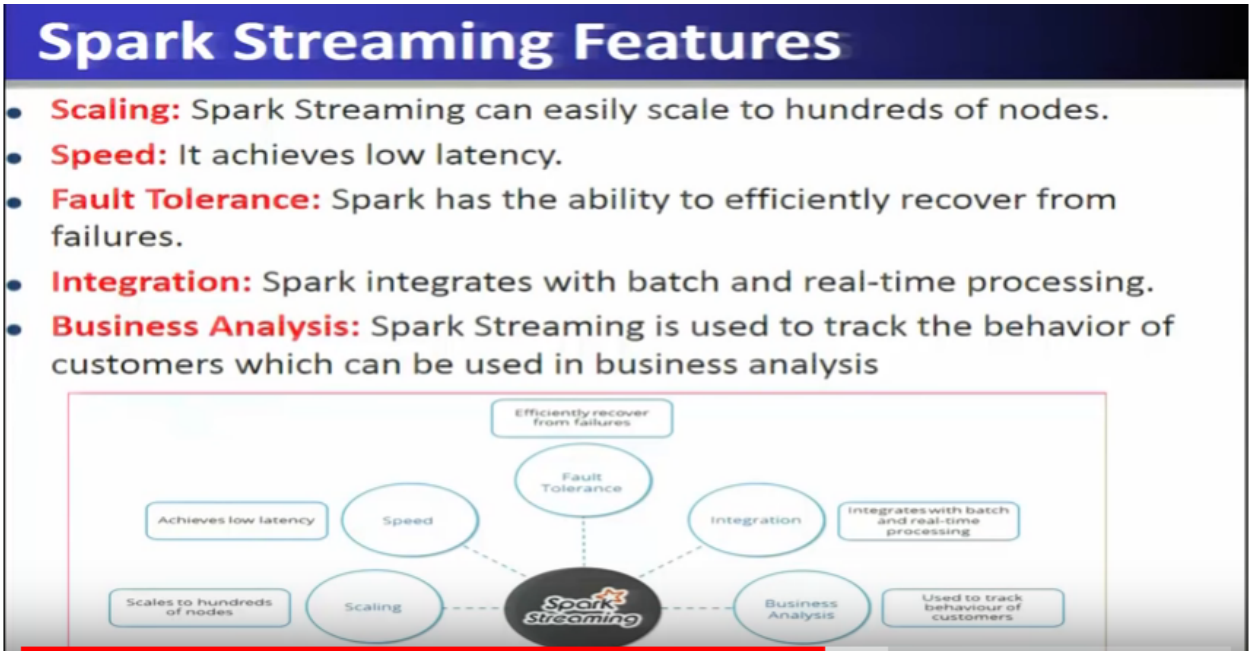
So, we can use the Spark is Streaming to, stream the real-time data from various sources, like Twitter, a stock market, geographical system, for doing the powerful analytics. So, Sparky sleeping is used to stream, the real-time data from various sources like Twitter, stock market, geographical systems, perform powerful, analytics to help various business.

Refer slide time: (25:51)



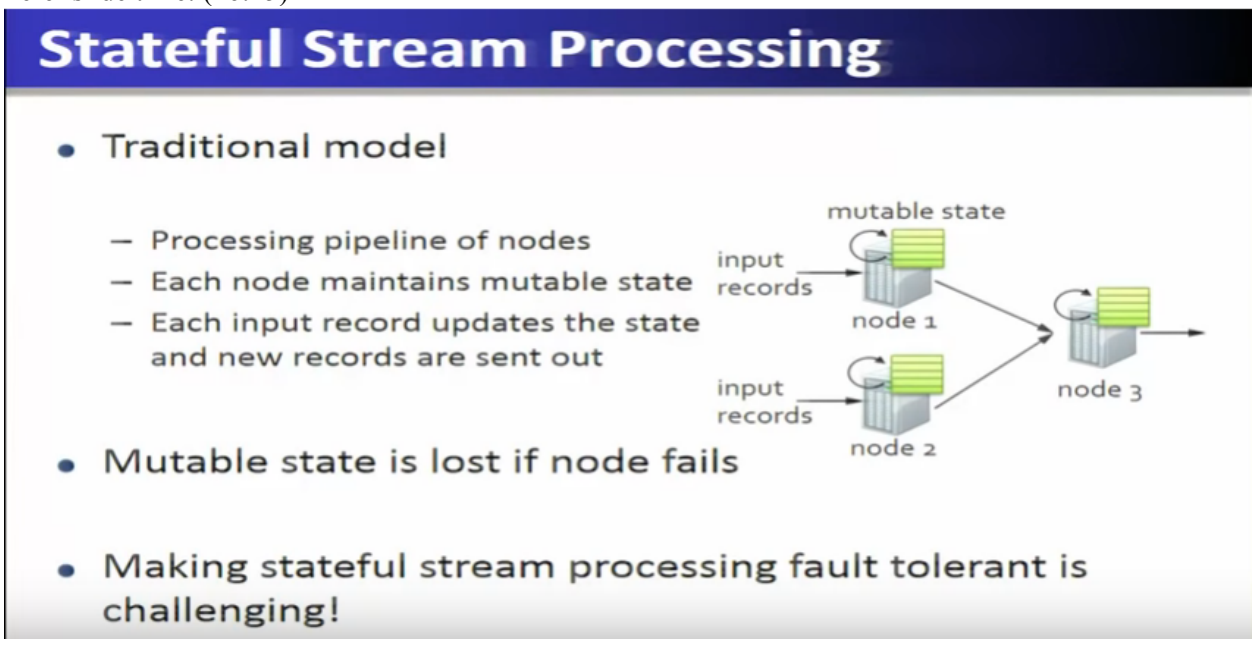
So therefore, there is a need of a framework for big data stream processing that, scales to the hundreds, thousands of the nodes, achieves second scale latency, sufficient to recover from the failure ante grade, with a batch and interactive processing.

Refer slide time: (26:06)



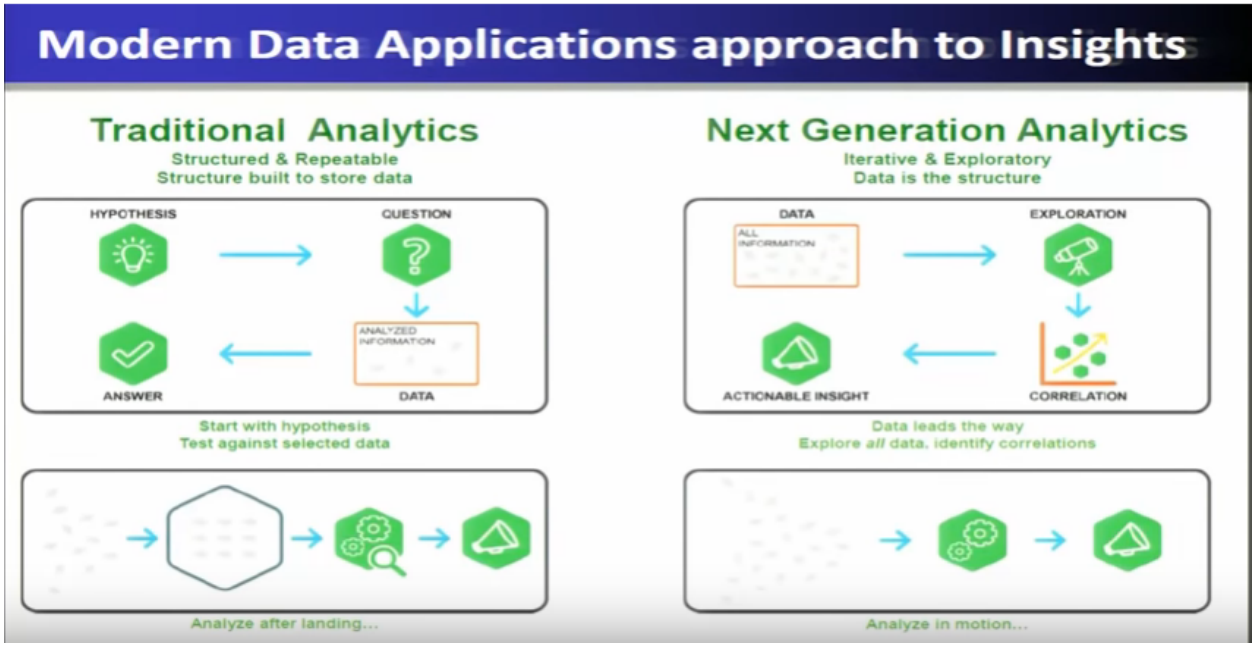
Let us see, how? What are the different features which are handled, which are able to cater all these parts? So, SPARK is tripping features, first is die scaling. So, Spark Streaming, can easily scale to hundreds and thousands, so speed also is a low latency and fault tolerance is achieved, here to recover from the failure, ending it is also, integrated with the real-time and Business Analytics is also supported.

Refer slide time: (26:45)



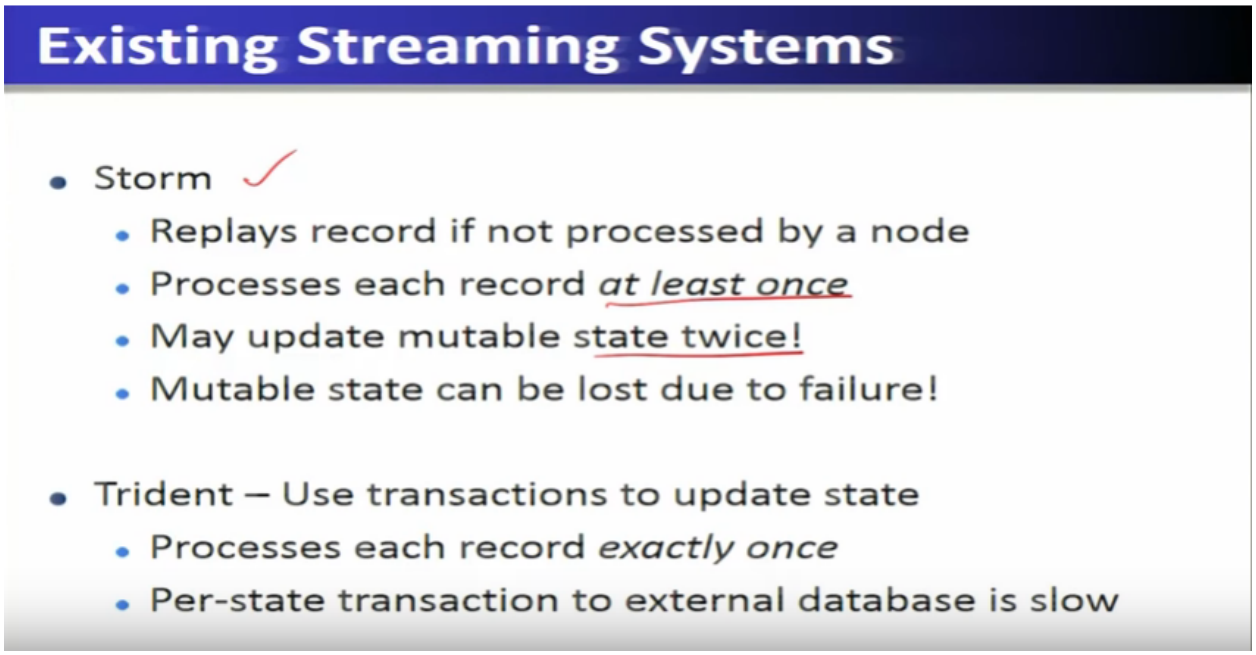
So, let us see, another part, besides the integration with the batch and the batch processing and the real-time Streaming data processing, other than that, we will see another requirement, which is about, stateful stream processing. In the traditional model, as we have seen that, it provides a pipeline and if the mutable state is lost, then it has to handle.

Refer slide time: (27:15)



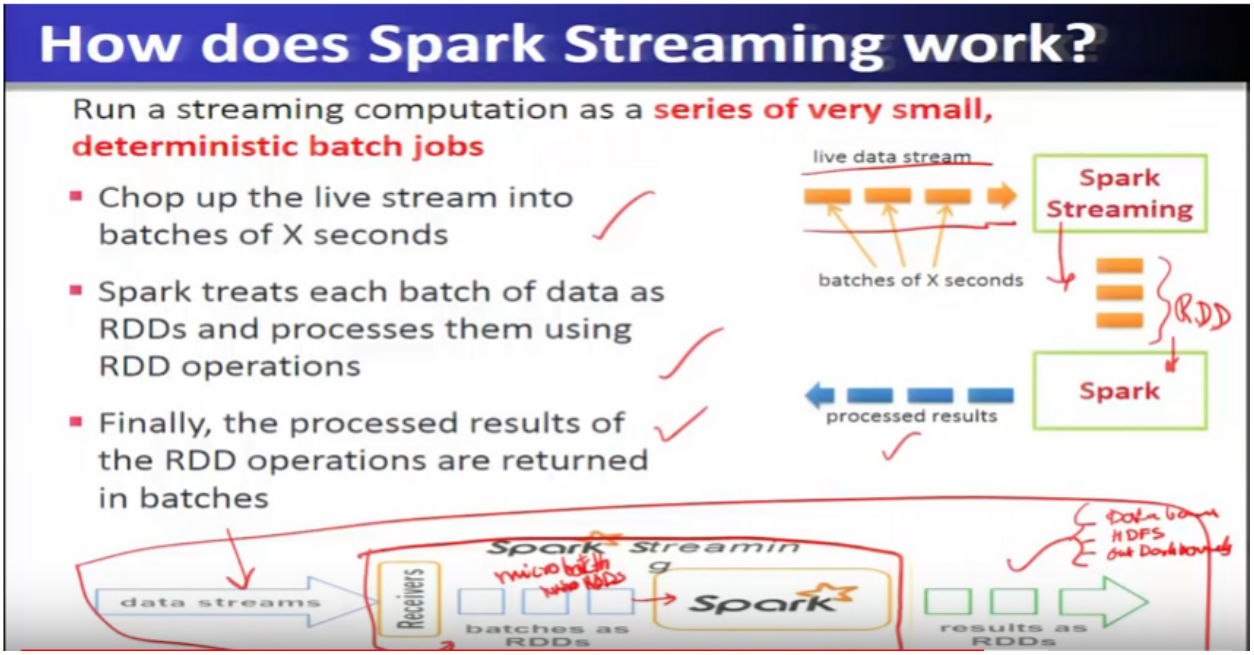
So, let us see, the modern data applications, approaches for, for more insights. So traditional analytics is, basically require, these kind of analysis.

Refer slide time: (27:32)



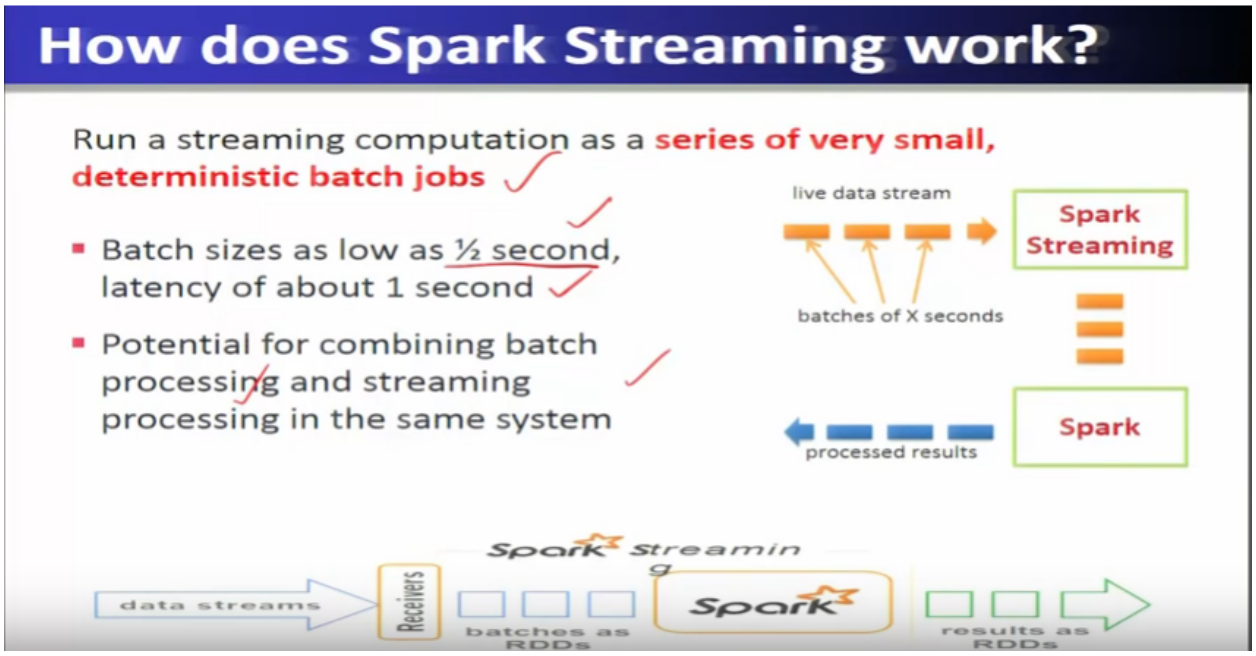
Now, the existing system for the Streaming data analysis is called a, 'Storm'. And it replaced the record, if not processed by the node and therefore, sometimes it provides the, at least ones semantics. So that means, some of the updates, if the mutable states and the nodes are filled, to achieve the updates in the mutable state, so that means, it will update twice, so that becomes a problem in, at least one semantics and the mutable states, can be lost due to the failures. So, this at least once semantics and will create problems in some of the times, where the updates are to be done twice and in the, existing systems, like storm and does achieve only up to that, state of the art, which is called at least once. So, exactly once is the semantics, which is required and it is supported in the SPARK Streaming system. There are other Streaming system, such as Trident, which use the transactions to update the state and there also, has exactly ones, semantics and for estate transaction to, external database is slow.

Refer slide time: (28:58)



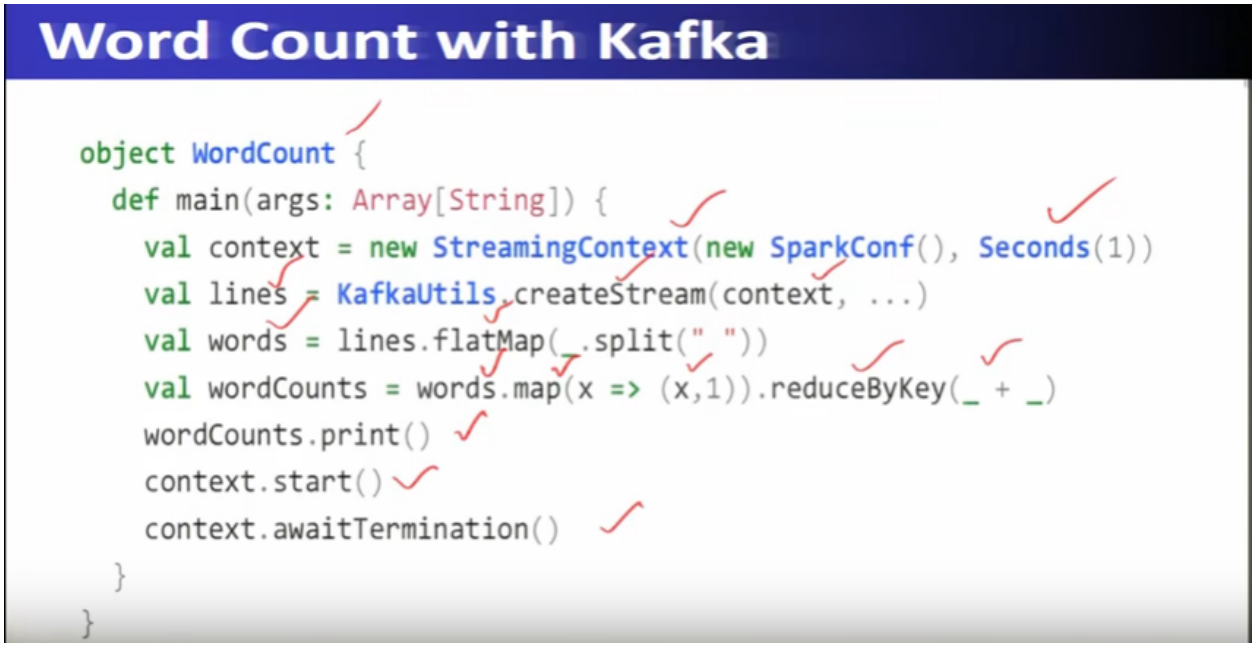
Now, how does this Spark Streaming work? Let us understand this. So, it runs a Streaming computations, as the series of small, deterministic bad jobs and the live streams which are, taken up or which are considered in the system is, to be divided into the X seconds. And the Spark treats each batch of data is, an RDD and process them using, an RDD operation. Finally, the process results of RDD operations are returned in the batches. So, let us see, all these things using the diagram, so here, the live stream data, when it is entered, it will be divided into the batch of, of X seconds. And this particular Spark Streaming system, will now, divide into the batches, I earned this batches in the form of RDD is, will be given to the Spark engine for processing and the process, result will finally be emitted. So, again let us, discuss this entire scenario, which says that, you to run this Spark Streaming system requires, the, the Streaming data to be entered into the system, that is called data streams. Data streams are, received by the receivers of the Spark Streaming system and then, it will divide into the, batches of X seconds and they are called, 'Micro Batches'. Or a, 'Micro RDDs'. So, after dividing into the batches of X seconds, this particular micro batches will be given to the Spark. And Spark reads each batch of data as RDD and process them using the different RDD operations, which are provided by the Spark engine. and finally the process result of the RDD operations are returned, in the form of batches and will be taken care, either they are, to be stored in the database or they are stored in HDFS or they are output on, dashboards. So, there are different ways, this output can be or the result of, this Spark Streaming can be, now used indifferent applications.

Refer slide time: (31:47)



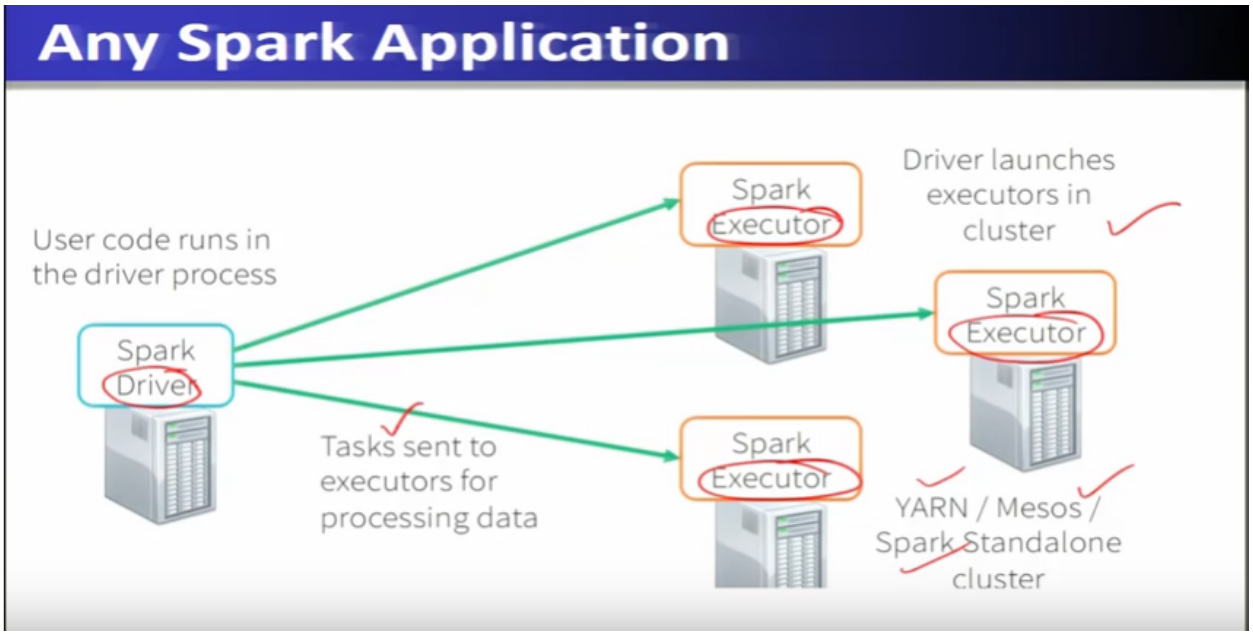
So therefore, we have seen that, to run a Spark Streaming computation, as a series of very small, deterministic bad jobs. And these batch sizes are, as low as half of the second. So it cannot be lesser than, half a second latency, why because this is to beset by the system. So let us see, if let us say batch size is a half of a second, it will finally end-to-end latency, it comes out to be only one second. So therefore, seconds of latency is achieved, due to this batch size of a half a second, size is being computed and processed. so therefore, the potential for combining batch and Streaming data processing in the same system, exists why because, finally these batches are also micro batch, RDD's and which are given to, the SPARK system and the batch data also is processed to the SPARC system, so finally they have the single engine, that is the SPARK engine, which process both the batch ,as well as the Streaming data, of after the processing, of the Spark is Streaming. So therefore, it is possible to combine or integrate both batch processing and Streaming, Streaming processing in the same system.

Refer slide time: (33:12)



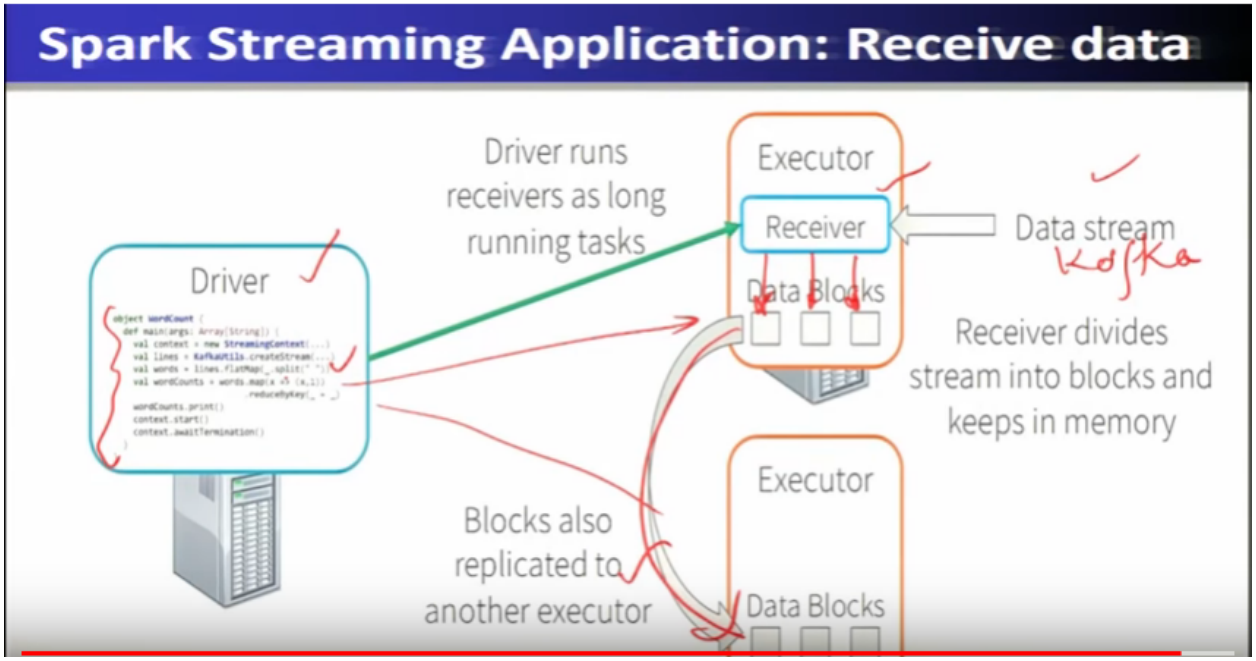
And now, let us see, an example of a word count, with Kafka as the, live input source. And how this is all done, in the Spark Streaming system. So this word count application, we can see, we have to create the Spark Streaming context. and with a, with a one-second, size of batch sizes and the Sparkle and then, this particular context will be bound to the, to the Kafka utilities and which will give, the live stream in the form of lines and this particular line, will be further, processed by the, by the Spark Streaming using flat map. Which will split this line, into the words and this word will now, be applied on a map function. So, map function is doing the transformation, on all the words ,which will be received by the Kafka live stream. And this map function will now perform, the transformation that for every word, it will emit the word and a value 1, which in turn will be taken reduced by key. and for every word, it will now try to count or he will or it will now, do the sum of all such words appearing number of ones, that means and therefore every word will now, have a particular count, which will be printed. And then, we have to do this context start and finally the context of wait, termination also these. So, you see that what count with the Kafka is quite, easy to understand and you can write down the program, address of the things, that is the entire operation of, this stream computation is completely abstracted, from the user.

Refer slide time: (35:12)



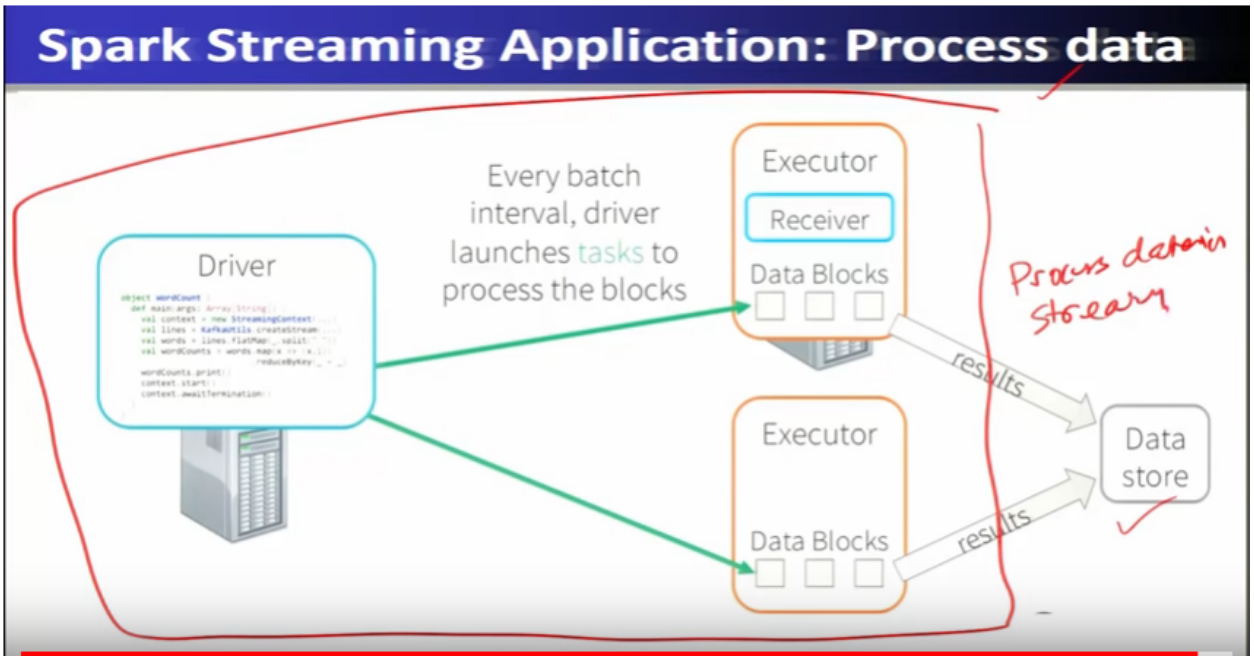
Now, let us see the internal details, of the calf, the Spark stream execution, before that let us understand first the, Spark application. So, in the Spark application that is, a Spark engine, will have the processing in the terms of, the driver program. Which will now, communicate with the executors running on different nodes, on to the cluster systems. And this particular driver, will communicate to the executor and this particular driver will assign the tasks, send to the executor for processing this particular data. Now, this particular driver, launches the executors, in this particular cluster and this can be done, with the help of yarn and maces or Spark stand alone clusters. So, this way the Spark launches using, the driver program and driver and executors' driver will, assign the task to the executors and executors in turn will, perform the execution of the processing of data, in the Spark system.

Refer slide time: (36:25)



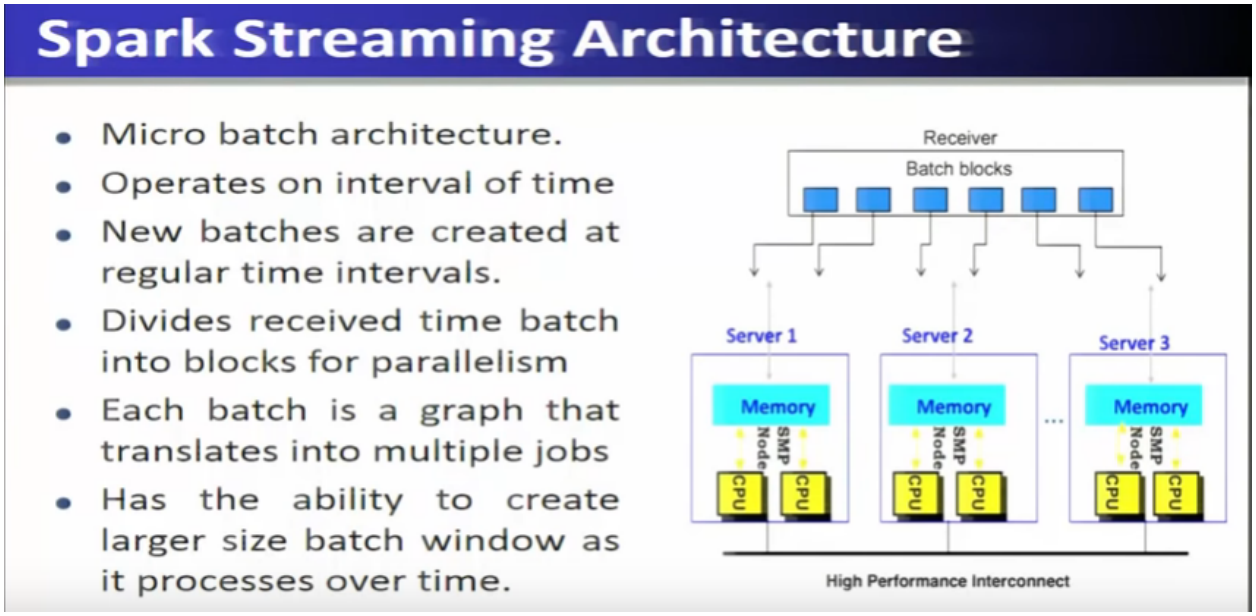
Now in Spark Streaming runs on top of a Spark engine. So let us see, how the Spark Streaming application, will now progress in this scenario, again there will be a driver program. and the same word count program, which we have written is given to the driver and Driver will now, in turn runs, the receiver as long as, as long as, running the task. So, this particular tasks are given to the executor and executors, in turn will have the receiver, which will receive the live data stream from Kafka. So, after receiving the data from, the Kafka now the data will be, means, this particular live stream data, will be received in the form of lines and these lines will be now, done in and that will be in the form of data blocks. Now, then it will assign the task of a flat map, on these executors ,these data blocks are to be replicated, on to the other executors. So, replication also is required, for achieving the fault tolerance ,of these data blocks. Now, further the driver will also assign, the tasks to be performed by the executors, on these data blocks. And for example, we have seen the flat map and performing the, the Map Reduce operation on the word counts, all these tasks will be executed, at the executor. So, so all these are the components of, the SPARK Streaming system. So, here we see that, how it will receive, the data using the receiver.

Refer slide time: (38:12)



And then, we have also seen, now see that, how the data will be processed, so each batch interval the driver launches, the task to process these particular data blocks. And these processing will now, give other results to the, results back and that result will be stored in the, in the data store. So, these parts we have seen and now, we have also seen, internally how the data is to be processed, in the SPARK in the Streaming, in the SPARK Streaming system.

Refer slide time: (38:49)



So let us, review the entire scenario, as the SPARK is slimming architecture. So, it is based on micro batch architecture, operates in the interval of time, whenever a new batch, are created at a regular time

interval, it divides, the received time batch into, the blocks for parallelism. And each batch is draft that, translates into the multiple jobs. And has the ability to create, the larger sized batch window as, it processes over the time.