**Final Exam – Cloud Computing Fall 2019**

**Greeshmika Korrapati, UID: U00932594**

**1.Major Features of Spark and Message Passing Interface**

**i. Features of APACHE SPARK:**

**Lightning Fast Processing:**

Apache Spark is predominantly is a framework that process data parallelly which works with [Hadoop](http://hadoop.apache.org/) making it very easy for developing, Big Data applications merging batches, streaming services, and data analytics. Fast processing is significant feature of [Apache Spark](https://intellipaat.com/blog/what-is-apache-spark/) which influenced the technology world to choose this over other technologies. It is using in-memory computations that is making the process fast. It is developed in such a way for supporting comprehensive applications. In case of Hadoop Map Reduce it needs additional IO space as each map process reads from disk and writes to disk, this applies for reduce process as well. This results in degrading performance, when we consider iterative process such as K means clustering the amount of using MapReduce is high this would lead to excess usage of disc instead stores data in memory. Spark is using a concept known as Resilient distributed datasets (RDD) this stores data in memory and stores data in disc whenever needed. Spark maintains a record known as Directed acyclic graph (DAG) in which changes done on a dataset is recorded. Spark will run application in Hadoop cluster where its speed is hundred times faster in case of memory and ten times faster in case of disk IO.

**Ease of Use:**

Parallel processing data and concurrency of data and distributed data processing are important processes followed in Spark Framework. In case of Hadoop MapReduce users must change code for each operation and this could be more challenging when there are dealing with complex projects. But Spark handles this issue well as it supports multi-languages and has libraries that assist execution with eighty high-level operators with RDD. Developers can develop and run their application in any desired language such as Java, Python, R, Scala and high-level operators are used for interactive querying of data in shell.

**Fault tolerance:**

Spark maintains record of changes performed on dataset in form of linear graph which is helpful in providing fault tolerance. As RDD is immutable dataset, background of every operation performed for creation of fault-tolerant input dataset is remembered by RDD and if RDD is lost due to work node failure then that part of RDD can be performed again from original dataset with help of background of operations. There exist two types of data that must be recovered at time of failure. They are data that is received and replicated on one of node and this data can be recovered at time of failure occurrence. Second type is data is not replicated and it can only be retrieved from source after fault occurs. This process of recovering lost data will enhance performance with reduction of maintaining replicas and updating them. Spark runs over cluster managers such as Yarn which is responsible in tracking health of nodes and if founds that some node is not healthy it immediately assigns job to new node.

**Other Features:**

Spark not only supports Hadoop MapReduce but also supports Machine learning, SQL Queries, Streaming data, Graph algorithms. With all this included spark can perform analytics in advanced way. Spark also supports real-time streaming process and produces results instantly. Spark has compatibility with both versions of Hadoop.

**ii. Features of Message Passing Interface:**

MPI is a message passing library specification which primarily address message passing parallel programming model. It is basically preferred as it is having various implementations in both private and public domain. There much flexibility as it is portable, and it requires very less changes when an application is deployed in another platform that supports MPI. With huge data that is getting generated processing these huge data is of greater demand for deriving insights by analyzing them and for processing this data it requires efficient resources to compute. Developing such effective computing devices will cost more and this results in development of cluster programming. MPI is a protocol through which data can be moved from one process to another process.

**Performance of MPI:**

In a parallel computing programming model, communication with more continuance is a main factor that influence the performance of a system. MPI is used to achieve synchronization in a distributed system. A developer can be able to access the cache and shared memory of various processors in a distributed environment with the help of MPI library and this allows the developer to manage the location of data and increase continuance in message passing. This feature of MPI results in the increasing fame and usage of MPI in parallel computing areas. MPI is highly utilized in High performance computing when compared to big data processing. But, practically the results of experiments conducted show that MPI surpassed spark in executing analysis on twitter data.

**Ease of Use**:

MPI is portable and if an application is designed by using MPI library it can be implemented in various parallel programming architectures where same MPI library is available and it can be executed in various programming languages such a C++ and C, FORTAN which are utilized for high-performance computing. The MPI libraries mostly use the low-level details and make it easy for common binding with any language and adopting parallel programming. MPI Provides a virtual computing model which hides differences in architecture and enables implementation on heterogeneous systems

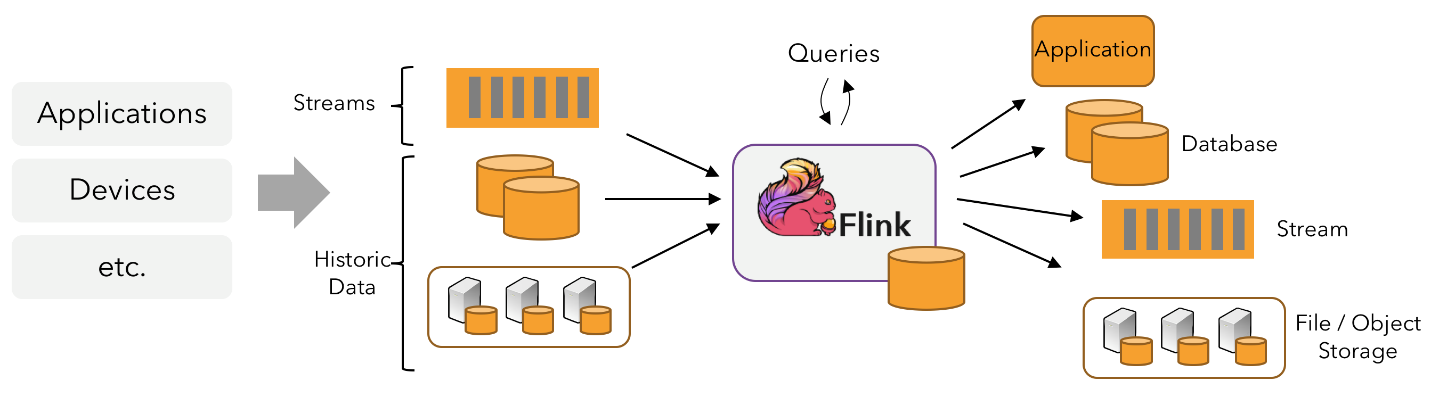
**Fault tolerance:**

Checkpointing is used in MPI for fault tolerance. In is a process the state of the execution is stored to the disk and it can be retrieved at the time of restarting process. This Checkpointing when utilized properly would provide a good fault recovery system. Checkpointing also would be facilitating migration of process, that is it interrupts the implementation of a process on one node and eventually continues its implementation on another node. But, checkpointing a parallel application is difficult when compared with every processor taking checkpoints separately. For avoiding this domino effect and other issues after recovery, the status of inter-process communication must be maintained a track and the global checkpoint have to be maintained consistently.

**2. Differences between Apache Flink and Apache spark streaming and Fault tolerance:**

**Apache Flink Streaming**:

Flink is an open source streaming platform and could be executed using REST API. Stream processing will involve unbounded stream inputs and bounded stream inputs as well. Flink depends on data-based model which means that it processes data as soon as it receives it instead of waiting for other data to segregate them into batches. So, this allows Flink to process millions of data in seconds. It can also integrate with multiple data sources and services such as Apache Cassandra, Elasticsearch, Kafka, RabbitMQ, Hive. It also supports very high -level API such as Map/Reduce, Windows, Sort, joins, Group by, Filters. Flink follow exactly once processing and stateful processing. For few tasks it depends upon previously accumulated tasks so for such events we need state processing, State can be executed locally so this helps Flink application to achieve high efficiency and less downtime.



**Fault tolerance in Flink:**

Fault tolerance in Flink is achieved with help of checkpointing. As part of checkpointing data is backed up by saving eventually to sources such as HDFS. Checkpointing in Flink will save only transformations made but not entire data this will decrease reduces the quantity of data to be stored in HDFS and improves performance. The overhead of checkpointing is almost trivial and this allows developers to retain substantial states within Flink. Flink has high availability setup with the help of zookeeper and this helps for re-generating the task when there is failure occurred because of few errors and crashing of drivers. Flink also enables fault-tolerant and exactly once processing with help of combining of snapshots of states and replaying stream. The snapshots of state capture the complete state of the distributed cluster, capturing balance to the input queues and the state all over the task graph which results from having put away the data updated. During failures, the sources are reversed and state as well as processing are reinstated. As stated above, the state snapshots will be captured serially, without disturbing the uninterrupted processing.

**Working of Spark Streaming:**

As stated by Apache Software Foundation, developing extremely expandable streaming applications with fine capacity and fault-tolerance is streamlined with help of Spark Streaming API. Spark Streaming has the tools for joining streams and for querying the data in streams, these are pivotal for building applications that are interactive and real-time and for this it requires analyzing data real-time. API’s of spark streaming can be combined with various data services such as AWS kinesis, Apache Kafka, HDFS, Twitter. Basically, spark will be partitioning the data streams into time stamped batches and sends this to the spark engine and here analyzing and querying of data is done. Spark Streaming utilizes a process known as Discretized Stream processing. Spark Engine will be facilitating the utilization of high-level changes such as Map, Filter, Flat Map and tasks such as count, reduce and so on. These tasks are implemented on input data which are segregated as batches and they are delivered in order of batches. Spark runs on cluster managers such as Yarn. Yarn handles jobs that are interrupted due to failure of nodes by immediately assigning such jobs to another worker node. Spark streaming will be reading data from a stream, if a data file that has records with no structured relationships is provides for streaming to the Spark Stream, then it splits the input data into bits and transforms it into Discretized Streams.



**Fault tolerance of Spark:**

• Spark streaming develops a lineage graph known as Directed acyclic graph (DAG) which has track of changes applied on dataset and this is useful for fault tolerance in spark.

• Spark Streaming uses a protocol in which processing of each data record is done only once and if there is any failure of this process then it can only retrieve data only from source, this enables each and every record in dataset process without failure.

• In Spark, as RDD’s will be read from fault-tolerant systems such as HDFS and S3, the designed RDD will also be considered as fault tolerant. Although in Spark Streaming as data is read from various types of data, the data is buffered by API on a worker node and RDD’s are created.

• For some reason data of RDD’s are lost they can be computed again with help of lineage graph but if worker node fails and data is not stored in this case still, we can retrieve data from source.

• Sliding window approach is used for recollecting data from source, every worker node of spark will maintain track for entire duration of a window when retrieving data from the stream. When it is identified that particular window’s data is lost, then instantly it can be retrieved from source.

**3.Phishing and spamming in AWS EC2:**

**Identification of the problem in case of EC2:**

Phishing and spamming attacks are arduous for identifying them to impede them, if spammer choose cloud services such as Amazon Ec2 then his presence could be in any geographical place virtually during attack. If an area is occluded, then malicious user could turn around a new instance to evade blocking. Such phishing and spamming could not be recognized by tracking congestion from some specified ports. If spammer is using Amazon's email, then there is a chance that these spams might be restricted to some extent.

**Prevention:**

There are many analyses going on to discover new techniques for avoiding phishing and spamming as these are considered as biggest cyber security threats. Many companies are depending on restricting spam by using filters and alerting users to report any spam such that they can block these malicious spammers, but this is not avoiding or controlling spams completely. So, some companies such as Amazon, Google, and so on started implementing verifying users, authenticating emails and utilizing advanced machine learning techniques.

**Phishing and Spamming Detection by Amazon:**

Amazon's email service facilitates an authenticating technique by complying with Domain Based. Message Authentication Reporting and Conformance. So DMARC policy can be set up with SPF or DKIM on domains, this means that email must satisfy condition set up with these policies. Policies include that a valid DKIM signature should be present, domain from which email is being sent should contain this sender email in the list as well as recipient in the domain list. DMARC record will include DNS records as well. In addition to this Amazon SES has few other restrictions for users in their SES Sandbox such as sending emails are limited to 200 per 24 hours, email sending rate is maximized to 1 per second. It also has restrictions on size of message, attachments and headers. All recipient addresses are to be verified from, source, return paths, sender is to be verified. In sandbox AWS user can send emails within SES and it has a special form to be filled to get access for sending emails to outsource.

**Reputation Maintenance of Sender by AWS:**

Amazon SES console contains a reputation dashboard tab where entire data of mail sender's reputation can be viewed. It consists of all details regarding sending if there are any complaints raised against his mails and if some of his mails are bounced all these mounts for reputation of sender. Sender will have few limitations to send emails related to promotions or marketing by default and if it is needed then a request is to be raised and amazon would consider request by doing thorough background verifications and considering their reputation. This reputation would get effected if an email is restricted by some spam traps or email message contains links or reference for a blacklisted domain. Amazon has CloudWatch and this sends notifications when sender's reputation is worsening and if it is reaching thresholds. It permits to handle issues instantly that are causing harm for sender reputation.

**Methods that could be implemented:**

This could mainly include creating awareness about cyber security threats such as phishing and spamming to all users while they sign up their accounts and to encourage them to report any spam mails, use some spam filters to avoid such spams. More restrictions could be inculcated for senders in such a way that they could send emails to receipts only when receipts accept senders as safe sender. Though some privacy policy restricts they should be a technique to proofread email to some extent and immediately block the sender if some malicious content or reference of block listed domains is found in email. Enabling a technique to track accounts with help of IP address of instance to track activities of user with help of his network congestion such that if some malicious activities are tracked then account could be blocked instantly.

**4.Design patterns of Cloud:**

**Cloud Intensive Process:**

In this technique very less resources of mobiles are used, and cloud is utilized for rest of flow. Except capture image other steps pre-process image, extract features from images, run feature matching algorithm, retrieve information of the most matched persons from the person database, present the information to the user is performed using cloud. This decreases the utilization of capacity and storage of mobiles and helps to transfer such advanced qualities to mobiles with finite resources. This technique was most popularly embraced proposal for some period, but it started transposing in contemporary days.

Pros:

Keep up extensively with extensive gadgets with finite resources, it is extremely expandable, it is very simple to operate over various platforms since except capture image other steps are performed using cloud. the storage price of cloud is very affordable and there are no limitations when compared with storage in mobiles this be more beneficial. The Capacity of processing data is very high when compared to mobile.

Cons:

In terms of communication costs this is expensive technique as capturing of image is done in mobile that should be uploaded to cloud and it might take considerable amount of time and it varies with size of image as well and later other stages are performed in cloud which would result in some delay. In addition to this in cloud storage is done with help of network file system and this would add more delay time because of its read/ write operations on disc. Most of operations depends upon internet connectivity in this technique and response time depends on bandwidth of network and speed of data transmission.

**Cloud-Mobile Mix process:**

In this contemporary day’s technology is so updated that mobiles are upgraded with better processing capacity and improved scalable storage. Also advanced Integrated chips and other data processing units are arranged as in built to enable advanced features process impeccably on smartphones instead of association with external resources such as cloud it can produce ample outcome. So, this helps user and developer to protect their sensitive data such as their passwords, biometrics on public clouds which could lead to some malicious activities. when we consider this face recognition application by using cloud mobile mix (hybrid) we can achieve doing some part of flow such as capturing image, pre- processing image and extracting features could be performed on mobile and other parts such as running feature matching algorithm and retrieving information of matched persons from person database could be done on cloud and result of this could be displayed in mobile.

Pros:

With respect to communication costs it is better when compared to previous approach as some part of workflow is performed on mobile. Even storage costs would be minimal as feature matching algorithm and retrieving data is performed on data, so database is kept in cloud, so this does not use most of mobile's storage. Response time would be improved when compared to previous method. Private data such as biometrics of user will be securely stored in his mobile. Even there is decrease in network bandwidth and consuming of data.

Cons:

Even now it depends on connectivity of data as part of workflow is performed on cloud. This technique could work only for mobiles that supports all necessary computing resources. Notable amount of power and storage of mobile would be utilized.

**Mobile Intensive Model:**

In this process mobile plays a pivotal role as every step of workflow capturing image, pre-processing image, extract features from images, run feature matching algorithm, retrieving data from person database, showing results to user on mobile is performed on mobile. So, this requires mobile to be very productive with all necessary computing resources also would require more storage and processing features included.

Pros:

This technique will have minimal or no communication costs as every task is performed on mobile and its response time will be improved when compared to other techniques but should take care of processing setup as it requires advanced features to handle workload. Provides more security for user’s data as everything is in his mobile. Does not require connectivity of data.

Cons:

Requires more storage as all information of person is stored in mobile and this could be made better if we can use cloud for just storing data of user. This could be achieved only smartphones with advanced features and extensive processing features.

**5.Handling Limited Memory Issue in SGX:**

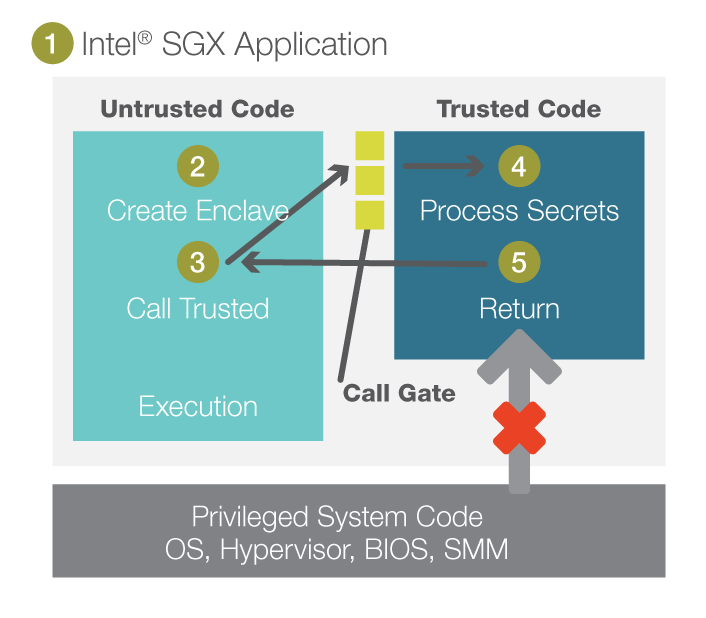
As enclave in SGX is trusted area where execution of hardware is backed up and this is where we implement our code so our code should use less memory, in addition to this each enclave is executed once in SGX and memory allocated to it can only utilized by it due to these reasons memory is limited to 128 MB by Intel. However, SGX has capability of dynamically adding and removing pages from an enclave. Exceeding memory demand would lead for re-encrypting pages and copying them to memory of regular system this would impact performance. To process 1 GB of data it should be loaded into enclave as smaller blocks.

**Handling the integrity of the part of your program running in untrusted area:**

Integrity of part of program running in untrusted area can be handled by SGX with help of attestation. The enclave has a secure coprocessor, known as Trusted Platform Module (TPM), this helps in performing encrypting and decrypting of data and other operations like generating keys, storage of keys in a protected environment. TPM also performs attesting the state of platform and configuring to various parties that are interested. The party which is assessing is known as challenger and party that is assessed is known as attester. Depending on the attestation, the challenger will be able to decide either to trust the platform or not and for this it uses some reference values and compare with attester’s value. There are two kinds of attestation performed in SQX they are: local attestation and remote attestation.

Local attestation permits both attester and challenger exist on same platform to attest each other. The enclaves will perform this attestation by sending a local message so that both parties will be able to exchange to assess their identities as they are on same platform. Only Local attestation is not sufficient for deciding either platform is trustworthy or not for this we also need remote attestation where the challenger would be on different platform.

In our case if we consider our word count program in single CPU then just local attestation is required. So as discussed before in this type of attestation enclave sends local message and this is exchanged between challenger and attested enclaves and these two will get some encrypted keys as well as their reports from system and they analyze the results by comparing with reference value and decide whether they are trust worthy to each other. As discussed in lecture slides if we consider two enclaves’ w, c measurement of both is done by CPU. Initially for generating key c’s enclave measurement is needed. If c sends a message to w, then w requests for measurement report of c to CPU. Then CPU will fill the report and create Mac with help of report key and sends to w. Then it is sent to c from w. c requests CPU to send key for report verification, then c verifies MAC that is received along with report and trusts w if it matches. When we consider that our program runs on two different platforms then we also require remote attestation to be done.



**Designing Word Count Program on SGX:**

We use trusted Execution Environments in SGX to design a program where there exists application data, code, Enclave, OS and end client. Data is basically sent to Enclave and we can access enclaves only with help of its enclave code in addition to this we can also access application’s secrets with this code. As we discussed earlier Enclave should be minimal, so we load data as smaller blocks to Enclave page which has dynamic memory as we are designing 1GB word count program. Next part would be handling attestation in our case if we consider that we are using single CPU then local attestation is enough for building trust between trusted enclave and untrusted area, after establishing trust between enclaves sealing process is started that is encrypting data, there are two ways of encrypting data one is enclave identity and other method is sealing identity. By using either of methods data can be supplied to Main Enclave where word count program code exits and as data is sent in streaming method it runs code and send backs results and this is stored in disk and every time as streaming of data is done it updates the result and send to output file according to word count code (Mapper/Reducer).

**RESOURCES:**

<https://dzone.com/articles/6-sparkling-features-apache>

<https://intellipaat.com/blog/tutorial/spark-tutorial/spark-features/>

<https://data-flair.training/blogs/apache-spark-features/>

<https://towardsdatascience.com/a-beginners-guide-to-apache-spark-ff301cb4cd92>

<https://databricks.com/spark/about>

<https://en.wikipedia.org/wiki/Message_Passing_Interface>

<https://computing.llnl.gov/tutorials/mpi/>

<https://www.mcs.anl.gov/research/projects/mpi/tutorial/gropp/node11.html#Node11>

Checkpointing and Migration of parallel processes based on Message Passing Interface Zhang Youhui, Wang Dongsheng, Zheng Weimin Department of Computer Science, Tsinghua University, China.

<https://spark.apache.org/research.html>

<https://www.mcs.anl.gov/~lusk/papers/fault-tolerance.pdf>

<https://spark.apache.org/docs/2.2.0/streaming-programming-guide.html>

<https://hackernoon.com/what-makes-apache-flink-the-best-choice-for-streaming-applications-fc377858a53>

<https://training.ververica.com/intro/intro-1.html>

<https://docs.aws.amazon.com/ses/latest/DeveloperGuide/limits.html>

<https://docs.aws.amazon.com/ses/latest/DeveloperGuide/request-production-access.html>

<https://docs.aws.amazon.com/ses/latest/DeveloperGuide/dmarc.html>

<https://aws.amazon.com/ses/reputationmanagement/>

<https://www.ibm.com/cloud/learn/what-is-mobile-cloud-computing>

<https://software.intel.com/en-us/forums/intel-software-guard-extensions-intel-sgx/topic/737218>

<https://link.springer.com/chapter/10.1007/978-3-319-59665-5_13>

<https://arxiv.org/ftp/arxiv/papers/1801/1801.05863.pdf>

<http://www.sgx101.com/portfolio/application_design/>