Group-34
Gautam Kumar Singh
Harsh Singhal

CS 838 Assignment-3 Part-B

(1)Application's completion time

	Application's completion time (minutes)
Spark	10.17
GraphX	2.59

GraphX perform much better than Spark because programming abstraction provided by GraphX are optimized for iterative graph algorithms.

(2)Network and Storage Read/ Write Bandwidth used during the application lifetimes

	Storage Read(GB)	Storage Write(GB)	Network Receive(GB)	Network Transmit(GB)
Spark	0.100	23.088	30.931	30.159
GraphX	1.348	4.640	3.140	3.297

GraphX provide mechanism such as Vertex Mirroring , Multicast joins , variable Integer Encoding and Automatic join elimination due to which Network bandwidth usage is very low in GraphX. Low storage write overhead in case of GraphX seems to be due memory based shuffle and some other optimizations in GraphX.

(3)Number of tasks for every execution.

	Number of tasks for every execution.
Spark	2060
GraphX	560

(4)Does GraphX provide additional benefits while implementing the PageRank algorithm? Explain and reason out the difference in performance, if any.

Implementation of PageRank algorithms using graph abstraction (eg graph operators) provided by GraphX is easier than using general purpose dataflow operators, which requires complex joins to implement iterative graph algorithms.

GraphX PageRank implementation outperforms Spark implementation of PageRank due to following possible reasons :-

(i) GraphX abstraction and graph operators are optimized for computational patterns in iterative graph algorithms .

(ii) Index reuse may reduces per-iteration runtime of PageRank algorithm, as graph operators try to maximize index reuse.

(iii) GraphX exploits immutability of RDD to reduce memory overhead and thus improve system performance.

(iv) Lesser communication overhead due to Vertex Mirroring , Multicast joins , variable Integer Encoding and Automatic join elimination . Network Read/Write data in table above shows very less communication overhead in GraphX framework.

(v) Memory based shuffle in GraphX reduces write overhead and increases performances. Data obtained from from disk-stats shows only 4.64 GB of storage write in GraphX as compared to 23 GB for Spark.

(vi) The triplets operator in GraphX returns the triplets view of of the graph. If a triplet view already exists , the previous triplets are incrementally maintained to avoid a full join. This join optimization on triplet view reduces communication and computation significantly.

Moreover, general-purpose join and aggregation strategies defined in Spark are not optimized for iterative graph algorithms. Whereas, GraphX recasts graph-specific optimizations as distributed join optimizations and materialized view maintenance.