BACKGROUND

Iterative Algorithms
- Common in the fields of machine learning, and data mining.
- Repeatedly read (iterate) over the input dataset, applying one or more functions to improve, or add to, the final solution.
- Terminate (converge) on some condition, e.g. once a maximum number of iterations is reached, or when further iterations would yield little or no improvement to the final solution.

BigData Frameworks
- Provide tools to implement and execute complex algorithms in parallel on clusters of networked machines.
- Programmers do not have to manage the parallel execution of their algorithm on the cluster, or faults with individual machines.

MapReduce
- Programming model breaks computation into map and reduce phases.
- Designed for batch processing, not iterations.

PROBLEM AND MOTIVATION

Iterations with MapReduce
- Iterations can be implemented by chaining MapReduce jobs together. However, this can be inefficient.

Methodology

Select two new big data frameworks – Spark and Flink – in addition to Hadoop MapReduce.
- Implement two popular iterative machine learning algorithms; the K-Means clustering algorithm and the Apriori frequent item set mining algorithm.
- Compare how each framework supports the implementation of these algorithms.
- Investigate each framework’s runtime performance and fault tolerance features in a cluster environment.

CONTRIBUTIONS

Evaluation of MapReduce and these new big data frameworks. Comparison of their support for; sharing state, convergence, execution, and fault tolerance.
- Recommendation guide examining a range of criteria such as; understandability, usability, practicality, and performance.

IMPLEMENTATIONS

Requirements and Support
- State – is shared state available throughout the algorithms execution.
- Convergence – does the framework provide mechanism to manage convergence, or must it be custom built.
- Execution – is the algorithm run at an iteration level where the programmer must collect information from each iteration before starting the next, or does the framework execute the entire algorithm and handle iterations as part of its API.
- Fault Tolerance – is the framework able to recover the iterative algorithm’s state.

EXPERIMENT CONFIGURATION

Nodes
- Amazon EC2 M3.large [1]
- 2 x vCPU, Intel Xeon E5-2670 (2.5-2.6GHZ)
- 7.5GB Memory
- 1 x 32GB SSD, 1 x 10GB EBS SSD

Cluster
- Hadoop YARN 2.2.0
- K-Means experiment ran on two clusters using 50 and 100 nodes respectively.
- Apriori ran on a 10 node cluster.

Data
- Synthetic transactions 2.5GB. Average transaction length of five.
- Synthetic two dimensional points. 50GB (25 centroids). 100GB (50 centroids).
- Fixed set of initial centroids defined to ensure same execution across all runs.

PERFORMANCE RESULTS

K-Means
- Flink consistently performed best.
- Spark faster than Hadoop on 50GB, 50 nodes. Slower on 100GB, 100 nodes.

REFERENCES