Competitive Assessment of Scala vs. PySpark

Max Moro

Nikhil Gupta

Distributed Computing

The Need	The Solution	The Evolution
As the size of data increases, it needs to be stored and computed over distributed systems	**MapReduce and Hadoop** Uses a network of many computers for massive amounts of data and computation	**Spark (Apache)** Faster, Distributed, in Memory

Spark

Latest Technology

Quickly and easily handle Big Data

Flexible Alternative to MapReduce

Works also with AWS, Cassandra, and more

100x faster then MapReduce

Keeps data in memory (vs. hard disk for MapReduce)

Distributed data collection

Scala

General purpose programming Language

- From Object-Oriented to Functional Programming
- Java VM

Overcome Java's shortcomings

- No boilerplates
- Easy to code

Used for

- Web App
- Data Streaming
- Parallel Processing
- Analysis using Spark

PySpark

What?

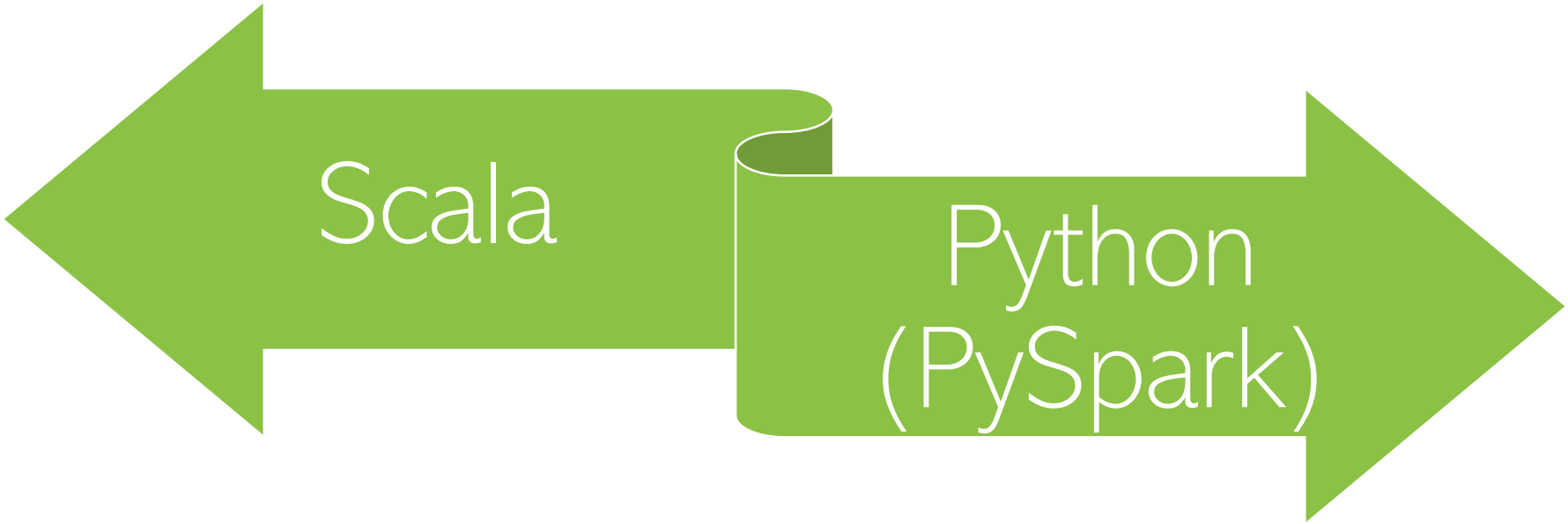
- Spark API's for Python programming language

Why?

- Python is the most popular language for machine learning
- Allows easier interfacing to existing applications built in Python

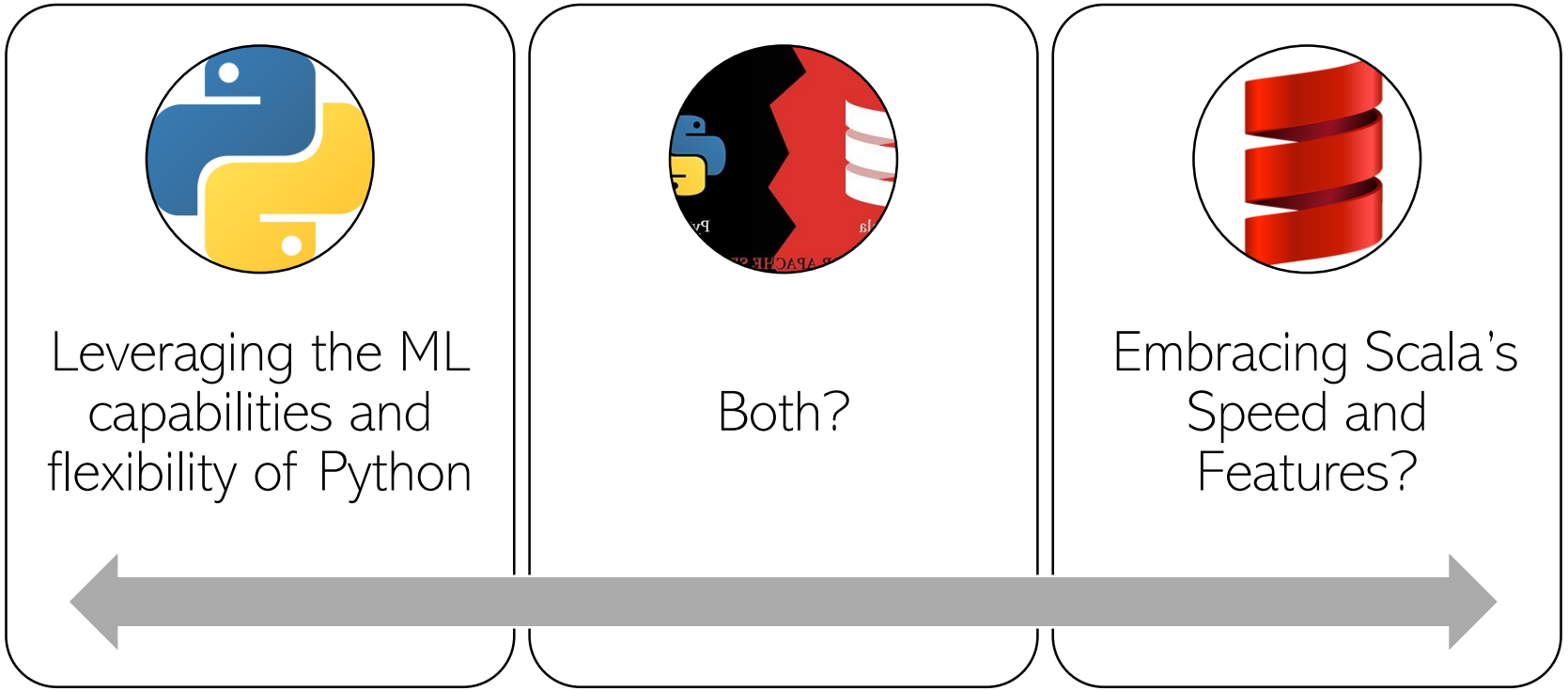
Project Goal 1

Which one is Faster?



Project Goal 2

Should a company invest in a new environment like Scala?



Design of Experiment

Data Generation

Random Data Generator in Python

Query Execution

Executed in Random order in each run.

Blocking

By Machine

By Setup

By Data Size

By Operation

Design Metrics

Query Execution Time

Avoid Human Error

Automated Timing Code

Logging Capability

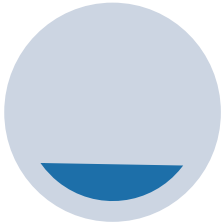
Throughput

Does performance change with data size?

Execution Time normalized for data size

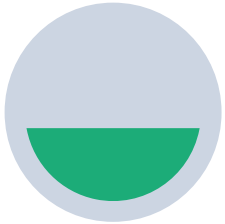
MB processed / second

Datasets - Rows



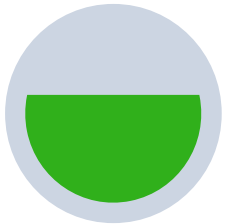
10MBs

10K rows



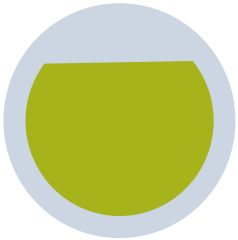
100MBytes

100K rows



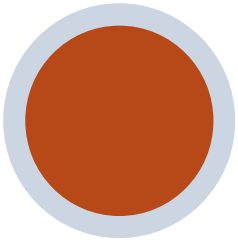
200MBytes

200K rows



300MBytes

300k rows



500MBytes

500k rows

Datasets - Columns

Integers

- 20 Columns
- Random positive and negative numbers

Floating Point

- 20 columns
- Random positive and negative numbers

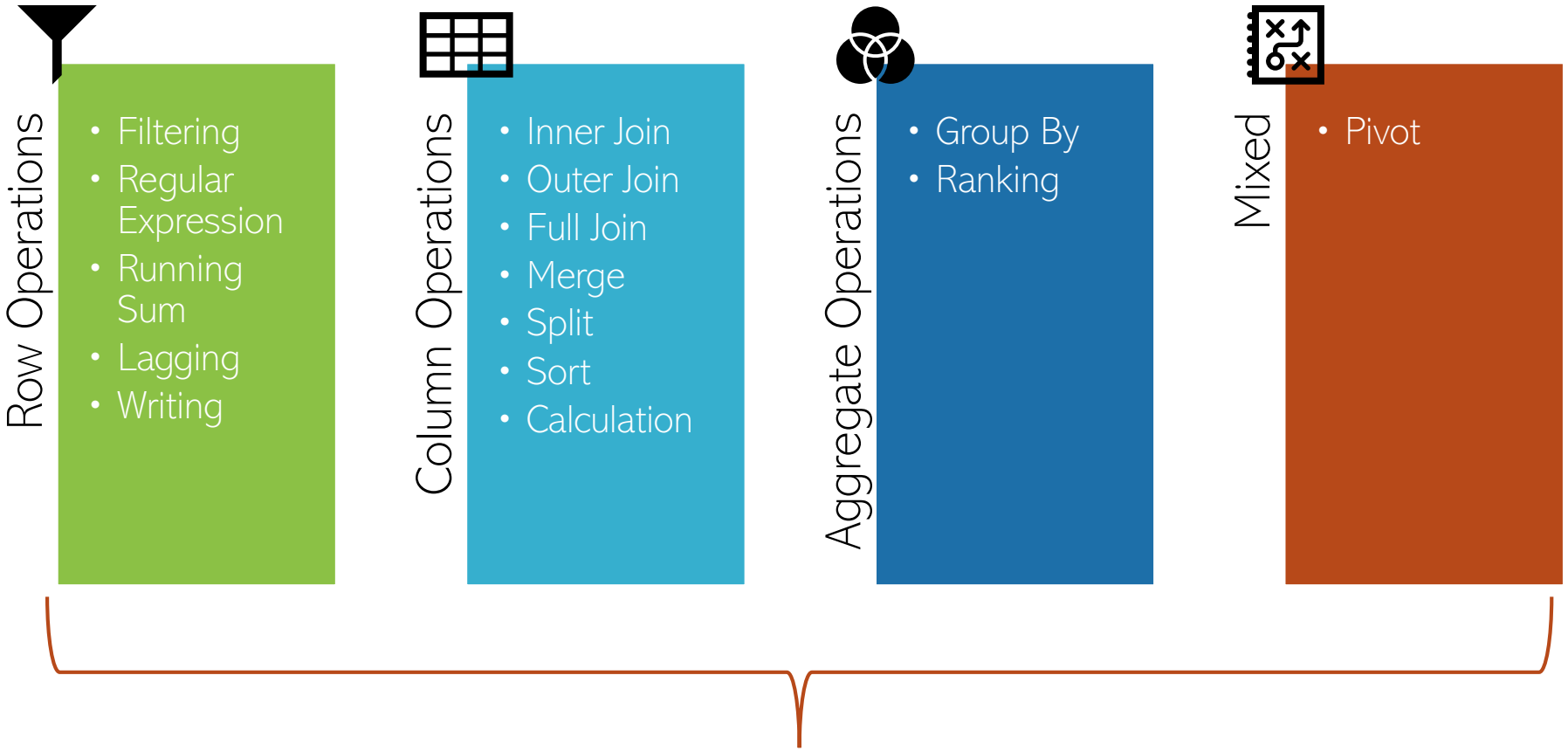
Characters

- 10 columns
- From 5 to 10 words per sentence
- Randomly Generated from a 400k word dictionary

Groups

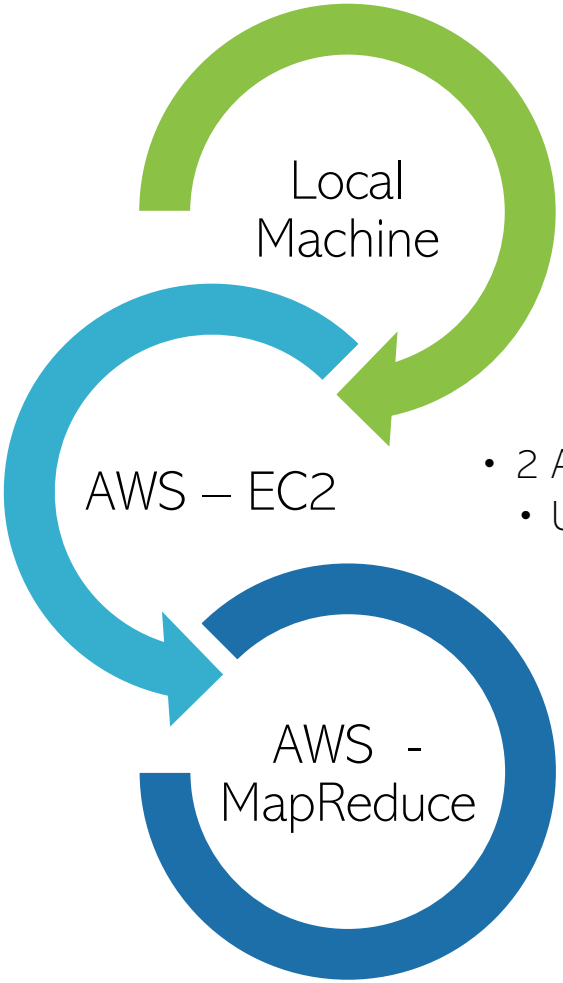
- 10 columns
- From 1k to 50k groups per dataset
- 2 to 3 word identifiers (factors) used for grouping
- Randomly Generated from a 400k word dictionary

Queries



6 runs per each Dataset

Environments



- 2 Local Virtual Machines
- 2 AWS Virtual Machines
  - Using Databricks
- Virtual Cluster Setup
  - Using Databricks
  - 1 master and 3 slaves

Data Points

We completed a total of **300** tests, with 33 queries each, obtaining **9,900** data points used to complete the analysis.

* Cluster environment analysis was performed on only 1 machine

Statistical Analysis

Memory Flush

Warm Cache Runs

Discard first run

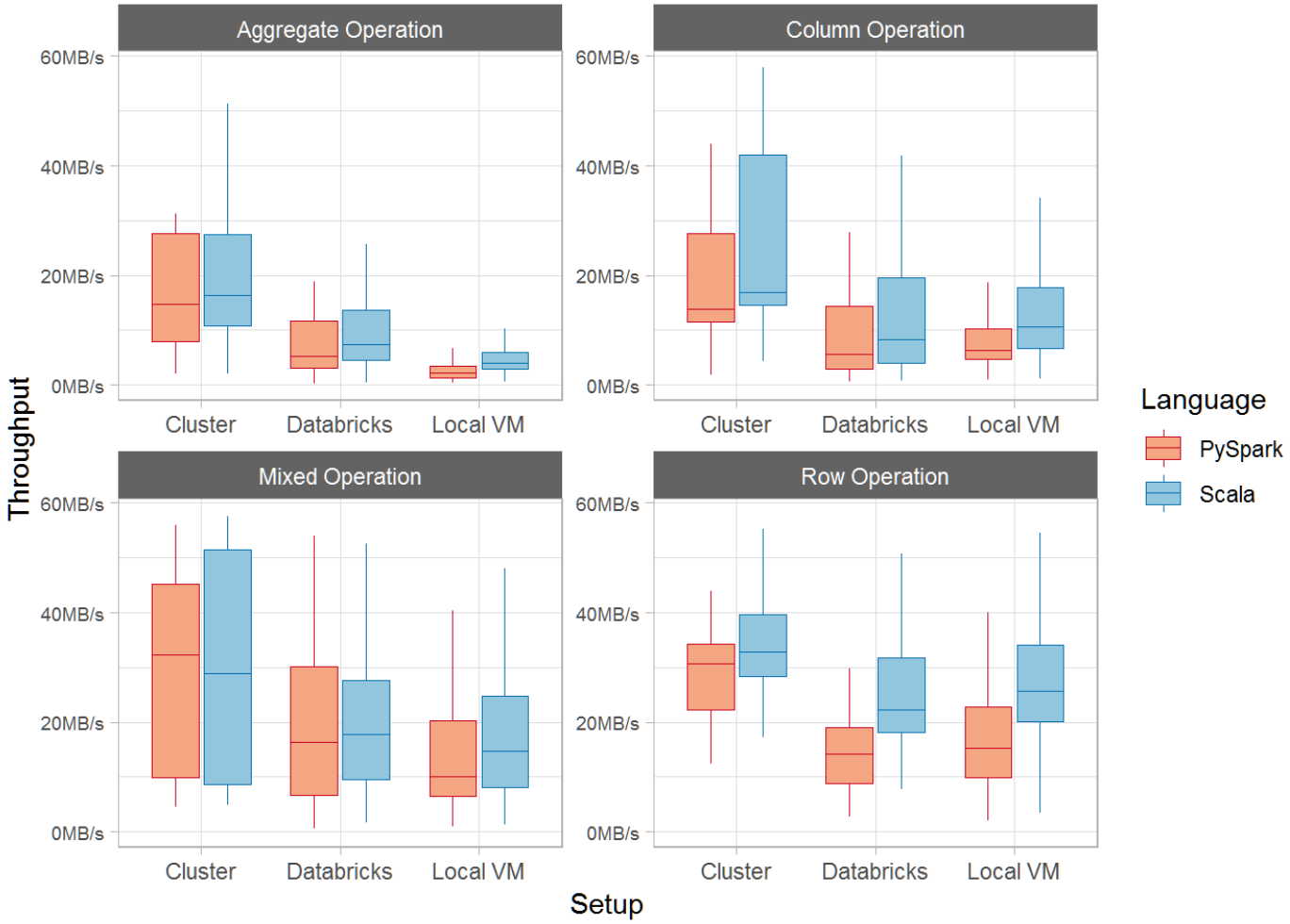
Statistical Analysis

Right Skewed Data

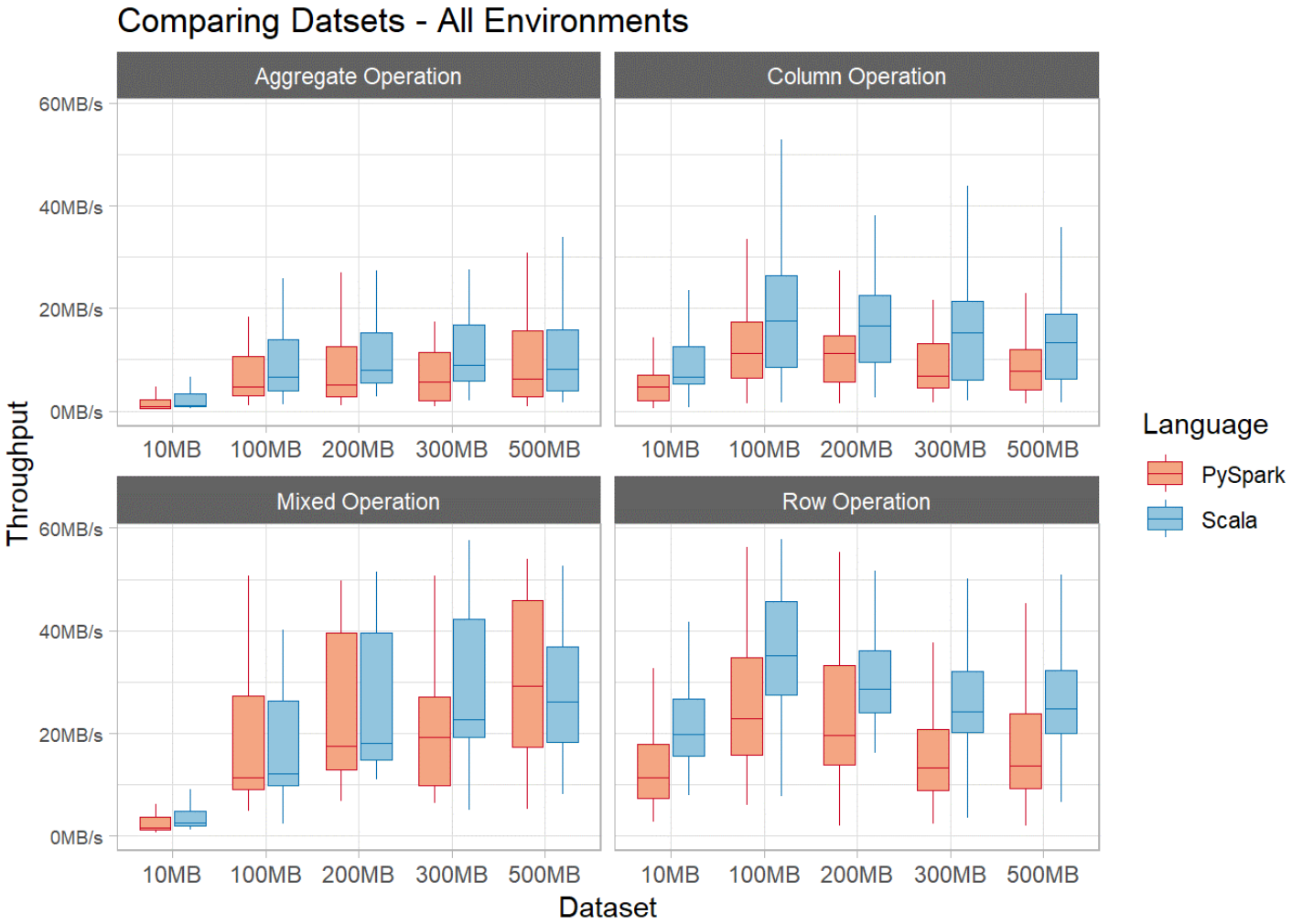
Unequal Variance

Wilcoxon Rank Sum Test

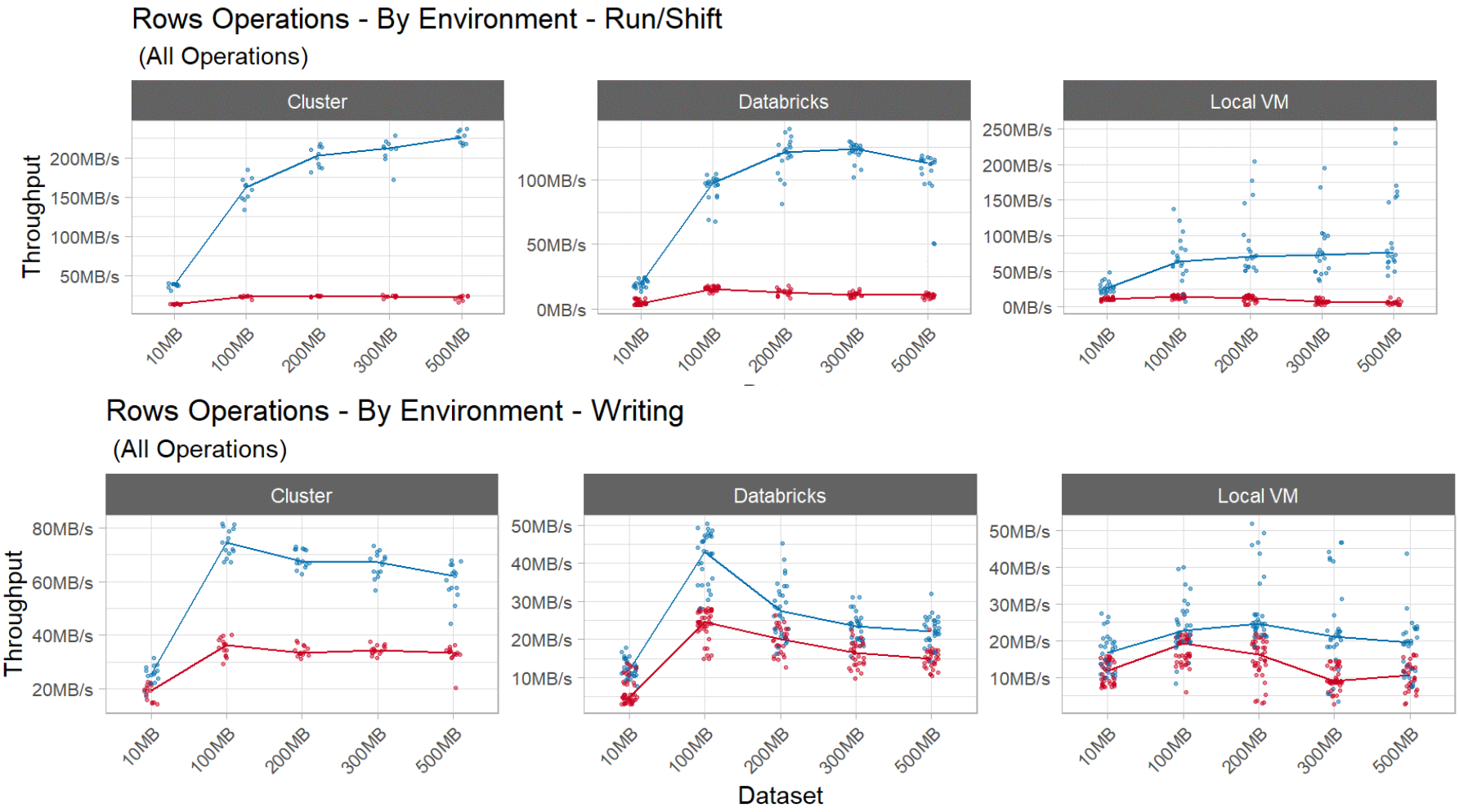
Results – Environments



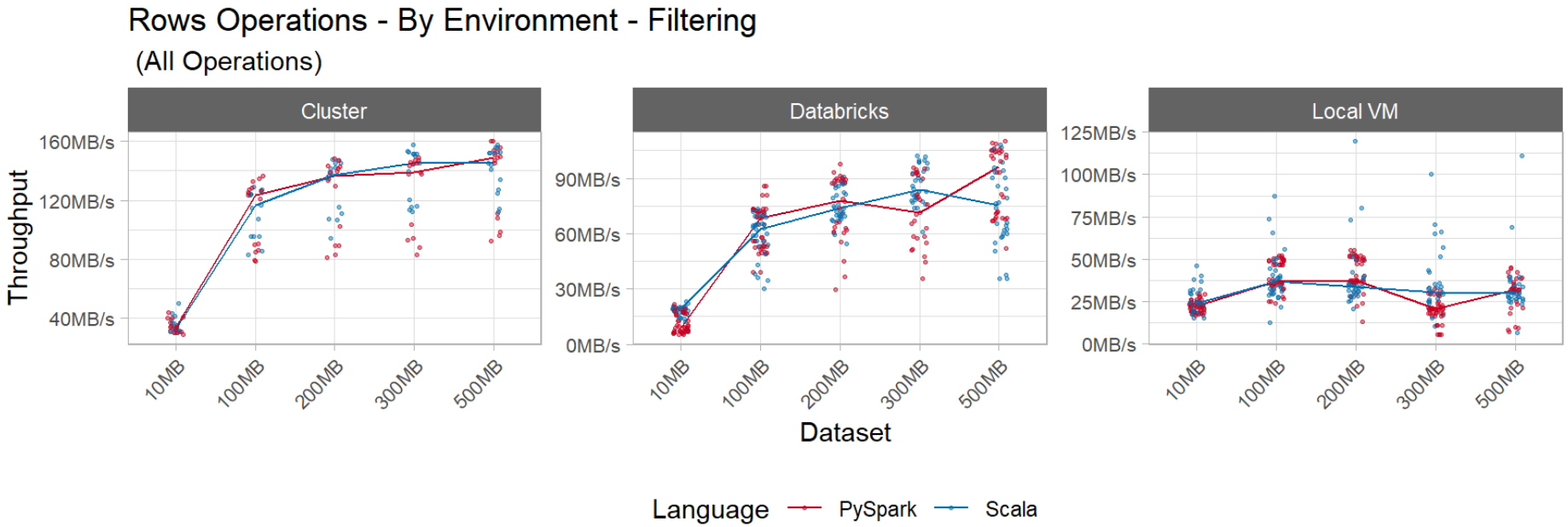
Results – Datasets



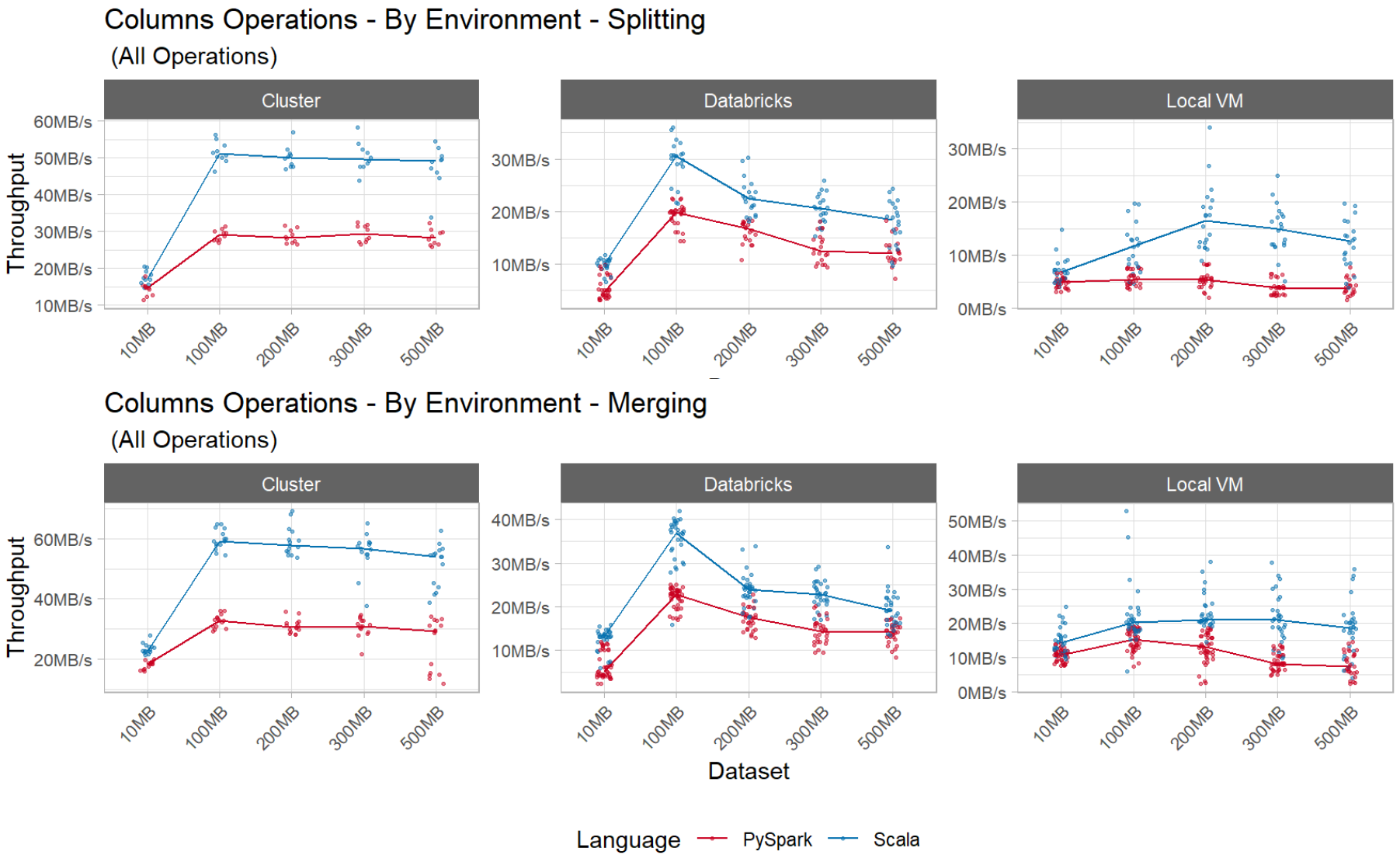
Results – Row Operations



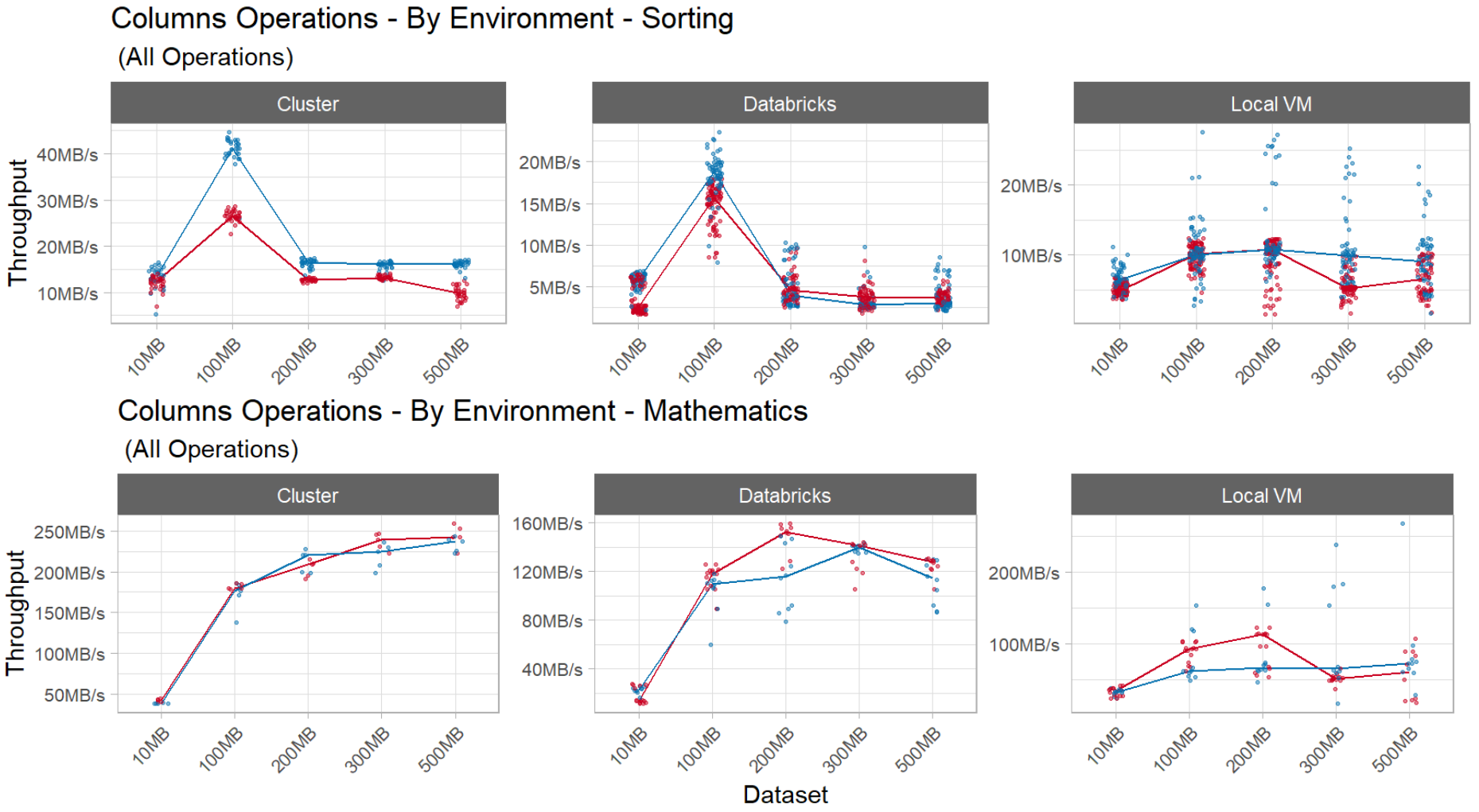
Results – Row Operations



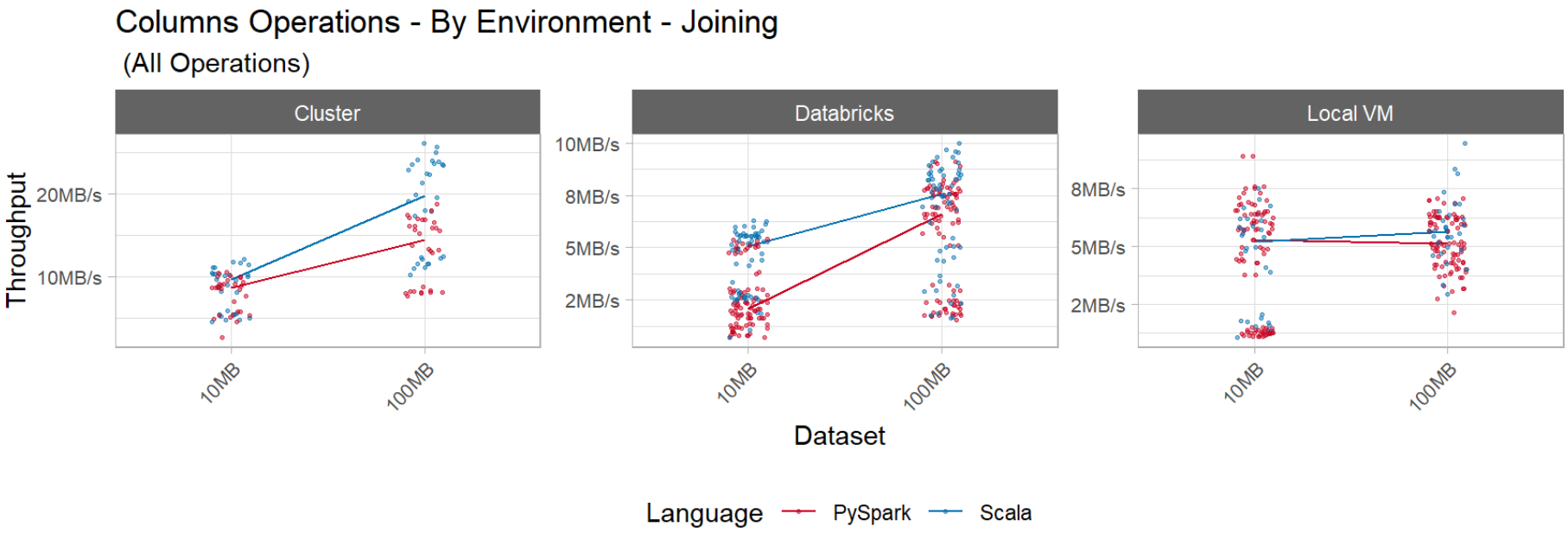
Results – Column Operations



Results – Column Operations

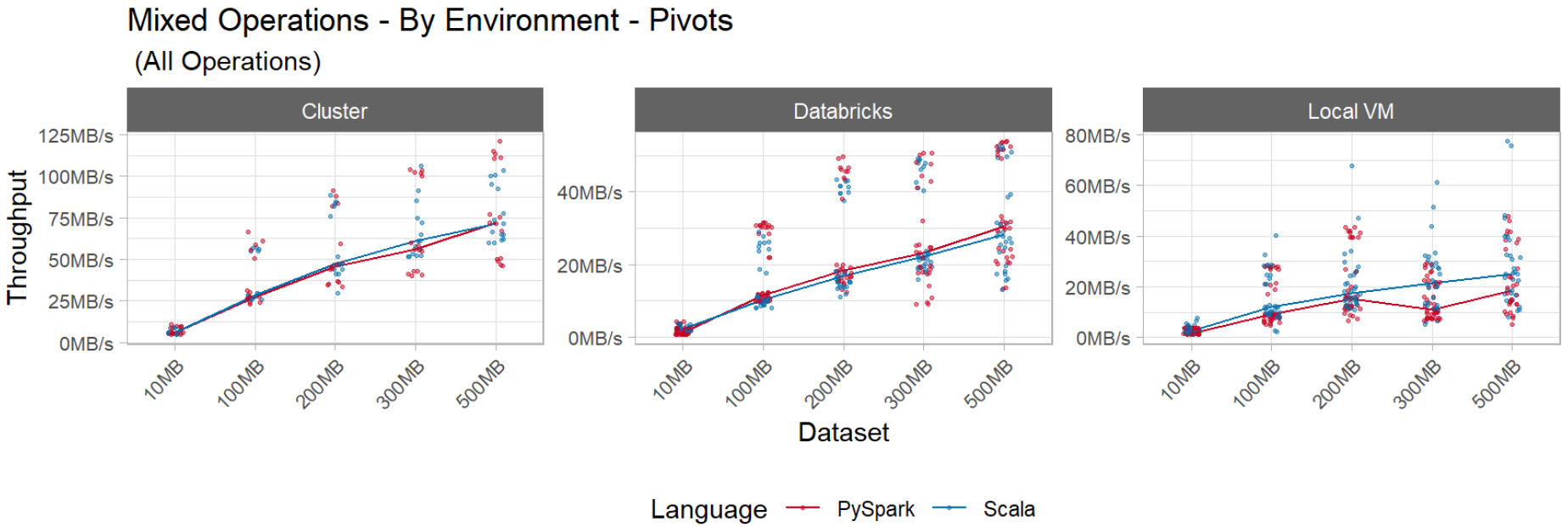


Results – Column Operations



Results – Aggregate Operations

Results – Mixed Operations



Result Summary

Row Operations	Filtering	Tie
	Running Sum	Scala Faster
	Lagging	Scala Faster
	Writing	Scala Faster
Column Operations	Split & Merge	Scala Faster
	Sort	Scala Faster
	Join	Scala slightly faster
	Calculation	Tie
Aggregate and Mixed Operations	Group By	Scala slightly faster
	Ranking	Scala faster
	Pivot	Tie

Key Findings

Setup

- AWS has some complexity to setup Spark and libraries
- Databricks makes the process much easier

Memory

- Spark uses a lot of memory
- A 500MB dataset requires 8GB of memory to run complex queries

Scala

- Similar to Java
- Fast but some limitations on libraries
- Many functions are still in beta

PySpark

- Complex to setup in Linux
- Scala-like commands + flexibility of Python
- API data transfer has a performance penalty

Conclusion

For Data Scientists, what is the best Big Data Solution?

Which one is faster?

- Scala is faster (compared to PySpark) for most operations
- Difference in performance increases as machines become more powerful (e.g. on distributed cluster)

Is it worth learning a new faster language?

- Scala is not yet mature as Python
- Use Scala for larger datasets if ML algorithms are available
- Use PySpark for data manipulation + Python if ML algorithms not available in Scala.

Reproducible Research

Code for queries and complete analysis is available on GitHub