



Data  
Schools

# Day 3 - Module 1: Spark Framework

## Introduction

# Learning Objectives

Upon successful completion of this lecture, you will have a good understanding on the Spark framework, motivation and programming model.

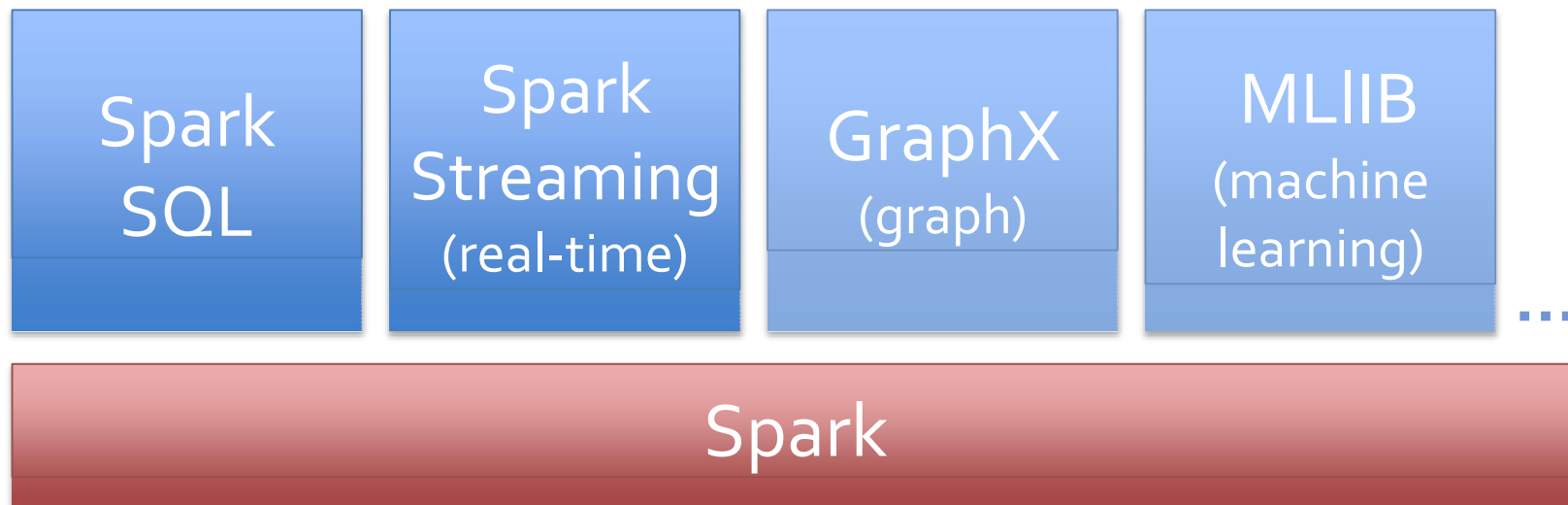
# Spark

# What is Spark?

- Fast and expressive cluster computing system interoperable with Apache Hadoop
  - Improves efficiency through:
    - In-memory computing primitives
    - General computation graphs
  - Improves usability through:
    - Rich APIs in Scala, Java, Python
    - Interactive shell
- Up to 100 × faster  
(2-10 × on disk)
- Often 5 × less code

# The Spark Stack

- Spark is the basis of a wide set of projects in the Berkeley Data Analytics Stack (BDAS)



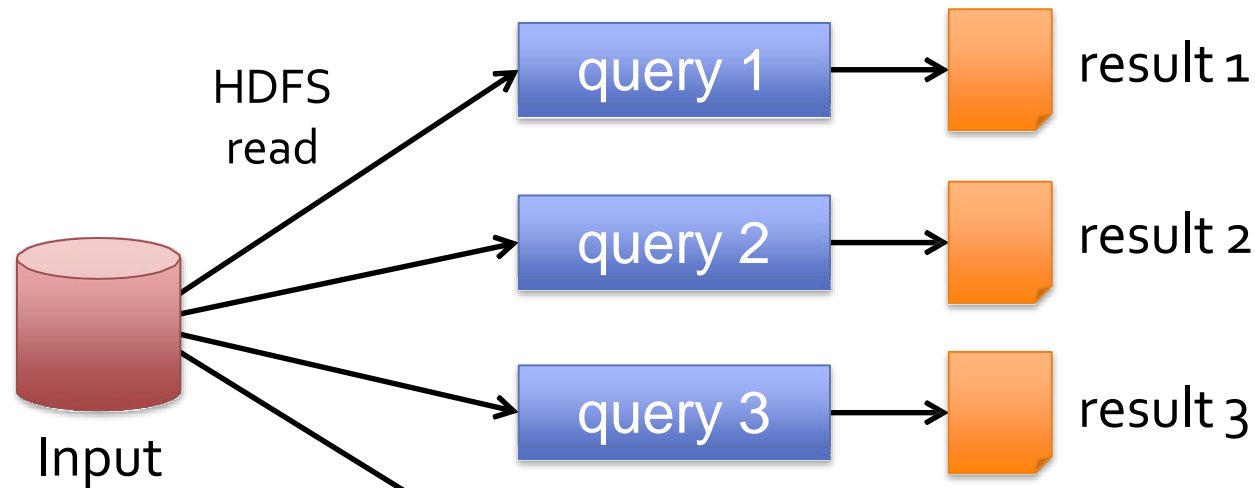
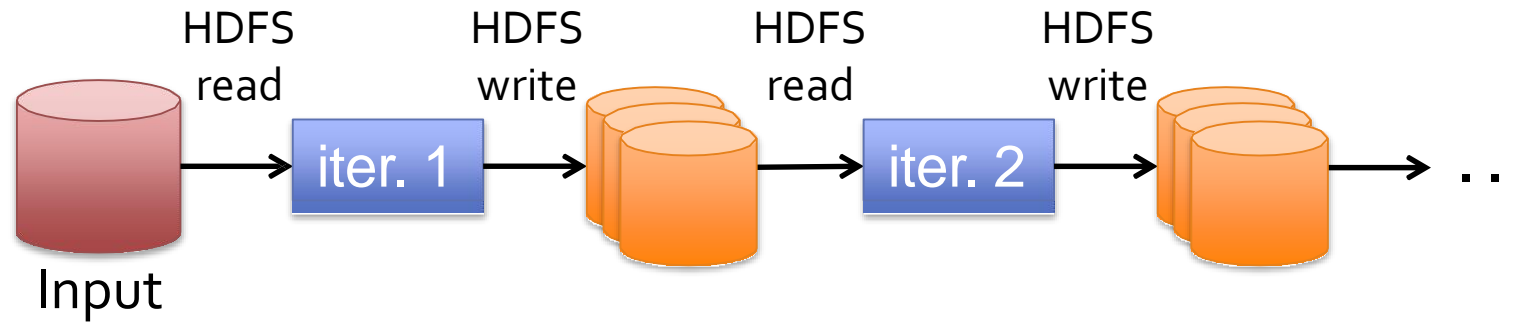
More details: [amplab.berkeley.edu](http://amplab.berkeley.edu)

IoT and Big Data Analytics

# Why a New Programming Model?

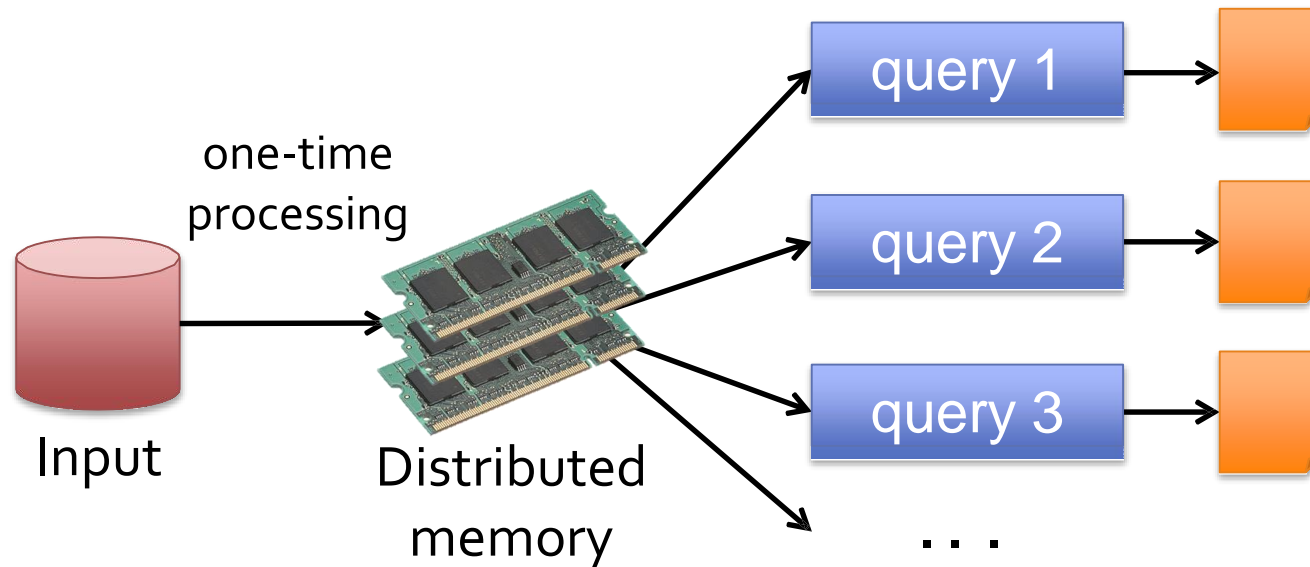
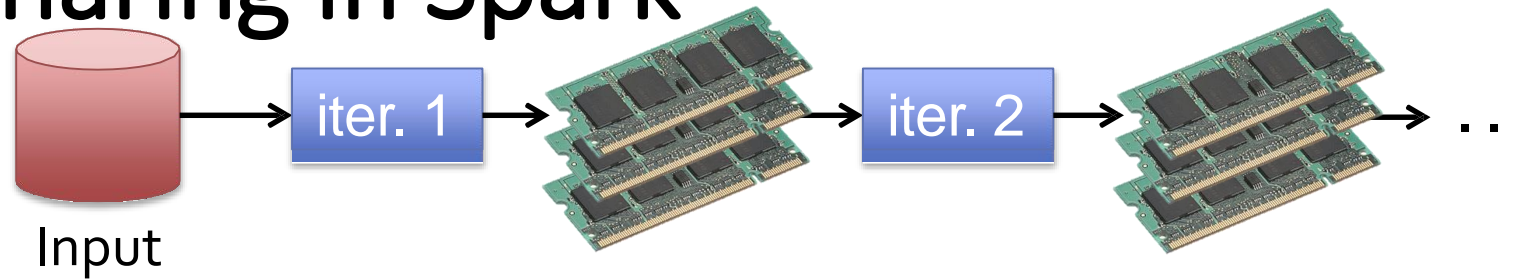
- MapReduce greatly simplified big data analysis
- But as soon as it got popular, users wanted more:
  - More **complex**, multi-pass analytics (e.g. ML, graph)
  - More **interactive** ad-hoc queries
  - More **real-time** stream processing
- All 3 need faster **data sharing** across parallel jobs

# Data Sharing in MapReduce



**Slow** due to replication, serialization, and disk IO

# Data Sharing in Spark



**~10 × faster than network and disk**



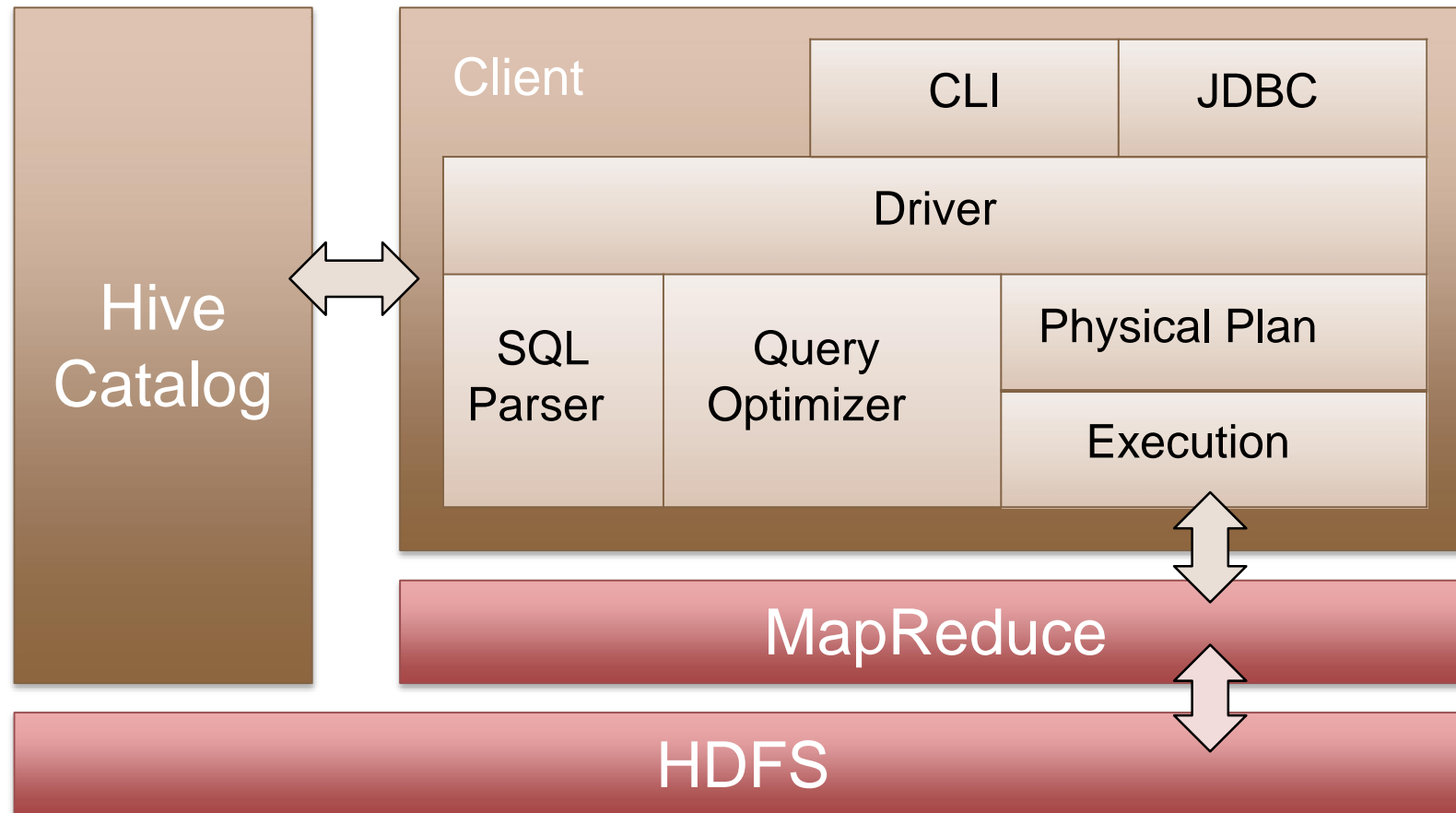
# Spark Programming Model

- Key idea: resilient distributed datasets (RDDs)
  - Distributed collections of objects that can be cached in memory across the cluster
  - Manipulated through parallel operators
  - Automatically recomputed on failure
- Programming interface
  - Functional APIs in Scala, Java, Python
  - Interactive use from Scala shell

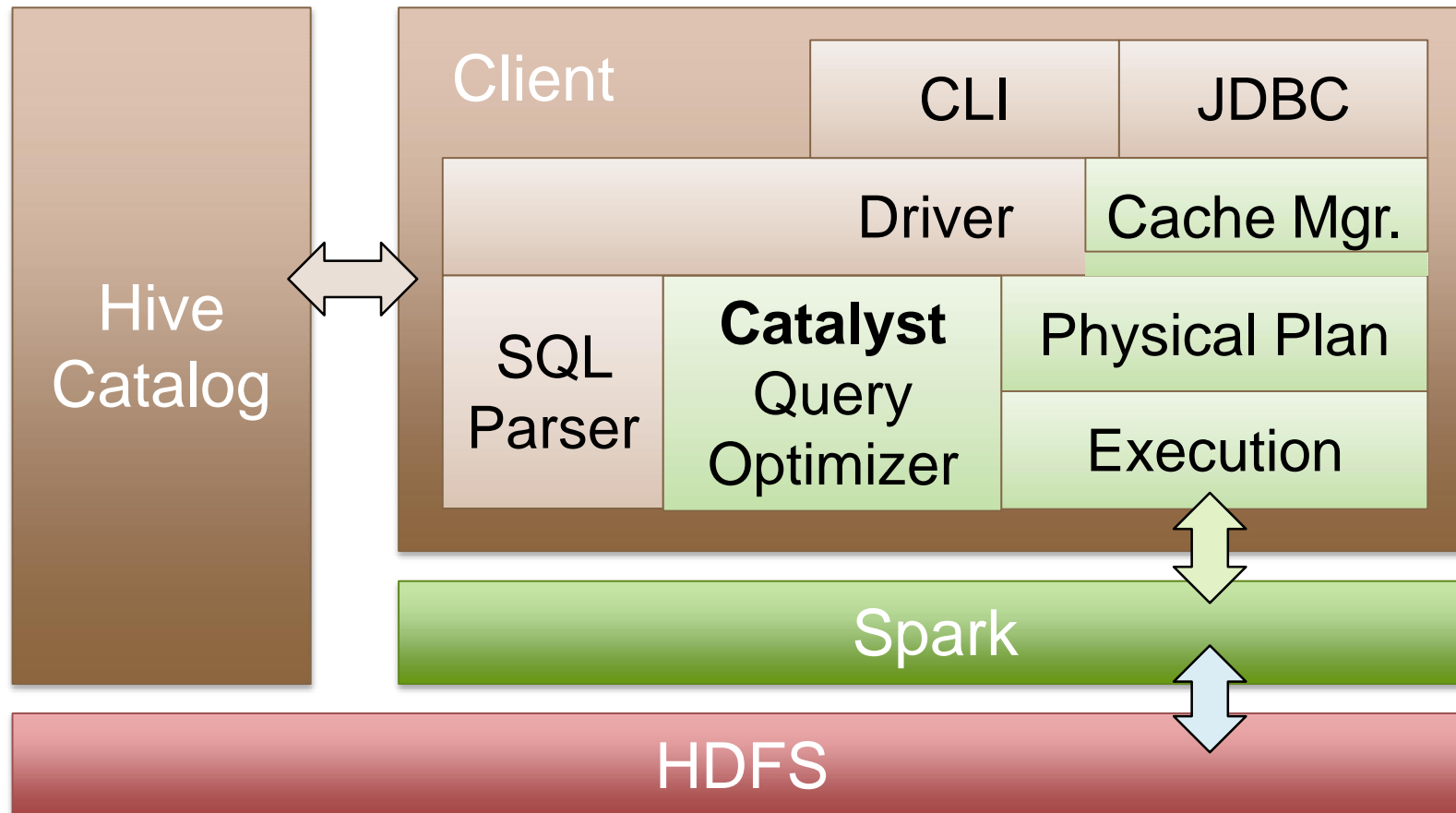
# Spark SQL

- Columnar SQL analytics engine for Spark
  - Support both SQL and complex analytics
  - Columnar storage, JIT-compiled execution, Java/Scala/Python UDFs
  - Catalyst query optimizer (also for DataFrame scripts)

# Hive Architecture



# Spark SQL Architecture



[Engle et al, SIGMOD 2012]

# What is MLLIB?

- MLLib is a Spark subproject providing machine learning primitives:
  - initial contribution from AMPLab, UC Berkeley
  - shipped with Spark since version 0.8



# What is MLLIB?

## Algorithms:

- **classification:** logistic regression, linear support vector machine (SVM), naive Bayes
- **regression:** generalized linear regression (GLM)
- **collaborative filtering:** alternating least squares (ALS)
- **clustering:** k-means
- **decomposition:** singular value decomposition (SVD), principal component analysis (PCA)



# Conclusion

- Big data analytics is evolving to include:
  - More **complex** analytics (e.g. machine learning)
  - More **interactive** ad-hoc queries
  - More **real-time** stream processing
- Spark is a fast platform that *unifies* these apps
- More info: [spark-project.org](http://spark-project.org)



