2017 SDSC Summer Institute
Scalable Machine Learning

SDSC SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego

Scalable Machine Learning Agenda

1:30 - 2:15 – R in HPC

2:15 - 3:15 – Machine Learning with Spark

3:15 - 3:30 – Break

3:30 - 4:15 – Deep Learning Overview

4:15 - 4:45 – CNN Transfer Learning with Keras

4:45 - 5:00 – Wrap-up

Machine Learning with Spark

Mai H. Nguyen

Spark Topics

- Spark Overview
- Programming in Spark
- MLlib

Spark Overview

What is Spark?

- General framework for distributed computing
- Provides built-in data parallelism and fault-tolerance for big data processing on a cluster
- Goals: speed, ease of use, generality
  - Multiple analytics applications, data sources, platforms
- Open-source

Basics of Distributed Processing with Spark

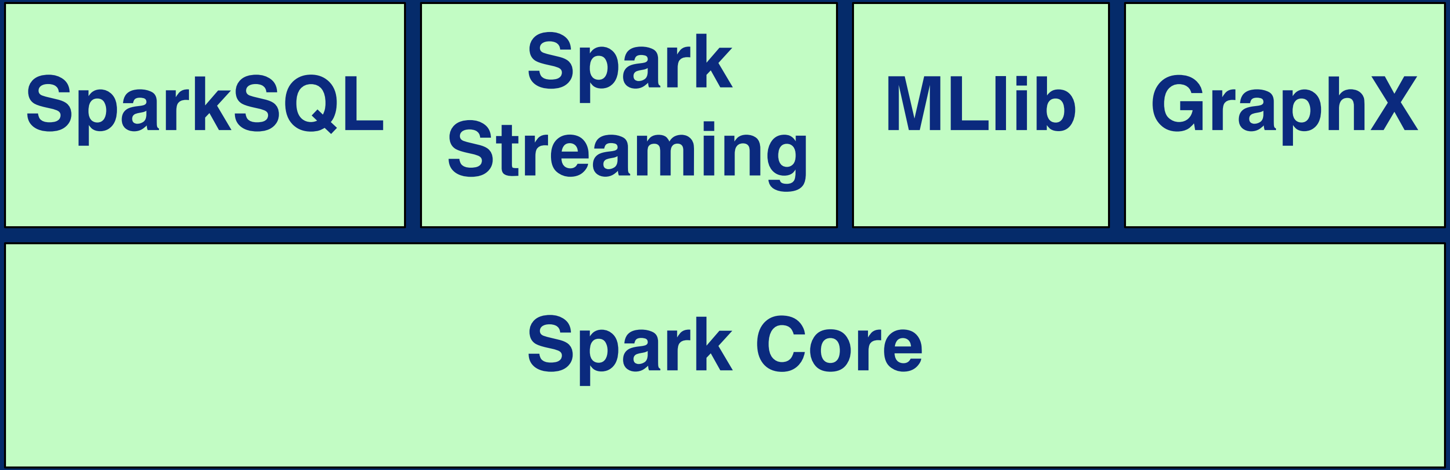
Expressive programming environment

In-memory processing

Support for diverse workloads

Interactive shell

The Spark Stack



SparkSQL

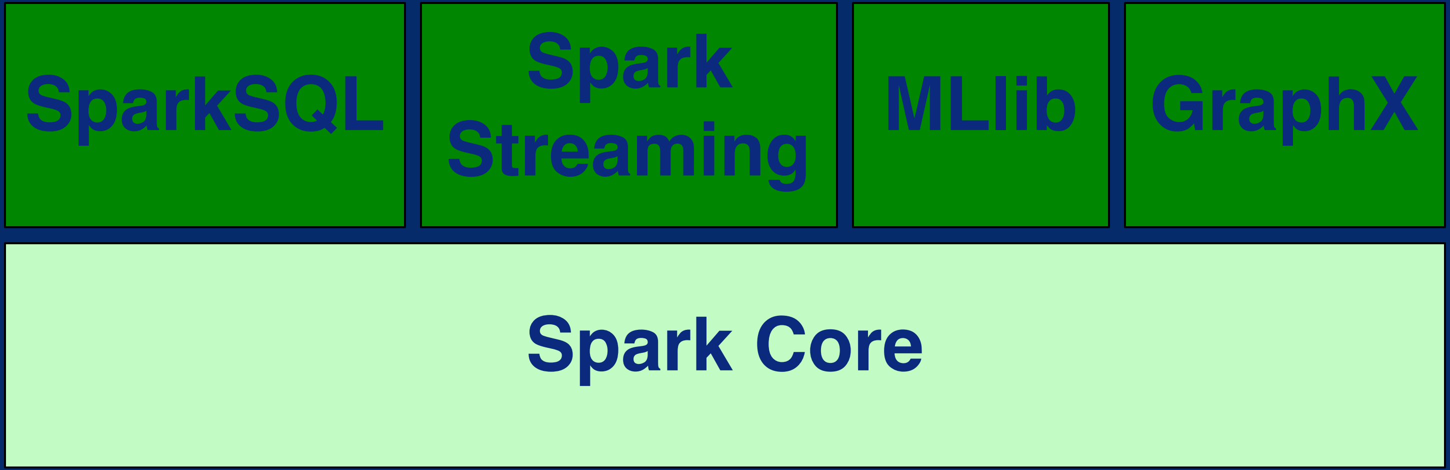
Spark Streaming

MLlib

GraphX

Spark Core

The Spark Stack



SparkSQL

Spark Streaming

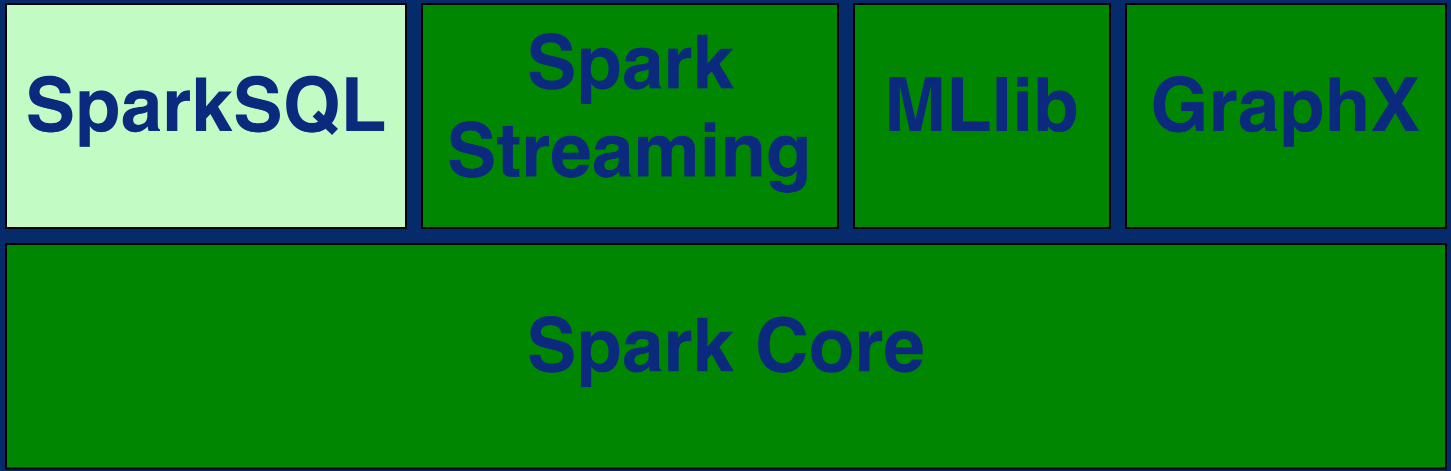
MLlib

GraphX

Spark Core

Distributed computing

The Spark Stack



SparkSQL

Spark Streaming

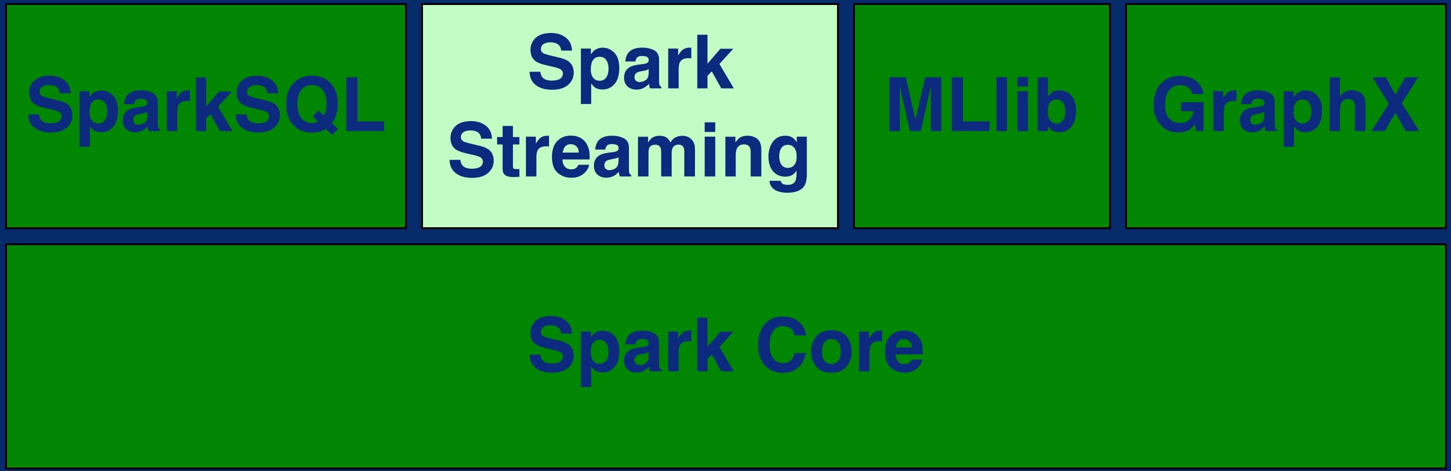
MLlib

GraphX

Spark Core

SQL-like querying

The Spark Stack



SparkSQL

Spark Streaming

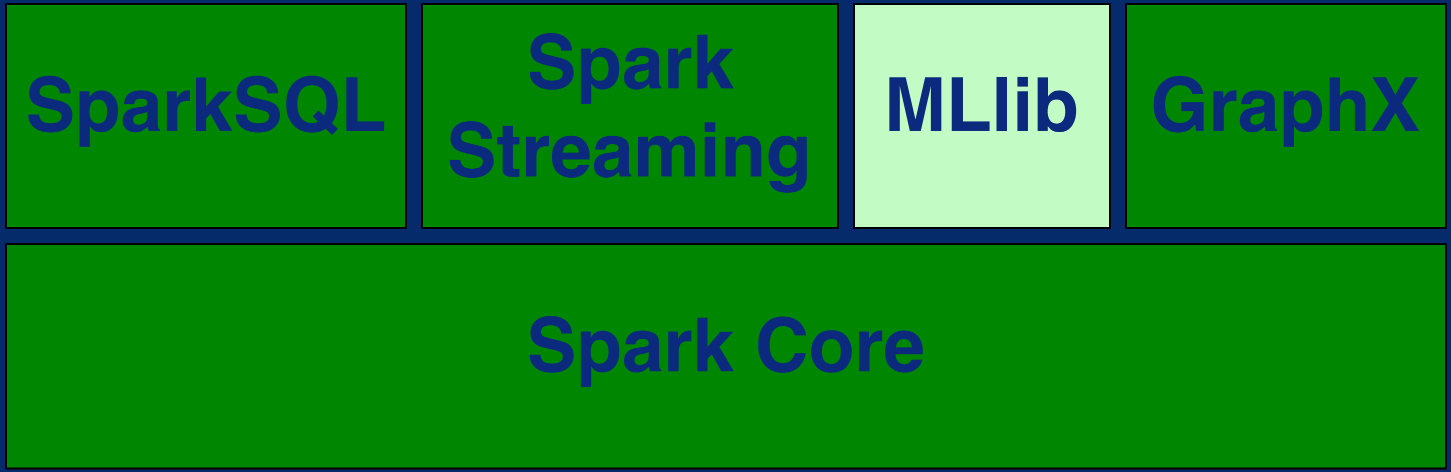
MLlib

GraphX

Spark Core

Streaming processing

The Spark Stack



SparkSQL

Spark Streaming

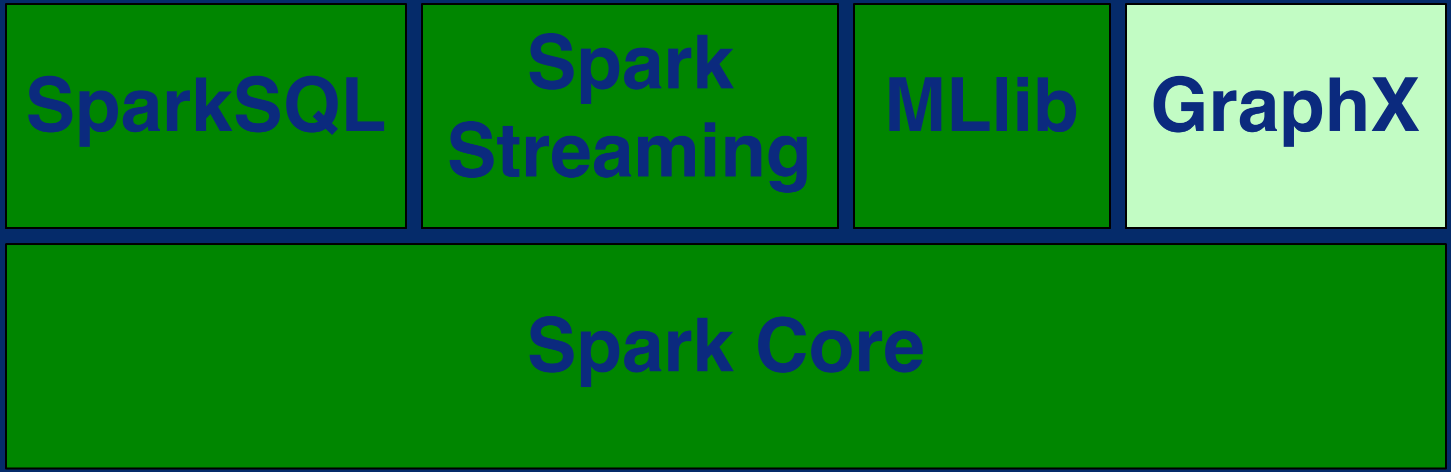
MLlib

GraphX

Spark Core

Machine learning

The Spark Stack



SparkSQL

Spark Streaming

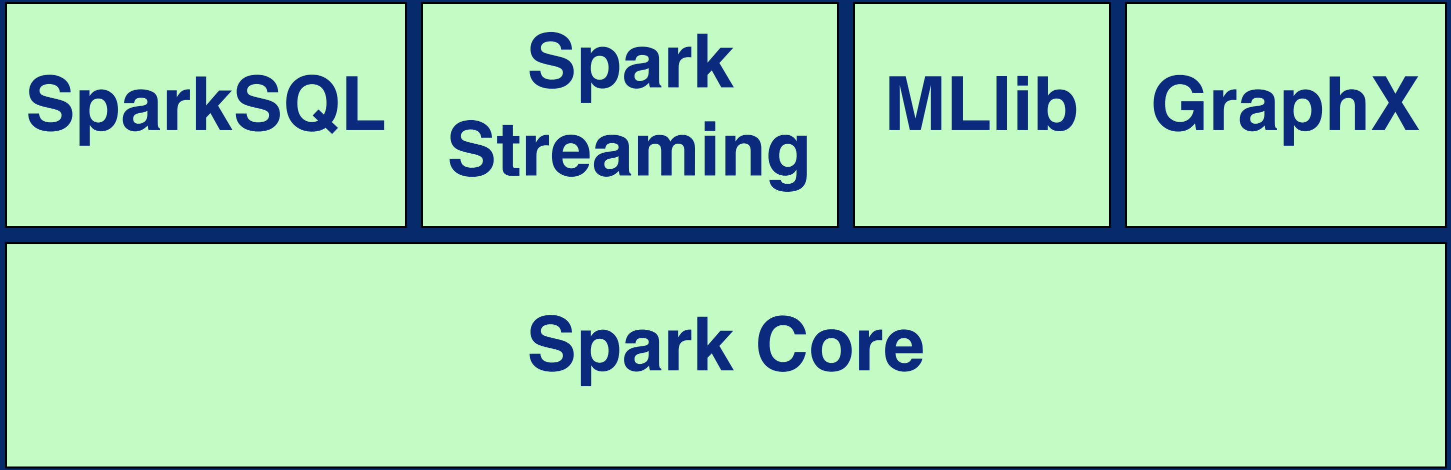
MLlib

GraphX

Spark Core

Graph analytics

The Spark Stack



SparkSQL

Spark Streaming

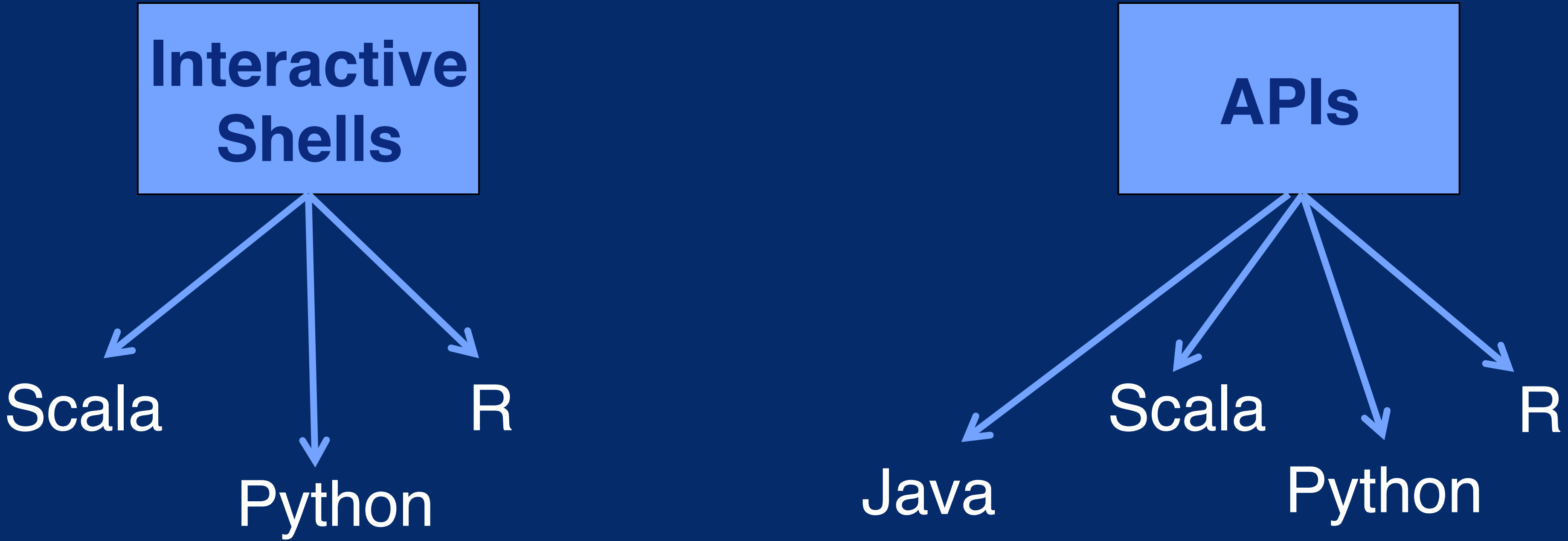
MLlib

GraphX

Spark Core

Supports diverse analytics applications

Spark Interface



Interactive Shells

Scala

Python

R

APIs

Java

Scala

Python

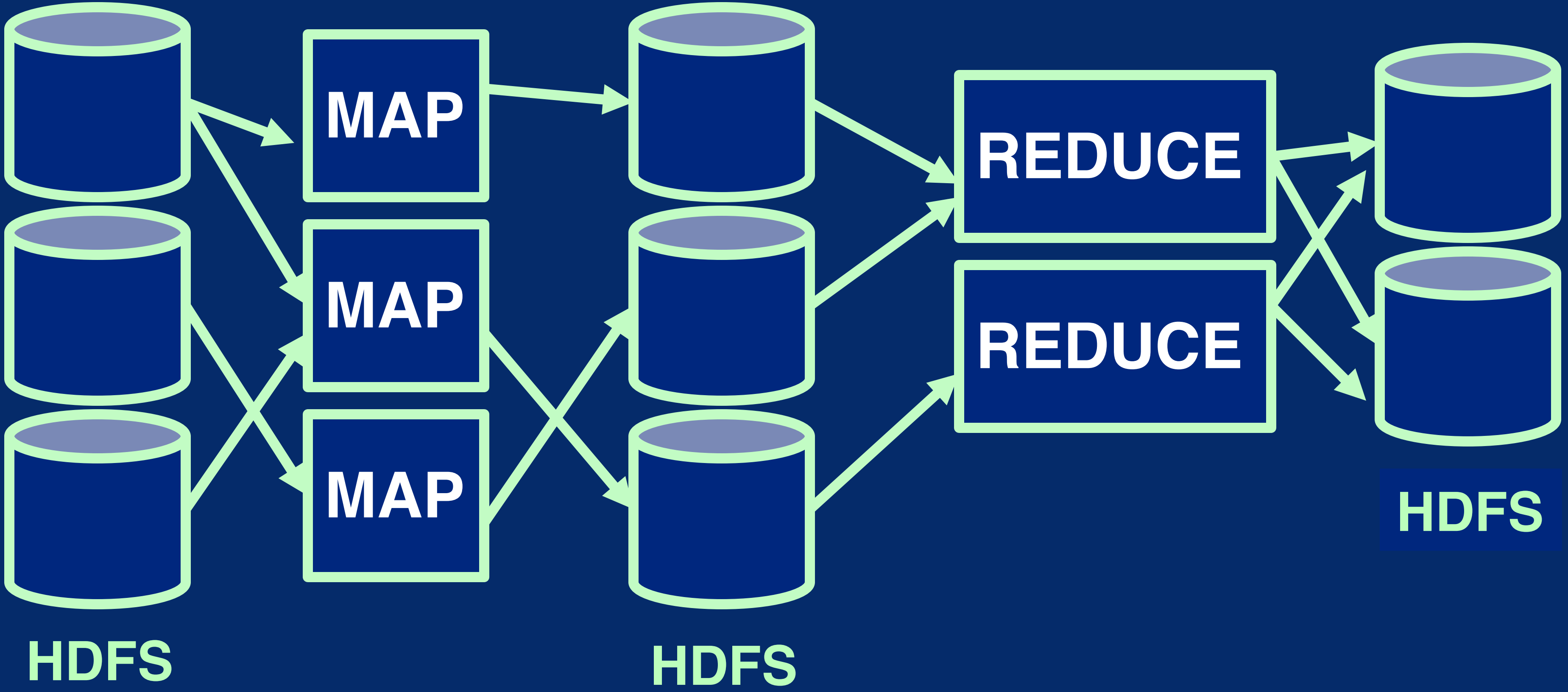
R

Provides ease of use

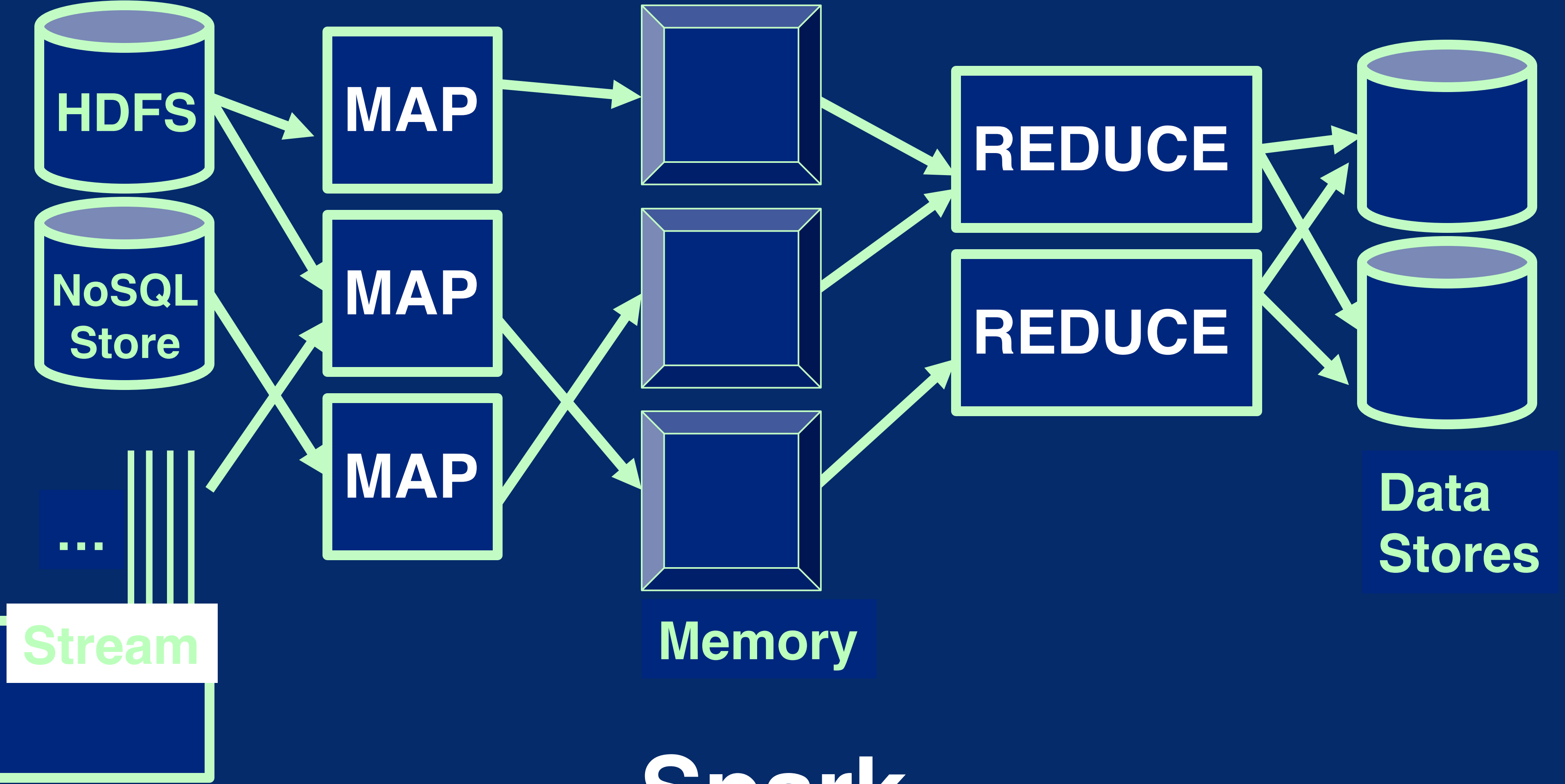
In Memory Processing

Provides speed

What does in memory processing mean?



MapReduce

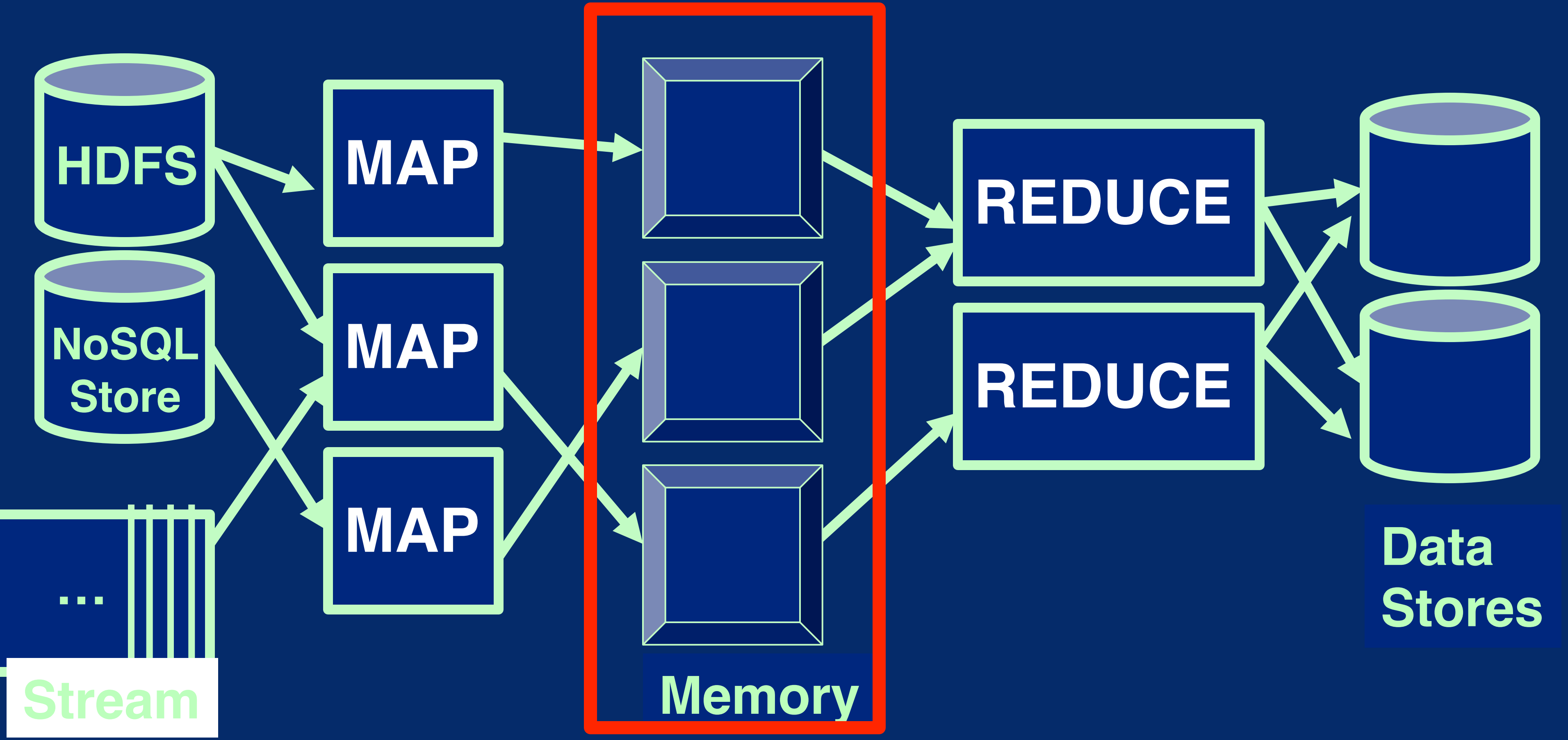


Spark

SDSC SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego

Resilient Distributed Datasets



Spark

SDSC SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego

Resilient Distributed Datasets

Dataset

Data storage created from: HDFS, S3, HBase, JSON, text, Local hierarchy of folders

Or created transforming another RDD

Resilient **Distributed** Datasets

Distributed

Distributed across the cluster of machines

Divided in partitions, atomic chunks of data

Resilient Distributed Datasets

Resilient

Recover from errors, e.g. node failure, slow processes

Track history of each partition, re-run

DataFrames & DataSets

DataFrame

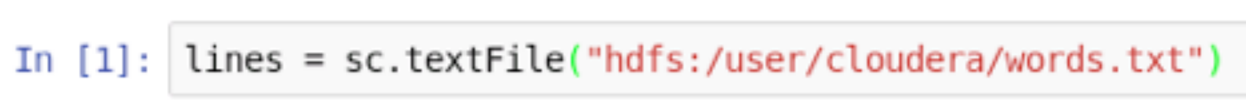
DataSet

- Extensions to RDDs
- Provide higher-level abstractions, improved performance, better scalability

Programming in Spark

Creating RDDs

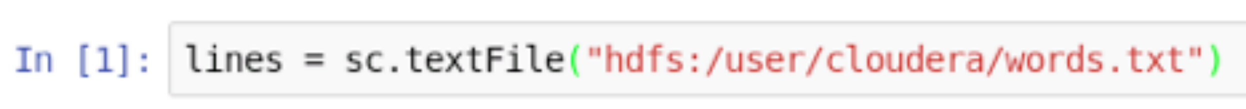
Driver Program



```
In [1]: lines = sc.textFile("hdfs:/user/cloudera/words.txt")
```

Creating RDDs

Driver Program

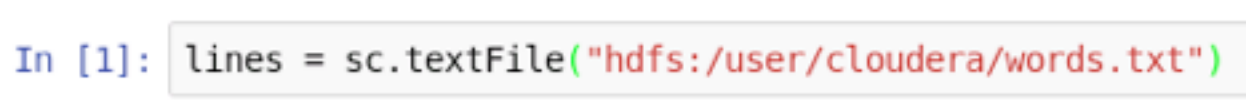


```
In [1]: lines = sc.textFile("hdfs:/user/cloudera/words.txt")
```

```
lines = sc.parallelize([“big”, ”data”])
```

Creating RDDs

Driver Program



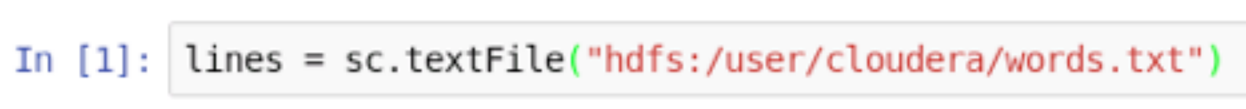
```
In [1]: lines = sc.textFile("hdfs:/user/cloudera/words.txt")
```

```
lines = sc.parallelize([“big”, ”data”])
```

```
numbers = sc.parallelize(range(10), 3)
```

Creating RDDs

Driver Program



```
In [1]: lines = sc.textFile("hdfs:/user/cloudera/words.txt")
```

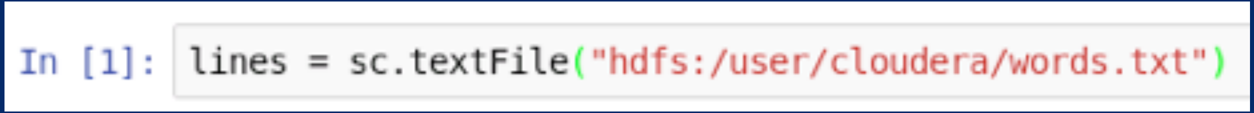
```
lines = sc.parallelize([“big”, ”data”])
```

```
numbers = sc.parallelize(range(10), 3)
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Creating RDDs

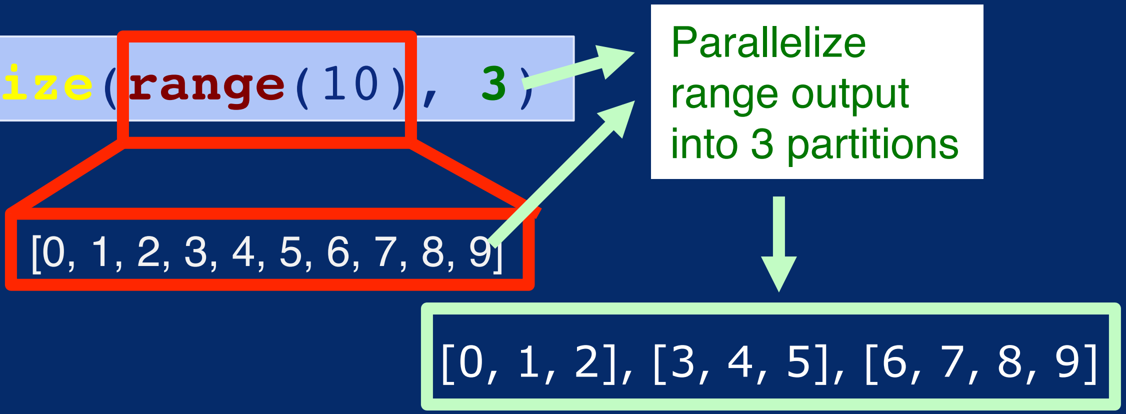
Driver Program



```
In [1]: lines = sc.textFile("hdfs:/user/cloudera/words.txt")
```

```
lines = sc.parallelize([“big”, ”data”])
```

```
numbers = sc.parallelize(range(10), 3)
```



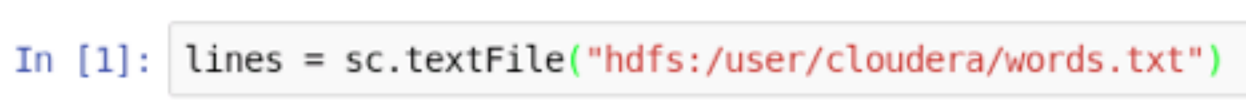
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Parallelize range output into 3 partitions

[0, 1, 2], [3, 4, 5], [6, 7, 8, 9]

Creating RDDs

Driver Program



```
In [1]: lines = sc.textFile("hdfs:/user/cloudera/words.txt")
```

```
lines = sc.parallelize([“big”, ”data”])
```

```
numbers = sc.parallelize(range(10), 3)
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

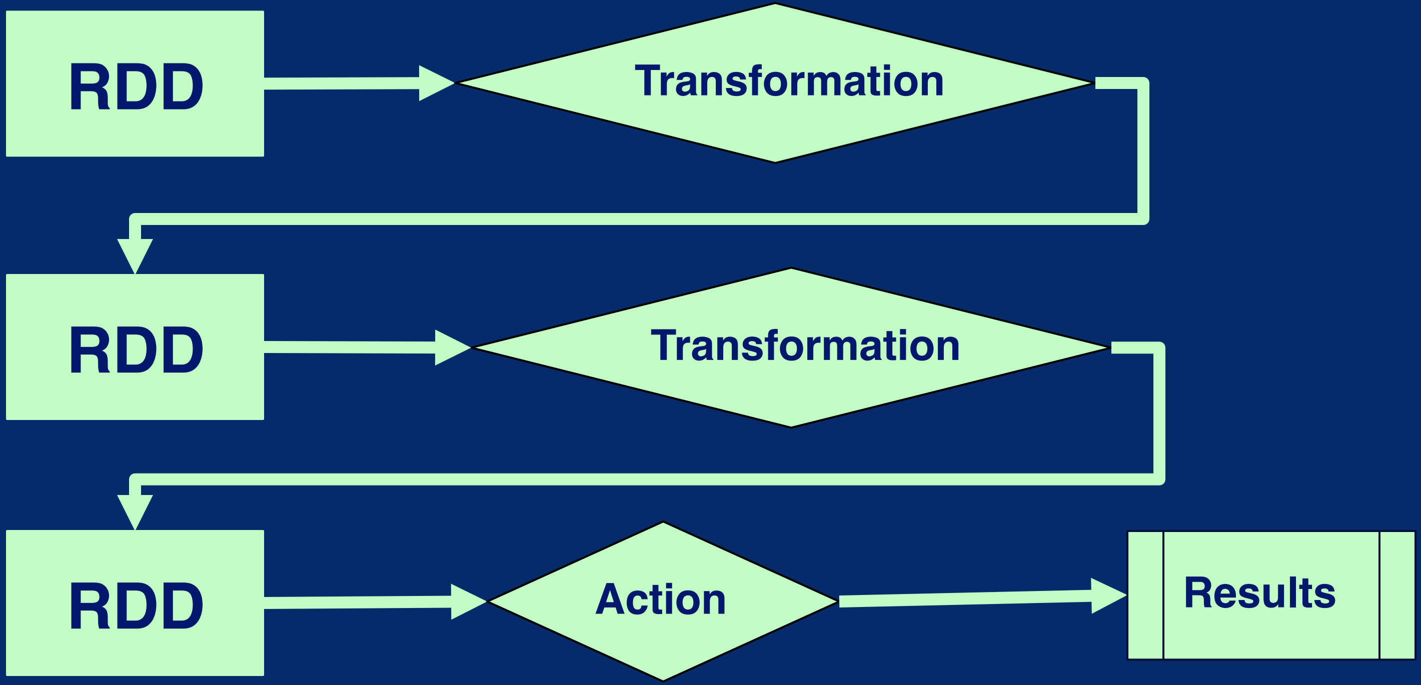
Parallelize range output into 3 partitions

[0, 1, 2], [3, 4, 5], [6, 7, 8, 9]

```
numbers.collect()
```

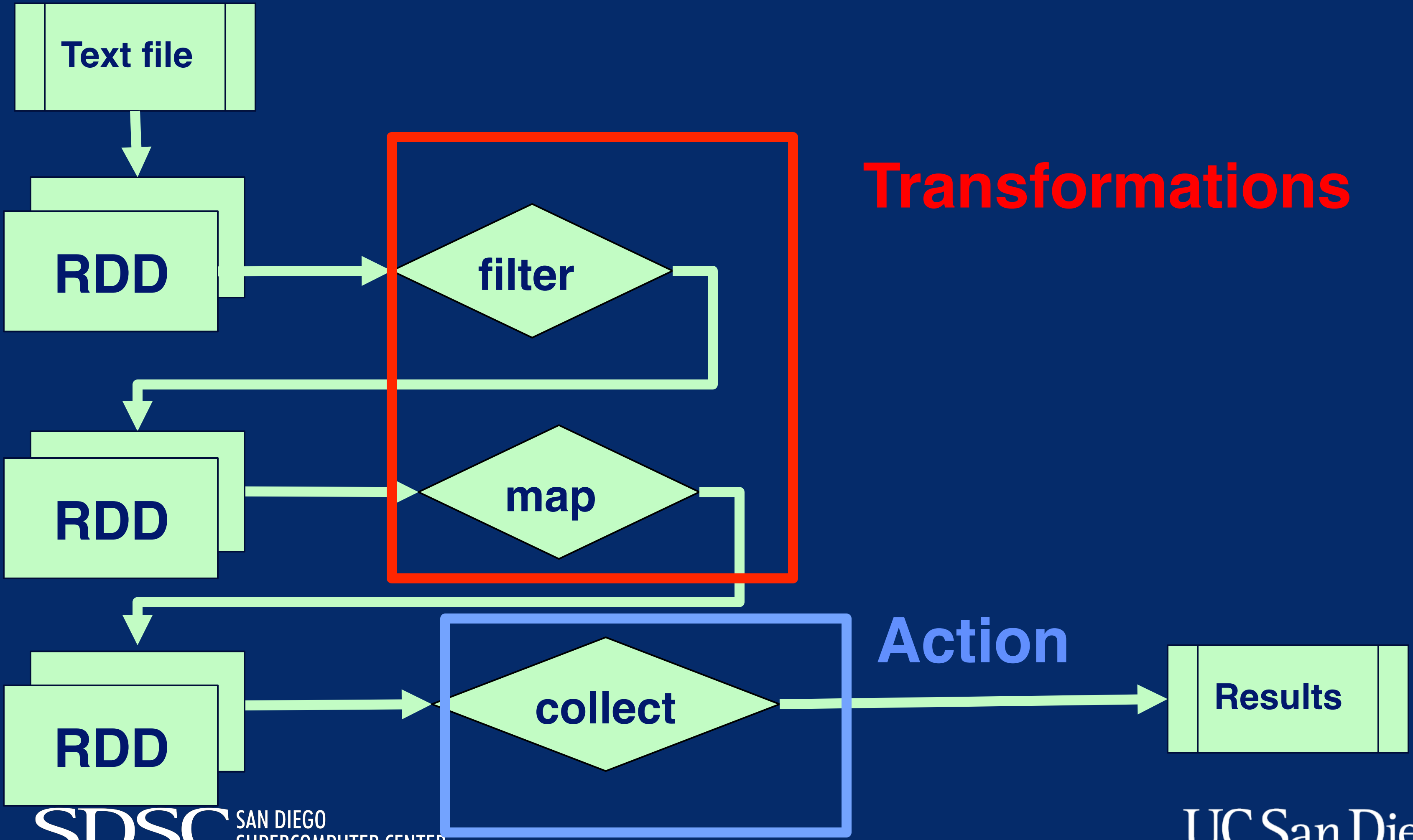
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

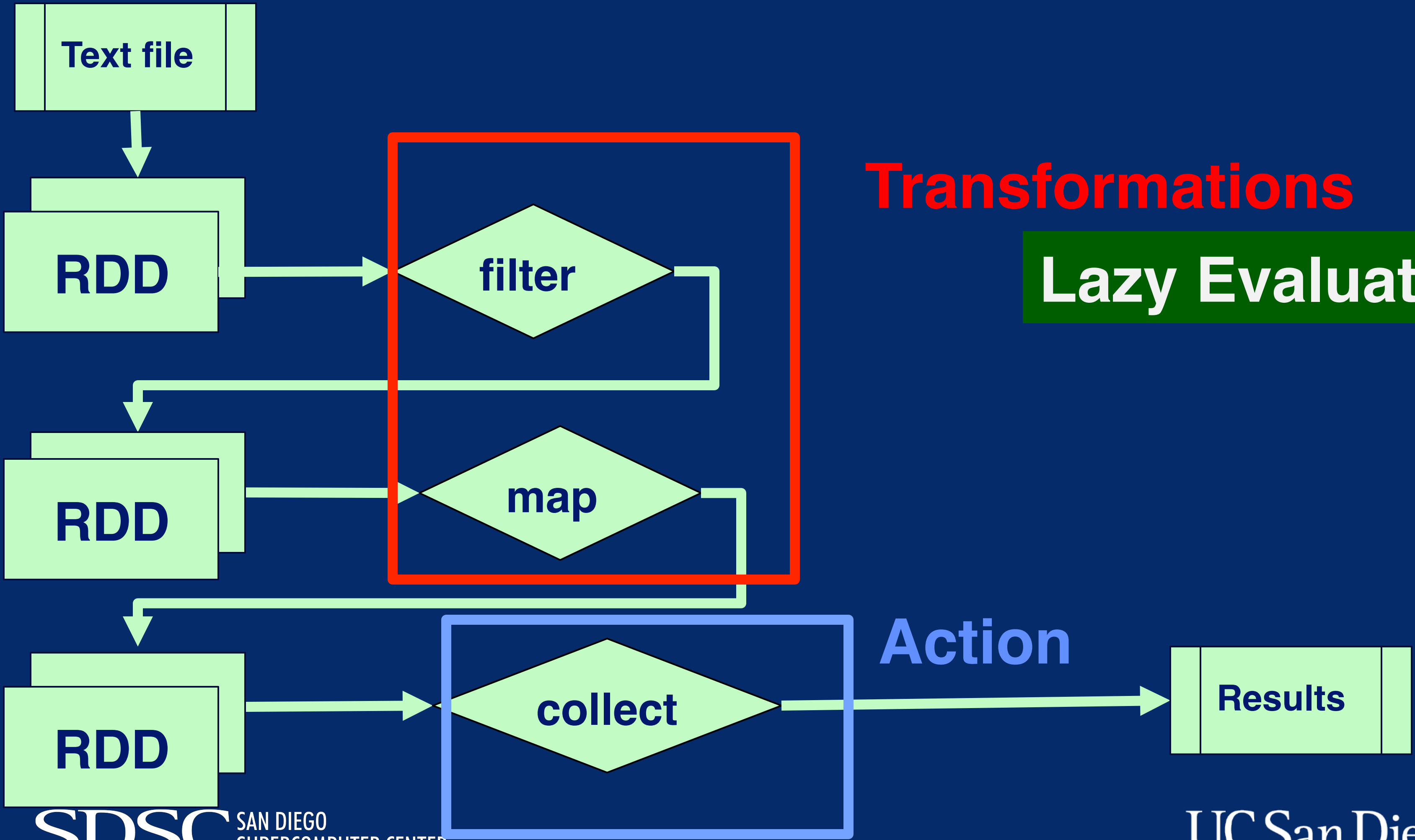
Processing RDDs



SDSC SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego


Text file
RDD
filter
RDD
map
RDD
collect
Results
Transformations
Action


Text file
RDD
filter
RDD
map
RDD
collect
Results
Transformations
Lazy Evaluation
Action

Transformations & Actions

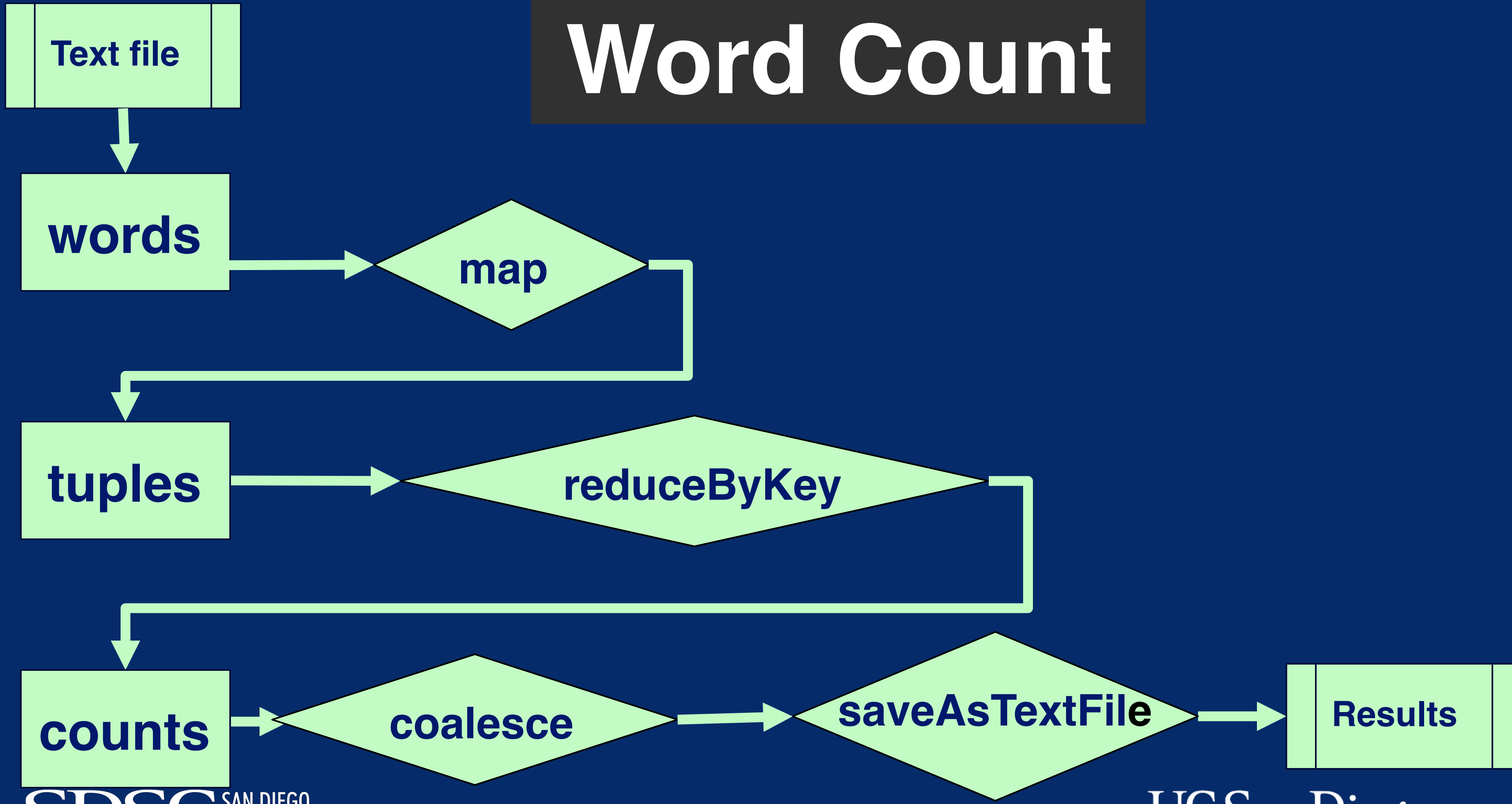
Transformations

- map
- filter
- coalesce
- reduceByKey

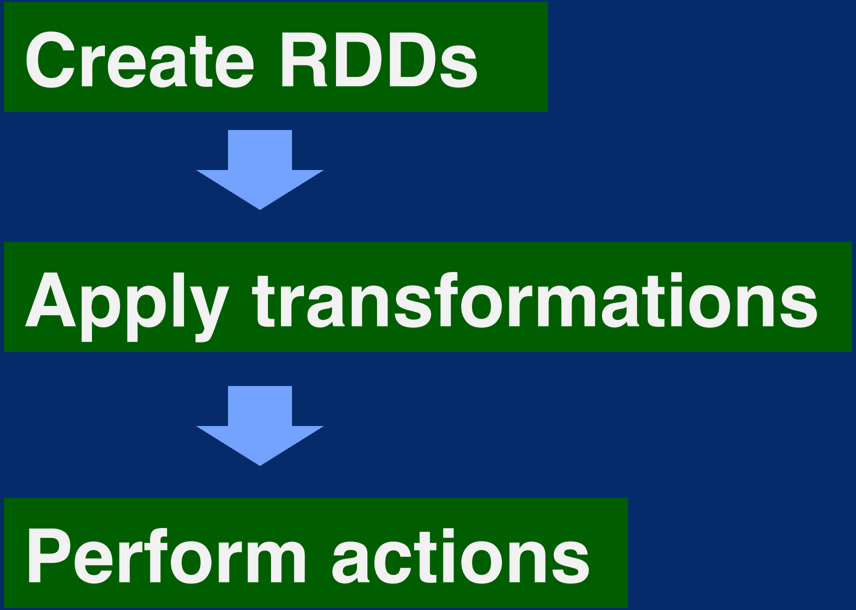
Actions

- collect
- take
- reduce
- saveAsText

Word Count


Text file
words
map
tuples
reduceByKey
counts
coalesce
saveAsTextFile
Results

Programming in Spark

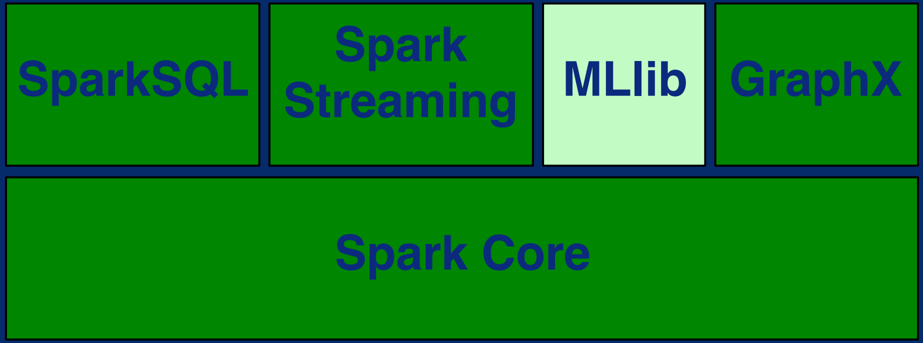


Create RDDs

Apply transformations

Perform actions

Spark MLlib:
Machine Learning



SparkSQL

Spark Streaming

MLlib

GraphX

Spark Core

Spark MLlib

- Scalable machine learning library
- Provides distributed implementations of common machine learning algorithms and utilities
- Has APIs for Scala, Java, Python, and R

MLlib Algorithms & Techniques

- Machine Learning
  - Classification, regression, clustering, etc.
  - Evaluation metrics
- Statistics
  - Summary statistics, sampling, etc.
- Utilities
  - Dimensionality reduction, transformation, etc.

MlLib Example – Summary Statistics

- Compute column summary statistics

```
from pyspark.mllib.stat import Statistics
```

1

2

```
# Data as RDD of Vectors
dataMatrix = sc.parallelize([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]])


# Compute column summary statistics.
summary = Statistics.colStats(dataMatrix)
print(summary.mean())
print(summary.variance())
print(summary.numNonzeros())
```

3

4

MLlib Example – Clustering

- Build k-means model for clustering

1

```
from pyspark.mllib.clustering import KMeans, KMeansModel
from numpy import array
```

2

```
# Read and parse data
data = sc.textFile(“data.txt”)
parsedData = data.map(lambda line:
             array([float(x) for x in line.split(‘ ‘)]))
```

3

```
# k-means model for clustering
clusters = Kmeans.train (parsedData, k=3)
```

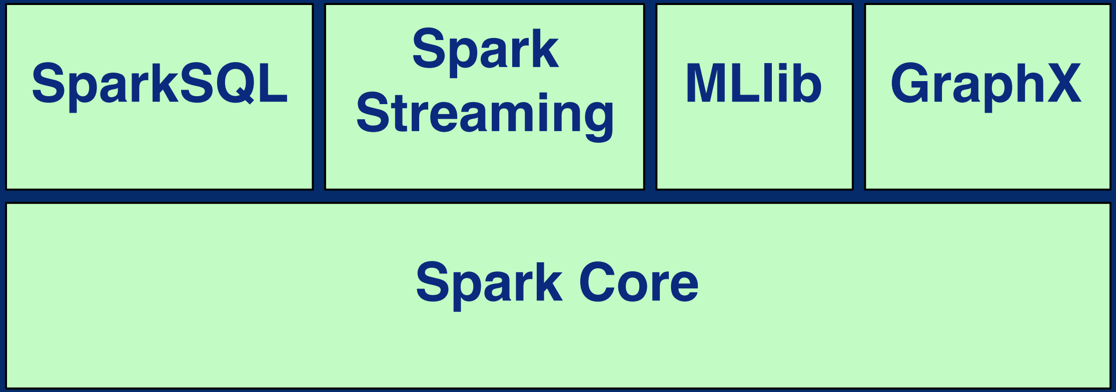
4

```
print(clusters.centers)
```

5

Spark MLlib

- MLlib is Spark’s machine learning library.
  - Distributed implementations
- Main categories of algorithms and techniques:
  - Machine learning
  - Statistics
  - Utilities for data preparation



SparkSQL

Spark Streaming

MLlib

GraphX

Spark Core

Spark

- Spark core provides distributed computing
- Libraries support multiple analytics applications and workloads
- RDDs provide data parallelism & fault-tolerance
- MLlib provides scalable machine learning

Q&A

SDSC SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego