2018 SDSC Summer Institute
Scalable Machine Learning

SDSC SAN DIEGO SUPERCOMPUTER CENTER

UC San Diego

Scalable Machine Learning Agenda

1:30 - 3:00 – R in HPC

3:00 - 3:15 – Break

3:15 - 3:45 – Machine Learning with Spark

3:45 - 4:15 – PySpark Hands-On

4:15 - 4:45 – SparkR Hands-On

4:45 - 5:00 – Wrap-Up

Machine Learning with Spark

Mai H. Nguyen, Ph.D.

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

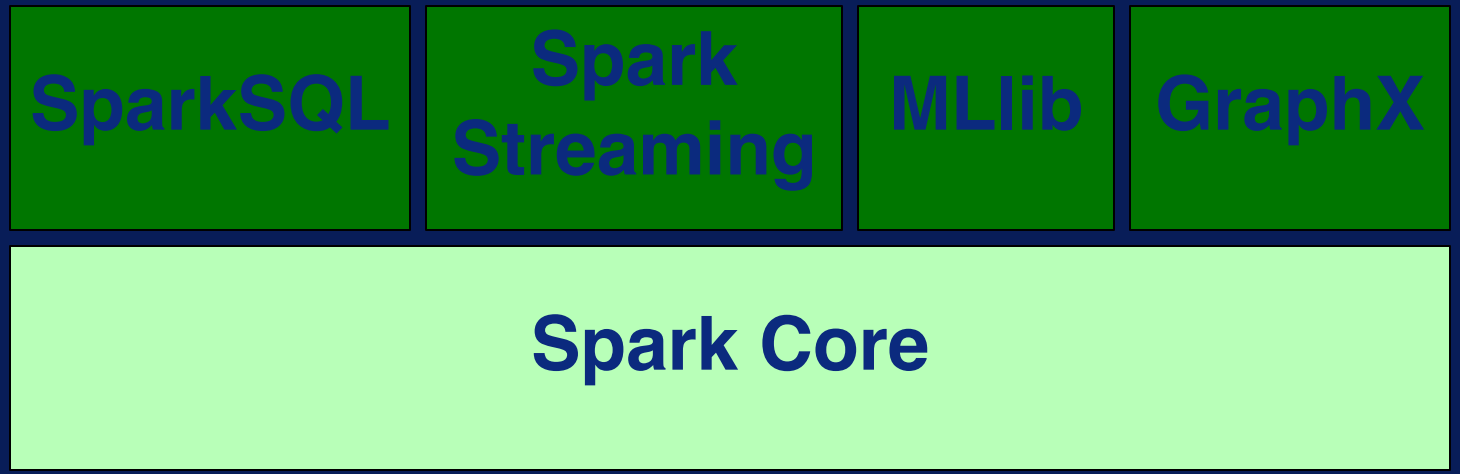
Interactive shell

The Spark Stack

SparkSQL	Spark Streaming	MLlib	GraphX

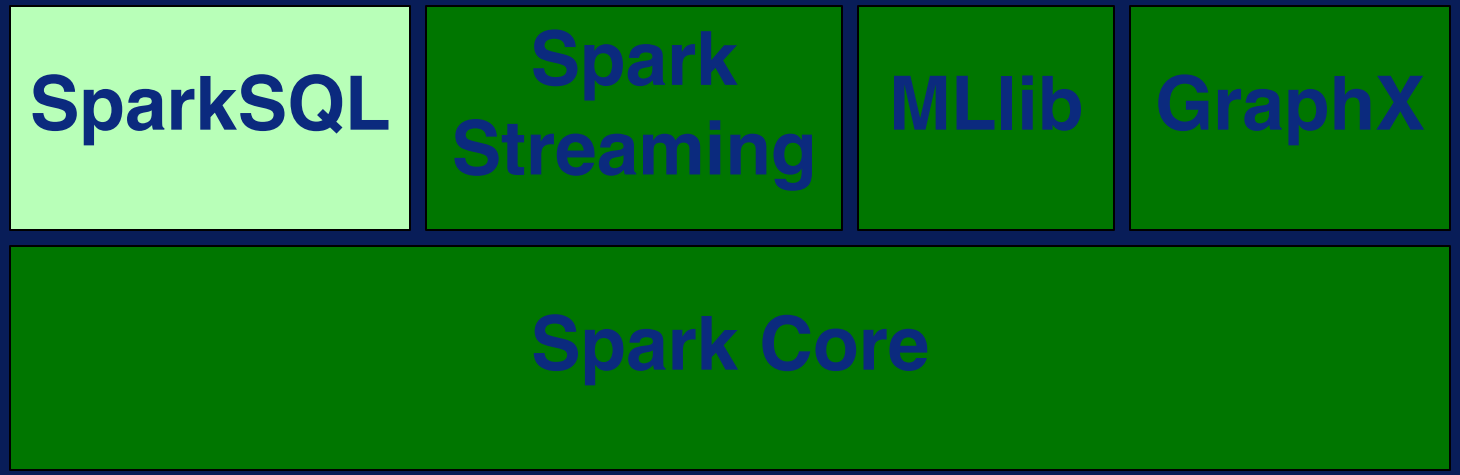
Spark Core

The Spark Stack



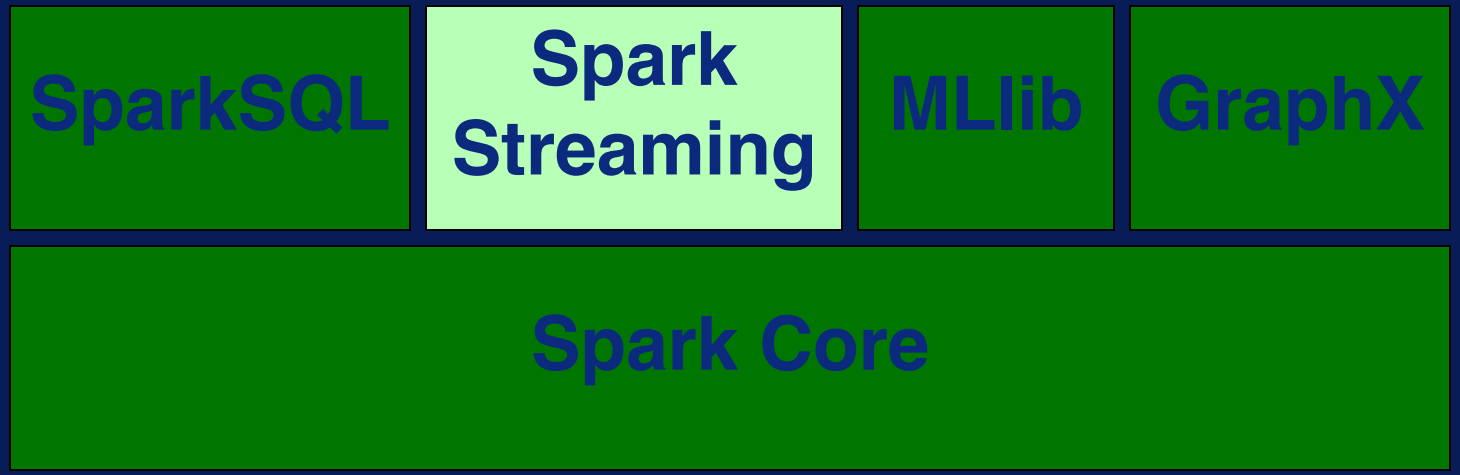
Distributed computing

The Spark Stack



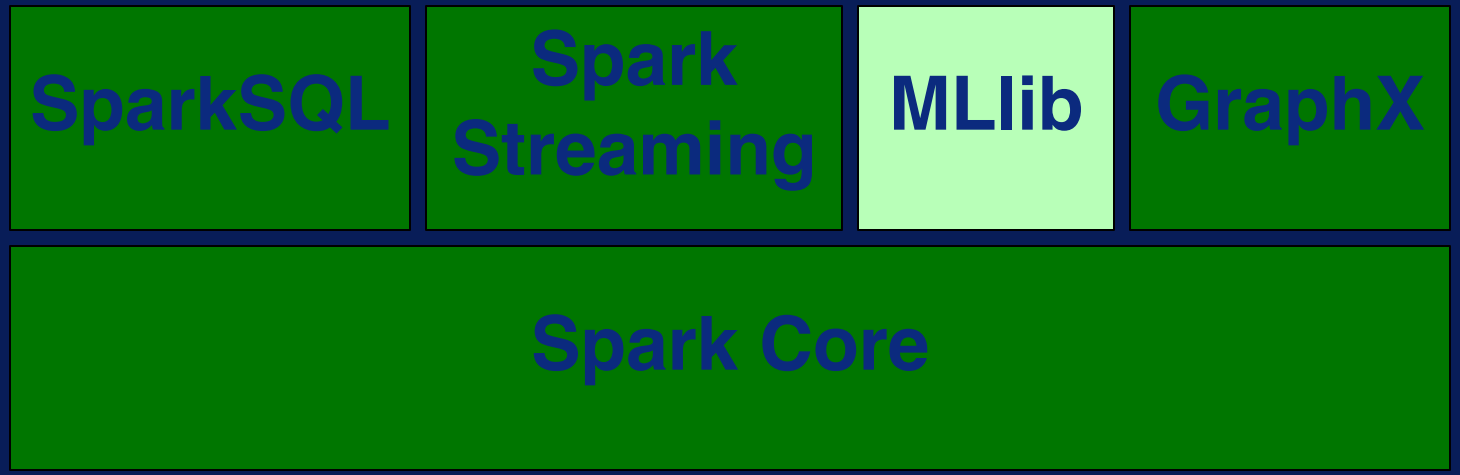
SQL-like querying

The Spark Stack



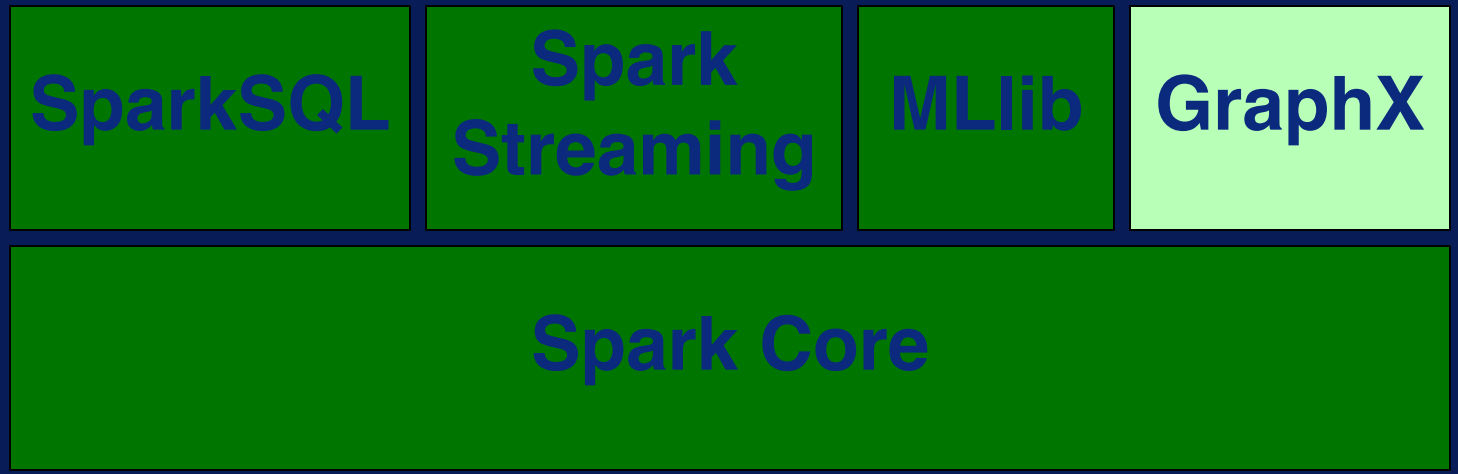
Streaming processing

The Spark Stack



Machine learning

The Spark Stack



Graph analytics

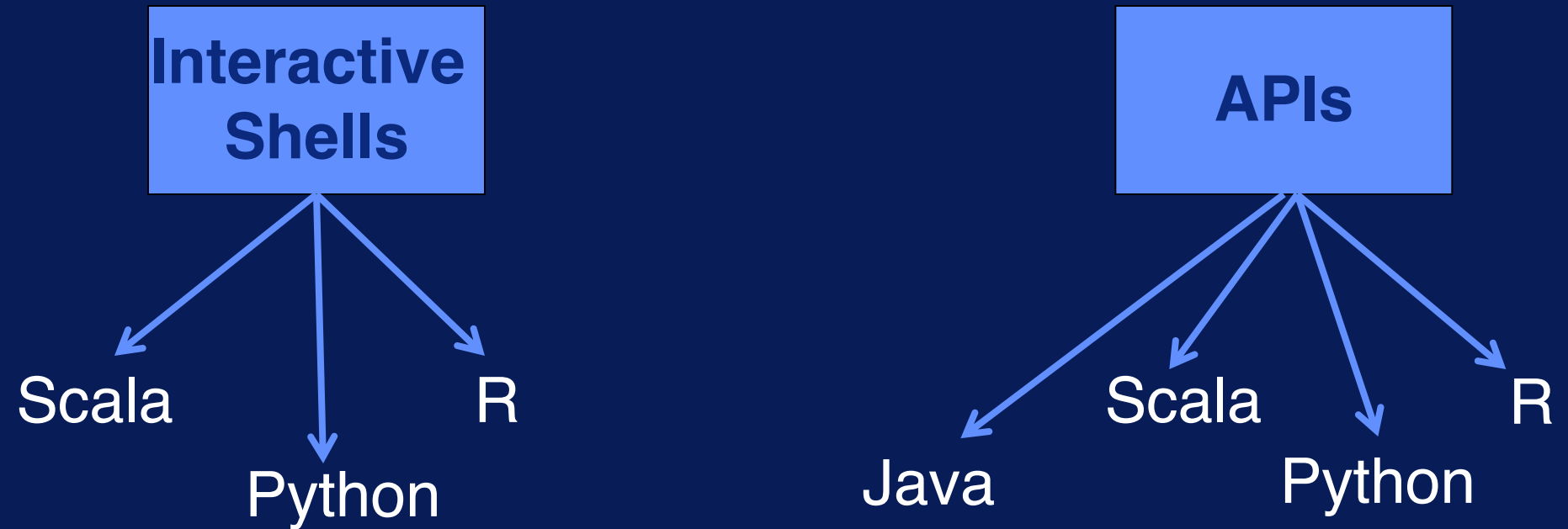
The Spark Stack

SparkSQL	Spark Streaming	MLlib	GraphX

Spark Core

Supports diverse analytics applications

Spark Interface

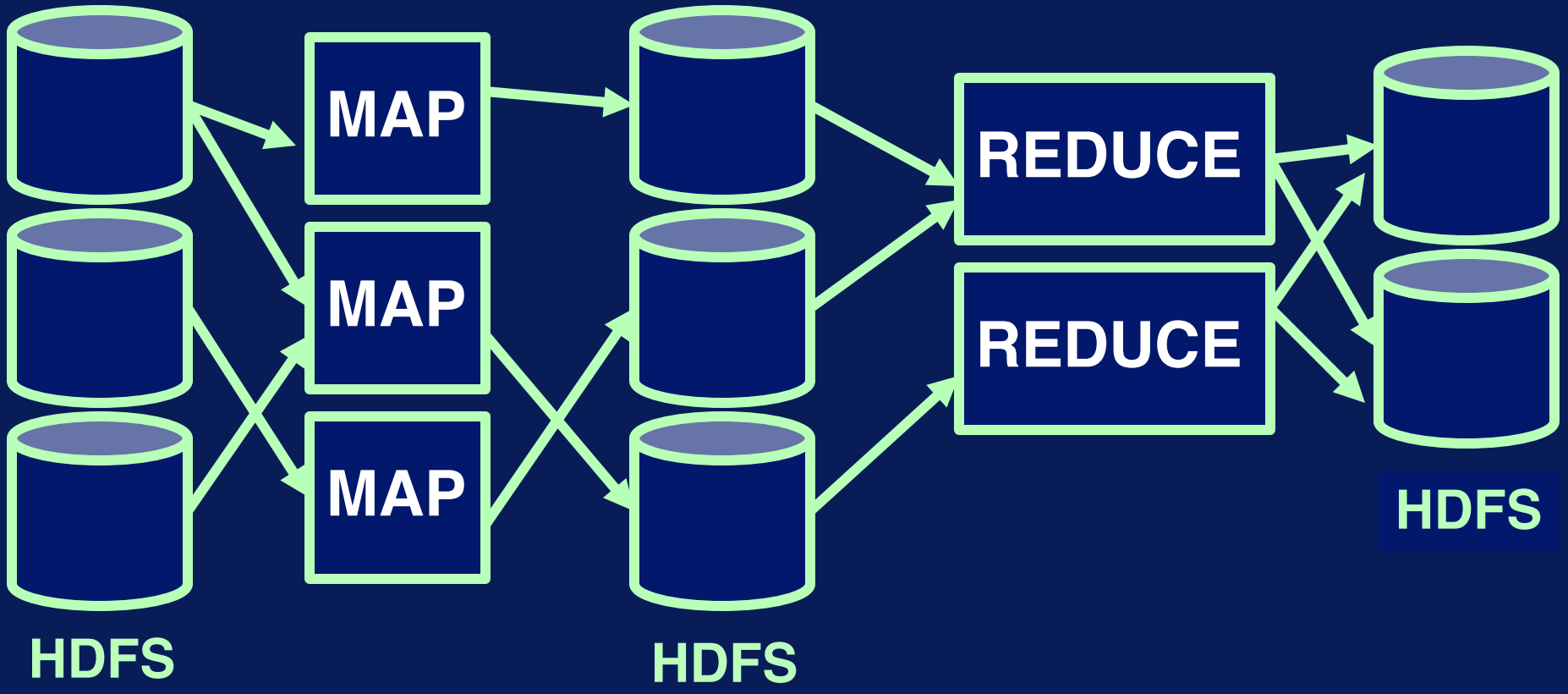


Provides ease of use

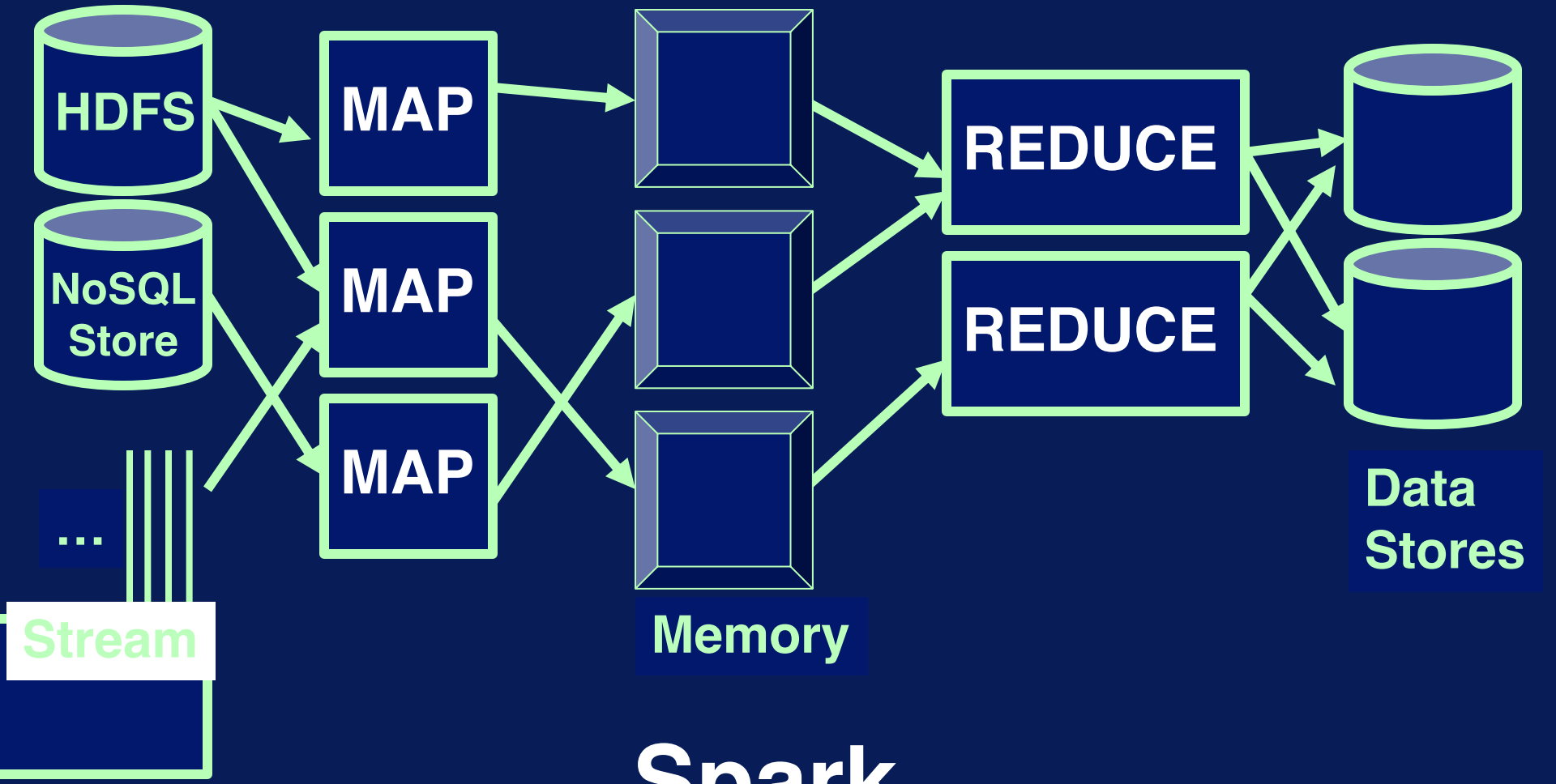
In Memory Processing

Provides speed

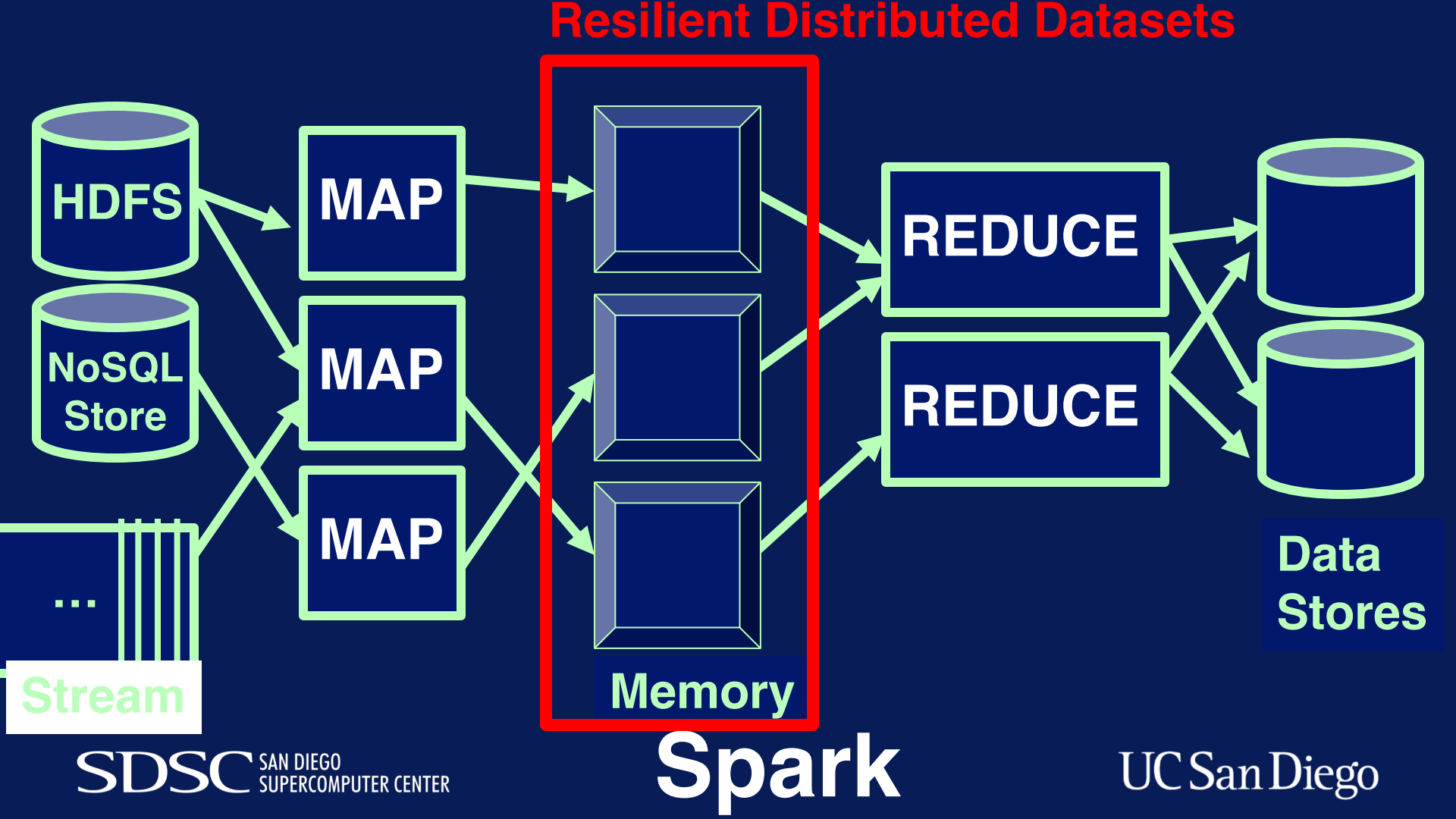
What does in memory processing mean?



MapReduce


HDFS
NoSQL Store
...
Stream
MAP
MAP
MAP
Memory
REDUCE
REDUCE
Data Stores

Spark


Resilient Distributed Datasets
HDFS
NoSQL Store
...
Stream
MAP
MAP
MAP
Memory
REDUCE
REDUCE
Data Stores
SDSC SAN DIEGO SUPERCOMPUTER CENTER
Spark
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]

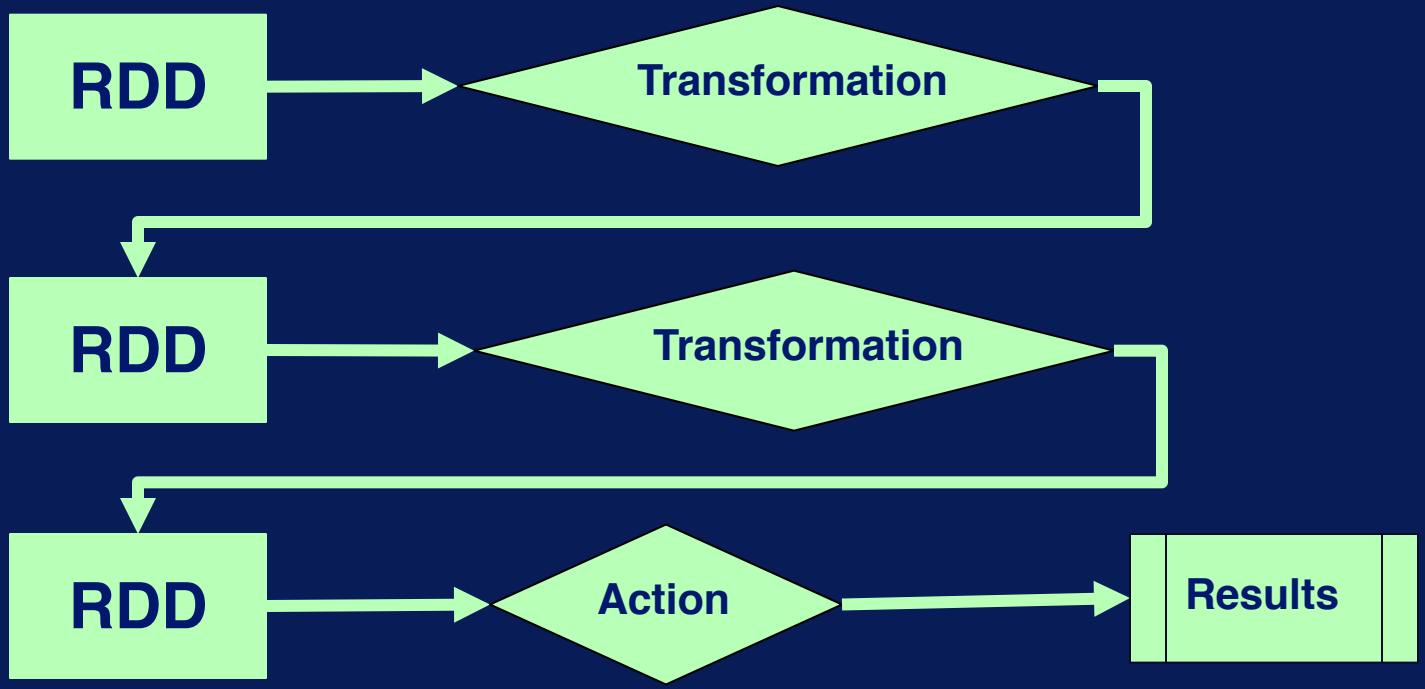
Parallelize range output into 3 partitions

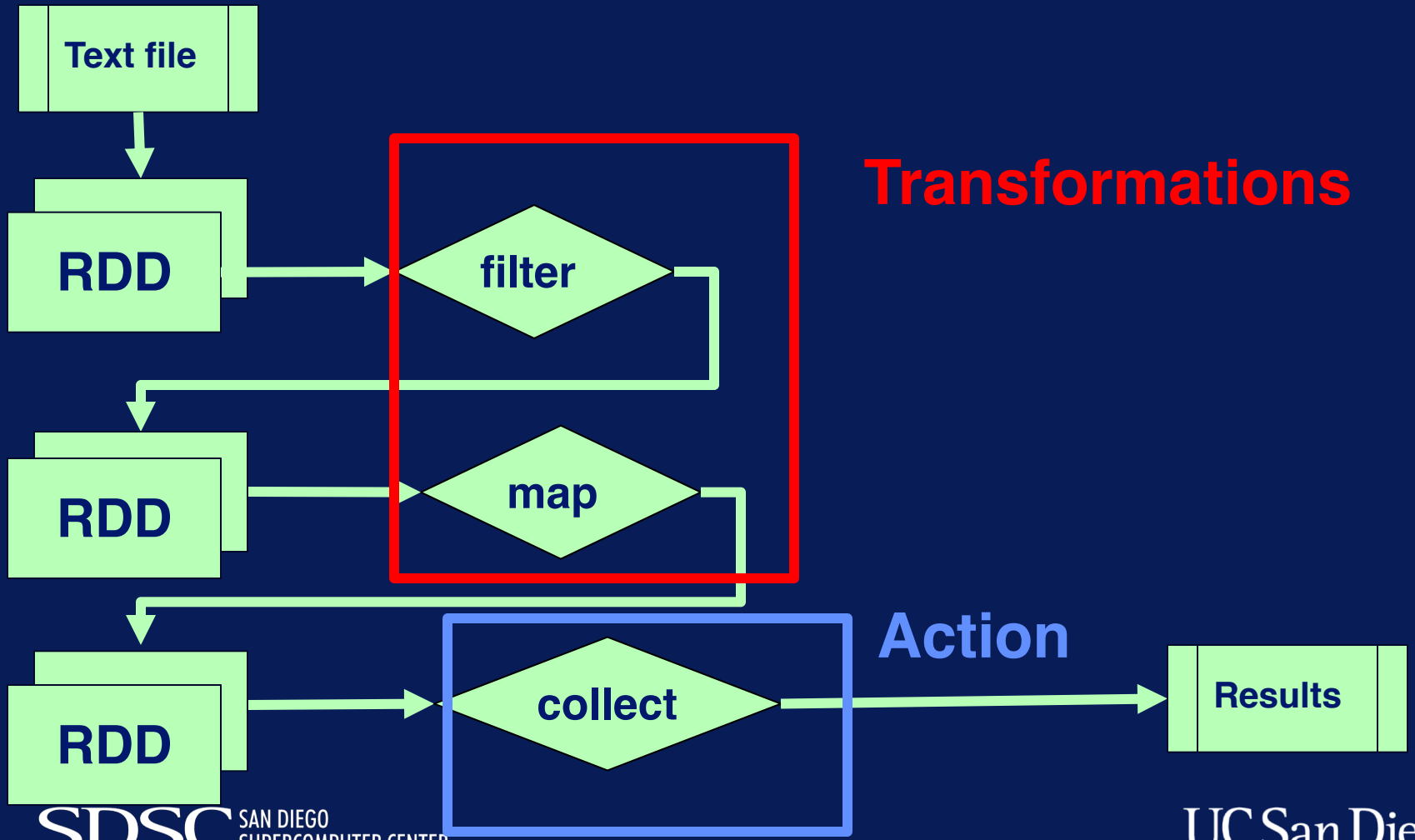
[0, 1, 2], [3, 4, 5], [6, 7, 8, 9]

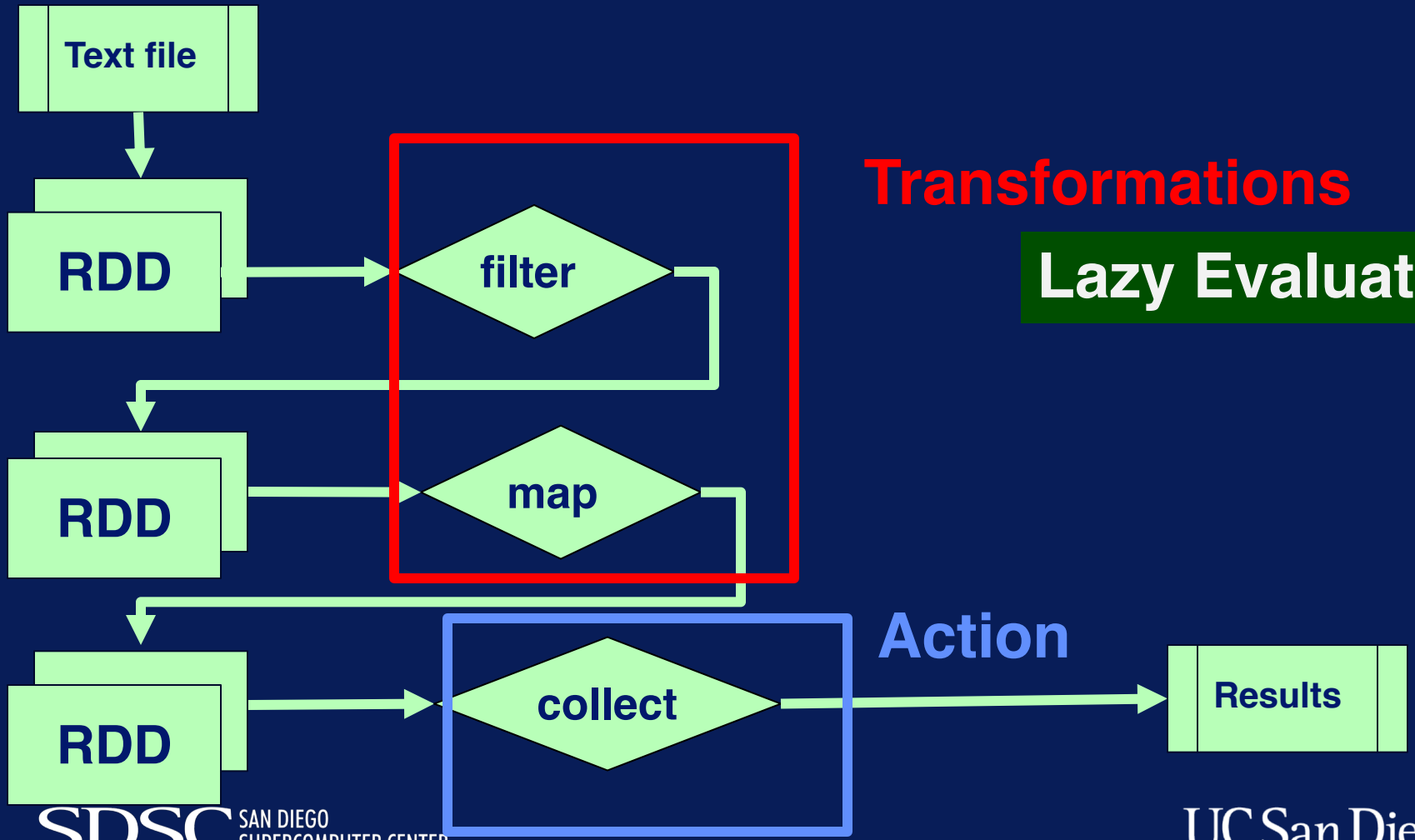
```
numbers.collect()
```

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

Processing RDDs




Text file
RDD
filter
RDD
map
RDD
collect
Transformations
Action
Results


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

Transformations & Actions

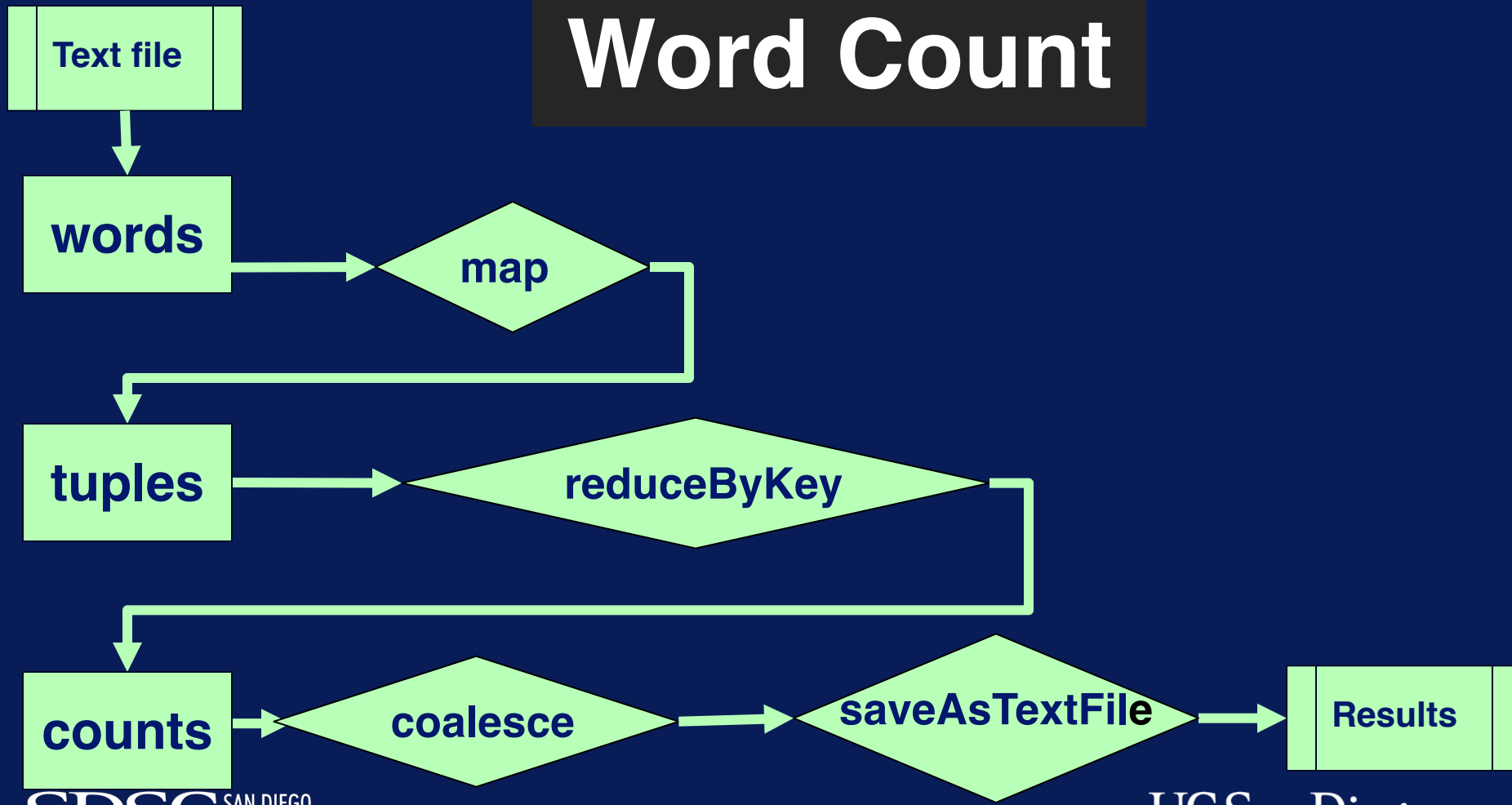
Transformations

- map
- filter
- coalesce
- reduceByKey

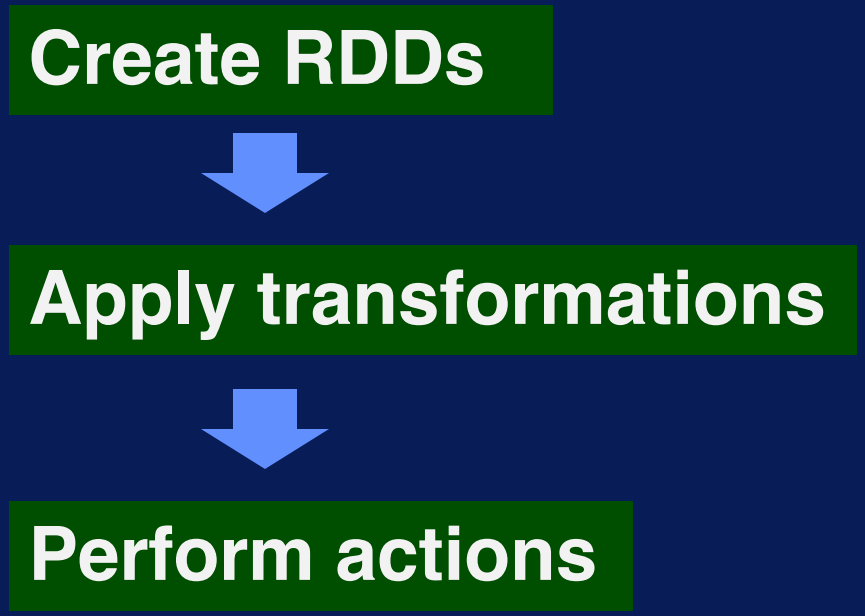
Actions

- collect
- take
- reduce
- saveAsText

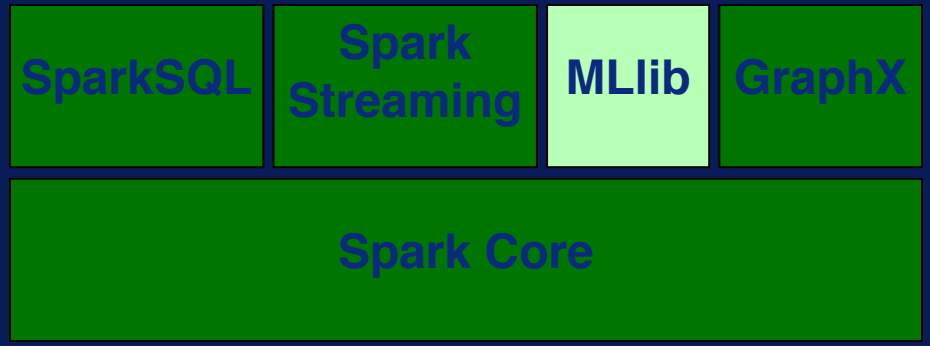
Word Count


Text file
words
map
tuples
reduceByKey
counts
coalesce
saveAsTextFile
Results

Programming in Spark



Spark MLlib: Machine Learning



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 –Statistics

```
from pyspark.sql.functions import rand

# Generate random numbers
df = sqlContext.range(0,10)
                .withColumn('rand1', rand(seed=10))
                .withColumn('rand2', rand(seed=27))


# Show summary statistics
df.describe().show()


# Compute correlation
df.stat.corr('rand1','rand2')
```

MLlib Example – Clustering

```
from pyspark.ml.clustering import KMeans

# Read and parse data
data = spark.read.csv("data.csv",inferSchema="true",
                      header="true")


# k-means model for clustering
kmeans = Kmeans().setK(3).setSeed(123)
model = kmeans.fit (data)
for center in model.clusterCenters()
   print (center)
```

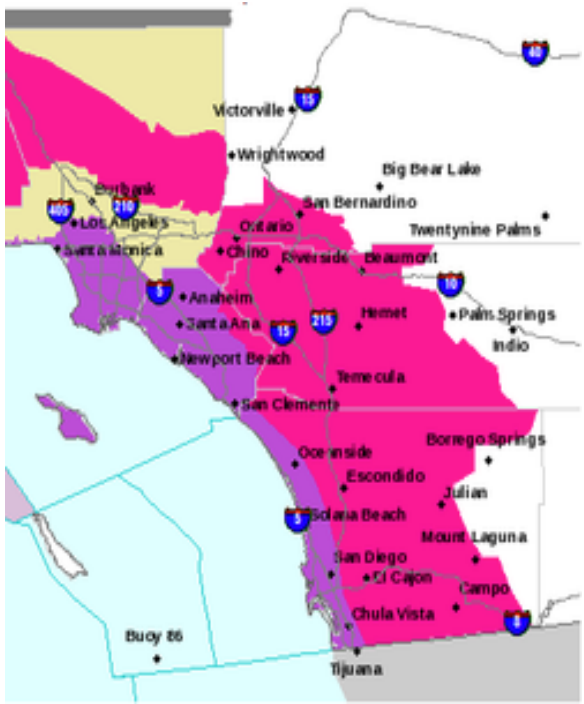
Spark MLlib

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

Spark Use Case

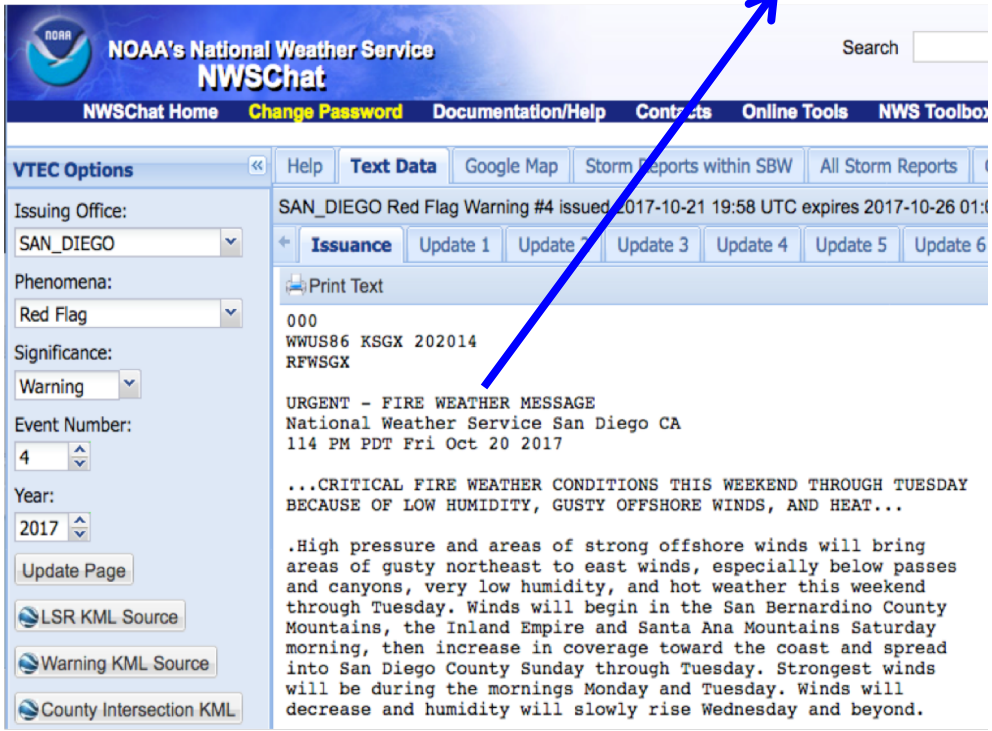
- **Santa Ana conditions**
  - Dry, windy weather patterns
  - Significantly increase dangers of wildfires
- **Research Goal: Detect Santa Ana conditions**
  - Location-specific & time-specific detection
  - To focus firefighting efforts on region with increased risks

Red Flag Warnings "… issued 2017-10-21 19:58 UTC expires 2017-10-26 01:00 UTC"



pink: red flag warning
purple: high surf advisory

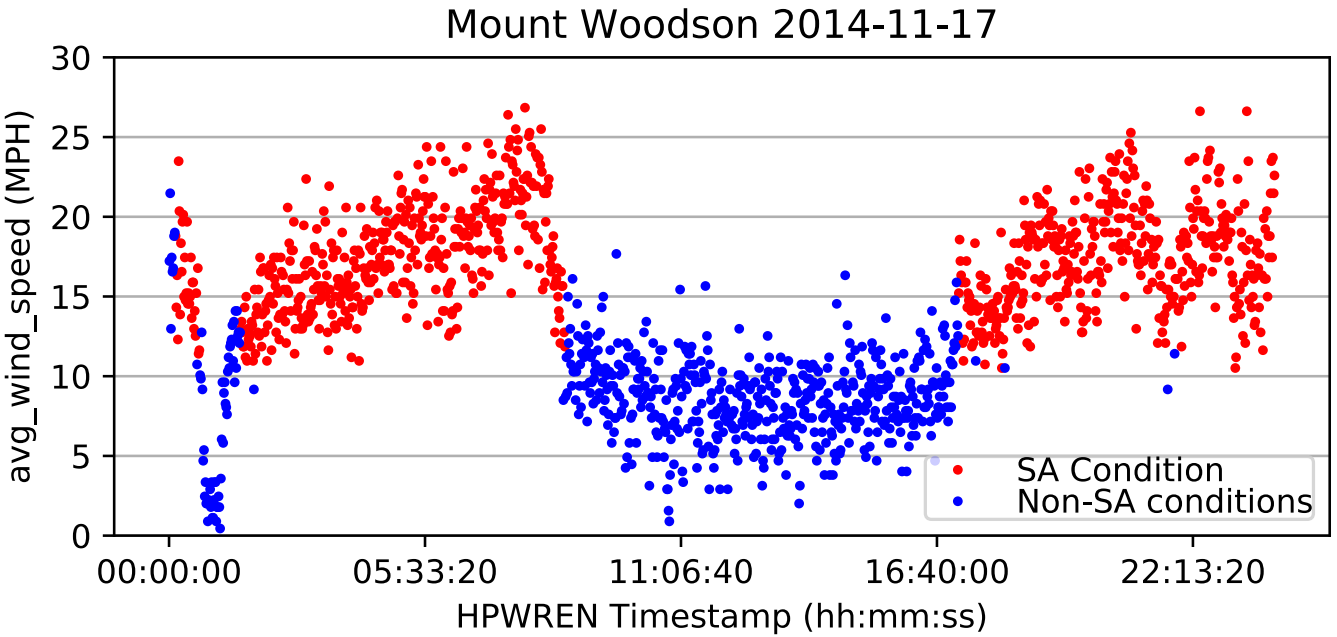
Source: http://www.kpbs.org/news/2014/jan/24/warnings-wildfire-risk-high-surf-continue-san-dieg/



Source: https://nwschat.weather.gov/vtec/

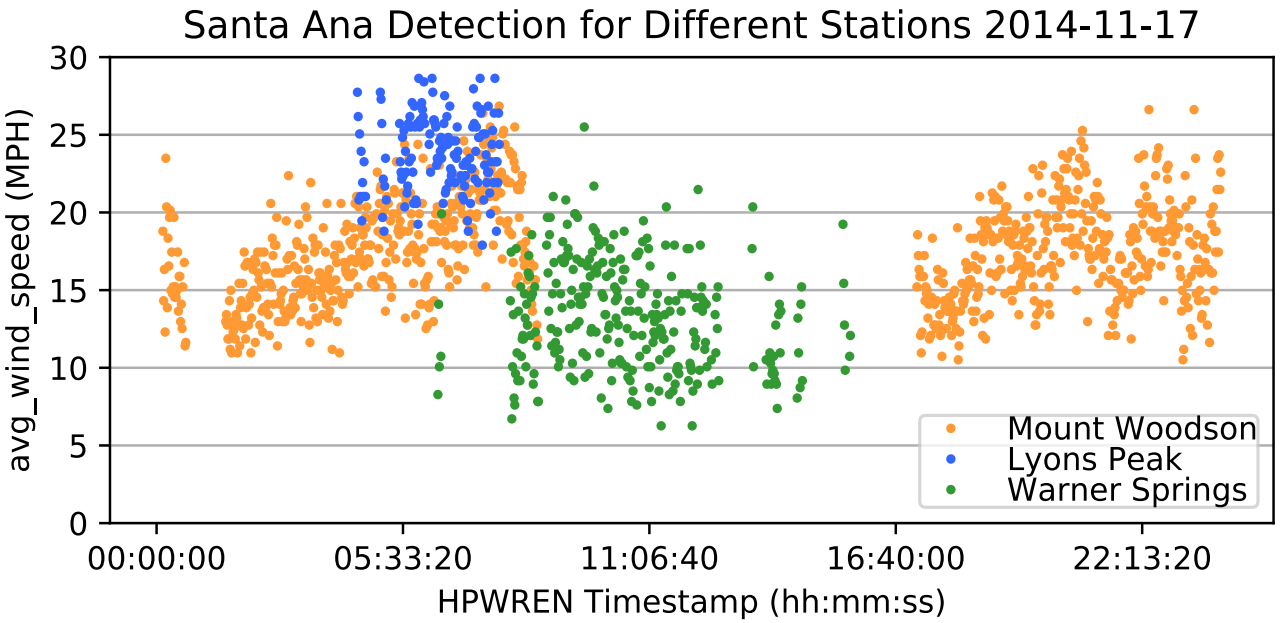
Detecting Santa Ana Conditions

2014-11-17: Red Flag warning issued

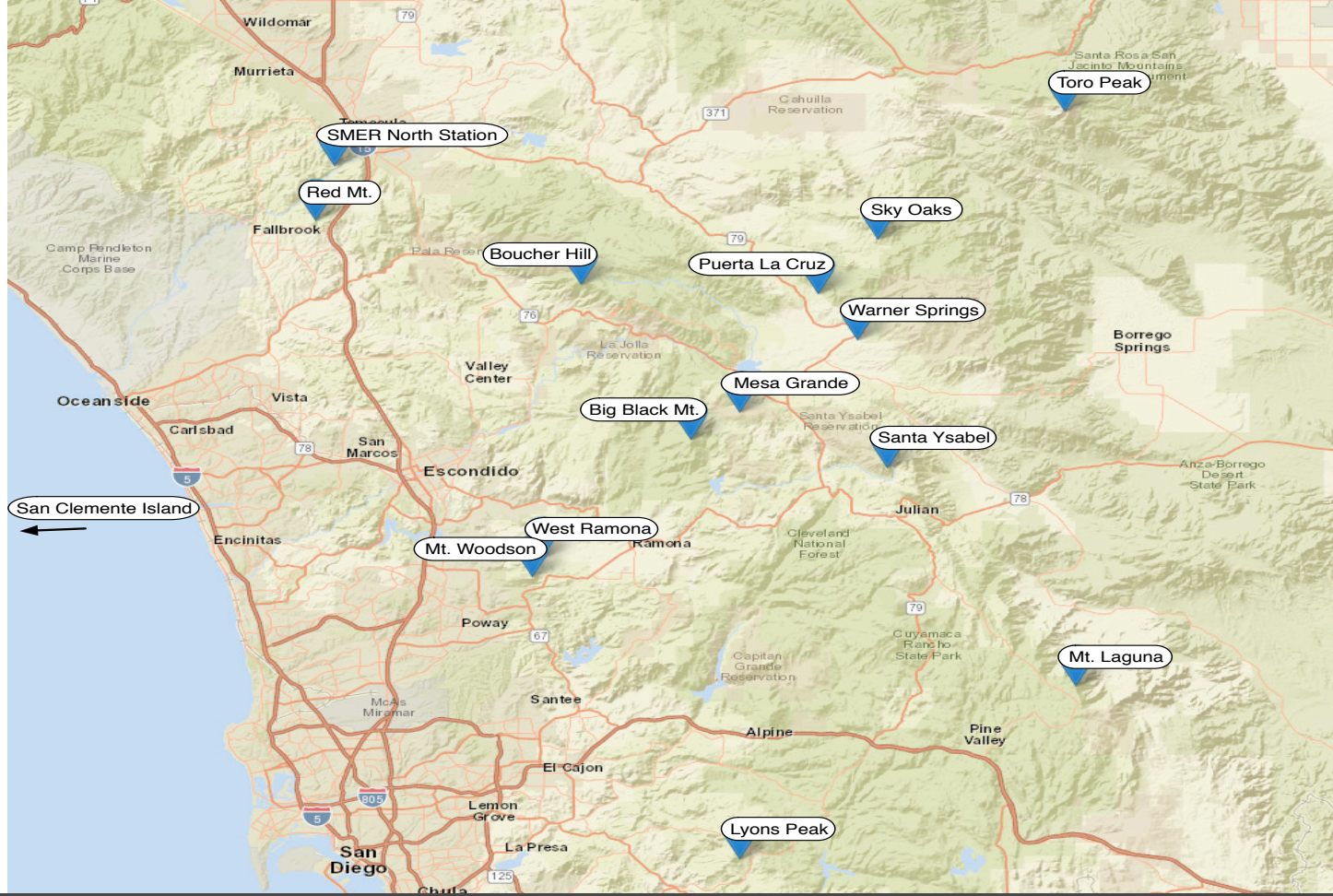


Santa Ana Detection – Multiple Stations

Mt Woodson, Lyons Peak, Warner Springs on 2014-11-17

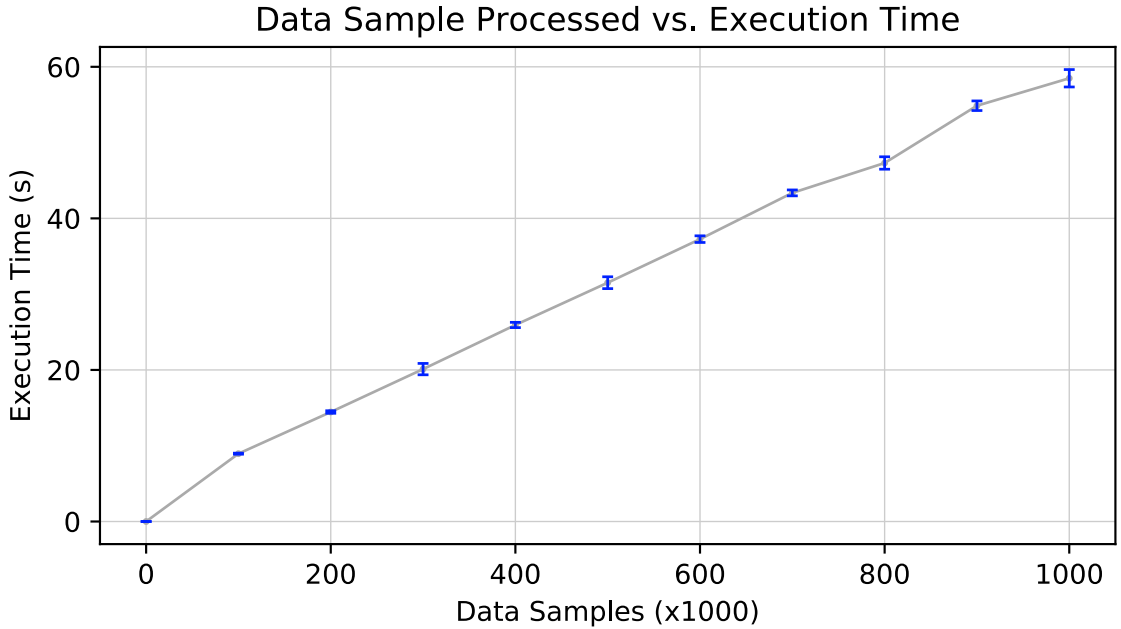


Map of HPWREN Weather Stations

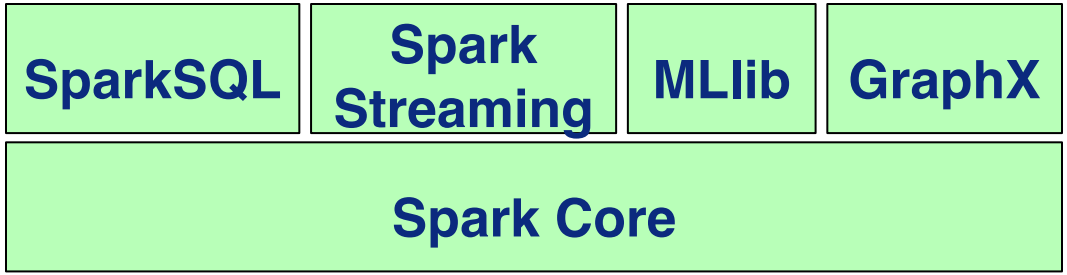


Scalability of System

- System can process up to 1M samples in 58.49 seconds (i.e., 1M stations every minute)



Scalable Machine Learning Summary



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

Spark Resources

- Spark
  - https://spark.apache.org/
- MLlib
  - https://spark.apache.org/mllib/
- Mastering Apache Spark
  - https://jaceklaskowski.gitbooks.io/mastering-apache-spark/content/

Questions?