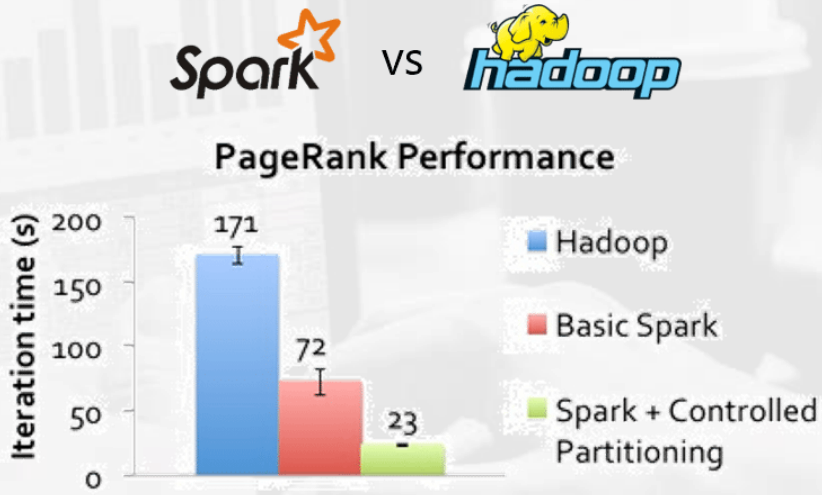
BIG DATA ANALYTICS

MAP REDUCE

Course Outline

- Introduction – Distributed Systems
- MapReduce – How does it work?
- Programming in Spark

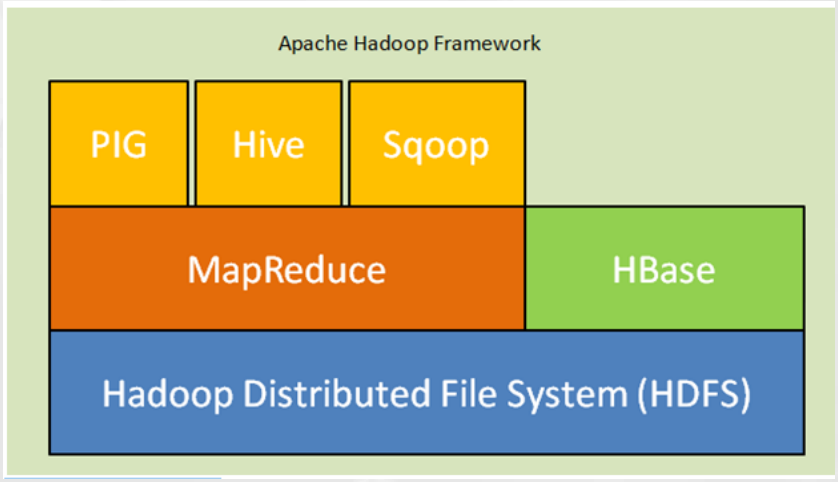


Introduction – Distributed Computing Systems

Hadoop is a distributing computing framework.

Uses HDFS to solve distributed data problem.

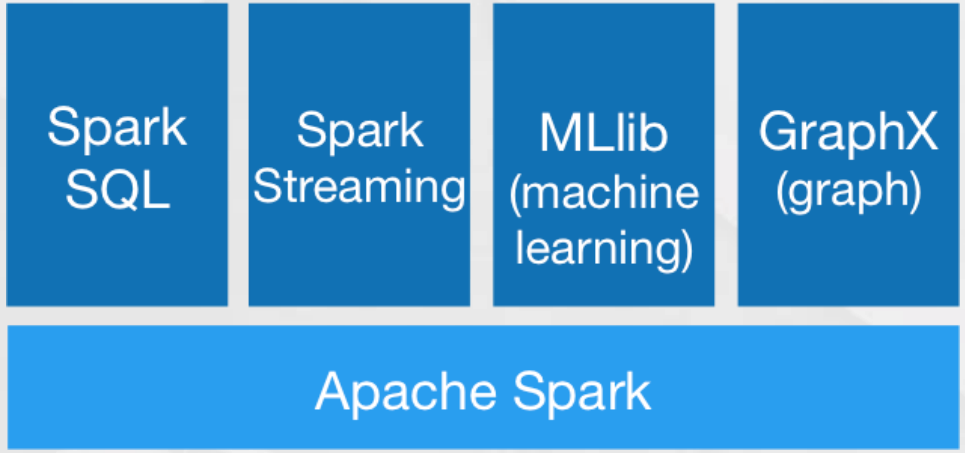
Uses MapReduce to provide effective distributed computation.



Spark is a general purpose cluster computing framework.

Supports MapReduce but provides additional functionality.

Very useful for machine learning and optimization.



MapReduce – What it is?

Definition:

- It is a software framework for processing large datasets using several machines.

Objective:

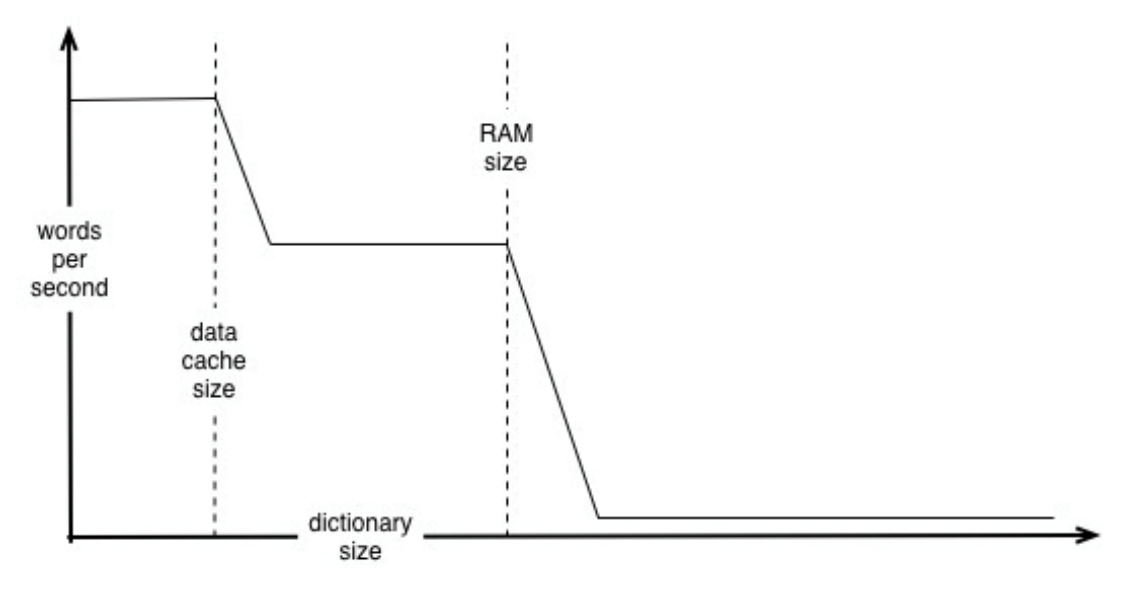
- Map data into (key, value) pairs and reduce all pairs with **same key.**

- **Use Cases:**
  - **Counting Word Frequencies**
  - **Distributed Sorting**
  - **Machine Learning**

Counting Words – Single Machine vs MapReduce

A central data structure – In a single computer, can be your RAM, Disk, etc.

Computation - Your program will run on only one processor, -- speed is constant

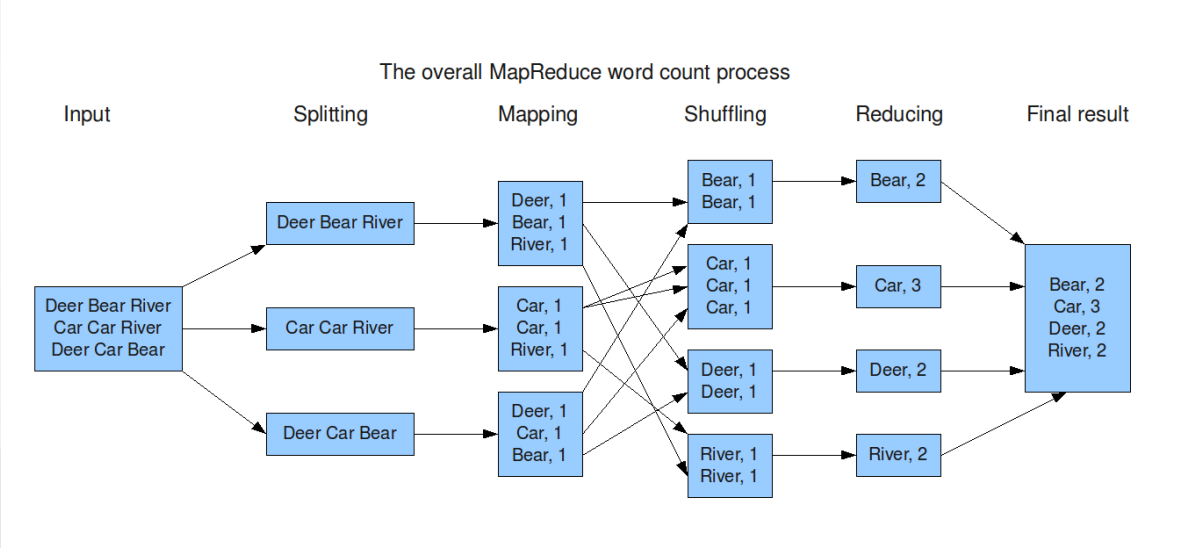


Counting Words – MapReduce Approach

Data – No central storage, therefore there will be no memory issues

MapReduce has 3 steps:

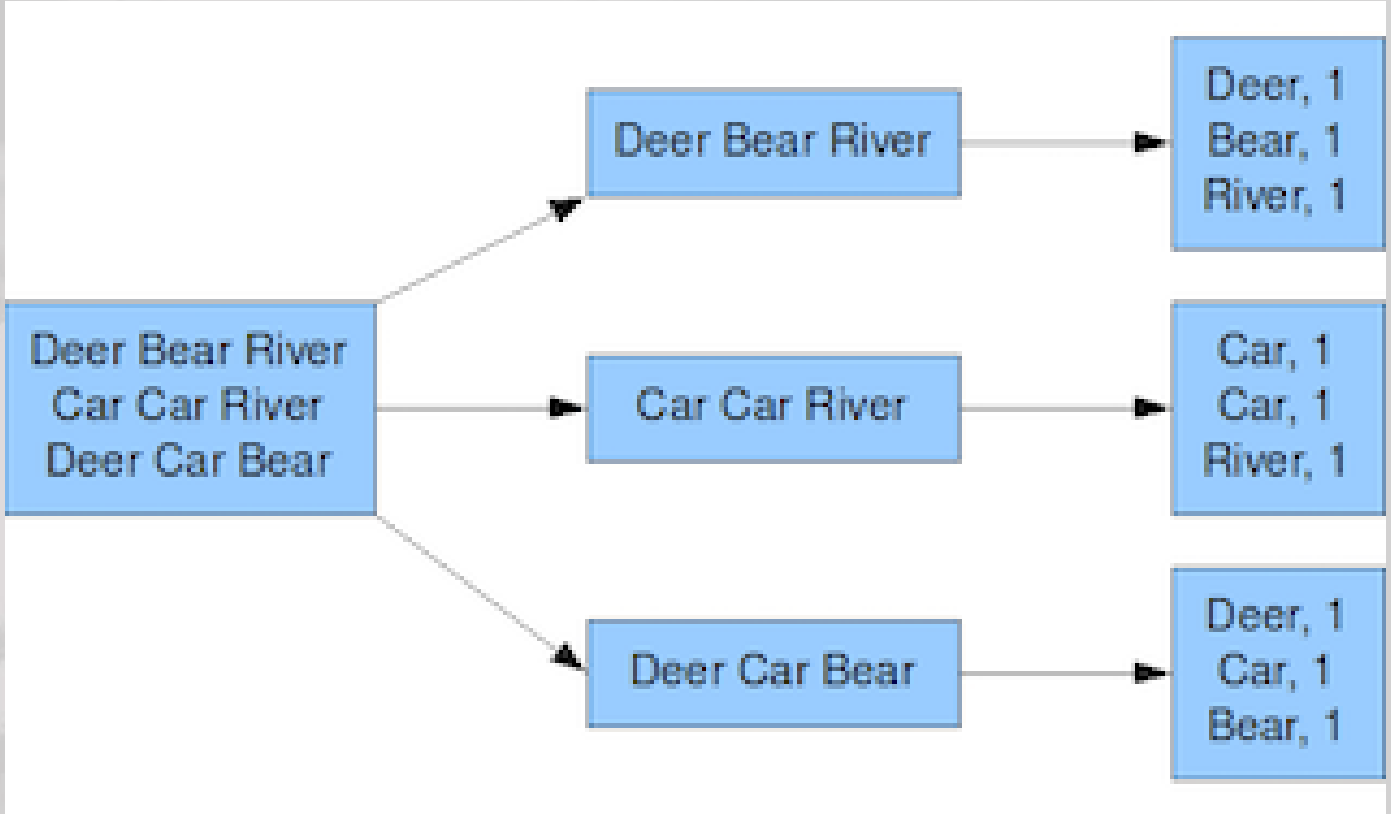
- Mapping step – Produce intermediate results & associates with an output **key**
- Shuffling step – Group intermediate results with same output **key**
- Reducing step – Process groups of intermediate results with same output **key**



Counting Words: Mapping

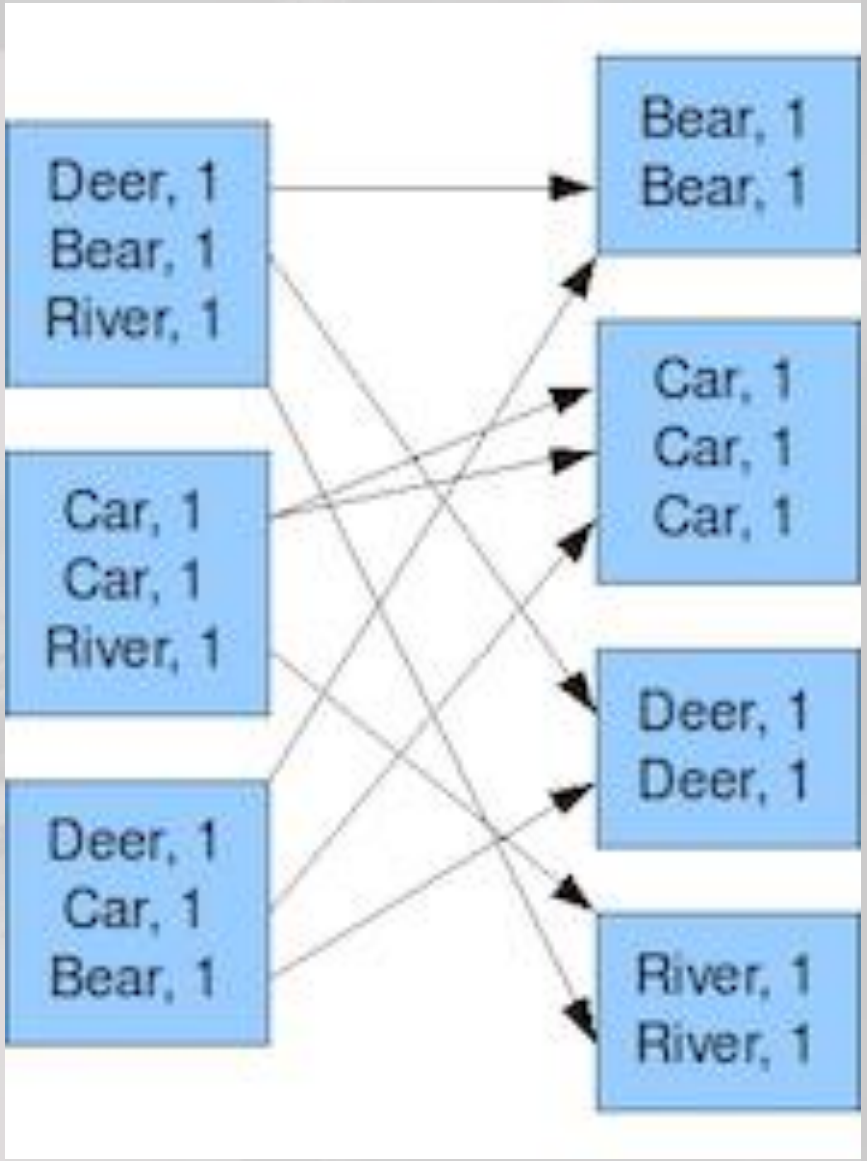
- Apply a function to each element.
- In this case function is:

  # The Word and the occurrence



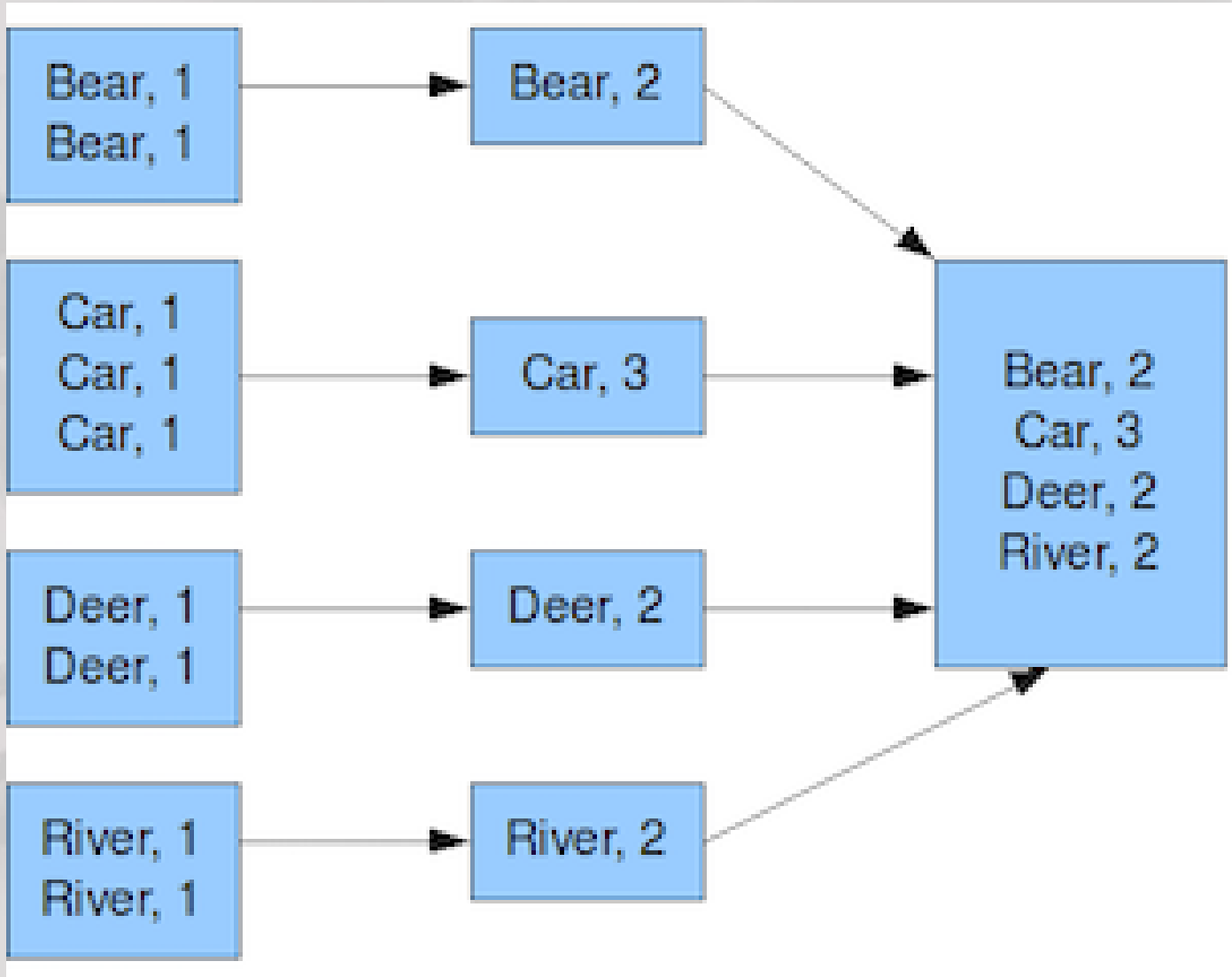
Counting Words: Shuffling

- Gather all the groups.
- In this case, sort them.



Counting Words: Reducing

- Finally, count the number of values with the same key.
- Sum all same keys together, restart sum when key changes



MAP REDUCE ALGORITHM

Advantages:

- Each three steps, they are absolutely separable. No need to run in the same computer.
- Mapping function, can be split into many independent parallel tasks.
- Shuffling and reducing can also be parallelized easily and write results into an output file.

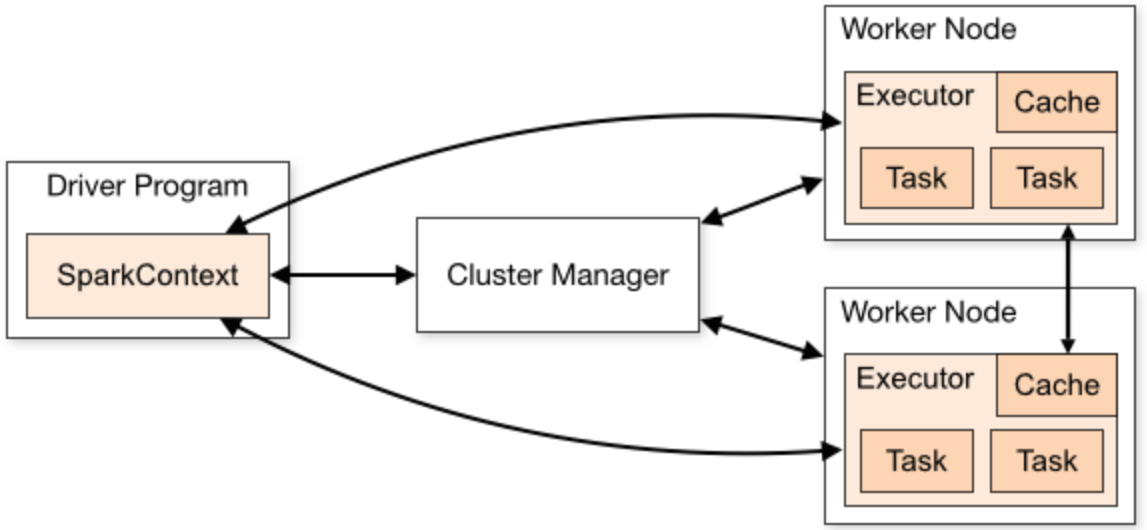
Disadvantages:

- Not for all purposes. Problem can be broken up to map-reduce paradigm.

```
from functools import reduce
from itertools import groupby

# ========= Mapping ==========
words = ['Deer', 'Bear', 'River', 'Car',
         'Car', 'River', 'Deer', 'Car', 'Bear']
mapping = map((lambda x: (x, 1)), words)
print(mapping)
# output:
# [('Deer', 1), ('Bear', 1), ('River', 1), ('Car', 1),
# ('Car', 1), ('River', 1), ('Deer', 1), ('Car', 1), ('Bear'
, 1)]

# ========= Shuffling ==========
sorted_mapping = sorted(mapping)
print(sorted_mapping)
# output:
# [('Bear', 1), ('Bear', 1), ('Car', 1), ('Car', 1),
# ('Car', 1), ('Deer', 1), ('Deer', 1), ('River', 1), ('Rive
r', 1)]

# ========= Reducing ==========
grouper = groupby(sorted_mapping, lambda p: p[0])
final = map(lambda l: (l[0], reduce(lambda x, y: x +
                                    y, map(lambda p: p[1], l
[1]))), grouper)
print(list(final))
# output:
# [('Bear', 2), ('Car', 3), ('Deer', 2), ('River', 2)]
```

Next Week

- RDD is a Resilient Distributed Data set
- Setting up Spark Environment
- First Spark App – Counting Words



Thank You!

Feel free to email me for any questions. Reach me at msenturk@saintpeters.edu. Download class material at the following link.