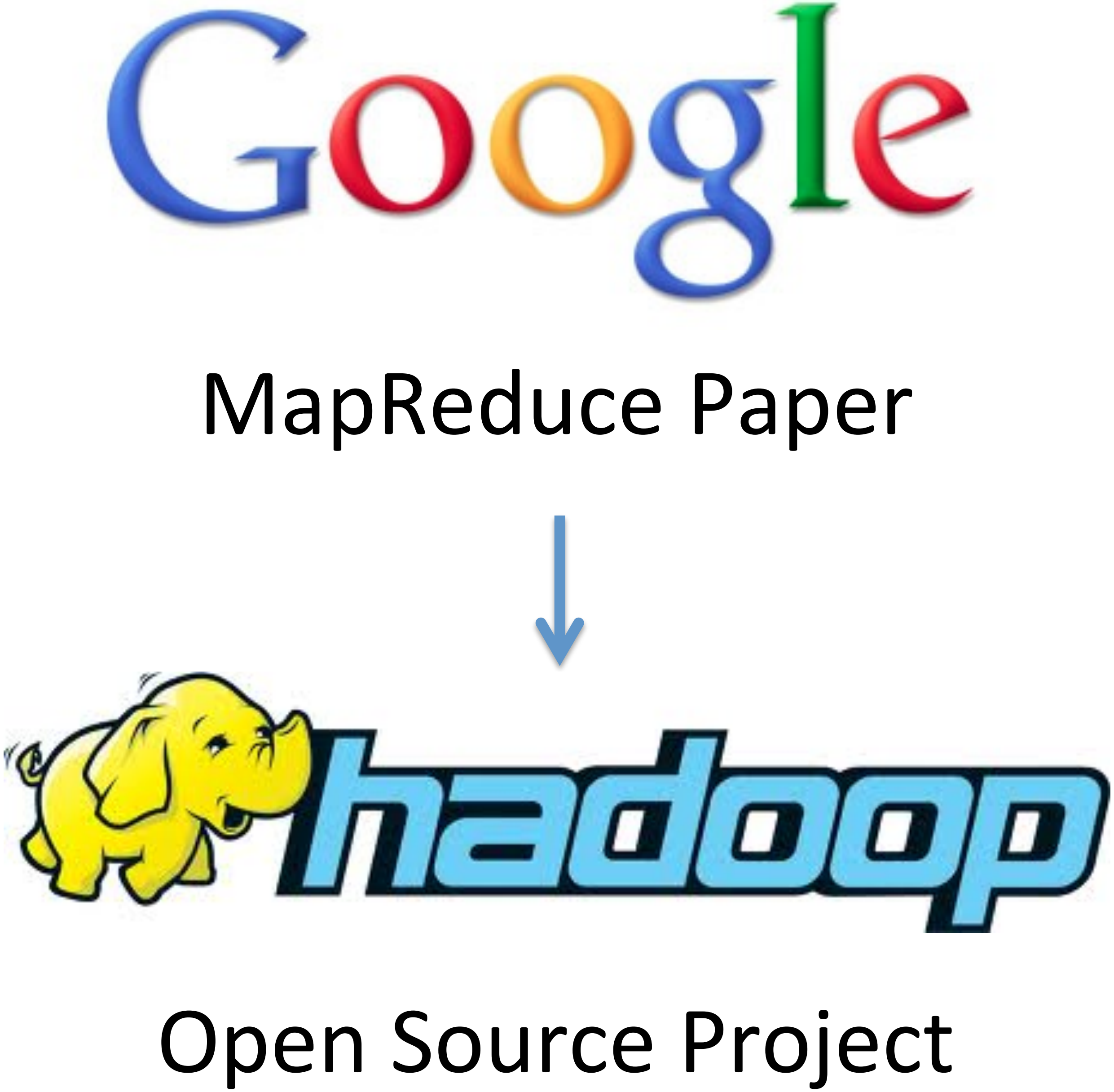
# Scaling Up with MapReduce, Hadoop, and Amazon

# Term Frequency

| | |
|---|---|
| Kenneth Lay | 15 MB |
| Enron | 1,300 MB |
| GMail | >1,000,000,000 MB |

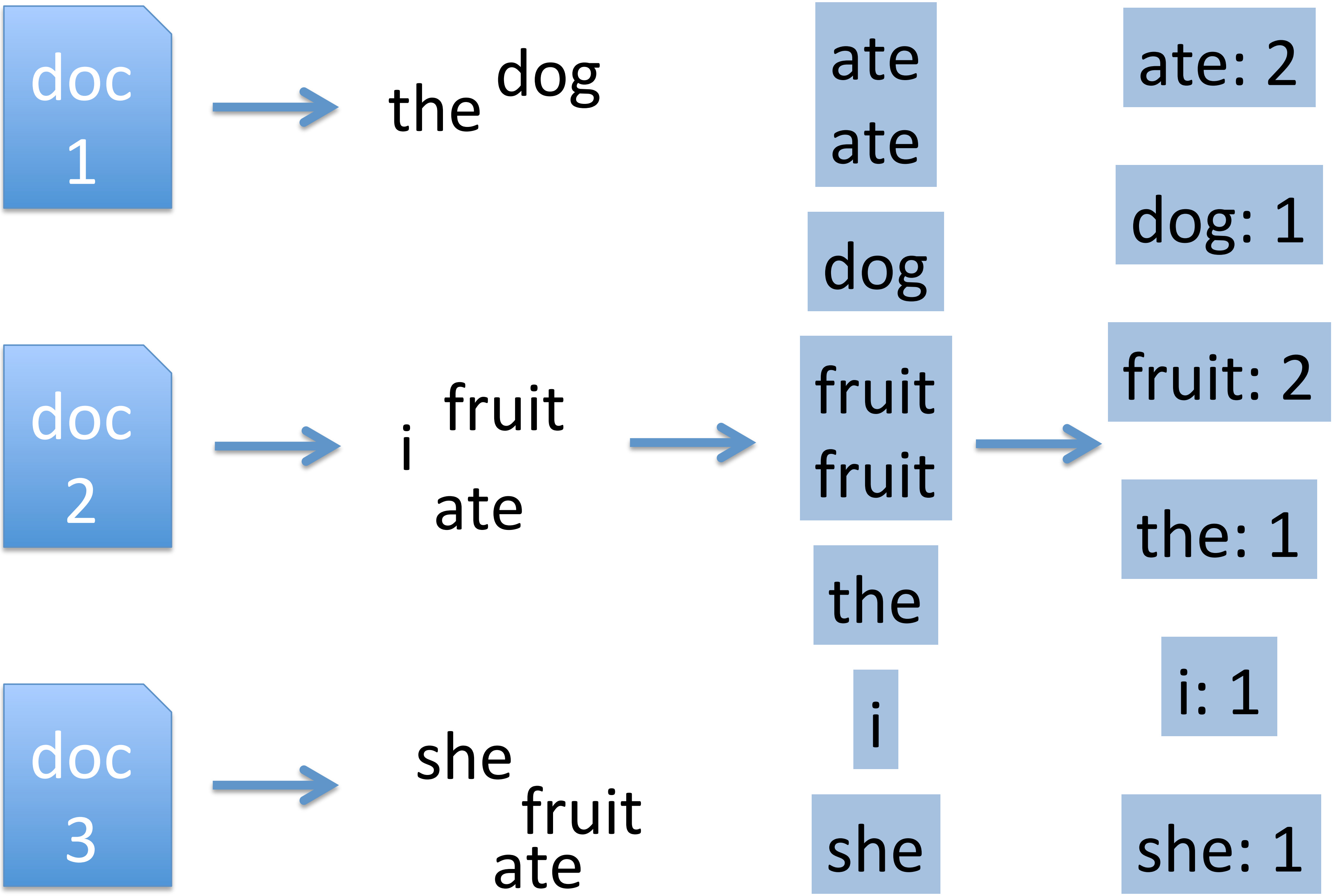
# Parallelism



MapReduce Paper

Open Source Project

# Common Pattern



doc 1

the dog

doc 2

i fruit ate

doc 3

she fruit ate

ate
ate

dog

fruit
fruit

the

i

she

ate: 2

dog: 1

fruit: 2

the: 1

i: 1

she: 1

Loop

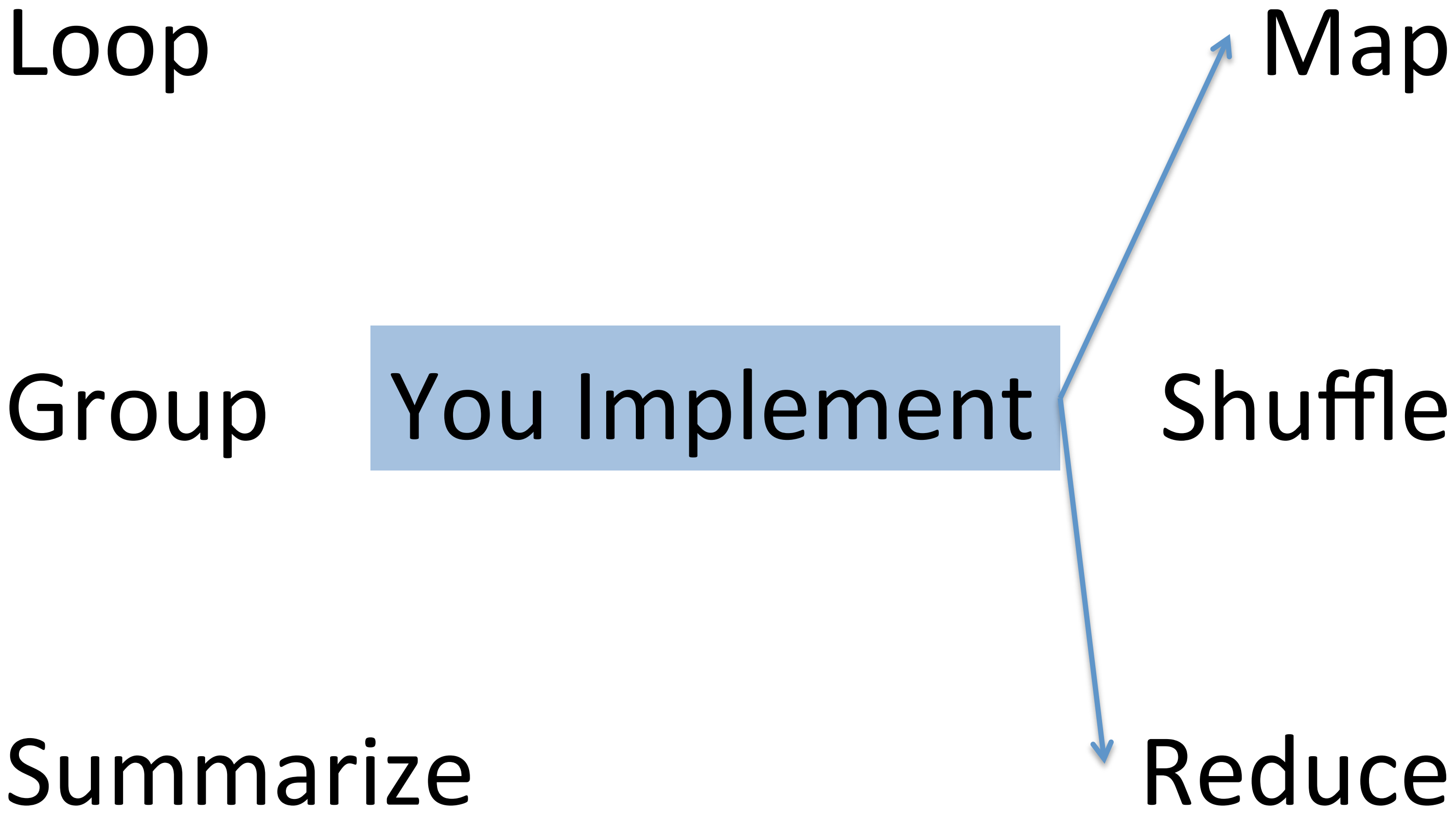
Group

Summarize

| | Word Count |
|---|---|
| **Loop** | words in documents |
| **Group** | instances of a word |
| **Summarize** | count instances of a word |

|  | Word Count | Candidates |
|---|---|---|
| **Loop** | words in documents | lines in csv |
| **Group** | instances of a word | candidate, day |
| **Summarize** | count instances of a word | sum of contributions by candidate, day |

# MapReduce



Loop

Map

Group

You Implement

Shuffle

Summarize

Reduce

# Amazon Provides Compute Power

Elastic MapReduce (EMR): Computers

Simple Storage Service (S3): Files

# S3 stores files

- Create bucket (unique name)
- Files in bucket
- Access via amazon web console
- Access via programmatic API

# Amazon Charges Us Money

- S3: 14 cents per GB per month
- EMR: 10 cents per machine per hour
  - 1 minute = 1 hour
  - Ask us if you use more than 200 hours
  - Excess gets charged to our credit card

# Slower Than You Think

Scale, not Performance

1) Account: http://shoutkey.com/private

2) git pull

3) Lab: http://dataiap.github.com/dataiap/day5/mapreduce