

## Overview Programming Model

- ▮ Inspired by functional programming languages
- ▮ **Implicit parallelism** (abstracts distributed storage and processing)
- ▮ **Map** function: key/value pair → set of intermediate key/value pairs
- ▮ **Reduce** function: merge all intermediate values by key

## Example

- given

Name	Dep
X	CS
Y	CS
A	EE
Z	CS

- sql equivalent

```
SELECT Dep, count(*) FROM csv_files GROUP BY Dep
```

- MapReduce

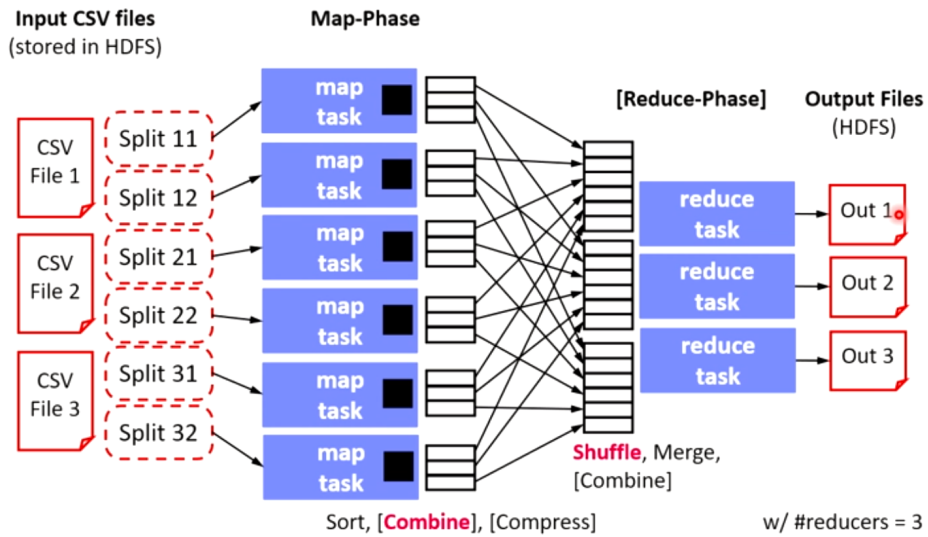
```
map(Long pos, String line) {  
  parts ← line.split(",")  
  emit(parts[1], 1)  
}
```

CS	1
CS	1
EE	1
CS	1

```
reduce(String dep,  
        Iterator<Long> iter) {  
  total ← iter.sum();  
  emit(dep, total)  
}
```

CS	3
EE	1

## Execution Model



## Criticism

- performance
  - high latency
- low level [[API]]
- many different systems