

Chapter 4: Parallel Program Structures V

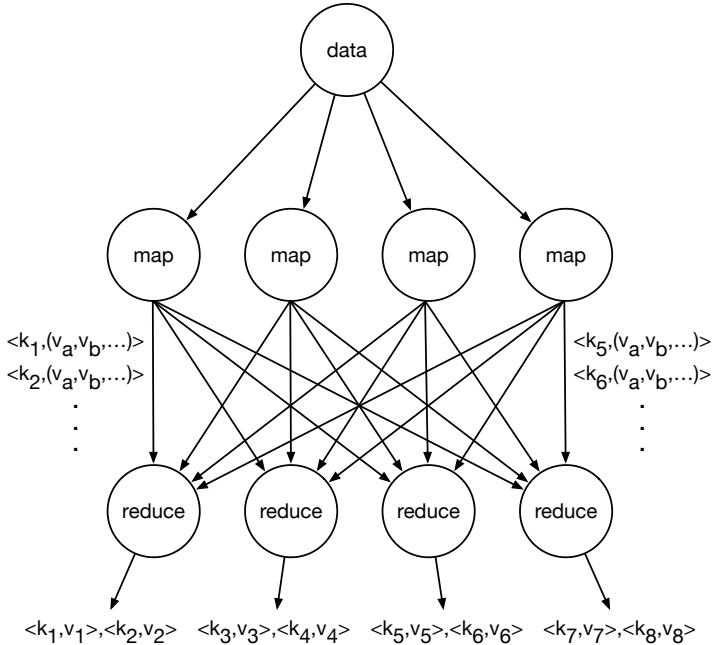
Elements of Parallel Computing

Eric Aubanel

MapReduce

- ▶ Google, 2004: “Simplified data processing on large clusters”
- ▶ Hadoop and other frameworks

- ▶ *Map*: applied to set of key-value pairs
- ▶ *Shuffle*: gathers key-value pairs with the same key into key-list(value) pairs. This phase not visible to the programmer.
- ▶ *Reduce*: aggregates results for each key



Word Count

```
Procedure map(line)
    while line has more words do
        output(word, 1)
    end
end
```

```
Procedure reduce(key, list(value))
    sum  $\leftarrow$  0
    foreach value in list do
        sum  $\leftarrow$  sum + value
    end
    output(key, sum)
end
```

Combine phase

- ▶ Mapper can produce too many tasks
- ▶ Combine phase: combines all key-value pairs from each map with a user-defined function, which for word count is the same as the reduce function
- ▶ Lessens workload of shuffle phase

K-means Clustering with MapReduce

```
Procedure map(vector, cluster)  
    find closest cluster center to vector  
    output(centerIndex, vector)  
end
```

```
Procedure reduce(centerIndex, list(vector))  
    sum  $\leftarrow$  0  
    foreach vector in list do  
        sum  $\leftarrow$  sum + vector  
    end  
    output(centerIndex, sum, sizeof(list))  
end
```

Improved K-means Mapper

```
Procedure map(list(vector), cluster)  
    foreach vector in list do  
        find closest cluster center to vector  
        output(centerIndex, vector)  
    end  
end
```

Guidelines for Distributed Memory Programming

1. Data distribution should minimize communication
2. Global communication routines should be used when appropriate