

# Map-Reduce Design Patterns

**Venkatesh Vinayakarao**

venkateshv@cmi.ac.in

<http://vvtesh.co.in>

---

Chennai Mathematical Institute

---

**Finding patterns is the essence of wisdom. – Dennis Prager**

# I have a story for you!

Patterns here, Patterns there,  
Patterns, Patterns everywhere...

# Are beauty and quality objective?

- Can we agree some things are beautiful and some are not?



# Christopher Alexander Asked... What is a good design?

What makes a  
bad  
architectural  
design?

What makes a  
good  
architectural  
design?

Can we  
recognize  
good  
design?

Are beauty  
and quality  
objective?

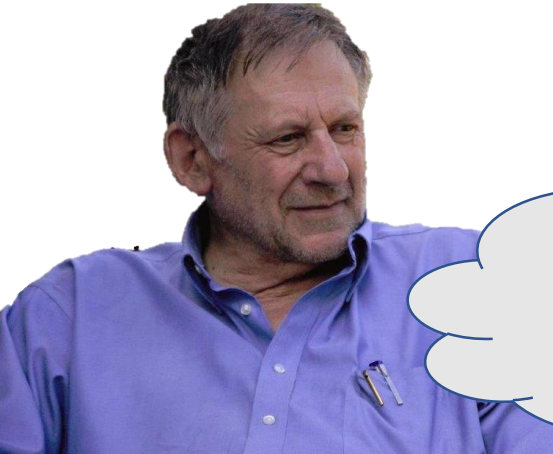
Is there a  
basis for  
describing  
common  
consensus



# Good Design

**Cultural Anthropology: Within a culture, individuals agree what is good design, what is beautiful.**

**Example: Symmetry is good**



**Beauty can be objectively measured.**

# Patterns

- Good design structures had similarities between them.
- Alexander called these similarities *patterns*.

*A pattern is a solution to a problem in a context.*

- "Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over..."

Christopher Alexander, A Pattern Language:  
Towns/Buildings/Construction, 1977

# A question

- Can you tell me one design that is absolutely symmetrical?

**Equivalent ideas exist in software design**

# Good Design

- What according to you are the two biggest factors that determine a good/bad design?



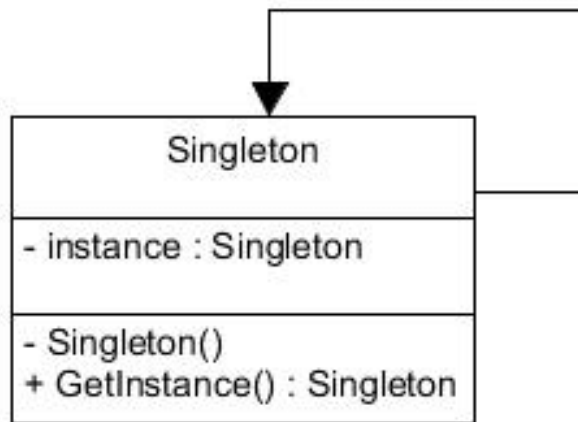
# Good and Bad Design

- What are the commonalities in what is viewed as good (and what is viewed as bad)?
  - A software system that is easy to maintain is considered good
    - A fragile software system is considered bad
  - A software system that is easy to understand is considered good
    - Obfuscated “spaghetti code” is considered bad

# Quiz

- Have you come across software patterns? Can you give one example?

# Singleton Pattern

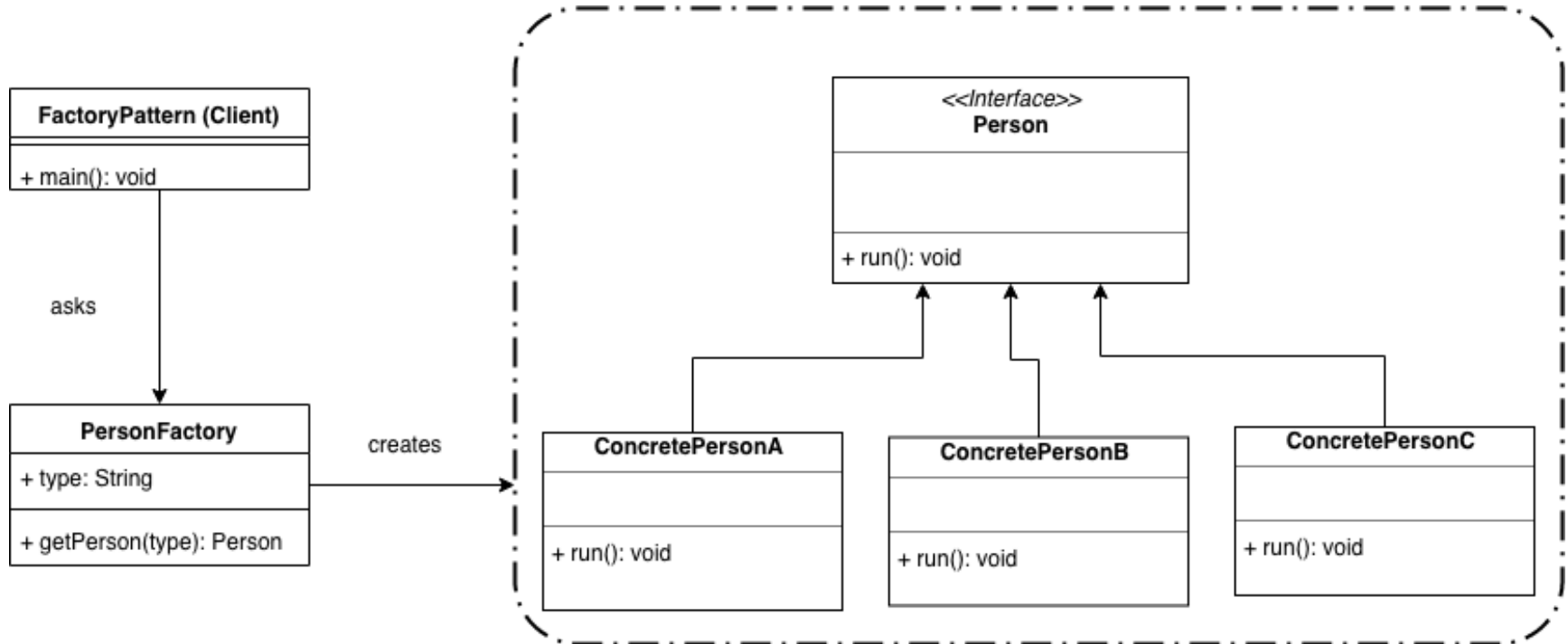


```
public class CMI
{
    private static final CMI instance = new CMI();

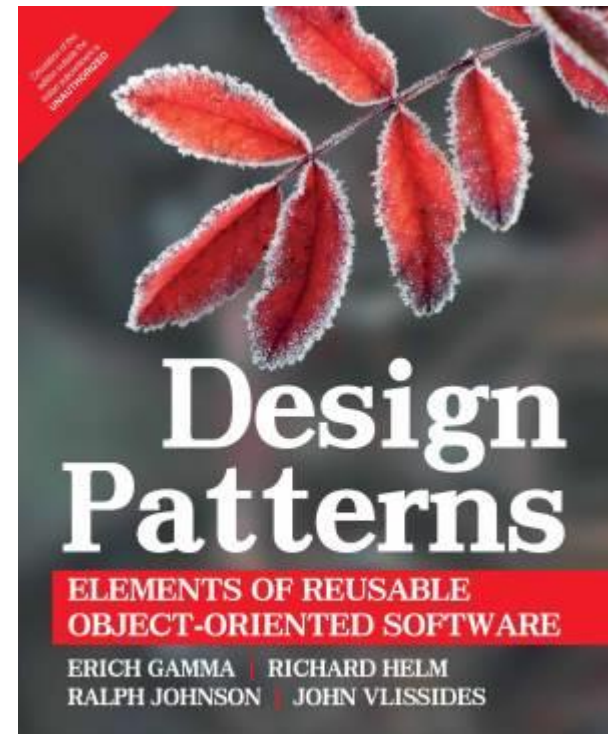
    private CMI()
    {
        // private constructor
    }

    public static CMI getInstance(){
        return instance;
    }
}
```

# Factory Pattern



# For More on Design Patterns



We shall now look at some Map-Reduce design patterns.

# Recap

## Hadoop Architecture

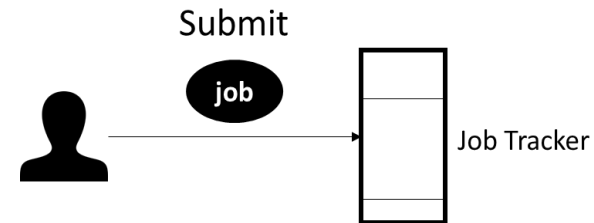
Application  
(map-reduce)

Application  
(pig)

Application  
(nosql db)

**YARN**  
(Resource Management – Job Scheduling/Monitoring)

**HDFS**  
(Replicated Reliable Storage)

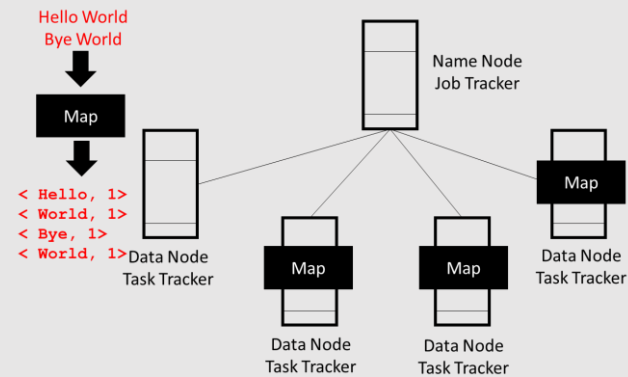


## Map-Reduce Model

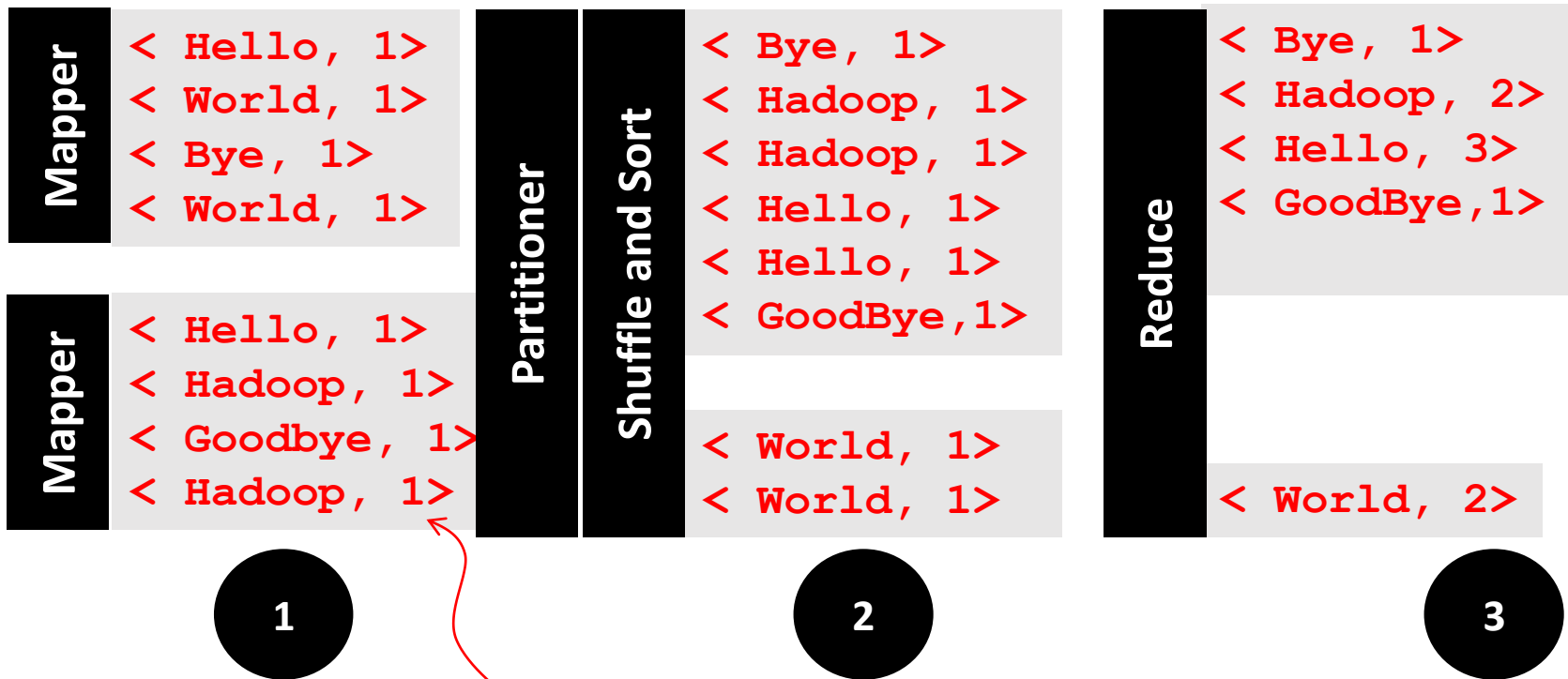
Map

Shuffle and Sort

Reduce

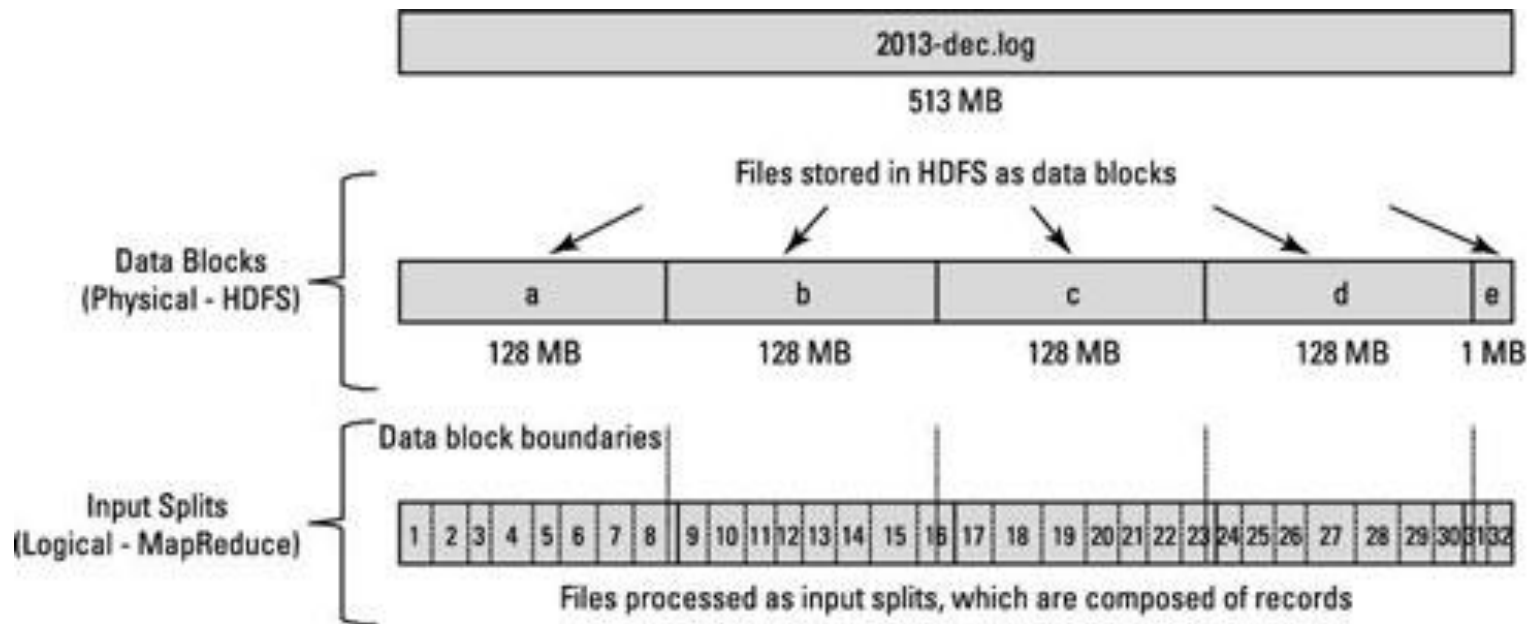


# Map-Reduce Processing



Combiner can be used to summarize locally per mapper.

# Input Splits



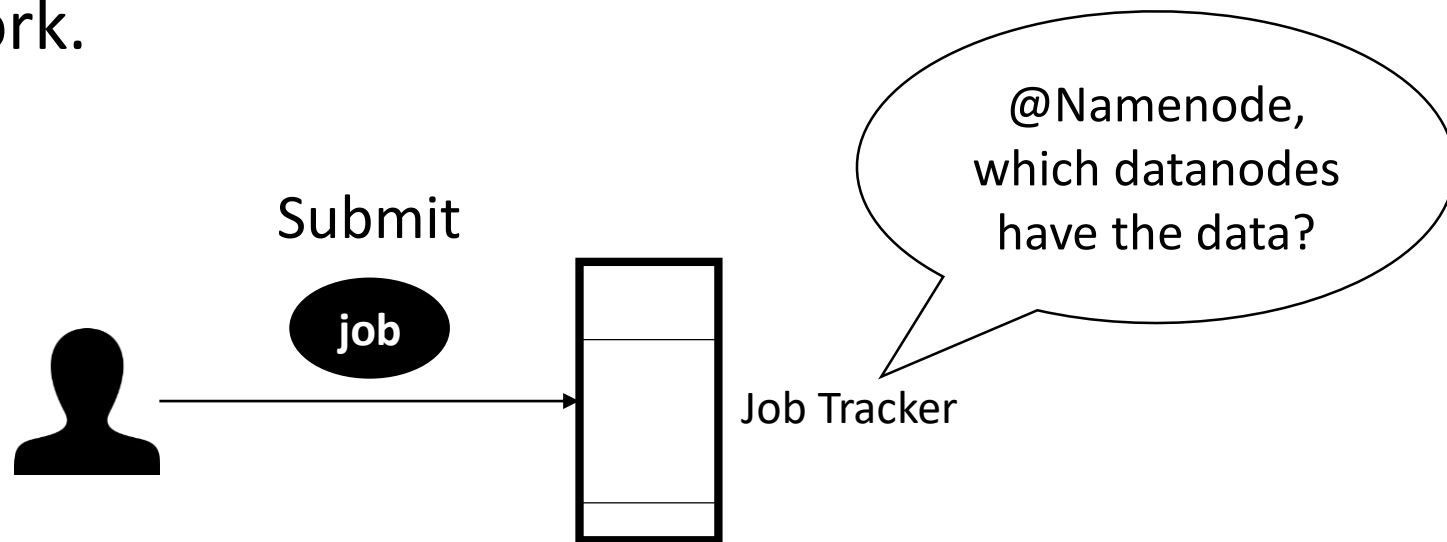
Note that a remote read may be required at block boundaries.

map:  $(K1, V1) \rightarrow \text{list}(K2, V2)$   
reduce:  $(K2, \text{list}(V2)) \rightarrow \text{list}(K3, V3)$



# A Hadoop Map-Reduce Developer

- Writes the “map” code
- Writes the “reduce” code
- Submits the map and reduce code to Hadoop framework.



# Submitting a Map-Reduce Job

hadoop jar

`/usr/joe/wordcount.jar`

`org.myorg.WordCount`

`/usr/joe/wordcount/input`

`/usr/joe/wordcount/output`

# Mapper

```
public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, one);
    }
}
```

See <https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html> for details.

# Reducer

```
public void reduce(Text key, Iterable<IntWritable> values,  
                  Context context  
                  ) throws IOException, InterruptedException {  
    int sum = 0;  
    for (IntWritable val : values) {  
        sum += val.get();  
    }  
    result.set(sum);  
    context.write(key, result);  
}
```

# Create a Job

```
public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```

# Submit Job to Hadoop

```
$ bin/hadoop jar wc.jar WordCount /user/joe/wordcount/input  
/user/joe/wordcount/output
```

```
$ bin/hadoop fs -ls /user/joe/wordcount/input/  
/user/joe/wordcount/input/file01  
/user/joe/wordcount/input/file02
```

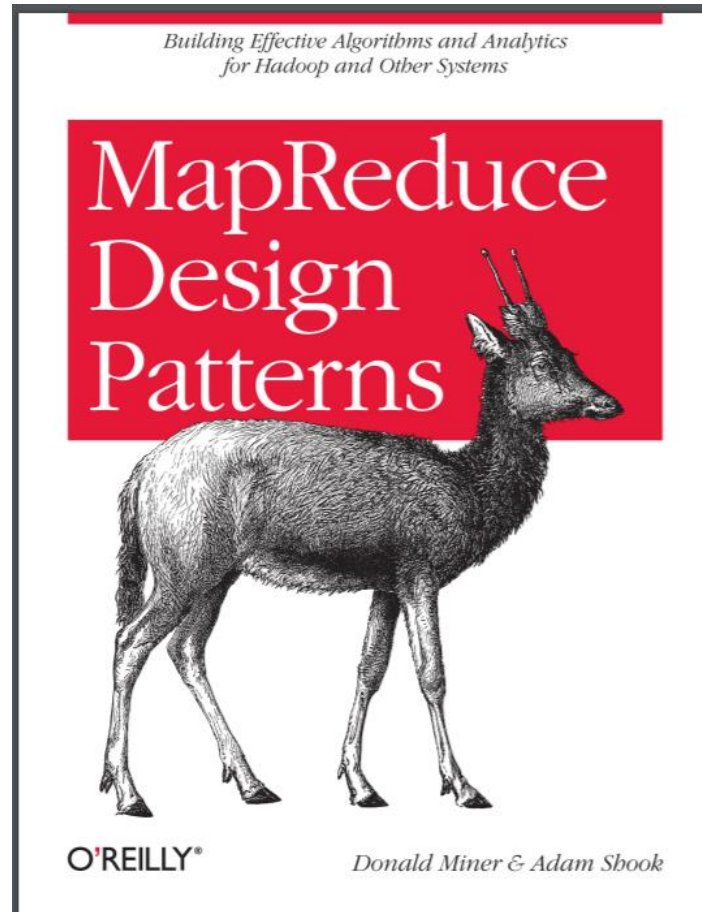
```
$ bin/hadoop fs -cat /user/joe/wordcount/input/file01  
Hello World Bye World
```

```
$ bin/hadoop fs -cat /user/joe/wordcount/input/file02  
Hello Hadoop Goodbye Hadoop
```

# Output

```
$ bin/hadoop fs -cat /user/joe/wordcount/output/part-r-00000  
Bye 1  
Goodbye 1  
Hadoop 2  
Hello 2  
World 2
```

# Readings

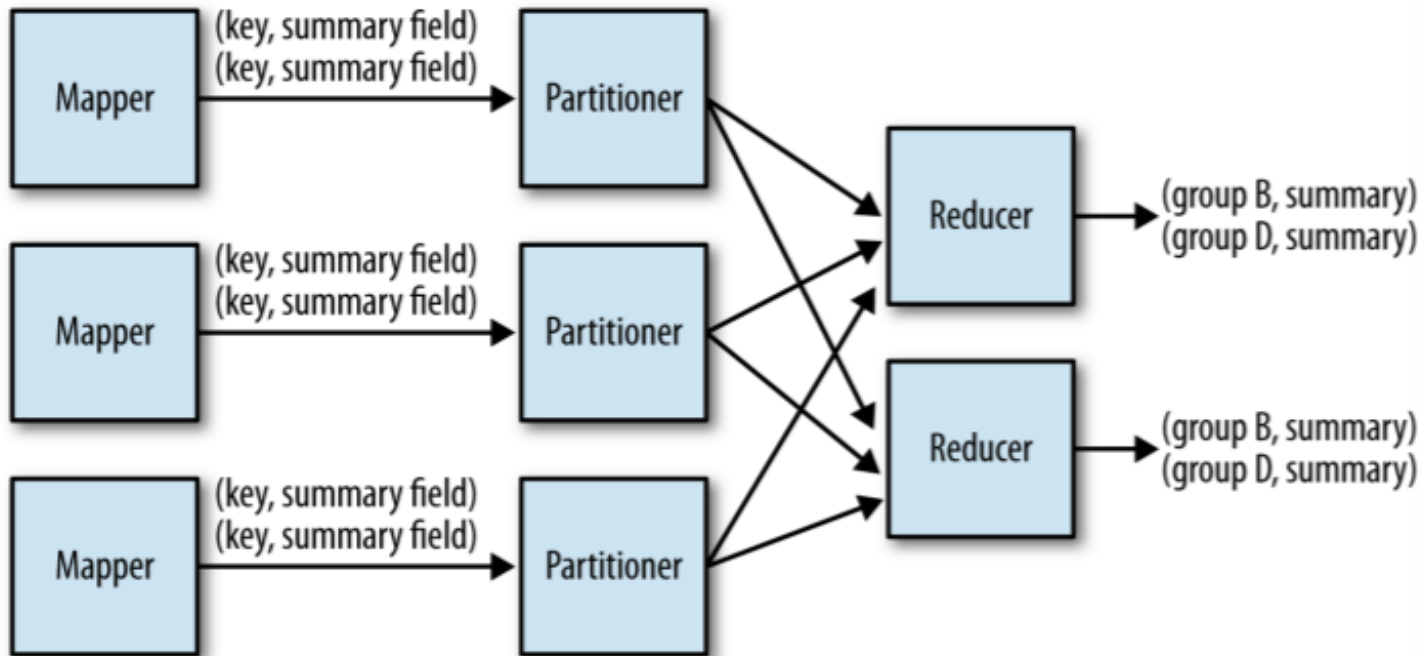




# How Will You Implement These With Map Reduce?

- Min/Max
- Average
- Count
- Median
- Filtering
- Top 10
- Convert key-values to hierarchy
- Partioning
- Sorting

# Summarization Pattern



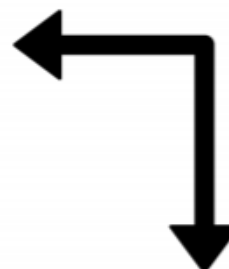
A **partitioner** controls the logical grouping keys of the intermediate map output.

# Min/Max/Count

Map Output / Combiner Input

		Input Key	Input Value		
		User	Minimum	Maximum	Count
Group 1		12345	10	10	1
		12345	8	8	1
		12345	21	21	1
Group 2		54321	1	1	1
		54321	47	47	1
		99999	7	7	1
		99999	12	12	1

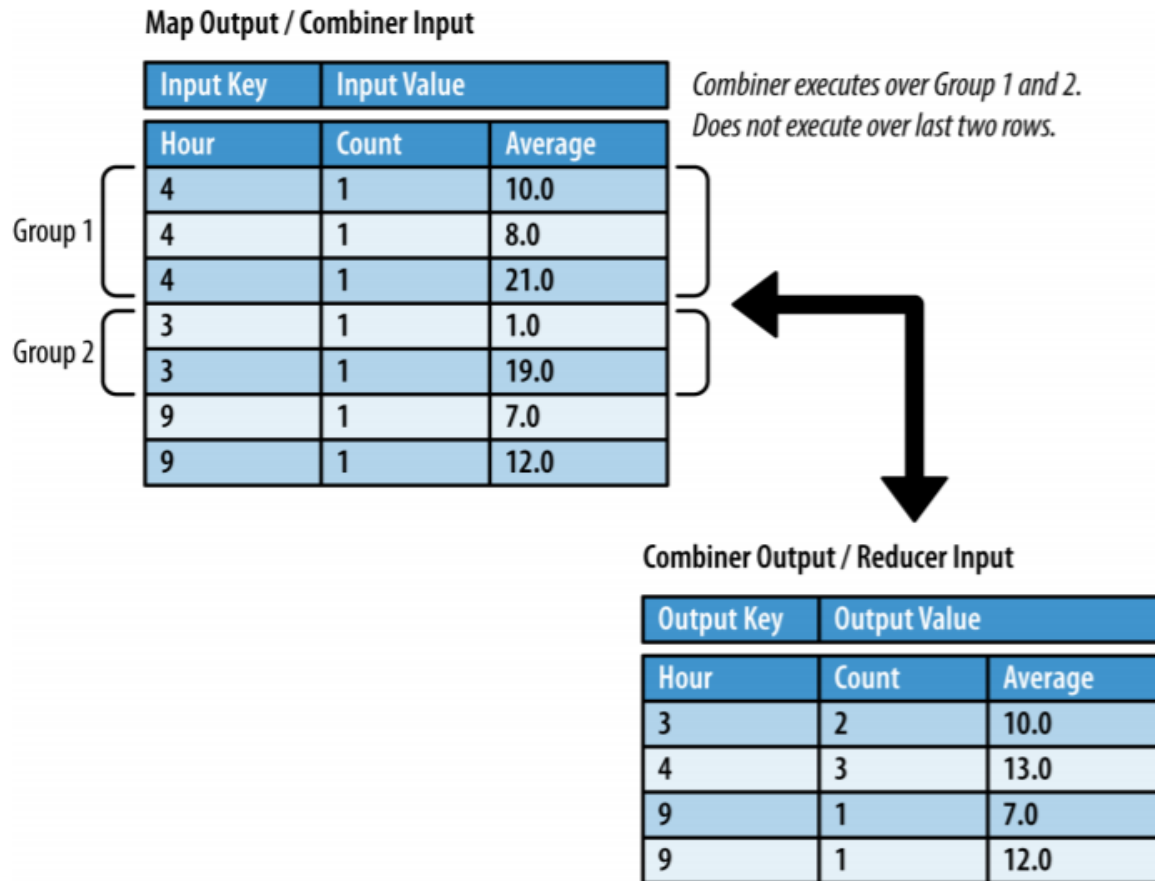
Combiner executes over Group 1 and 2.  
Does not execute over last two rows.



Combiner Output / Reducer Input

		Output Key	Output Value		
			Minimum	Maximum	Count
		12345	8	21	3
		54321	1	47	2
		99999	7	7	1
		99999	12	12	1

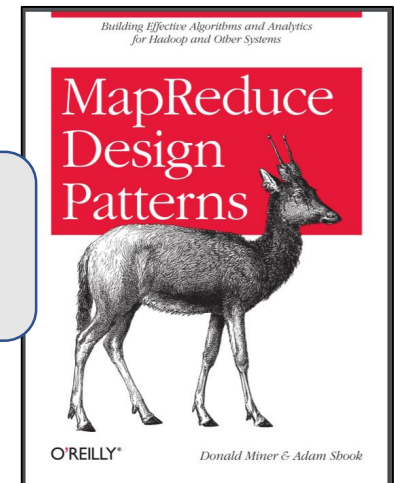
# Average



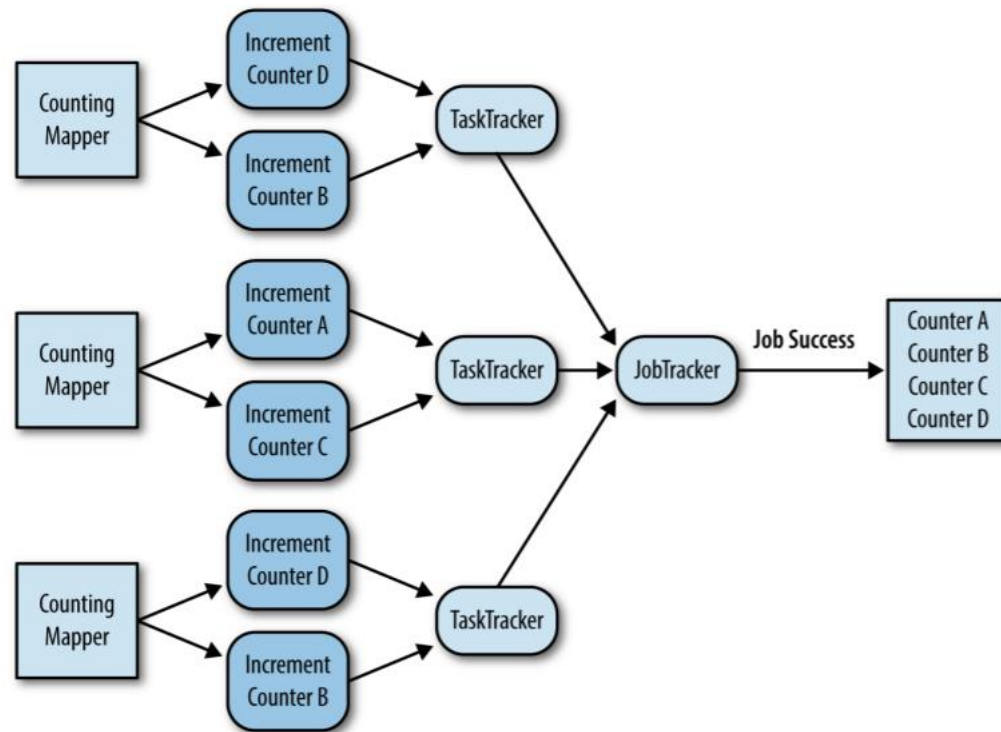
# Flex Your Brain!

- How will you compute the median?

Refer to Chapter 2 of MR Design Patterns Book.

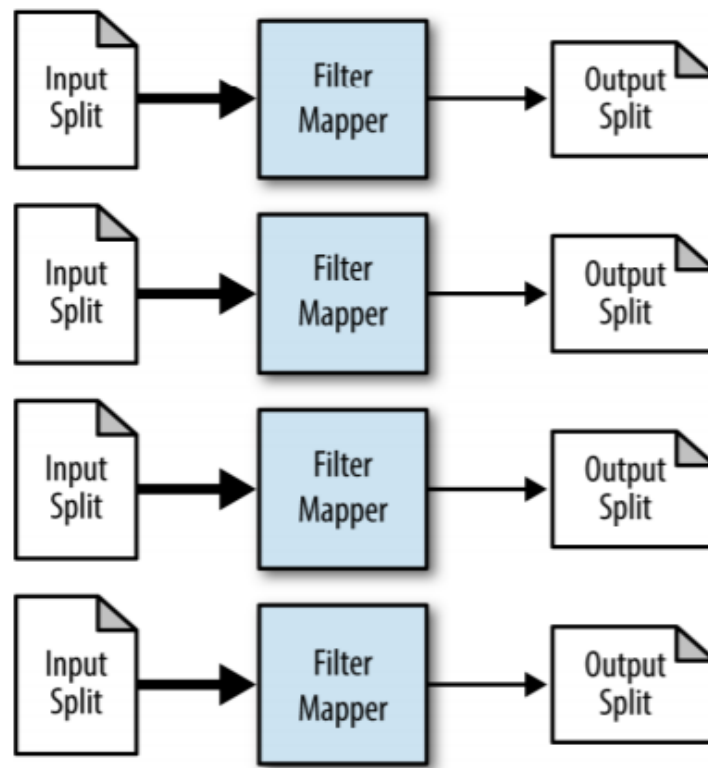


# Counting Mappers



Global counters belong to job-tracker. Use responsibly.

# Filtering



No  
Reducer  
Required.

# Filter Example

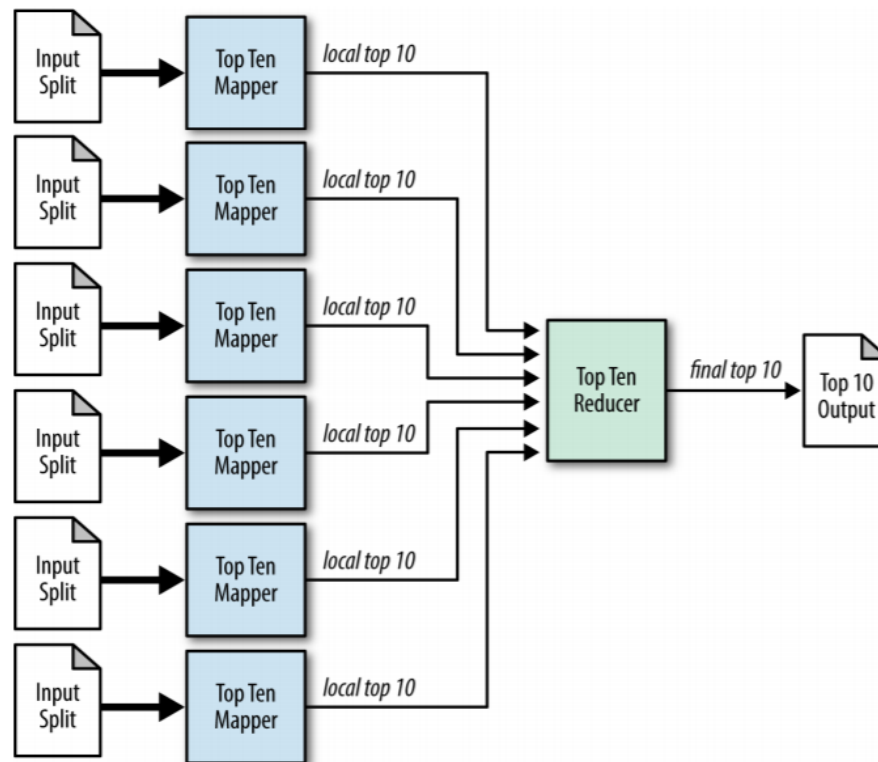
```
public void map(Object key, Text value, Context context)
    throws IOException, InterruptedException {

    if (rands.nextDouble() < percentage) {
        context.write(NullWritable.get(), value);
    }
}
```



# Top 10 Pattern

- How will you determine the top 10 numbers in petabytes of numbers?



# Structure to Hierarchy

How to store this  
in RDBMS?

Posts

Post

Comment

Comment

Post

Comment

Comment

Comment