Introduction to Big Data

ECAP456

Dr. Rajni Bhalla

Associate Professor

Learning Outcomes

After this lecture, you will be able to

- learn MapReduce types
- learn input formats and output formats in hadoop and mapreduce

Map Reduce Types

The map and reduce functions in Hadoop MapReduce have the following general form:

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

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

Map Reduce Types

```
public void map(LongWritable key,
Text value, Context context)
 { .....
 .....
context.write(new Text(year), new
IntWritable(airTemperature));
}
public void reduce(Text key, Iterable
values, Context context)
 {
..................
context.write(key, new
IntWritable(maxValue));
}
public void combiner(Text key,
Iterable values, Context context)
 {
 ..................
context.write(key, new
IntWritable(maxValue));
}
```

Map Reduce Types

If a combine function is used, then it is the same form as the reduce function (and is an implementation of Reducer), except its output types are the intermediate key and value types (K2 and V2), so they can feed the reduce function:

map: (K1, V1) → list(K2, V2)

combine: (K2, list(V2)) → list(K2, V2)

reduce: (K2, list(V2)) → list(K3, V3)

Often the combine and reduce functions are the same, in which case, K3 is the same as K2, and V3 is the same as V2.

Map Reduce Types

How input types are set?

Map Reduce Types

```
job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);
```

Map Reduce Types

So if K2 and K3 are the same, you don’t need to call setMapOutputKeyClass(), since it falls back to the type set by calling setOutputKeyClass().

Similarly, if V2 and V3 are the same, you only need to use setOutputValueClass().

Configuration of MapReduce in the new API

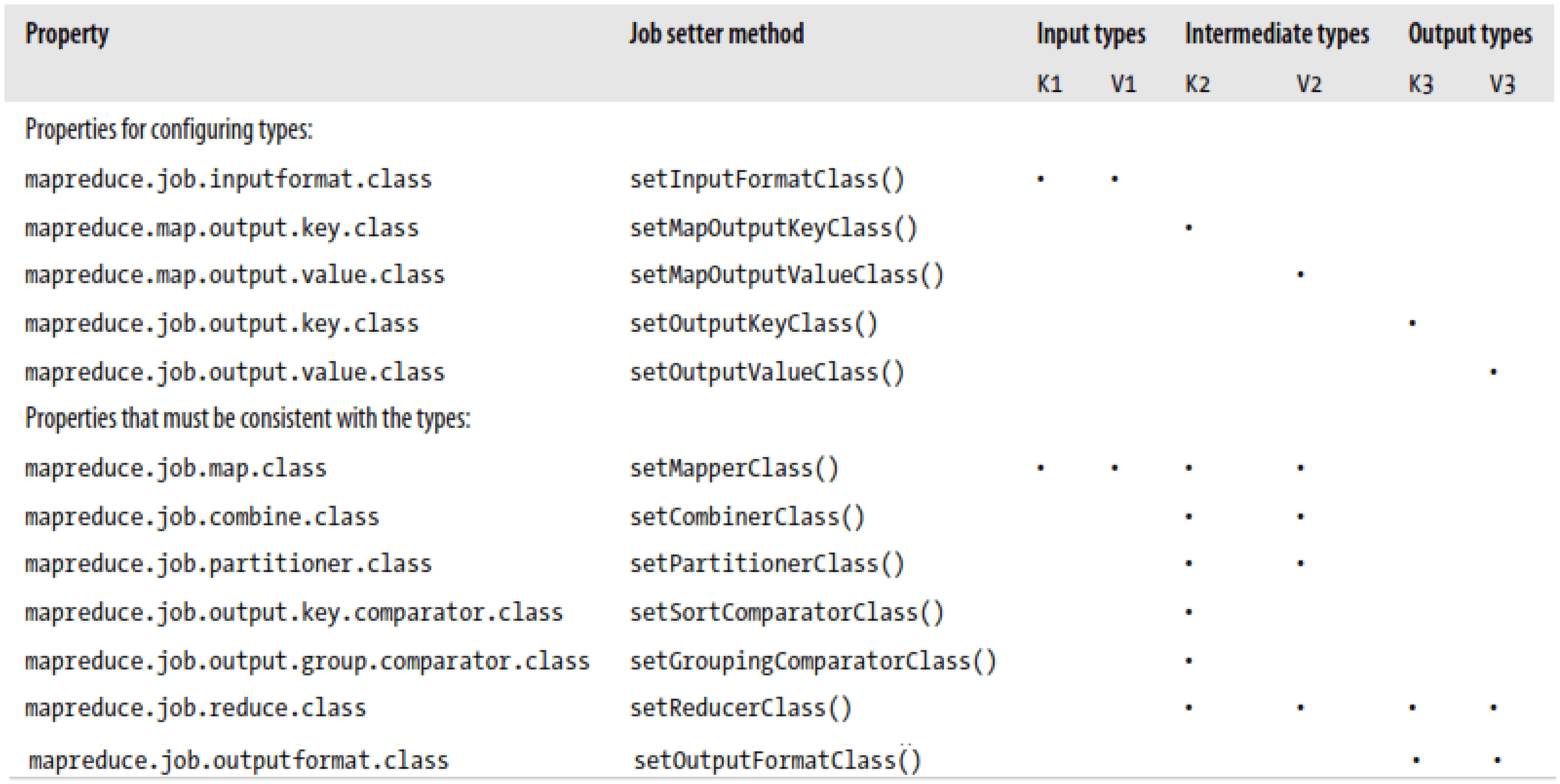


Table 7-1. *Configuration of MapReduce types in the new API*

Property	Job setter method	Input types		Intermediate types		Output types	
		K1	V1	K2	V2	K3	V3
Properties for configuring types:							
`mapreduce.job.inputformat.class`	`setInputFormatClass()`	•	•				
`mapreduce.map.output.key.class`	`setMapOutputKeyClass()`			•			
`mapreduce.map.output.value.class`	`setMapOutputValueClass()`				•		
`mapreduce.job.output.key.class`	`setOutputKeyClass()`					•	
`mapreduce.job.output.value.class`	`setOutputValueClass()`						•
Properties that must be consistent with the types:							
`mapreduce.job.map.class`	`setMapperClass()`	•	•	•	•		
`mapreduce.job.combine.class`	`setCombinerClass()`			•	•		
`mapreduce.job.partitioner.class`	`setPartitionerClass()`			•	•		
`mapreduce.job.output.key.comparator.class`	`setSortComparatorClass()`			•			
`mapreduce.job.output.group.comparator.class`	`setGroupingComparatorClass()`			•			
`mapreduce.job.reduce.class`	`setReducerClass()`			•	•	•	•
`mapreduce.job.outputformat.class`	`setOutputFormatClass()`					•	•

Configuration of MapReduce types in the old API

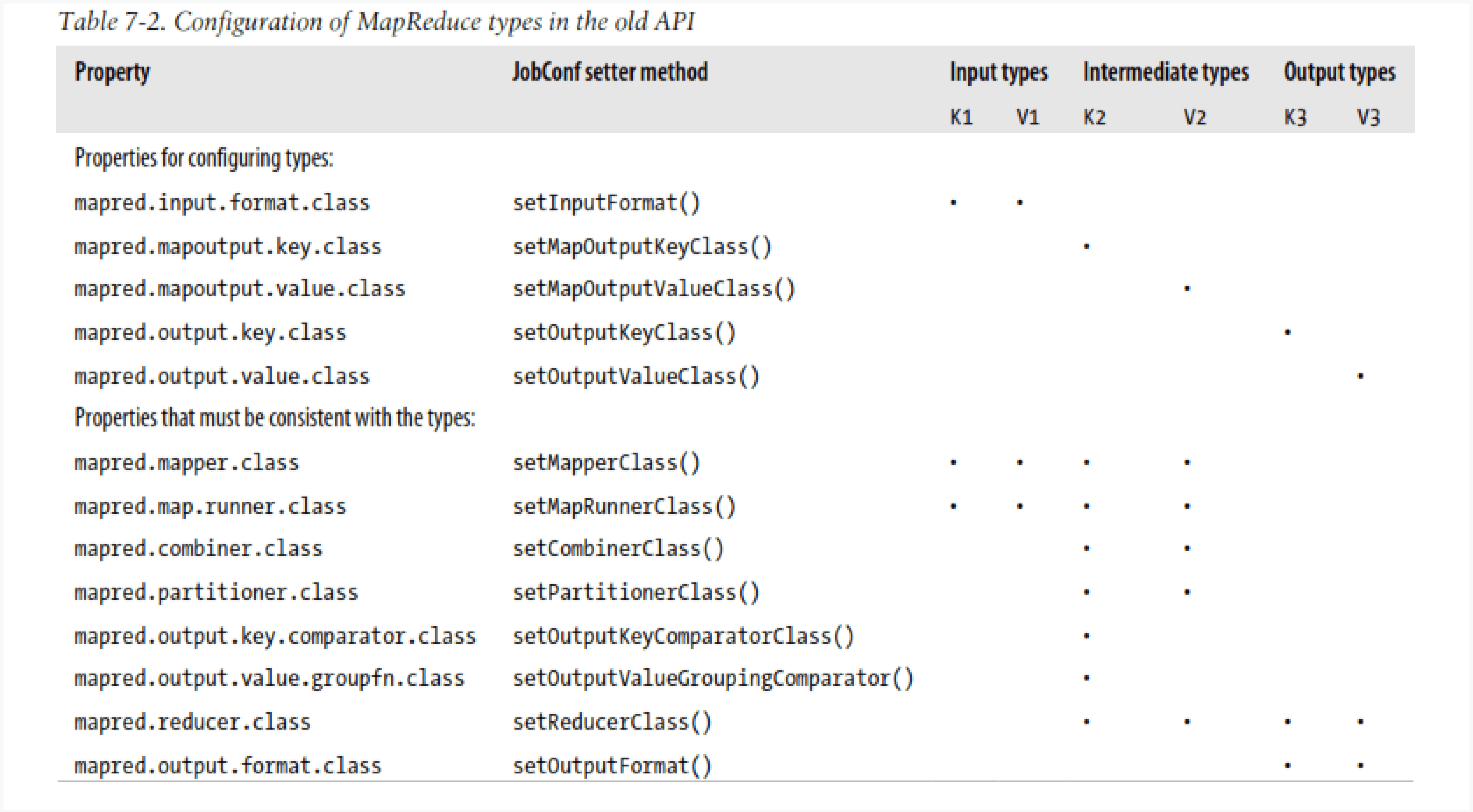


Table 7-2. Configuration of MapReduce types in the old API

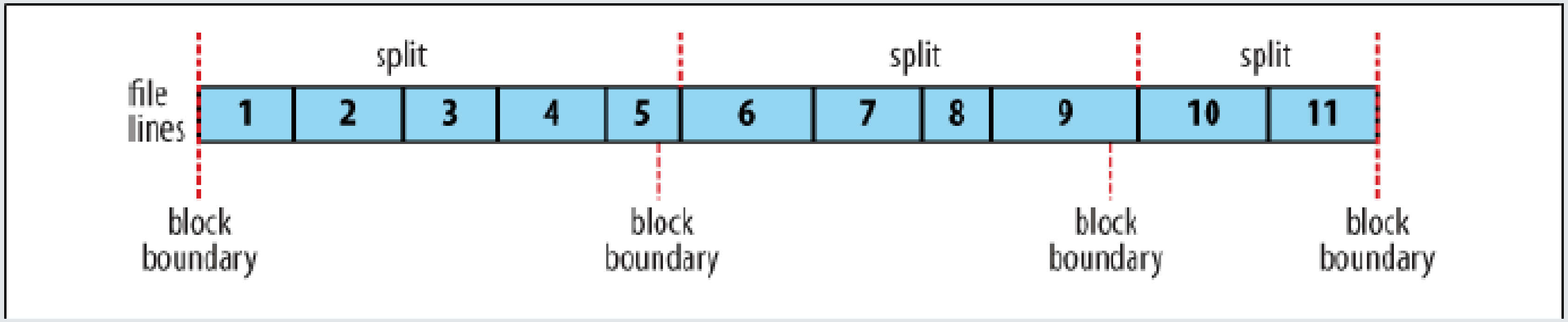
Property	JobConf setter method	Input types		Intermediate types		Output types	
		K1	V1	K2	V2	K3	V3
Properties for configuring types:							
`mapred.input.format.class`	`setInputFormat()`	•	•				
`mapred.mapoutput.key.class`	`setMapOutputKeyClass()`			•			
`mapred.mapoutput.value.class`	`setMapOutputValueClass()`				•		
`mapred.output.key.class`	`setOutputKeyClass()`					•	
`mapred.output.value.class`	`setOutputValueClass()`						•
Properties that must be consistent with the types:							
`mapred.mapper.class`	`setMapperClass()`	•	•	•	•		
`mapred.map.runner.class`	`setMapRunnerClass()`	•	•	•	•		
`mapred.combiner.class`	`setCombinerClass()`			•	•		
`mapred.partitioner.class`	`setPartitionerClass()`			•	•		
`mapred.output.key.comparator.class`	`setOutputKeyComparatorClass()`			•			
`mapred.output.value.groupfn.class`	`setOutputValueGroupingComparator()`			•			
`mapred.reducer.class`	`setReducerClass()`			•	•	•	•
`mapred.output.format.class`	`setOutputFormat()`					•	•

Input Formats

Hadoop can process many different types of data formats, from flat text files to databases

Input Formats

The Relationship Between Input Splits and HDFS Block



Text Input

So a file containing the following text:
On the top of the Crumpetty Tree The Quangle Wangle sat, But his face you could not see, On account of his Beaver Hat.

Text Input

So a file containing the following text:	The records are interpreted as the following key-value pairs:
On the top of the Crumpetty Tree The Quangle Wangle sat, But his face you could not see, On account of his Beaver Hat.	(0, On the top of the Crumpetty Tree) (33, The Quangle Wangle sat,) (57, But his face you could not see,) (89, On account of his Beaver Hat.)

Text Input

KeyValueTextInputFormat

Text Input

mapreduce.input.keyvaluelinerecodreader.key.valu
e.separator property

Or

key.value.separator.in.input.line

in the old API

Text Input

Consider the following input file, where → represents a (horizontal) tab character:

```
line1→On the top of the Crumpetty Tree
line2→The Quangle Wangle sat,
line3→But his face you could not see,
line4→On account of his Beaver Hat.
```

Text Input

NLineInputFormat

Text Input

On the top of the Crumpetty Tree

The Quangle Wangle sat,

But his face you could not see,

On account of his Beaver Hat.

Text Input

If, for example, N is two, then each split contains two lines.

One mapper will receive the first two key-value pairs:

(0, On the top of the Crumpetty Tree)

(33, The Quangle Wangle sat,)

Text Input

And another mapper will receive the second two key-value pairs:

(57, But his face you could not see,)

(89, On account of his Beaver Hat.)

Text Input

XML

set your input format to StreamInputFormat and

set the stream.recordreader.class property to

org.apache.hadoop.streaming.StreamXmlRecordReader

to use xml as an input format.

Text Input

Binary Input :

SequenceFileInputFormat

Text Input

Binary Input :

SequenceFileInputFormat

SequenceFileAsTextInputFormat

Text Input

Binary Input :

SequenceFileInputFormat

SequenceFileAsTextInputFormat

SequenceFileAsBinaryInputFormat

Text Input

SequenceFileInputFormat

Hadoop's sequence file format stores sequences of binary key-value pairs. Sequence files are well suited as a format for MapReduce data since they are splittable they support compression as a part of the format.

Text Input

SequenceFileAsTextInputFormat

SequenceFileAsTextInputFormat is a variant of SequenceFileInputFormat that converts the sequence file's keys and values to Text objects.

Text Input

SequenceFileAsBinaryInputFormat

SequenceFileAsBinaryInputFormat is a variant of SequenceFileInputFormat that retrieves the sequence file's keys and values as opaque binary objects.

Text Input

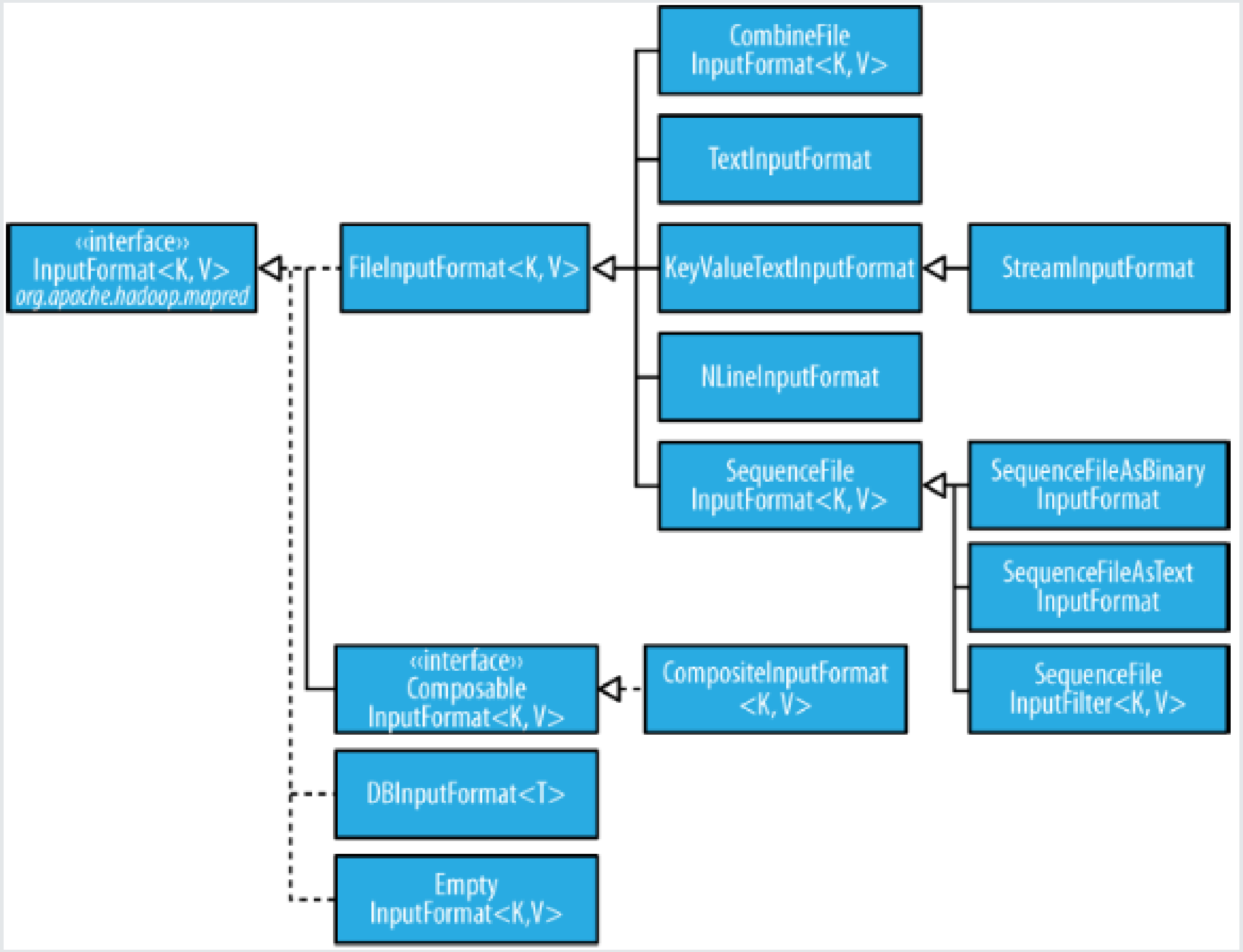
Multiple Inputs

The input to a MapReduce job may consist of multiple input files. This case is handled elegantly by using the MultipleInputs class.

Text Input

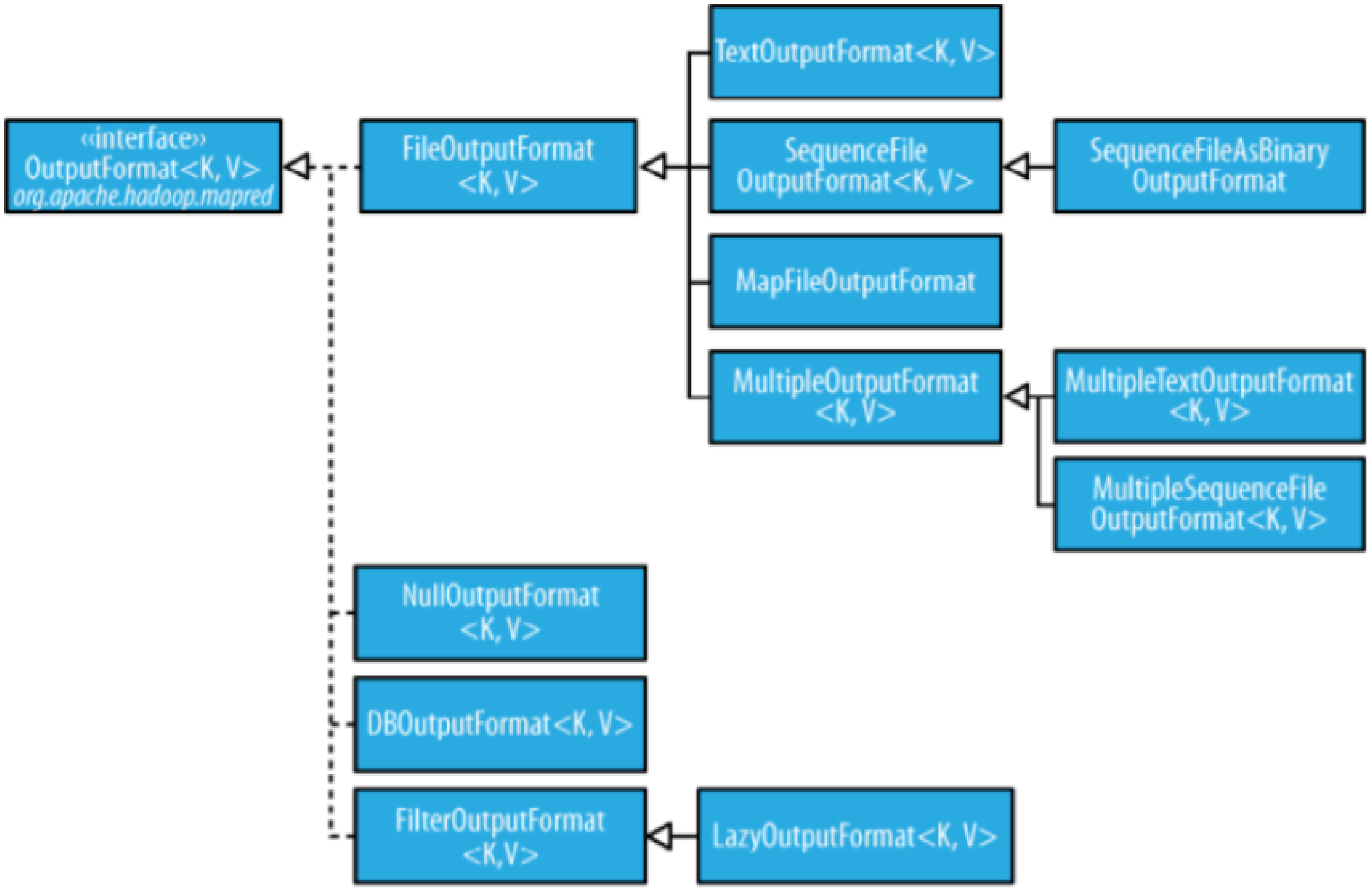
```
MultipleInputs.addInputPath(job, ncdcInputPath,
TextInputFormat.class,
MaxTemperatureMapper.class);
 MultipleInputs.addInputPath(job, metOfficeInputPath,
TextInputFormat.class,
MetOfficeMaxTemperatureMapper.class);
```

Database Input


«interface» InputFormat<K, V> org.apache.hadoop.mapred
FileInputFormat<K, V>
CombineFile InputFormat<K, V>
TextInputFormat
KeyValueTextInputFormat
StreamInputFormat
NLineInputFormat
SequenceFile InputFormat<K, V>
SequenceFileAsBinary InputFormat
SequenceFileAsText InputFormat
SequenceFile InputFilter<K, V>
«interface» Composable InputFormat<K, V>
CompositeInputFormat <K, V>
DBInputFormat<T>
Empty InputFormat<K,V>

Input format class hierarchy

Database Input



Output format class hierarchy

Text Output

- The default output format, TextOutputFormat, writes records as lines of text.
- Its keys and values may be of any type, since TextOutputFormat turns them to strings by calling toString() on them.
- Each key-value pair is separated by a tab character. The counterpart to TextOutput Format for reading in this case is KeyValueTextInputFormat.

Binary Output

- SequenceFileOutputFormat

Binary Output

- SequenceFileOutputFormat
- SequenceFileAsBinaryOutputFormat

Binary Output

- SequenceFileOutputFormat
- SequenceFileAsBinaryOutputFormat
- MapFileOutputFormat

Hadoop InputFormat

Hadoop InputFormat

- Describes the input-specification for execution.
- Describes how to split up and read input files.
- InputFormat is the first step.
- Responsible for creating the input splits and dividing them into records.
- Input files reside in HDFS.

Hadoop InputFormat

Hence, In MapReduce, InputFormat class is one of the fundamental classes which provides below functionality:

- It selects the files or other objects for input.
- It also defines the Data splits.
- It defines the RecordReader.

Hadoop InputFormat

How we get the data from Mapper?

```
public abstract class InputFormat<K, V>
{
public abstract List<InputSplit> getSplits(JobContext context)
throws IOException, InterruptedException;
public abstract RecordReader<K, V>
createRecordReader(InputSplit split,
TaskAttemptContext context) throws IOException,
InterruptedException;
}
```

Hadoop InputFormat

Types of InputFormat in MapReduce

FileInputFormat

TextInputFormat

Key

Value

KeyValueTextInputFormat

SequenceFileInputFormat

SequenceFileAsTextInputFormat

SequenceFileAsBinaryInputFormat

NlineInputFormat

DBInputFormat

Hadoop InputFormat

Types of InputFormat in MapReduce

1. FileInputFormat

Hadoop InputFormat

Types of InputFormat in MapReduce

1. FileInputFormat
2. TextInputFormat

Hadoop InputFormat

Types of InputFormat in MapReduce

1. FileInputFormat
2. TextInputFormat
3. KeyValueTextInputFormat

Hadoop InputFormat

Types of InputFormat in MapReduce

1. FileInputFormat
2. TextInputFormat
3. KeyValueTextInputFormat
4. SequenceFileInputFormat

Hadoop InputFormat

Types of InputFormat in MapReduce

1. FileInputFormat
2. TextInputFormat
3. KeyValueTextInputFormat
4. SequenceFileInputFormat
5. NlineInputFormat

Hadoop InputFormat

Types of InputFormat in MapReduce

1. FileInputFormat
2. TextInputFormat
3. KeyValueTextInputFormat
4. SequenceFileInputFormat
5. NlineInputFormat
6. DBInputFormat

Hadoop InputFormat

Types of InputFormat in MapReduce

1. FileInputFormat
2. TextInputFormat
3. KeyValueTextInputFormat
4. SequenceFileInputFormat
5. NlineInputFormat
6. DBInputFormat
7. Sequence FileAs BinaryInputFormat

That’s all for now…