org.apache.hadoop.mapreduce

Class Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT>

java.lang.Object
org.apache.hadoop.mapreduce.Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT>

Direct Known Subclasses:

ChainReducer, FieldSelectionReducer, IntSumReducer, LongSumReducer, ValueAggregatorCombiner, ValueAggregatorReducer, WrappedReducer

---

```
@Checkpointable
 @InterfaceAudience.Public
 @InterfaceStability.Stable
public class Reducer<KEYIN,VALUEIN,KEYOUT,VALUEOUT>
extends Object
```

Reduces a set of intermediate values which share a key to a smaller set of values.

`Reducer` implementations can access the `Configuration` for the job via the `JobContext.getConfiguration()` method.

`Reducer` has 3 primary phases:

1. **Shuffle**

   The `Reducer` copies the sorted output from each `Mapper` using HTTP across the network.

2. **Sort**

   The framework merge sorts `Reducer` inputs by `keys` (since different `Mappers` may have output the same key).

   The shuffle and sort phases occur simultaneously i.e. while outputs are being fetched they are merged.

   SecondarySort

   To achieve a secondary sort on the values returned by the value iterator, the application should extend the key with the secondary key and define a grouping comparator. The keys will be sorted using the entire key, but will be grouped using the grouping comparator to decide which keys and values are sent in the same call to reduce.The grouping comparator is specified via `Job.setGroupingComparatorClass(Class)`. The sort order is controlled by `Job.setSortComparatorClass(Class)`.

   For example, say that you want to find duplicate web pages and tag them all with the url of the "best" known example. You would set up the job like:
   - Map Input Key: url
   - Map Input Value: document
   - Map Output Key: document checksum, url pagerank
   - Map Output Value: url
   - Partitioner: by checksum

- OutputKeyComparator: by checksum and then decreasing pagerank
- OutputValueGroupingComparator: by checksum

3. **Reduce**

   In this phase the `reduce(Object, Iterable, org.apache.hadoop.mapreduce.Reducer.Context)` method is called for each `<key, (collection of values)>` in the sorted inputs.

   The output of the reduce task is typically written to a `RecordWriter` via `TaskInputOutputContext.write(Object, Object)`.

The output of the `Reducer` is **not re-sorted**.

Example:

```
public class IntSumReducer<Key> extends Reducer<Key,IntWritable,
                                                Key,IntWritable> {
  private IntWritable result = new IntWritable();

  public void reduce(Key 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);
  }
}
```

See Also:

`Mapper`, `Partitioner`

Constructor Summary

Constructors

Constructor and Description
`Reducer()`

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method and Description
`protected void`	`cleanup(org.apache.hadoop.mapreduce.Reducer.Context context)` Called once at the end of the task.
`protected void`	`reduce(KEYIN key, Iterable<VALUEIN> values, org.apache.hadoop.mapreduce.Reducer.Context context)` This method is called once for each key.

void	run(org.apache.hadoop.mapreduce.Reducer.Context context) Advanced application writers can use the run(org.apache.hadoop.mapreduce.Reducer.Context) method to control how the reduce task works.
protected void	setup(org.apache.hadoop.mapreduce.Reducer.Context context) Called once at the start of the task.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Reducer

```
public Reducer()
```

Method Detail

setup

```
protected void setup(org.apache.hadoop.mapreduce.Reducer.Context context)
              throws IOException,
                     InterruptedException
```

Called once at the start of the task.

Throws:

IOException

InterruptedException

reduce

```
protected void reduce(KEYIN key,
                      Iterable<VALUEIN> values,
                      org.apache.hadoop.mapreduce.Reducer.Context context)
               throws IOException,
                      InterruptedException
```

This method is called once for each key. Most applications will define their reduce class by overriding this method. The default implementation is an identity function.

Throws:

IOException

InterruptedException

cleanup

```
protected void cleanup(org.apache.hadoop.mapreduce.Reducer.Context context)
               throws IOException,
                      InterruptedException
```

Called once at the end of the task.

Throws:

`IOException`

`InterruptedException`

run

```
public void run(org.apache.hadoop.mapreduce.Reducer.Context context)
         throws IOException,
                InterruptedException
```

Advanced application writers can use the `run(org.apache.hadoop.mapreduce.Reducer.Context)` method to control how the reduce task works.

Throws:

`IOException`

`InterruptedException`