



University of
South Australia

Design Patterns, Combiners and Dictionaries

1

What is a design pattern?

Resources for MapReduce
Design Patterns are plentiful

Donald Miner and Adam Shook's
book 'MapReduce Design Patterns'
lists six types of patterns specifically
for MapReduce

1. Summarization patterns



2

Pattern format in Miner & Shook

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Summarization patterns

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Intent

Group records together by a key field and calculate a numerical aggregate per group

Consider f to be a generic numerical summarization function, given a list of values (v_1, v_2, \dots, v_n) , we'd be trying to find a value α i.e. $\alpha = f(v_1, v_2, \dots, v_n)$

Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Motivation

Data sets are large these days! Too hard to discern meaning without a top-level view of the data.

Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Applicability

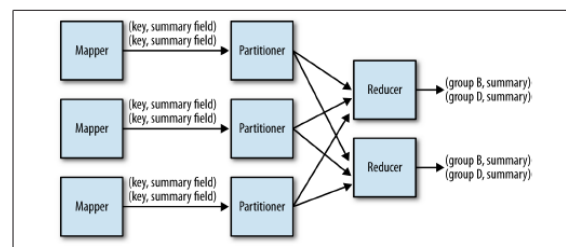
Should be used when both of the following are true:

- You are dealing with numerical data
- The data can be grouped by specific fields

Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Structure



MapReduce Design Patterns – Donald Miner & Adam Shook (2013)

Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Consequences

The output of the job will be a set of files containing a single record per reducer input group. Each record will consist of the key and all aggregate values.

Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Resemblances

Analogous patterns in both SQL and Pig

Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Known uses

Word count, Min/Max/Count,
Average/Median/Standard Deviation

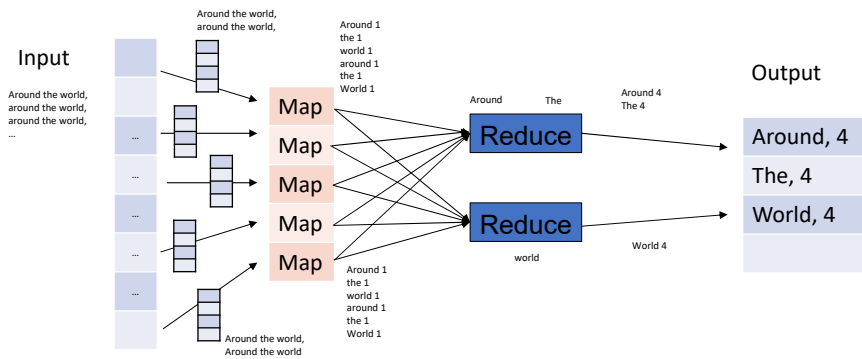
Numerical summarizations

- Intent
- Motivation
- Applicability
- Structure
- Consequences
- Resemblances
- Known uses
- Performance Analysis

Performance analysis

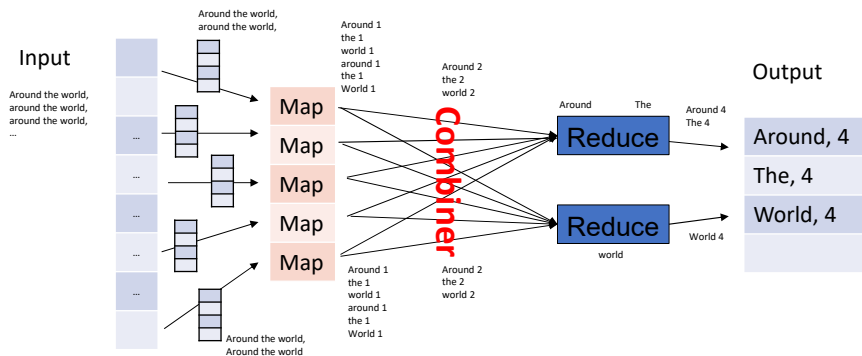
MapReduce was built for these types of jobs, it performs well. However there are still factors to be cautious of. The authors note that mastery of the combiner can significantly effect performance in some scenarios.

On the topic of combiners...



13

On the topic of combiners...



14

No combiner

```

cloudera@quickstart:~$
File Edit View Search Terminal Help
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=36638
Total time spent by all maps in occupied slots (ms)=23872
Total time spent by all map tasks (ms)=36638
Total time spent by all reduce tasks (ms)=23872
Total vcore-milliseconds taken by all map tasks=36638
Total vcore-milliseconds taken by all reduce tasks=23872
Total megabyte-milliseconds taken by all map tasks=37091728
Total megabyte-milliseconds taken by all reduce tasks=23377728

Map-Reduce Framework
Map input records=479829
Map output records=4473879
Map output bytes=17895488
Map output materialized bytes=2884232
Input split bytes=248
Combine input records=0
Combine output records=0
Reduce shuffle bytes=2884232
Reduce input records=4473879
Reduce output records=0
Spilled Records=8947748
Shuffled Maps = 2
Failed Shuffles=0
Partial file=0
Map task elapsed time=1398
CPU time spent (ms)=2338
Physical memory (bytes) snapshot=92682384
Virtual memory (bytes) snapshot=4798234288
Total committed heap usage (bytes)=754513248

Shuffle Errors
BAD ID=0
CONNECTION=0
IO ERROR=0
WRONG LENGTH=0
WRONG MAP=0
WRONG REDUCE=0
File Input Format Counters
Bytes Read=4957795
Bytes Written=543
28/08/24 06:12:23 INFO Streaming-StreamJob: Output directory: /user/cloudera/test/output_test1
cloudera@quickstart:~$

```

With combiner

```

[cloudera@quickstart ~]$ hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-mr.jar -mapper /home/cloudera/test/src/mapper.py -combiner /home/cloudera/test/src/reducer.py -reducer /home/cloudera/test/src/reducer.py -input /user/cloudera/test/input/* -output /user/cloudera/test/output_test2

```

```

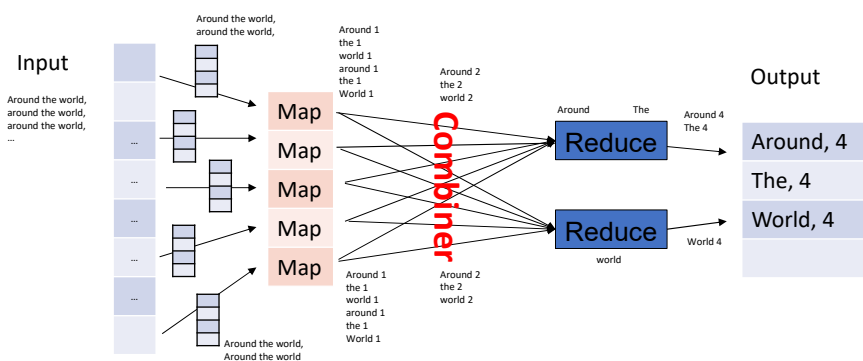
cloudera@quickstart:~$
File Edit View Search Terminal Help
Data-local map tasks=2
Total time spent by all maps in occupied slots (ms)=37199
Total time spent by all maps in occupied slots (ms)=5259
Total time spent by all map tasks (ms)=37199
Total time spent by all reduce tasks (ms)=5259
Total vcore-milliseconds taken by all map tasks=37199
Total vcore-milliseconds taken by all reduce tasks=5259
Total megabyte-milliseconds taken by all map tasks=38991776
Total megabyte-milliseconds taken by all reduce tasks=5385216

Map-Reduce Framework
Map input records=479829
Map output records=4473879
Map output bytes=17895488
Map output materialized bytes=1232
Input split bytes=248
Combine input records=4473879
Combine output records=137
Reduce input records=0
Reduce output records=137
Reduce shuffle bytes=1232
Reduce input records=137
Reduce output records=137
Spilled Records=274
Shuffled Maps = 2
Failed Shuffles=0
Partial file=0
Map task elapsed time=1389
CPU time spent (ms)=27186
Physical memory (bytes) snapshot=795734816
Virtual memory (bytes) snapshot=4683771904
Total committed heap usage (bytes)=731381760

Shuffle Errors
BAD ID=0
CONNECTION=0
IO ERROR=0
WRONG LENGTH=0
WRONG MAP=0
WRONG REDUCE=0
File Input Format Counters
Bytes Read=4957795
Bytes Written=785
28/08/24 06:14:36 INFO Streaming-StreamJob: Output directory: /user/cloudera/test/output_test2
[cloudera@quickstart ~]$

```

On the topic of combiners...

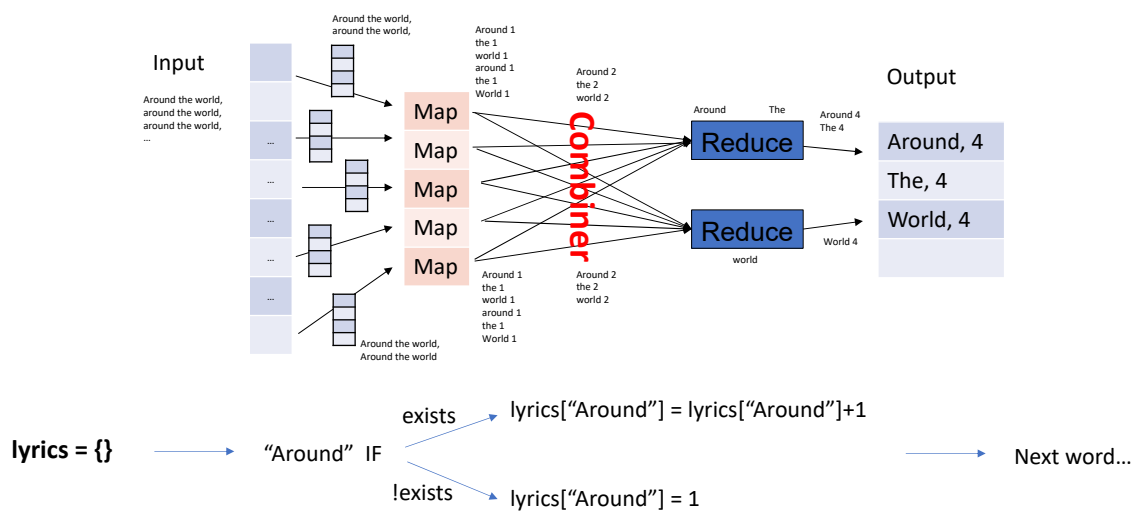


Dictionaries

In Python a dictionary is a collection which is unordered, changeable and indexed. Dictionaries are written with curly brackets, and they are also structured with key/value pairs

Define dictionary	<pre>myDictionary = { "Name": "Mr Robot", "Year": "2015", "Seasons": "4"} </pre>	<pre>myDictionary["Year"] = 2019 </pre>	Change items
Access items	<pre>x = myDictionary["Year"] or x = myDictionary.get("Year") </pre>	<pre>for x in myDictionary: print(x) print(myDictionary[x]) </pre> <pre>for x in myDictionary.values(): print(x) </pre>	Loop through keys and values

Combiners using dictionaries



WARNING

This material has been reproduced and communicated to you by or on behalf of the **University of South Australia** in accordance with section 113P of the *Copyright Act 1968* (**Act**).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice