
hadoop
Map Reduce

Zanim zaczniemy ćwiczenia musimy zainstalować:

- PIP:
  yum install python-pip
- MRJob:
  pip install google-api-python-client==1.6.4
  pip install mrjob==0.5.11
- Nano:
  yum install nano

oraz ściągnąć nasz zbiór danych:

- Utwórz nowy folder w lokalnym systemie plików:
  mkdir twoje_nazwisko
- Wejdź do folderu:
  cd twoje_nazwisko
- Pobierz plik:
  wget https://danilewicz.blob.core.windows.net/public/1987.zip
- Rozpakuj archiwum:
  unzip 1987.zip

	Name	**Description**
1	Year	1987-2008
2	Month	01-Dec
3	DayofMonth	Jan-31
4	DayOfWeek	1 (Monday) - 7 (Sunday)
5	DepTime	actual departure time (local, hhmm)
6	CRSDepTime	scheduled departure time (local, hhmm)
7	ArrTime	actual arrival time (local, hhmm)
8	CRSArrTime	scheduled arrival time (local, hhmm)
9	UniqueCarrier	unique carrier code
10	FlightNum	flight number
11	TailNum	plane tail number
12	ActualElapsedTime	in minutes
13	CRSElapsedTime	in minutes
14	AirTime	in minutes
15	ArrDelay	arrival delay, in minutes
16	DepDelay	departure delay, in minutes
17	Origin	origin IATA airport code
18	Dest	destination IATA airport code
19	Distance	in miles
20	TaxiIn	taxi in time, in minutes

21	TaxiOut	taxi out time in minutes
22	Cancelled	was the flight cancelled?
23	CancellationCode	reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
24	Diverted	1 = yes, 0 = no
25	CarrierDelay	in minutes
26	WeatherDelay	in minutes
27	NASDelay	in minutes
28	SecurityDelay	in minutes
29	LateAircraftDelay	in minutes

Ćwiczenie 1

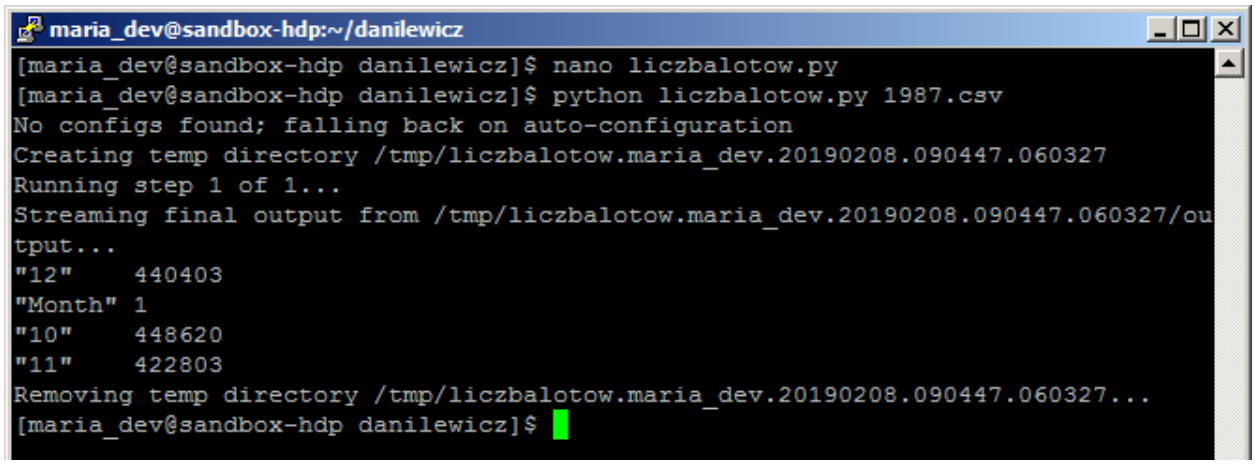
Stworzymy prosty program MapReduce, który zliczy liczbe lotów w poszczególnych miesiącach.

1. Stwórz nowy plik w którym będzie nasz kod:
   nano liczbalotow.py
2. W edytorze tekstu wpisz poniższy kod:

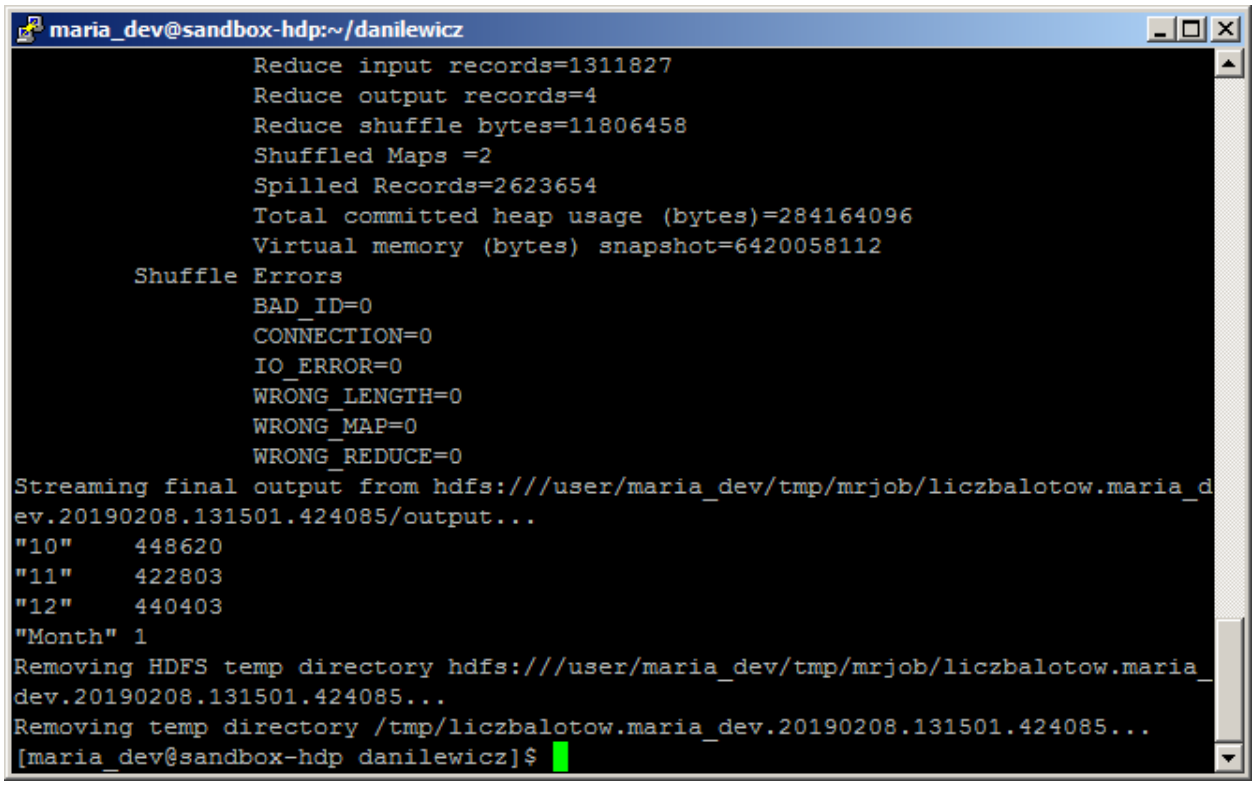
```
from mrjob.job import MRJob
from mrjob.step import MRStep

class LiczbaLotow(MRJob):
	def steps(self):
		return [
			MRStep(mapper=self.mapper_get_months,
			reducer=self.reducer_count_months)
		]

	def mapper_get_months(self, _, line):
		fields = line.split(',')
		yield fields[1], 1

	def reducer_count_months(self, key, values):
		yield key, sum(values)

if __name__ == '__main__':
    LiczbaLotow.run()
```

3. Zamknij edytor zapisując zmiany:
   - *Ctrl + X*
   - *Y*
   - *Enter*

4. Najpierw uruchomimy nasz program lokalnie: python liczbalotow.py 1987.csv



5. Następnie uruchomimy nasz program korzystając z hadoop:
python liczbalotow.py -r hadoop --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar 1987.csv



Ćwiczenie 2

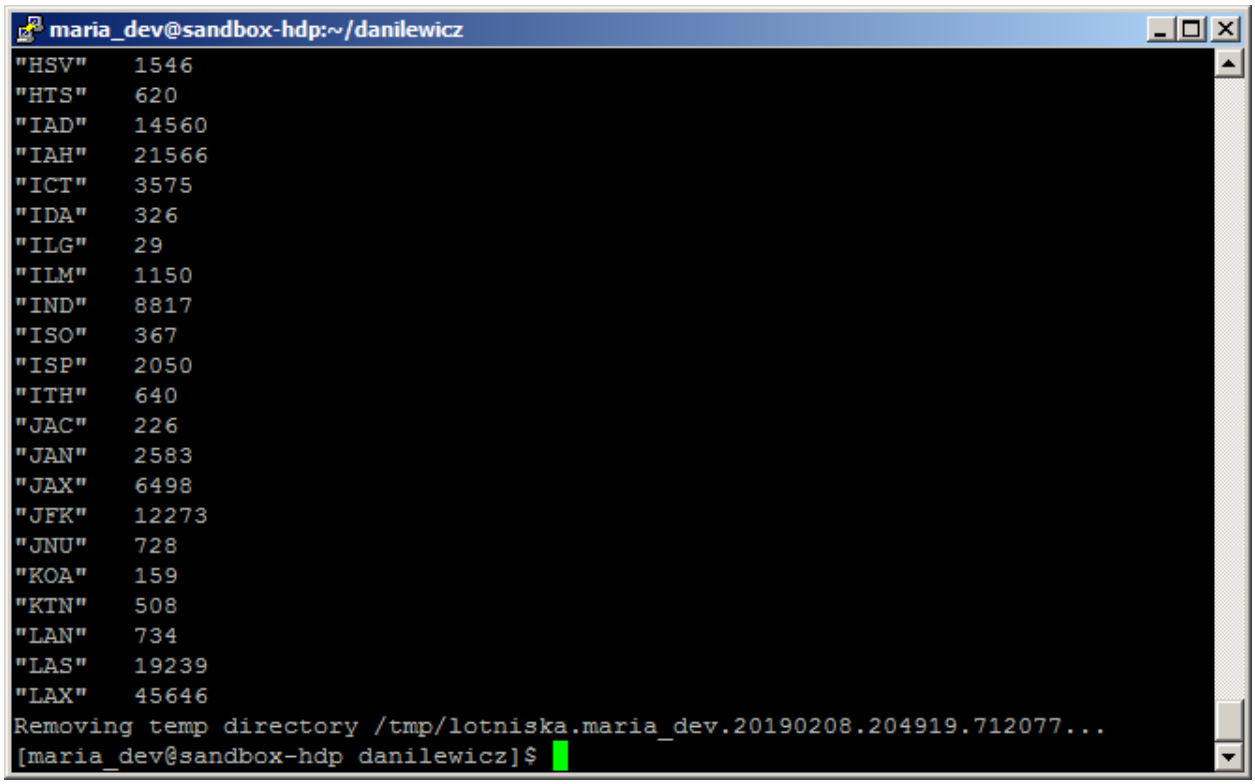
Stworzymy program, który znajdzie nam lotniska z których startowało najwięcej samolotów.

1. Zaczniemy od stworzenia programu podobnego do ćwiczenia 1:
   Stwórz nowy plik w którym będzie nasz kod:
   nano lotniska.py
2. W edytorze tekstu wpisz poniższy kod:

```
from mrjob.job import MRJob
from mrjob.step import MRStep

class Lotniska(MRJob):
	def steps(self):
		return [
			MRStep(mapper=self.mapper_get_airports,
			reducer=self.reducer_count_airports)
		]

	def mapper_get_airports(self, _, line):
		fields = line.split(',')
		yield fields[16], 1

	def reducer_count_airports(self, key, values):
		yield key, sum(values)

if __name__ == '__main__':
    Lotniska.run()
```

3. Zamknij edytor zapisując zmiany:
   - *Ctrl + X*
   - *Y*
   - *Enter*

4. Uruchom program lokalnie: python lotniska.py 1987.csv



5. Zamień kolejność w *reducer_count_airports*:

```
from mrjob.job import MRJob
from mrjob.step import MRStep

class Lotniska(MRJob):
	def steps(self):
		return [
			MRStep(mapper=self.mapper_get_airports,
			reducer=self.reducer_count_airports)
		]

	def mapper_get_airports(self, _, line):
		fields = line.split(',')
		yield fields[16], 1

	def reducer_count_airports(self, key, values):
		yield sum(values), key

if __name__ == '__main__':
    Lotniska.run()
```

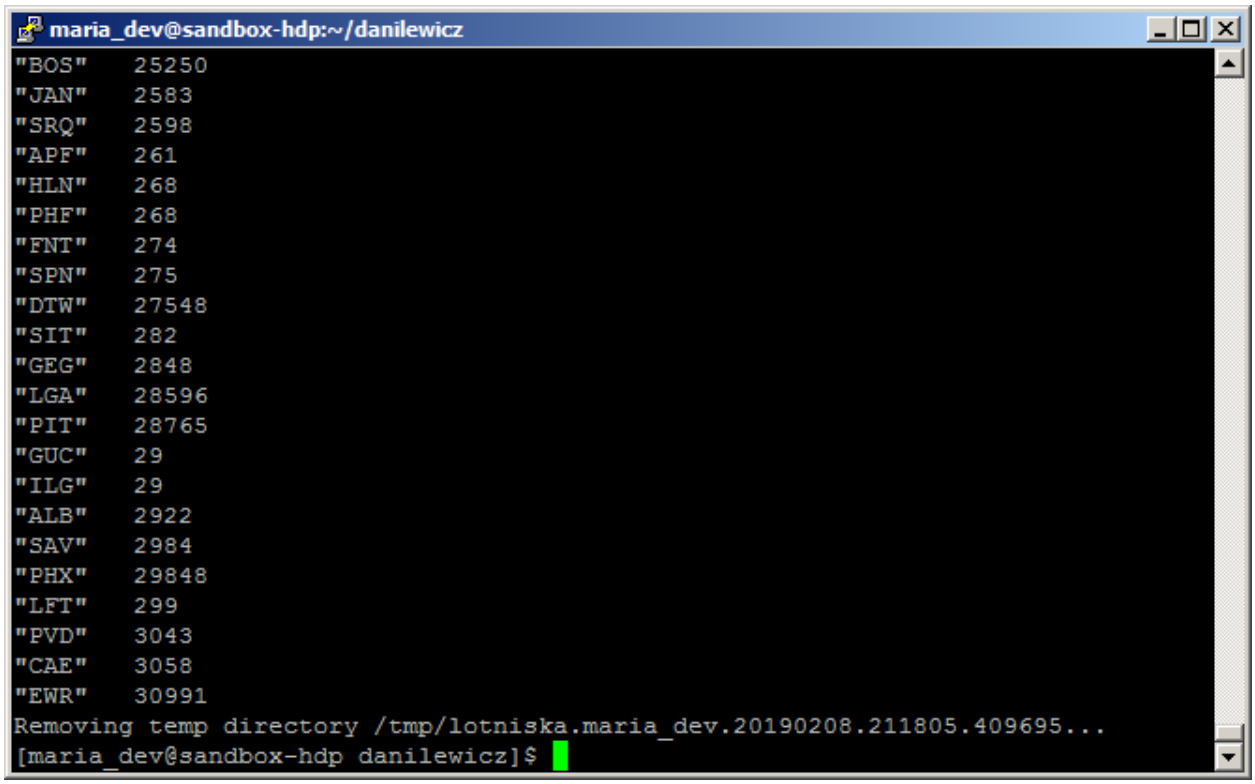
6. Sprawdźmy teraz wynik: python lotniska.py 1987.csv

```
maria_dev@sandbox-hdp:~/danilewicz
1546    "HSV"
620     "HTS"
14560   "IAD"
21566   "IAH"
3575    "ICT"
326     "IDA"
29      "ILG"
1150    "ILM"
8817    "IND"
367     "ISO"
2050    "ISP"
640     "ITH"
226     "JAC"
2583    "JAN"
6498    "JAX"
12273   "JFK"
728     "JNU"
159     "KOA"
508     "KTN"
734     "LAN"
19239   "LAS"
45646   "LAX"
Removing temp directory /tmp/lotniska.maria_dev.20190208.205759.251583...
[maria_dev@sandbox-hdp danilewicz]$
```

7. Aby ponownie posortować wynik, dopisz liniki zaznaczone za żółto:

```
from mrjob.job import MRJob
from mrjob.step import MRStep

class Lotniska(MRJob):
        def steps(self):
                return [
                        MRStep(mapper=self.mapper_get_airports,
                        reducer=self.reducer_count_airports),
                        MRStep(reducer=self.reducer_sort_output)
                ]

        def mapper_get_airports(self, _, line):
                fields = line.split(',')
                yield fields[16], 1

        def reducer_count_airports(self, key, values):
                yield sum(values), key

        def reducer_sort_output(self, count, airports):
                for airport in airports:
                        yield airport, count

if __name__ == '__main__':
    Lotniska.run()
```

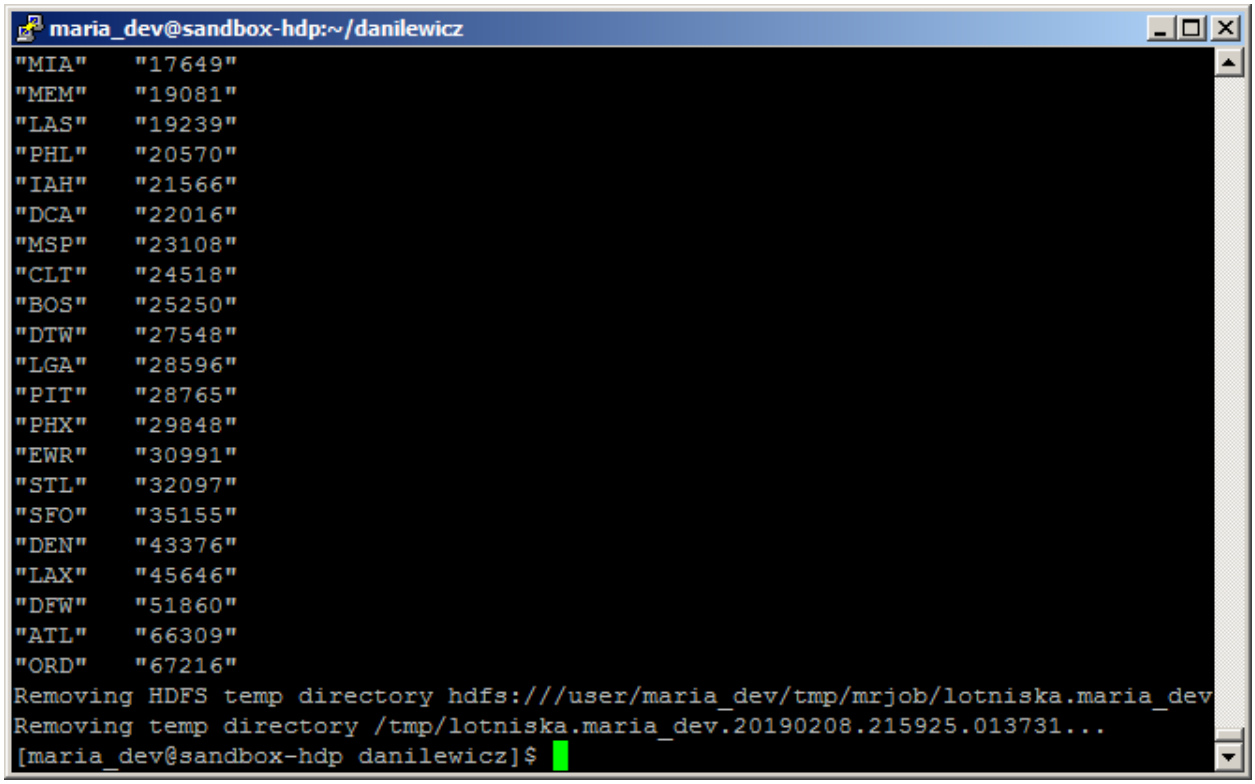
8. Sprawdźmy teraz wynik: python lotniska.py 1987.csv



9. Wynik jest źle posortowany bo liczby są posortowane jak tekst. Naprawimy to zmieniając jedną linikę:

```
from mrjob.job import MRJob
from mrjob.step import MRStep

class Lotniska(MRJob):
	def steps(self):
		return [
			MRStep(mapper=self.mapper_get_airports,
			reducer=self.reducer_count_airports),
			MRStep(reducer=self.reducer_sort_output)
		]

	def mapper_get_airports(self, _, line):
		fields = line.split(',')
		yield fields[16], 1

	def reducer_count_airports(self, key, values):
		yield str(sum(values)).zfill(5), key

	def reducer_sort_output(self, count, airports):
		for airport in airports:
			yield airport, count

if __name__ == '__main__':
    Lotniska.run()
```

10. Teraz możemy uruchomić nasz program na Hadoop:
    python lotniska.py -r hadoop --hadoop-streaming-jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-streaming.jar 1987.csv



Najwięcej samolotów wystartowało z lotniska ORD – Chicago O'Hare International Airport.