Zadanie 7

Ivan Filipchuk

Dask

```
import dask.dataframe as dd
import pandas as pd
import sys

if __name__ == "__main__":
    if len(sys.argv) != 4:
        print("Usage: dask_wordcount_no_rdd <input_file_path> <word_length>
<min_word_length>")
        sys.exit(1)

    input_file_path = sys.argv[1]
    word_length = int(sys.argv[2])
    min_word_length = int(sys.argv[3])
    with open(input_file_path, 'r') as file:
        lines = file.readlines()

    dask_df = dd.from_pandas(pd.DataFrame({'lines': lines}), npartitions=1)
    words = dask_df['lines'].str.split().explode().str.lower()
    words = words.apply(lambda word: ''.join(char for char in word if
char.isalnum()))
    words_filtered = words[words.str.len() == word_length]
    word_counts =
words_filtered.value_counts().compute().reset_index(name='count')
    word_counts = word_counts.sort_values(by='count', ascending=False)
    print(f"Words with length {word_length}:")
    for index, row in word_counts.iterrows():
        print(f"({row[0]}, {row[1]})")

    words_min_length = words[words.str.len() >= min_word_length]
    word_counts_min_length =
words_min_length.value_counts().compute().reset_index(name='count')
    word_counts_min_length = word_counts_min_length.sort_values(by='count',
ascending=False)
    print(f"\nWords with minimum length {min_word_length}:")
    for index, row in word_counts_min_length.iterrows():
        print(f"({row[0]}, {row[1]})")
```

Dockerfile

```
FROM python:latest
RUN pip install dask[complete]
COPY wordcount.py /
ENTRYPOINT ["python", "/wordcount.py"]
CMD []
```

docker build -t dask-wordcount-2 .

docker run -v C:\Users\filip\Desktop\Ztp-lab6\dask-wordcount-2\example.txt:/input.txt -v C:\Users\filip\Desktop\Ztp-lab6\dask-wordcount-2/results:/output dask-wordcount-2 /input.txt 3 5

```
Words with length 3:
(see, 2)
(the, 2)
(and, 1)
(how, 1)
(for, 1)
(out, 1)
(use, 1)

Words with minimum length 5:
(appears, 1)
(count, 1)
(counted, 1)
(program, 1)
(results, 1)
(sample, 1)
(separately, 1)
(should, 1)
(testing, 1)
(times, 1)
```