universitätbonn

Data Science and Big Data

Group 13

Manoj Prabhakar Kannan Ravi (3069835)
Hemanth Kumar Reddy Mayaluru (3063488)
Deepansh Pandey (3057904)

universitätbonn

Tasks / Modules

1. Task 1: Horizontal Scalability (Submitted in paper)
2. Task 2: Implement a Spark Program “UserClicks”
3. Task 3: Implement a class UserSet
4. Task 4: Implement a Spark Program “UserSet”
5. Task 5: Implement a Spark Program “CloseToMark”

universitätbonn

Task 2: Implement a Spark Program "UserClicks"

1. Configuration for a Spark application: Used to set various Spark parameters as key-value pairs
2. Setting **SparkContext**
3. Reading the input file and storing as **RDD**
4. **Line-split** (tab separation) and return List array of the artist names
5. Turn the words into **(word,1) pairs** using **mapToPair**
6. **reduceByKey** to **aggregate** the count of each key
7. Saving the output as txt file
8. Using **lookup** to listen the number of event for "Mark Knopfler"

universitätbonn

Task 3: Implement a class UserSet

1. **Adding User to a userSet**

```
public static void add(UserSet u, String username) {
    u.backingSet.add(username);
}
```

2. **Adding userSet to userSet:**

```
public UserSet addUserSet(UserSet other){
    UserSet tmp = new UserSet();
    tmp.backingSet.addAll(other.backingSet);
    return tmp;
}
```

3. **Computing Jaccard Distance**

```
public double distanceTo(UserSet other) {
HashSet<String> union = new HashSet<String>(backingSet);
union.addAll(other.backingSet);
HashSet<String> intersection = new HashSet<String>(backingSet);
intersection.retainAll(other.backingSet);
double dist = 1.0 - (double) intersection.size() / union.size();
return dist; }
```

universitätbonn

Task 4: Implement a Spark Program "UserSet"

1. Setting Spark Configuration using SparkConf
2. Setting **SparkContext**
3. Reading the input file and storing as **RDD**
4. **Line-split** (tab separation) and return List array of the artist names
5. Turn the words into **(word,1) pairs** using **mapToPair**
6. **reduceByKey** to **aggregate** the count of each key
7. Create an RDD to store the **keys**(artist names) as String and corresponding users as **userSet**
8. Iterating through every key to collect the users for a particular userSet and display the output
9. Saving the output as txt file
10. Using **lookup** to listen the number of event for "Mark Knopfler"

universitätbonn

Task 5: Implement a Spark Program "CloseToMark"

1. Setting Spark Configuration using SparkConf
2. Setting **SparkContext**
3. Reading the input file and storing as **RDD**
4. **Line-split** (tab separation) and return List array of the artist names
5. Turn the words into **(word,1) pairs** using **mapToPair**
6. **reduceByKey** to **aggregate** the count of each key
7. **Create an RDD to store the keys(artist names) as String and corresponding users as userSet**
8. Calculate the Jaccard distance from **distanceTo()** function defined in task 3
9. Saving the output as txt file
10. Using **lookup** to listen the number of event for "Mark Knopfler"

universitätbonn

Development Environment

Programming Language:

- Java with Apache Spark

Data structures:

- Lists
- RDD
- Arrays
- HashSet

Libraries:

- JavaPairRDD
- JavaSparkContext
- SparkConf

universitätbonn

THANK YOU