Data Intensive Lab2

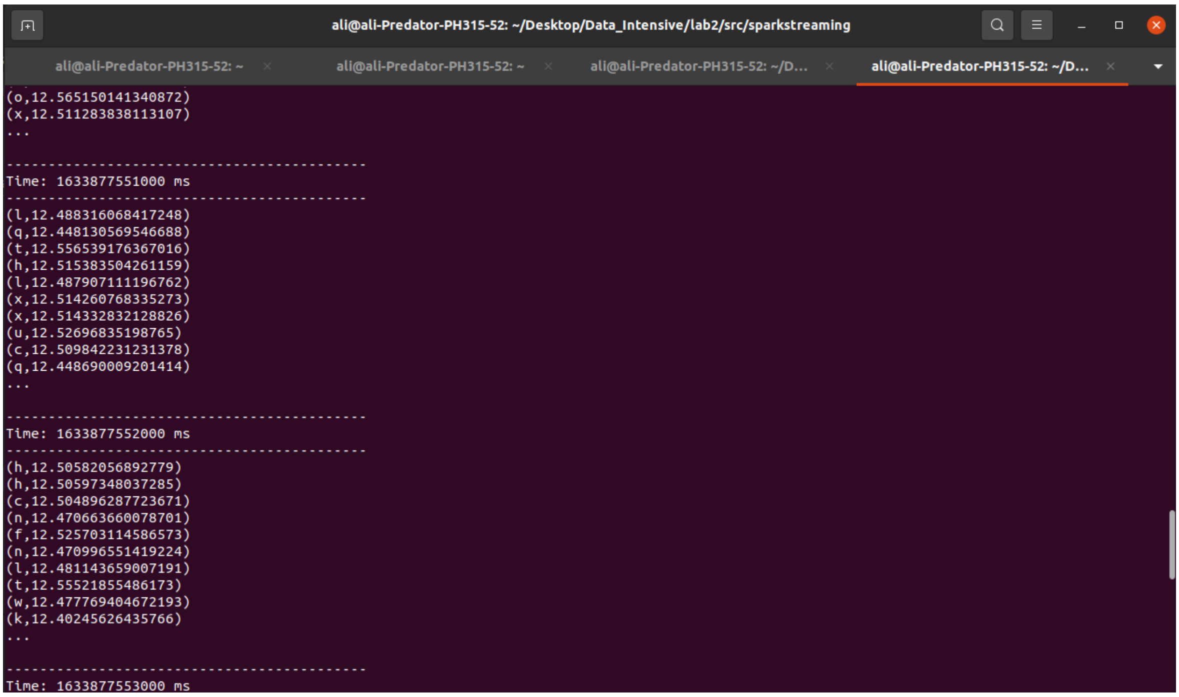
Ali Shibli
Carina Wickström

October 10, 2021

1 Results of Lab2

1.1 Task 1

Calculate the average of each key using spark streaming:

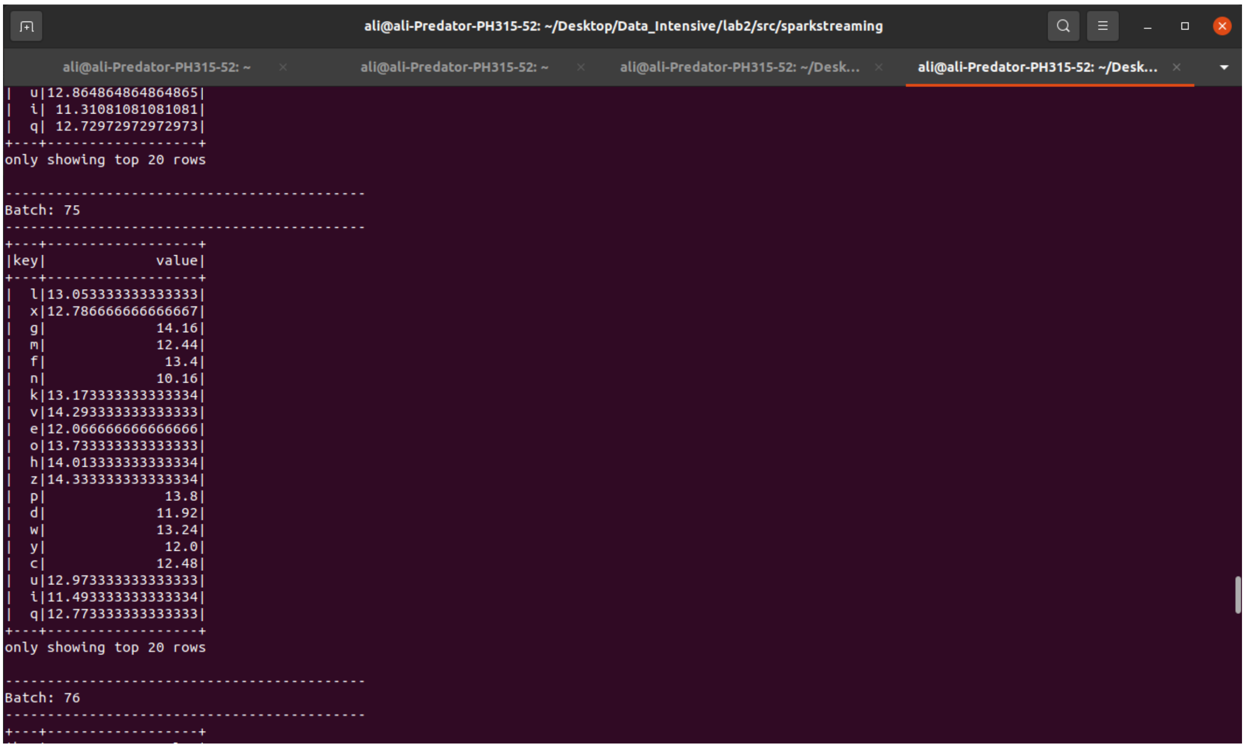

ali@ali-Predator-PH315-52: ~/Desktop/Data_Intensive/lab2/src/sparkstreaming
ali@ali-Predator-PH315-52: ~
ali@ali-Predator-PH315-52: ~
ali@ali-Predator-PH315-52: ~/D...
ali@ali-Predator-PH315-52: ~/D...

```
(o,12.565150141340872)
(x,12.511283838113107)
...

-------------------------------------------
Time: 1633877551000 ms
-------------------------------------------
(l,12.488316068417248)
(q,12.448130569546688)
(t,12.556539176367016)
(h,12.515383504261159)
(l,12.487907111196762)
(x,12.514260768335273)
(x,12.514332832128826)
(u,12.52696835198765)
(c,12.509842231231378)
(q,12.448690009201414)
...

-------------------------------------------
Time: 1633877552000 ms
-------------------------------------------
(h,12.50582056892779)
(h,12.50597348037285)
(c,12.504896287723671)
(n,12.470663660078701)
(f,12.525703114586573)
(n,12.470996551419224)
(l,12.481143659007191)
(t,12.55521855486173)
(w,12.477769404672193)
(k,12.40245626435766)
...

-------------------------------------------
Time: 1633877553000 ms
```

Figure 1: Results with Spark Streaming

1.2 Task 2

Calculate the average of each key using spark structured streaming:


ali@ali-Predator-PH315-52: ~/Desktop/Data_Intensive/lab2/src/sparkstreaming
ali@ali-Predator-PH315-52: ~
ali@ali-Predator-PH315-52: ~
ali@ali-Predator-PH315-52: ~/Desk...
ali@ali-Predator-PH315-52: ~/Desk...

```
|  u|12.864864864864865|
|  i| 11.31081081081081|
|  q| 12.72972972972973|
+---+------------------+
only showing top 20 rows

-------------------------------------------
Batch: 75
-------------------------------------------
+---+------------------+
|key|             value|
+---+------------------+
|  l|13.053333333333333|
|  x|12.786666666666667|
|  g|             14.16|
|  m|             12.44|
|  f|              13.4|
|  n|             10.16|
|  k|13.173333333333334|
|  v|14.293333333333333|
|  e|12.066666666666666|
|  o|13.733333333333333|
|  h|14.013333333333334|
|  z|14.333333333333334|
|  p|              13.8|
|  d|             11.92|
|  w|             13.24|
|  y|              12.0|
|  c|             12.48|
|  u|12.973333333333333|
|  i|11.493333333333334|
|  q|12.773333333333333|
+---+------------------+
only showing top 20 rows

-------------------------------------------
Batch: 76
-------------------------------------------
+---+------------------+
```

Figure 2: Results with Structured Spark Streaming

1.3 Task 3

1. Display the names of the users that are at least 30 years old.

    1. (4,(David,42))

    2. (6,(Fran,50))

    3. (3,(Charlie,65))

    4. (7,(Alex,55))

    5. (5,(Ed,55))

2. Display who likes who.

    1. Bob likes Alice

    2. Bob likes David

    3. Charlie likes Bob

    4. Charlie likes Fran

   5. David likes Alice
   6. Ed likes Bob
   7. Ed likes Charlie
   8. Ed likes Fran
   9. Alex likes Ed
   10. Alex likes Fran

3. If someone likes someone else more than 5 times than that relationship is getting pretty serious, so now display the lovers.
   1. Bob loves Alice
   2. Ed loves Charlie

4. Print the number of people who like each user (e.g., Alice is liked by 2 people).
   1. David is liked by 1
   2. Alice is liked by 2
   3. Fran is liked by 3
   4. Charlie is liked by 1
   5. Alex is liked by 0
   6. Ed is liked by 1
   7. Bob is liked by 2

5. Print the names of the users who are liked by the same number of people they like (e.g., Bob and David).
   1. David
   2. Bob

6. Find the oldest follower of each user (hint: use the aggregateMessages).
   1. Bob is the oldest follower of David
   2. David is the oldest follower of Alice
   3. Charlie is the oldest follower of Fran
   4. Ed is the oldest follower of Charlie
   5. Alex does not have any followers
   6. Alex is the oldest follower of Ed
   7. Charlie is the oldest follower of Bob