

Course: INFO7250 Big Data Systems Engineering

Name: Pramod Nagare

NUID: 001858910

Email: nagare.p@husky.ney.edu

Project: Streaming data pipeline for real-time analytics and Sentiment Analysis

GitHub: <https://github.com/pramodnagare/INFO7250BigDataSystems>

Highlights:

- GCP Streaming and Batch Processing data analytical pipeline
- Bloom filter
- Hadoop cluster setup
- Google data explorer interactive dashboard
- Optimization (Combiner, distributed caches, cluster, compact data type)
- Multithread/multiprocessor scripts for data preparation
- Map-Reduce in Python and Java

Dataset:

- <https://registry.opendata.aws/amazon-reviews/>
- <https://github.com/awslabs/open-data-registry/blob/master/datasets/amazon-reviews.yaml>
- <https://s3.amazonaws.com/amazon-reviews-pds/readme.html>
- 34+ GB Dataset .gzip format
- 130+ millions amazon customer reviews

Task 1: Data download and extraction

Implementation:

1. Downloading AWS Customer review data from source (AWS S3 bucket)
 - Data Size: 130+ millions of rows, ~80GB of data over 51 files
 - More details can be found on [index.txt file](#)
 - [Developed parallel processing python script](#)
2. Extracting downloaded .gzip file into tsv
 - [Developed parallel processing python script](#)

Screenshots:

Downloading dataset script: File is available [here](#)

```
In [1]: #This jupyter notebook is for downloading data parallelly from s3 bucket to local file system:

In [2]: import boto3
import botocore
import time
import os
import pandas as pd
from multiprocessing.pool import ThreadPool

In [3]: data_link = "amazon-reviews-pds"
output_dir = "../Data2/"

s3 = boto3.client('s3')
response = s3.list_objects_v2(Bucket=data_link)

In [12]: def downloadFile(fileItem):
head, file = os.path.split(fileItem)
filename = output_dir + file
s = time.time()
s3.download_file(Bucket=data_link, Key=fileItem, Filename=filename)
e = time.time()
print("Downloaded file {} in {} secs!".format(fileItem, e-s))
return None
```

```

In [5]: data = []
        for fileItem in response['Contents']:
            if str(fileItem['Key']).startswith('tsv'):
                head, file = os.path.split(fileItem['Key'])
                if file != "":
                    data.append(fileItem)

In [6]: df = pd.DataFrame(data)
        df = df.sort_values('Size').reset_index()

In [47]: batches = {}
        batch_num = 0
        data = []
        size_sum = 0
        for i in range(df.shape[0]):
            size_sum += int(df['Size'][i])
            data.append(df['Key'][i])
            if (size_sum/1024/1024 > 200) or (len(data) > 8):
                if len(data) > 1:
                    data.pop()
                    batches[batch_num] = data
                    size_sum = int(df['Size'][i])
                    data = [df['Key'][i]]
                else:
                    batches[batch_num] = data
                    size_sum = 0
                    data = []
                    batch_num += 1
        batches[batch_num] = data

In [13]: start = time.time()
        for batch in batches.values():
            results = ThreadPool(len(batch)).imap_unordered(downloadFile, batch)
            for result in results:
                while result != None:
                    continue
        end = time.time()
        print("Total Time for Downloading files is {} secs!".format(end-start))

```

Extracting dataset script: File is available [here](#)

```

In [1]: import concurrent.futures
        import os
        import glob
        import time
        import shutil
        import gzip

In [4]: input_folder = r"H:\INF07250_BigData\Project\Data/"
        output_folder = r"../Main_Data/"
        file_type = ".tsv.gz"

        NUM_WORKERS = int(os.environ['NUMBER_OF_PROCESSORS'])

In [5]: files = glob.glob(input_folder+"*.tsv.gz")
        print("Total Number of file to extract: {}".format(len(files)))

        Total Number of file to extract: 51

In [6]: def extractFile(filename):
        path, file = os.path.split(filename)
        file = file.replace(".gz", "")
        with gzip.open(filename) as file_input:
            with open(output_folder+file, "wb") as file_output:
                shutil.copyfileobj(file_input, file_output)
        return None

In [7]: start = time.time()

        with concurrent.futures.ThreadPoolExecutor(max_workers=NUM_WORKERS) as executor:
            futures = [executor.submit(extractFile, file) for file in files]
            concurrent.futures.wait(futures)

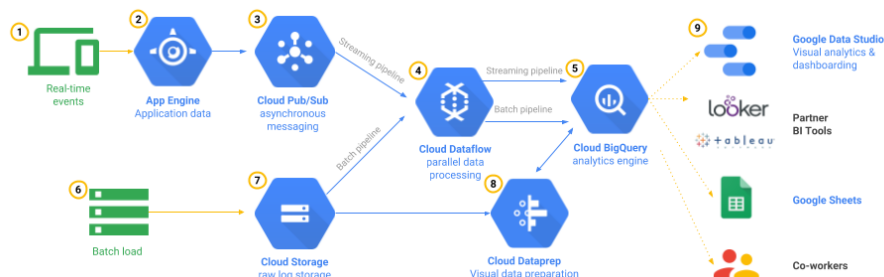
        end = time.time()
        print("Total Time for extraction of files is {} secs!".format(end-start))

        Total Time for extraction of files is 1601.342586517334 secs!

```

Task 2: Building streaming data analytics pipeline

Implementation:



1. Architecture implementation:

- Created streaming data analytical pipeline using GCP pub/sub and python
- Developed a multi-process python script for data upload to GCP Big Query
- This script can take .tsv.gz as an input, we are saving time of extracting multiple files

2. Data querying using Big Query:

- Implemented simple queried to explore Big Query

Screenshots: File is available [here](#)

```
# !pip install google-cloud
# !pip install google-cloud-pubsub
# !pip install google.cloud.bigquery
# !pip install google.cloud.storage
```

```
from google.oauth2 import service_account #For GCP Account connection
from google.cloud import pubsub_v1 # For PubSub Client
from google.cloud import bigquery # For BigQuery Client
from google.cloud import storage # For Cloud Storage Client
import json # For Message syntax
import os
```

```
#Setting up the credential file
cred = service_account.Credentials.from_service_account_file('../Configuration/INFO7250-b2d76e3086d3.json')
```

```
#Setting up the Configuration Variables:
project_id = "info7250"
project_name = "info7250"
topic_name = "info7250-topic"
bucket_name = "aws-review-data"
subscription_name = "info7374-subscription"
dataset_name = "AWS_Product_Review"
table_name = "Reviews"
job_name = "INFO7250_DataFlow"
```

```
#Creating a Cloud Storage Bucket:
storage_client = storage.Client(project=project_id,credentials=cred)
bucket = storage_client.create_bucket(bucket_name)
print('Bucket {} created.'.format(bucket.name))
```

Bucket aws-review-data created.

```
#Creating PubSub Topic:
publisher = pubsub_v1.PublisherClient(credentials=cred)
topic_path = publisher.topic_path(project_id, topic_name)
topic = publisher.create_topic(topic_path)
print('Topic created: {}'.format(topic))
```

Topic created: name: "projects/info7250/topics/info7250-topic"

```
#Creating PubSub Subscription:
subscriber = pubsub_v1.SubscriberClient(credentials = cred)
topic_path = subscriber.topic_path(project_id, topic_name)
subscription_path = subscriber.subscription_path(project_id, subscription_name)
subscription = subscriber.create_subscription(subscription_path, topic_path)
print('Subscription created: {}'.format(subscription))
```

```
Subscription created: name: "projects/info7250/subscriptions/info7374-subscription"
topic: "projects/info7250/topics/info7250-topic"
push_config {
}
ack_deadline_seconds: 10
message_retention_duration {
  seconds: 604800
}
expiration_policy {
  ttl {
    seconds: 2678400
  }
}
```

```
#Creating a BigQuery Dataset:
client = bigquery.Client(project=project_id, credentials=cred)
dataset_id = client.project+"."+dataset_name
dataset = bigquery.Dataset(dataset_id)
dataset.location = "US"
dataset = client.create_dataset(dataset)
print("Created dataset {}.{}".format(client.project, dataset.dataset_id))
```

Created dataset info7250.AWS_Product_Review

Dataset schema file available [here](#)

```
#Import Table schema from a JSON file:
schema = json.loads(open('../Configuration/Reviews').read())
table_schema = []
for line in schema:
    col = bigquery.SchemaField(name=line['name'], field_type=line['type'], mode=line['mode'], description=
line['description'])
    table_schema.append(col)
```

```
#Creating a BigQuery Table under the Dataset created:
table_id = project_id+"-dataset_name-"+table_name
table = bigquery.Table(table_id, schema=table_schema)
table = client.create_table(table)
print("Created table {}.{}.{}".format(table.project, table.dataset_id, table.table_id))
```

Created table info7250.AWS_Product_Review.Reviews

```
gcloud dataflow jobs run JOB_NAME \ --gcs-location gs://dataflow-templates/latest/PubSub_Subscription_to_BigQuery \ --
parameters \
    inputSubscription=projects/YOUR_PROJECT_ID/subscriptions/YOUR_SUBSCRIPTION_NAME\
    outputTableSpec=YOUR_PROJECT_ID:YOUR_DATASET.YOUR_TABLE_NAME
```

```
#Creating a Dataflow for PubSub to Big Query using template:
command = "gcloud dataflow jobs run {} --gcs-location gs://dataflow-templates/latest/PubSub_Subscription_t
o_BigQuery {}".format(job_name)
command += "--parameters "
command += "outputTableSpec={}.{}.{}".format(project_id, dataset_name, table_name)
command += "inputSubscription=projects/{}/subscriptions/{}".format(project_name, subscription_name)
```

```
print(command)
```

```
os.system(command)
```

```
gcloud dataflow jobs run INFO7250>Dataflow --gcs-location gs://dataflow-templates/latest/PubSub_Subscripti
on_to_BigQuery --parameters outputTableSpec=info7250:AWS_Product_Review.Reviews,inputSubscription=project
s/info7250/subscriptions/info7374-subscription
0
```

Data real-time streaming script

```
#Considering you have already created Streaming Dataflow manually using GCP Console for PubSub Subscriptio
n to BigQuery Table.
#Publish a real-time event/message on PubSub Topic.
```

```
message_data = {
    "marketplace": "Sunil",
    "customer_id": 18778586,
    "review_id": "RD1J57QYB6XMR",
    "product_id": "B00EDBY7X8",
    "product_parent": 122952789,
    "product_title": "Pramod",
    "product_category": "Toys",
    "star_rating": 5,
    "helpful_votes": 0,
    "total_votes": 0,
    "vine": False,
    "verified_purchase": True,
    "review_headline": "Five stars",
    "review_body": "Excellent!!!",
    "review_date": "2015-08-31"
}
```

```
#Formatting message data before publishing:
```

```
message_data = json.dumps(message_data)
message_data = message_data.encode('utf-8')
```

```
#Publishing a message on the PubSub Topic Created:
```

```
response = publisher.publish(topic_path, message_data, origin='python-sample')
```

Batch Processing script: File available [here](#)

```
import os
import glob
import time
import concurrent.futures
```

```
input_folder = r"D:\Semester4\INFO7250-BigData\Project\Data/"
file_type = ".tsv.gz"
```

```
NUM_WORKERS = int(os.environ["NUMBER_OF_PROCESSORS"])
print("There are {} number of processors".format(NUM_WORKERS))
```

There are 8 number of processors

```
files = glob.glob(input_folder+"*.tsv.gz")
print("Total Number of file to upload: {}".format(len(files)))
```

Total Number of file to upload: 51

```
def uploadFile(filename):
    path, file = os.path.split(filename)
    print("Uploading file {} on big query".format(file))
    command = "bg --location=us load --null-marker=NULL --quote '\"' "
    command += "-E UTF-8 --autodetect --source-format=CSV --field-delimiter '\\t' AWS_Product_Review.Reviews"
    + filename
    os.system(command)
    return None
```

```
for file in files[:1]:
    file = r"D:\Semester4\INFO7250-BigData\Project\Data\amazon_reviews_us_Personal_Care_Appliances_v1_00.t
sv.gz"
    uploadFile(file)
```

Uploading file amazon_reviews_us_Personal_Care_Appliances_v1_00.tsv.gz on big query

```
start = time.time()
```

```
with concurrent.futures.ThreadPoolExecutor(max_workers=NUM_WORKERS) as executor:
    futures = {executor.submit(uploadFile, file) for file in files}
    concurrent.futures.wait(futures)
```

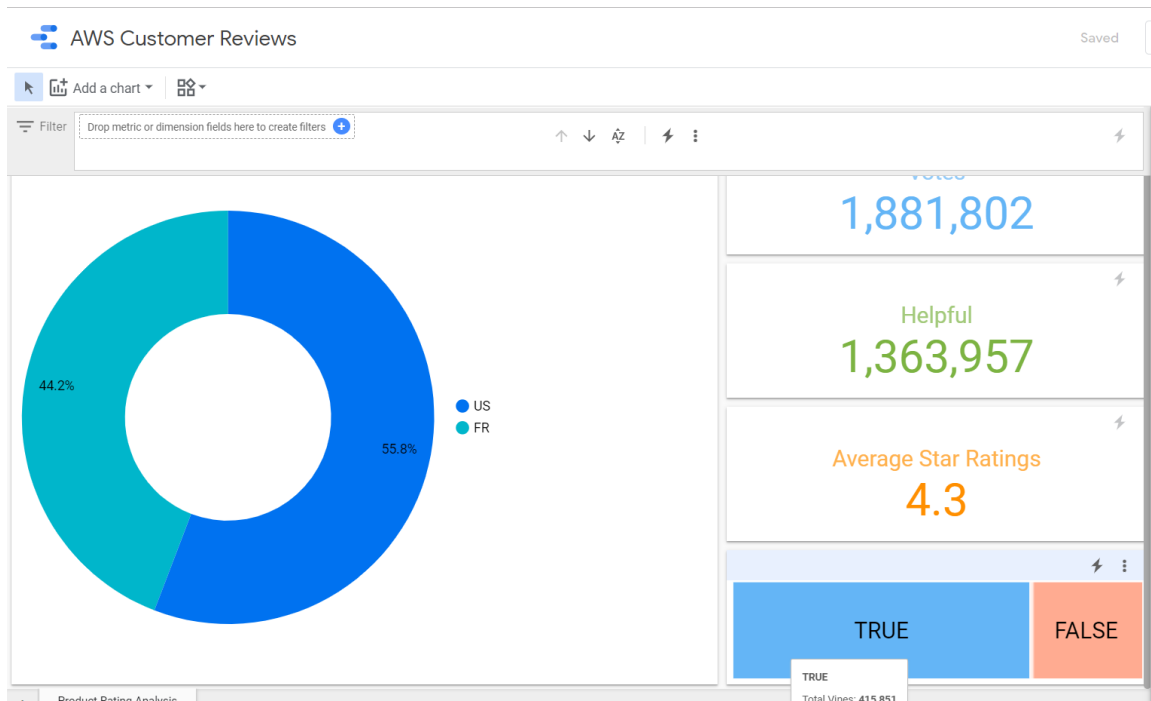
```
end = time.time()
print("Total Time for uploading of files is {} secs!".format(end-start))
```

Task 3: Real Time data visualization

Implementation:

1. Developed interactive dashboard using **Google Data explorer**
 - Created custom fields for detail data analysis
 - Data visualization using interactive graphs
 - Export and shared the dashboard

Screenshot:



Task 4: Bloom filter

Implementation:

1. Designed and developed bloom filter for stop word, positive and negative word analysis
2. Tried importing and exporting bloom filter using pickle file in python and data output stream in java

Screenshots: File is available [here](#)

JAVA Implementation:

```
1 import com.google.common.hash.BloomFilter;
2 import com.google.common.hash.Funnels;
3
4 import java.io.*;
5 import java.util.ArrayList;
6
7 import static com.google.common.base.Charsets.UTF_8;
8
9 public class MyBloomFilter {
10
11     public static ArrayList<String> getList(String filename) throws IOException {
12
13         ArrayList<String> words = new ArrayList<String>();
14
15         File wordFile = new File(filename);
16
17         BufferedReader br = new BufferedReader(new FileReader(wordFile));
18
19         String st;
20         while ((st = br.readLine()) != null)
21             words.add(st);
22
23         return words;
24     }
25 }
```

```

26     public static void main( String [] args) throws IOException {
27
28         ArrayList<String> positiveWords = getList("files/positive_words.txt");//getList(args[0]);
29         ArrayList<String> negativeWords = getList("files/negative_words.txt");//getList(args[1]);
30
31         float fpr = 0.0001f; //Float.parseFloat(args[2]);
32
33         BloomFilter<String> positiveWordsFilter = BloomFilter.create(Funnels.stringFunnel(UTF_8), positiveWords.size(), fpr);
34
35         for (String word : positiveWords)
36             positiveWordsFilter.put(word);
37
38         BloomFilter<String> negativeWordsFilter = BloomFilter.create(Funnels.stringFunnel(UTF_8), positiveWords.size(), fpr);
39
40         for (String word : negativeWords)
41             negativeWordsFilter.put(word);
42
43
44         System.out.println(positiveWordsFilter.mightContain("winner"));
45         System.out.println(positiveWordsFilter.mightContain("pramod"));
46         OutputStream os = new FileOutputStream("PositiveBloomFilter.txt");
47         positiveWordsFilter.writeTo(os);
48
49         InputStream bf = new FileInputStream("PositiveBloomFilter.txt");
50         BloomFilter loadBloomFilter = BloomFilter.readFrom(bf,Funnels.stringFunnel(UTF_8));
51
52         System.out.println(loadBloomFilter.mightContain("winner"));
53         System.out.println(loadBloomFilter.mightContain("Pramod"));
54
55
56         System.out.println(negativeWordsFilter.mightContain("absurd"));
57         System.out.println(negativeWordsFilter.mightContain("pramod"));
58         OutputStream os1 = new FileOutputStream("NegativeBloomFilter.txt");
59         negativeWordsFilter.writeTo(os1);
60
61         InputStream bfl = new FileInputStream("NegativeBloomFilter.txt");
62         BloomFilter loadBloomFilter1 = BloomFilter.readFrom(bfl,Funnels.stringFunnel(UTF_8));
63
64         System.out.println(loadBloomFilter1.mightContain("absurd"));
65         System.out.println(loadBloomFilter1.mightContain("Pramod"));
66
67
68     }
69 }

```

Python Implementation: File is available [here](#)

```

5  import math
6  import mmh3
7  from bitarray import bitarray
8
9  class BloomFilter(object):
10
11      ...
12      Class for Bloom filter, using murmur3 hash function
13      ...
14
15      def __init__(self, items_count,fp_prob):
16          ...
17          items_count : int
18              Number of items expected to be stored in bloom filter
19          fp_prob : float
20              False Positive probability in decimal
21          ...
22          # False possible probability in decimal
23          self.fp_prob = fp_prob
24
25          # Size of bit array to use
26          self.size = self.get_size(items_count,fp_prob)
27
28          # number of hash functions to use
29          self.hash_count = self.get_hash_count(self.size,items_count)
30
31          # Bit array of given size
32          self.bit_array = bitarray(self.size)
33
34          # initialize all bits as 0
35          self.bit_array.setall(0)
36

```

```

37     def add(self, item):
38         ...
39         Add an item in the filter
40         ...
41         digests = []
42         for i in range(self.hash_count):
43
44             # create digest for given item.
45             # i work as seed to mmh3.hash() function
46             # With different seed, digest created is different
47             digest = mmh3.hash(item,i) % self.size
48             digests.append(digest)
49
50             # set the bit True in bit_array
51             self.bit_array[digest] = True
52
53     def check(self, item):
54         ...
55         Check for existence of an item in filter
56         ...
57         for i in range(self.hash_count):
58             digest = mmh3.hash(item,i) % self.size
59             if self.bit_array[digest] == False:
60
61                 # if any of bit is False then,its not present
62                 # in filter
63                 # else there is probability that it exist
64                 return False
65         return True
66
67     @classmethod
68     def get_size(self,n,p):
69         ...
70         Return the size of bit array(m) to used using
71         following formula
72          $m = -(n * \lg(p)) / (\lg(2)^2)$ 
73         n : int
74             number of items expected to be stored in filter
75         p : float
76             False Positive probability in decimal
77         ...
78          $m = -(n * \mathbf{math.log}(p))/(\mathbf{math.log}(2)^2)$ 
79         return int(m)
80
81     @classmethod
82     def get_hash_count(self, m, n):
83         ...
84         Return the hash function(k) to be used using
85         following formula
86          $k = (m/n) * \lg(2)$ 
87
88         m : int
89             size of bit array
90         n : int
91             number of items expected to be stored in filter
92         ...
93          $k = (m/n) * \mathbf{math.log}(2)$ 
94         return int(k)

```

Analysis:

Bloom Filter files:

[NegativeBloomFilter.txt](#) Size: 4.7 KB

[PositiveBloomFilter.txt](#) Size: 4.7 KB

Word list files:

[negative words.txt](#) Size: 43.7 KB

[positive words.txt](#) Size: 16.8 KB

Task 5: Map Reduce for Sentiment Analysis

Implementation:

1. Created map-reduce jobs in python on Hadoop using Hadoop streaming jar
2. Tested map-reduce on local machine

3. Implemented both MR jobs with bloom filter and without bloom filter
4. Optimized the MR jobs with Hadoop best practices
 - Using compact data type for the job
 - Using combiner
 - Running MR jobs on Hadoop cluster on GCP and AWS
 - Implemented distributed caches

JAVA Implementation of MR job available here:

[INFO7250 Project](#)

Mapper

```
1  package Project;
2
3  import com.google.common.hash.BloomFilter;
4  import com.google.common.hash.Funnels;
5  import org.apache.hadoop.filecache.DistributedCache;
6  import org.apache.hadoop.fs.Path;
7  import org.apache.hadoop.io.IntWritable;
8  import org.apache.hadoop.io.LongWritable;
9  import org.apache.hadoop.io.Text;
10 import org.apache.hadoop.mapreduce.Mapper;
11
12 import java.io.FileInputStream;
13 import java.io.IOException;
14 import java.io.InputStream;
15
16 import static com.google.common.base.Charsets.UTF_8;
17
18 public class ProjectMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
19
20     BloomFilter positiveBloomFilter, negativeBloomFilter;
21     Text word = new Text();
22     IntWritable one = new IntWritable(1);
23
24     public ProjectMapper() throws IOException {
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
```


Reducer:

```
1 package Project;
2
3 import org.apache.hadoop.io.IntWritable;
4 import org.apache.hadoop.io.Text;
5 import org.apache.hadoop.mapreduce.Reducer;
6
7 import java.io.IOException;
8
9 public class ProjectReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
10
11
12     @Override
13     protected void reduce(Text key, Iterable<IntWritable> values, Reducer<Text, IntWritable, Text, IntWritable>.Context context) throws IO
14
15         int count = 0;
16
17         for(IntWritable val: values){
18             count += val.get();
19         }
20
21         IntWritable total = new IntWritable(count);
22         context.write(key, total);
23     }
24 }
25 }
```

Driver:

```
1 package Project;
2
3 import org.apache.hadoop.conf.Configuration;
4 import org.apache.hadoop.filecache.DistributedCache;
5 import org.apache.hadoop.fs.Path;
6 import org.apache.hadoop.io.IntWritable;
7 import org.apache.hadoop.io.Text;
8 import org.apache.hadoop.mapreduce.Job;
9 import org.apache.hadoop.mapreduce.Mapper;
10 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
11 import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
12 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
13 import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
14
15 import java.io.IOException;
16
17 public class Driver extends Mapper<Text, Text, Text, IntWritable> {
18
19     public static void main (String[] args) throws IOException, ClassNotFoundException, InterruptedException {
20
21         Configuration conf = new Configuration();
22
23         Job job = Job.getInstance(conf, "INF07250 Final Project");
24
25         job.setJarByClass(Driver.class);
26         job.setMapperClass(ProjectMapper.class);
27         // job.setMapperClass(NormalMapper.class);
28         job.setReducerClass(ProjectReducer.class);
29
30         job.setCombinerClass(ProjectReducer.class);
31
32         job.setInputFormatClass(TextInputFormat.class);
33         job.setOutputFormatClass(TextOutputFormat.class);
34
35         job.setOutputKeyClass(Text.class);
36         job.setOutputValueClass(IntWritable.class);
37
38         FileInputFormat.addInputPath(job, new Path(args[0]));
39         FileOutputFormat.setOutputPath(job, new Path(args[1]));
40
41         DistributedCache.addCacheFile(new Path(args[2]).toUri(), job.getConfiguration());
42         DistributedCache.addCacheFile(new Path(args[3]).toUri(), job.getConfiguration());
43
44         System.exit(job.waitForCompletion(true) ? 0:1);
45     }
46 }
```

Python Implementation of MR job is available here:

[Python](#)

Mapper

```
1 #!/usr/bin/env python
2 import sys
3 import re
4 from BloomFilter import BloomFilter
5
6 stop_words = []
7 sw_file = open('stop_words.txt', 'r')
8 for line in sw_file:
9     words = line.replace(' ','').replace('\n', '').replace('\r', '').split(',')
10    stop_words.extend(words)
11
12 n = len(stop_words) #no of items to add
13 p = 0.0001 #false positive probability
14
15 bloomf = BloomFilter(n,p)
16
17 for item in stop_words:
18     bloomf.add(item)
19
20 RE_D = re.compile('\d')
21
22 for line in sys.stdin:
23     #--- remove leading and trailing whitespace---
24     line = line.strip().split("\t")
25
26     if len(line) >= 13:
27
28         #--- split the line into words ---
29         review = re.findall(r'\w+', line[13])
30         for word in review:
31             if not bloomf.check(word) and len(word) > 1 and not RE_D.search(word): #not bloomf.check(word) and
32                 print('%s\t%s' % (word.lower(), "1"))
```

Reducer

```
1 #!/usr/bin/env python
2 # coding: utf-8
3
4 import sys
5 # maps words to their counts
6 word2count = {}
7
8 # input comes from STDIN
9 for line in sys.stdin:
10     # remove leading and trailing whitespace
11     line = line.strip()
12
13     # parse the input we got from mapper.py
14     word, count = line.split('\t', 1)
15     # convert count (currently a string) to int
16     try:
17         count = int(count)
18     except ValueError:
19         continue
20
21     try:
22         word2count[word] = word2count[word]+count
23     except:
24         word2count[word] = count
25
26 # write the tuples to stdout
27 # Note: they are unsorted
28 for word in word2count.keys():
29     print('%s\t%s' % ( word, word2count[word] ))
```

Task 6: Hadoop Cluster setup on GCP

Implementation:

1. Created a GCP Hadoop cluster with 2 worker nodes and 1 master
2. Submitted MR jobs on cluster using GCP utility
3. Implemented persistent storage for MR jobs
4. Ran some MR jobs on cluster with SSH to master node
5. Accessed the Hadoop dashboard for the job monitoring

MR Job performance Time:

1. On Local machine with 1 node: **5 hrs**
2. On GCP Hadoop cluster with 2 worker node and 1 master: **1 hr. 42 min**

Screenshots:

On local machine:

```
(base) pramodnagare@pramodnagare:~/INF07250_Project/BloomFilter$
(base) pramodnagare@pramodnagare:~/INF07250_Project/BloomFilter$
(base) pramodnagare@pramodnagare:~/INF07250_Project/BloomFilter$ time hadoop jar INF07250_Project_Normal.jar Project.Driver /AWS-Review-Dat
a/amazon_reviews_us_Digital_Ebook_Purchase_v1_00.tsv /test1 /book/positive_words.txt /book/negative_words.txt

2019-12-12 02:58:43,120 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-12-12 02:58:44,251 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface
and execute your application with ToolRunner to remedy this.
2019-12-12 02:58:44,265 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/pramodnagare/.stagi
ng/job_1576122566681_0010
2019-12-12 02:58:44,479 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-12 02:58:44,471 INFO input.FileInputFormat: Total input files to process : 1
2019-12-12 02:58:44,955 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-12 02:58:44,995 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-12 02:58:45,007 INFO mapreduce.JobSubmitter: number of splits:50
2019-12-12 02:58:45,235 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-12 02:58:45,258 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1576122566681_0010
2019-12-12 02:58:45,258 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-12-12 02:58:45,563 INFO conf.Configuration: resource-types.xml not found
2019-12-12 02:58:45,564 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-12-12 02:58:45,677 INFO impl.YarnClientImpl: Submitted application application_1576122566681_0010
2019-12-12 02:58:45,742 INFO mapreduce.Job: The url to track the job: http://pramodnagare:8088/proxy/application_1576122566681_0010/
2019-12-12 02:58:45,743 INFO mapreduce.Job: Running job: job_1576122566681_0010
2019-12-12 02:58:56,985 INFO mapreduce.Job: Job job_1576122566681_0010 running in uber mode : false
2019-12-12 02:58:56,986 INFO mapreduce.Job: map 0% reduce 0%
2019-12-12 03:01:20,950 INFO mapreduce.Job: map 1% reduce 0%
2019-12-12 03:05:23,017 INFO mapreduce.Job: map 2% reduce 0%
2019-12-12 03:09:13,700 INFO mapreduce.Job: map 3% reduce 0%
2019-12-12 03:13:18,335 INFO mapreduce.Job: map 4% reduce 0%
2019-12-12 03:17:27,052 INFO mapreduce.Job: map 5% reduce 0%
2019-12-12 03:21:56,085 INFO mapreduce.Job: map 6% reduce 0%
2019-12-12 03:25:53,661 INFO mapreduce.Job: map 7% reduce 0%
2019-12-12 03:30:01,248 INFO mapreduce.Job: map 8% reduce 0%
2019-12-12 03:31:22,490 INFO mapreduce.Job: map 9% reduce 0%
2019-12-12 03:32:06,580 INFO mapreduce.Job: map 10% reduce 0%
2019-12-12 03:32:41,656 INFO mapreduce.Job: map 11% reduce 0%
```

```
Total vcore-milliseconds taken by all map tasks=90019733
Total vcore-milliseconds taken by all reduce tasks=13462435
Total megabyte-milliseconds taken by all map tasks=92180206592
Total megabyte-milliseconds taken by all reduce tasks=13785533440
Map-Reduce Framework
  Map input records=12520723
  Map output records=59300440
  Map output bytes=1249507931
  Map output materialized bytes=1368109111
  Input split bytes=7650
  Combine input records=0
  Combine output records=0
  Reduce input groups=6559
  Reduce shuffle bytes=1368109111
  Reduce input records=59300440
  Reduce output records=6559
  Spilled Records=118600880
  Shuffled Maps =50
  Failed Shuffles=0
  Merged Map outputs=50
  GC time elapsed (ms)=183398
  CPU time spent (ms)=64477290
  Physical memory (bytes) snapshot=22984404992
  Virtual memory (bytes) snapshot=135886753792
  Total committed heap usage (bytes)=9673637888
  Peak Map Physical memory (bytes)=507383808
  Peak Map Virtual memory (bytes)=2676359168
  Peak Reduce Physical memory (bytes)=855752704
  Peak Reduce Virtual memory (bytes)=2675609600
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=6697186130
File Output Format Counters
  Bytes Written=154437

real    290m27.345s
user    0m41.153s
sys     0m7.869s
(base) pramodnagare@pramodnagare:~/INF07250_Project/BloomFilter$
(base) pramodnagare@pramodnagare:~/INF07250_Project/BloomFilter$
(base) pramodnagare@pramodnagare:~/INF07250_Project/BloomFilter$
(base) pramodnagare@pramodnagare:~/INF07250_Project/BloomFilter$
```

On GCP Hadoop cluster:

```
ssh.cloud.google.com/projects/info7250/zones/us-central1-c/instances/hadoop-info7250-m?authuser=1&hl=en_US&projectNumber=883502186074
parmodnagare1993@hadoop-info7250-m:~$
parmodnagare1993@hadoop-info7250-m:~$
parmodnagare1993@hadoop-info7250-m:~$
parmodnagare1993@hadoop-info7250-m:~$ time hadoop jar INFO7250_Project_Normal.jar Project.Driver /data/amazon_revie
ws_us_Digital_Ebook_Purchase_v1_00.tsv /normal /data/positive_words.txt /data/negative_words.txt
19/12/12 09:10:16 INFO client.RMProxy: Connecting to ResourceManager at hadoop-info7250-m/10.128.0.10:8032
19/12/12 09:10:16 INFO client.AHSProxy: Connecting to Application History server at hadoop-info7250-m/10.128.0.10:1
0200
19/12/12 09:10:16 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement t
he Tool interface and execute your application with ToolRunner to remedy this.
19/12/12 09:10:17 INFO input.FileInputFormat: Total input files to process : 1
19/12/12 09:10:17 INFO mapreduce.JobSubmitter: number of splits:50
19/12/12 09:10:17 INFO Configuration.deprecation: yarn.resourcemanager.system-metrics-publisher.enabled is deprecat
ed. Instead, use yarn.system-metrics-publisher.enabled
19/12/12 09:10:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1576131264570_0014
19/12/12 09:10:17 INFO impl.YarnClientImpl: Submitted application application_1576131264570_0014
19/12/12 09:10:17 INFO mapreduce.Job: The url to track the job: http://hadoop-info7250-m:8080/proxy/application_157
6131264570_0014/
19/12/12 09:10:17 INFO mapreduce.Job: Running job: job_1576131264570_0014
19/12/12 09:10:26 INFO mapreduce.Job: Job job_1576131264570_0014 running in uber mode : false
19/12/12 09:10:26 INFO mapreduce.Job: map 0% reduce 0%
19/12/12 09:11:22 INFO mapreduce.Job: map 1% reduce 0%
19/12/12 09:12:44 INFO mapreduce.Job: map 2% reduce 0%
19/12/12 09:14:08 INFO mapreduce.Job: map 3% reduce 0%
19/12/12 09:15:26 INFO mapreduce.Job: map 4% reduce 0%
19/12/12 09:16:48 INFO mapreduce.Job: map 5% reduce 0%
19/12/12 09:17:50 INFO mapreduce.Job: map 6% reduce 0%
19/12/12 09:18:06 INFO mapreduce.Job: map 7% reduce 0%
19/12/12 09:19:09 INFO mapreduce.Job: map 8% reduce 0%
19/12/12 09:20:32 INFO mapreduce.Job: map 9% reduce 0%
19/12/12 09:21:18 INFO mapreduce.Job: map 10% reduce 0%
19/12/12 09:21:27 INFO mapreduce.Job: map 11% reduce 0%
19/12/12 09:21:57 INFO mapreduce.Job: map 12% reduce 0%
19/12/12 09:23:23 INFO mapreduce.Job: map 13% reduce 0%
19/12/12 09:24:46 INFO mapreduce.Job: map 14% reduce 0%
19/12/12 09:25:19 INFO mapreduce.Job: map 15% reduce 0%
19/12/12 09:25:48 INFO mapreduce.Job: map 16% reduce 0%
19/12/12 09:27:12 INFO mapreduce.Job: map 17% reduce 0%
19/12/12 09:28:32 INFO mapreduce.Job: map 18% reduce 0%
19/12/12 09:29:56 INFO mapreduce.Job: map 19% reduce 0%
19/12/12 09:31:15 INFO mapreduce.Job: map 20% reduce 0%
19/12/12 09:32:17 INFO mapreduce.Job: map 21% reduce 0%
19/12/12 09:32:38 INFO mapreduce.Job: map 22% reduce 0%
19/12/12 09:32:49 INFO mapreduce.Job: map 23% reduce 0%
19/12/12 09:33:16 INFO mapreduce.Job: map 24% reduce 0%
19/12/12 09:33:48 INFO mapreduce.Job: map 25% reduce 0%
19/12/12 09:35:08 INFO mapreduce.Job: map 26% reduce 0%
19/12/12 09:36:35 INFO mapreduce.Job: map 27% reduce 0%
19/12/12 09:38:01 INFO mapreduce.Job: map 28% reduce 0%
19/12/12 09:39:26 INFO mapreduce.Job: map 29% reduce 0%
```

```
parmodnagare1993@hadoop-info7250-m: ~ - Google Chrome
ssh.cloud.google.com/projects/info7250/zones/us-central1-c/instances/hadoop-info7250-m?authuser=1&hl=en_US&projectNumber=883502186074
Killed reduce tasks=1
Launched map tasks=52
Launched reduce tasks=6
Data-local map tasks=52
Total time spent by all maps in occupied slots (ms)=117841276
Total time spent by all reduces in occupied slots (ms)=3993848
Total time spent by all map tasks (ms)=29460319
Total time spent by all reduce tasks (ms)=998462
Total vcore-milliseconds taken by all map tasks=29460319
Total vcore-milliseconds taken by all reduce tasks=998462
Total megabyte-milliseconds taken by all map tasks=60334733312
Total megabyte-milliseconds taken by all reduce tasks=2044850176
Map-Reduce Framework
  Map input records=12520723
  Map output records=59300440
  Map output bytes=1249507931
  Map output materialized bytes=1368110311
  Input split bytes=7250
  Combine input records=0
  Combine output records=0
  Reduce input groups=6559
  Reduce shuffle bytes=1368110311
  Reduce input records=59300440
  Reduce output records=6559
  Spilled Records=118600880
  Shuffled Maps =250
  Failed Shuffles=0
  Merged Map outputs=250
  GC time elapsed (ms)=79091
  CPU time spent (ms)=22372880
  Physical memory (bytes) snapshot=35660984320
  Virtual memory (bytes) snapshot=192930430976
  Total committed heap usage (bytes)=26222264320
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=6697196130
File Output Format Counters
  Bytes Written=154437
real    102m7.710s
user    0m22.320s
sys     0m4.880s
parmodnagare1993@hadoop-info7250-m:~$
parmodnagare1993@hadoop-info7250-m:~$
```

Bloom Filter Performance:

Bloom Filter files:

[NegativeBloomFilter.txt](#) Size: 4.7 KB

[PositiveBloomFilter.txt](#) Size: 4.7 KB

Word list files:

[negative words.txt](#) Size: 43.7 KB

[positive words.txt](#) Size: 16.8 KB

MR Job Performance

1. Without bloom filter: **41 min 9 sec**
2. With Bloom Filer: **33 min 13 sec**
3. With optimization: **20 min 37 sec**

Screenshots:

Without Bloom Filter

```
(base) pranodnagare@pranodnagare:~$
(base) pranodnagare@pranodnagare:~$
(base) pranodnagare@pranodnagare:~$ hadoop fs -rm -r /test2
Deleted /test2
(base) pranodnagare@pranodnagare:~$ time hadoop jar INFO7250_Project/FinalProject/INFO7250_Project.jar Project.Driver /AWS-Review-Data/Amazon_reviews_us_Digital_Ebook_Purchase_v1_00.tsv /test2 /book/stop-words.txt
2019-12-11 03:52:44,646 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-12-11 03:52:46,259 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2019-12-11 03:52:46,291 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/pranodnagare/.staging/job_1576050510038_0002
2019-12-11 03:52:46,555 INFO sasl.SaslDataTransferClient: SASL encryption trust check: LocalHostTrusted = false, remoteHostTrusted = false
2019-12-11 03:52:46,956 INFO input.FileInputFormat: Total input files to process : 1
2019-12-11 03:52:47,062 INFO sasl.SaslDataTransferClient: SASL encryption trust check: LocalHostTrusted = false, remoteHostTrusted = false
2019-12-11 03:52:47,091 INFO sasl.SaslDataTransferClient: SASL encryption trust check: LocalHostTrusted = false, remoteHostTrusted = false
2019-12-11 03:52:47,507 INFO mapreduce.JobSubmitter: number of splits:50
2019-12-11 03:52:48,076 INFO sasl.SaslDataTransferClient: SASL encryption trust check: LocalHostTrusted = false, remoteHostTrusted = false
2019-12-11 03:52:48,505 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1576050510038_0002
2019-12-11 03:52:48,505 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-12-11 03:52:49,182 INFO conf.Configuration: resource-types.xml not found
2019-12-11 03:52:49,183 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-12-11 03:52:49,294 INFO impl.YarnClientImpl: Submitted application application_1576050510038_0002
2019-12-11 03:52:49,421 INFO mapreduce.Job: The url to track the job: http://pranodnagare:8088/proxy/application_1576050510038_0002/
2019-12-11 03:52:49,422 INFO mapreduce.Job: Running job: job_1576050510038_0002
2019-12-11 03:53:04,690 INFO mapreduce.Job: Job job_1576050510038_0002 running in uber mode : false
2019-12-11 03:53:04,692 INFO mapreduce.Job: map 0% reduce 0%
2019-12-11 03:53:08,012 INFO mapreduce.Job: map 1% reduce 0%
2019-12-11 03:53:08,102 INFO mapreduce.Job: map 2% reduce 0%
2019-12-11 03:54:03,202 INFO mapreduce.Job: map 3% reduce 0%
2019-12-11 03:54:26,360 INFO mapreduce.Job: map 4% reduce 0%
2019-12-11 03:54:50,502 INFO mapreduce.Job: map 5% reduce 0%
2019-12-11 03:55:03,568 INFO mapreduce.Job: map 6% reduce 0%
2019-12-11 03:55:21,632 INFO mapreduce.Job: map 7% reduce 0%
2019-12-11 03:55:46,708 INFO mapreduce.Job: map 8% reduce 0%
2019-12-11 03:56:03,779 INFO mapreduce.Job: map 9% reduce 0%
2019-12-11 03:56:09,956 INFO mapreduce.Job: map 10% reduce 0%
2019-12-11 03:56:10,959 INFO mapreduce.Job: map 11% reduce 0%
2019-12-11 03:56:16,979 INFO mapreduce.Job: map 12% reduce 0%
2019-12-11 03:56:46,067 INFO mapreduce.Job: map 13% reduce 0%
2019-12-11 03:57:08,111 INFO mapreduce.Job: map 14% reduce 0%
2019-12-11 03:57:15,157 INFO mapreduce.Job: map 15% reduce 0%
2019-12-11 03:57:34,220 INFO mapreduce.Job: map 16% reduce 0%
2019-12-11 03:57:59,280 INFO mapreduce.Job: map 17% reduce 0%
2019-12-11 03:58:14,319 INFO mapreduce.Job: map 18% reduce 0%
2019-12-11 03:58:31,367 INFO mapreduce.Job: map 19% reduce 0%
2019-12-11 03:58:56,428 INFO mapreduce.Job: map 20% reduce 0%
2019-12-11 03:59:10,461 INFO mapreduce.Job: map 21% reduce 0%
```

```

Total time spent by all reduces in occupied slots (ms)=2057388
Total time spent by all map tasks (ms)=9251526
Total time spent by all reduce tasks (ms)=2057388
Total vcore-millisecods taken by all map tasks=9251526
Total vcore-millisecods taken by all reduce tasks=2057388
Total megabyte-millisecods taken by all map tasks=9473562624
Total megabyte-millisecods taken by all reduce tasks=2106765312
Map-Reduce Framework
  Map input records=12520723
  Map output records=407547175
  Map output bytes=4326086575
  Map output materialized bytes=5141181406
  Input split bytes=7650
  Combine input records=0
  Combine output records=0
  Reduce input groups=2468937
  Reduce shuffle bytes=5141181406
  Reduce input records=407547175
  Reduce output records=2468937
  Spilled Records=1316874350
  Shuffled Maps =50
  Failed Shuffles=0
  Merged Map outputs=50
  GC time elapsed (ms)=50803
  CPU time spent (ms)=6791920
  Physical memory (bytes) snapshot=26216099840
  Virtual memory (bytes) snapshot=13576232960
  Total committed heap usage (bytes)=22725269360
  Peak Map Physical memory (bytes)=519913472
  Peak Map Virtual memory (bytes)=2676498432
  Peak Reduce Physical memory (bytes)=957042688
  Peak Reduce Virtual memory (bytes)=2676445184
Shuffle Errors
  BAD_ID=0
  DELETED=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=6697186130
File Output Format Counters
  Bytes Written=32863155

```

```

real    41m9.809s
user    0m24.454s
sys     0m3.406s
(base) pranodnagare@pranodnagare:~$

```

With Bloom Filter

```

(base) pranodnagare@pranodnagare:~$ hadoop fs -rm -r /test1
Deleted /test1
(base) pranodnagare@pranodnagare:~$ time hadoop jar INFO7250_Project/FinalProject/INFO7250_Project_BF.jar Project.Driver /AWS-Review-Data/amazon_reviews_us_Digital_Ebook_Purchase_v1_00.tsv /test1 /book/JavaBloomFilter.txt
2019-12-11 03:09:11,628 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-12-11 03:09:12,093 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2019-12-11 03:09:13,114 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/pranodnagare/staging/job_1576050510038_0001
2019-12-11 03:09:13,782 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 03:09:15,247 INFO input.FileInputFormat: Total input files to process : 1
2019-12-11 03:09:15,400 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 03:09:15,447 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 03:09:15,465 INFO mapreduce.JobSubmitter: number of splits:50
2019-12-11 03:09:15,974 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 03:09:16,004 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1576050510038_0001
2019-12-11 03:09:16,004 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-12-11 03:09:16,523 INFO conf.Configuration: resource-types.xml not found
2019-12-11 03:09:16,524 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2019-12-11 03:09:17,847 INFO impl.YarnClientImpl: Submitted application application_1576050510038_0001
2019-12-11 03:09:17,982 INFO mapreduce.Job: The url to track the job: http://pranodnagare:8088/proxy/application_1576050510038_0001/
2019-12-11 03:09:17,983 INFO mapreduce.Job: Running job: job_1576050510038_0001
2019-12-11 03:09:39,477 INFO mapreduce.Job: Job job_1576050510038_0001 running in uber mode : false
2019-12-11 03:09:39,479 INFO mapreduce.Job:  map 0% reduce 0%
2019-12-11 03:10:18,126 INFO mapreduce.Job:  map 1% reduce 0%
2019-12-11 03:10:24,169 INFO mapreduce.Job:  map 2% reduce 0%
2019-12-11 03:10:30,209 INFO mapreduce.Job:  map 3% reduce 0%
2019-12-11 03:10:36,308 INFO mapreduce.Job:  map 4% reduce 0%
2019-12-11 03:11:01,434 INFO mapreduce.Job:  map 5% reduce 0%
2019-12-11 03:11:12,507 INFO mapreduce.Job:  map 6% reduce 0%
2019-12-11 03:11:19,556 INFO mapreduce.Job:  map 7% reduce 0%
2019-12-11 03:11:37,649 INFO mapreduce.Job:  map 8% reduce 0%
2019-12-11 03:11:49,729 INFO mapreduce.Job:  map 9% reduce 0%
2019-12-11 03:11:55,752 INFO mapreduce.Job:  map 10% reduce 0%
2019-12-11 03:12:01,830 INFO mapreduce.Job:  map 11% reduce 0%
2019-12-11 03:12:06,855 INFO mapreduce.Job:  map 12% reduce 0%
2019-12-11 03:12:30,989 INFO mapreduce.Job:  map 13% reduce 0%

```

```

Total time spent by all reduce tasks (ms)=1666474
Total vcore-millisecods taken by all map tasks=6829749
Total vcore-millisecods taken by all reduce tasks=1666474
Total megabyte-millisecods taken by all map tasks=6993662976
Total megabyte-millisecods taken by all reduce tasks=1706469376
Map-Reduce Framework
  Map input records=12520723
  Map output records=407433584
  Map output bytes=4325052456
  Map output materialized bytes=5139920105
  Input split bytes=7650
  Combine input records=0
  Combine output records=0
  Reduce input groups=2468614
  Reduce shuffle bytes=5139920105
  Reduce input records=407433584
  Reduce output records=2468614
  Spilled Records=1316747462
  Shuffled Maps =50
  Failed Shuffles=0
  Merged Map outputs=50
  GC time elapsed (ms)=52332
  CPU time spent (ms)=4942800
  Physical memory (bytes) snapshot=26140684288
  Virtual memory (bytes) snapshot=135700455424
  Total committed heap usage (bytes)=22860005376
  Peak Map Physical memory (bytes)=611602432
  Peak Map Virtual memory (bytes)=2668908544
  Peak Reduce Physical memory (bytes)=962953216
  Peak Reduce Virtual memory (bytes)=2679267328
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=6697186130
File Output Format Counters
  Bytes Written=32858684

```

```

real    33m13.193s
user    0m28.185s
sys     0m4.764s
(base) pranodnagare@pranodnagare:~$
(base) pranodnagare@pranodnagare:~$
(base) pranodnagare@pranodnagare:~$

```

With Bloom Filter and Optimization:

```
(base) pramodnagare@pramodnagare:~$
(base) pramodnagare@pramodnagare:~$ hadoop fs -ls /
Found 3 items
drwxr-xr-x - pramodnagare supergroup          0 2019-12-03 00:34 /AWS-Review-Data
drwxr-xr-x - pramodnagare supergroup          0 2019-12-10 21:56 /book
drwxr-xr-x - pramodnagare supergroup          0 2019-11-22 14:50 /hive
(base) pramodnagare@pramodnagare:~$
(base) pramodnagare@pramodnagare:~$ time hadoop jar INF07250_Project/FinalProject/Optmized/INF07250_Project_BF.jar Project.Driver /AWS-Review-Data/amazon_reviews_us_Digital_Ebook_Purchase_v1_00.tsv /test1/book/JavaBloomFilter.txt
2019-12-11 16:50:50,393 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2019-12-11 16:50:51,604 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2019-12-11 16:50:51,649 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/pramodnagare/.staging/job_1576099692672_0002
2019-12-11 16:50:51,931 INFO sasL.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 16:50:52,370 INFO Input.FileInputFormat: Total input files to process : 1
2019-12-11 16:50:52,467 INFO sasL.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 16:50:52,922 INFO sasL.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 16:50:52,931 INFO mapreduce.JobSubmitter: number of splits:50
2019-12-11 16:50:53,160 INFO sasL.SaslDataTransferClient: SASL encryption trust check: localhostTrusted = false, remoteHostTrusted = false
2019-12-11 16:50:53,190 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1576099692672_0002
2019-12-11 16:50:53,190 INFO mapreduce.JobSubmitter: Executing with tokens: []
2019-12-11 16:50:53,372 INFO conf.Configuration: resource-types.xml not found
2019-12-11 16:50:53,372 INFO resource.ResourceUtils: Unable to find 'resource-types.xml',
2019-12-11 16:50:53,696 INFO Impl.VarnClientImpl: Submitted application application_1576099692672_0002
2019-12-11 16:50:53,772 INFO mapreduce.Job: The url to track the job: http://pramodnagare:8088/proxy/application_1576099692672_0002/
2019-12-11 16:50:53,773 INFO mapreduce.Job: Running job: job_1576099692672_0002
2019-12-11 16:51:06,086 INFO mapreduce.Job: Job job_1576099692672_0002 running in uber mode : false
2019-12-11 16:51:06,087 INFO mapreduce.Job: map 0% reduce 0%
2019-12-11 16:51:35,424 INFO mapreduce.Job: map 1% reduce 0%
2019-12-11 16:51:52,692 INFO mapreduce.Job: map 2% reduce 0%
2019-12-11 16:51:59,740 INFO mapreduce.Job: map 3% reduce 0%
2019-12-11 16:52:11,835 INFO mapreduce.Job: map 4% reduce 0%
2019-12-11 16:52:36,005 INFO mapreduce.Job: map 5% reduce 0%
2019-12-11 16:52:43,045 INFO mapreduce.Job: map 6% reduce 0%
2019-12-11 16:52:55,131 INFO mapreduce.Job: map 7% reduce 0%
2019-12-11 16:53:18,283 INFO mapreduce.Job: map 8% reduce 0%
2019-12-11 16:53:23,361 INFO mapreduce.Job: map 9% reduce 0%
2019-12-11 16:53:26,372 INFO mapreduce.Job: map 10% reduce 0%
2019-12-11 16:53:28,380 INFO mapreduce.Job: map 11% reduce 0%
2019-12-11 16:53:30,387 INFO mapreduce.Job: map 12% reduce 0%
2019-12-11 16:53:55,500 INFO mapreduce.Job: map 13% reduce 0%
2019-12-11 16:54:03,536 INFO mapreduce.Job: map 14% reduce 0%
2019-12-11 16:54:11,575 INFO mapreduce.Job: map 15% reduce 0%
2019-12-11 16:54:23,633 INFO mapreduce.Job: map 16% reduce 0%
2019-12-11 16:54:45,729 INFO mapreduce.Job: map 17% reduce 0%
2019-12-11 16:54:55,770 INFO mapreduce.Job: map 18% reduce 0%
2019-12-11 16:55:08,820 INFO mapreduce.Job: map 19% reduce 0%
2019-12-11 16:55:26,903 INFO mapreduce.Job: map 20% reduce 0%
```

```
Total time spent by all reduces in occupied slots (ms)=938572
Total time spent by all map tasks (ms)=6109892
Total time spent by all reduce tasks (ms)=938572
Total vcore-milliseconds taken by all map tasks=6109892
Total vcore-milliseconds taken by all reduce tasks=938572
Total megabyte-milliseconds taken by all map tasks=6256529408
Total megabyte-milliseconds taken by all reduce tasks=961097728

Map-Reduce Framework
  Map input records=12520723
  Map output records=407433584
  Map output bytes=4325052456
  Map output materialized bytes=142078448
  Input split bytes=7650
  Combine input records=421693592
  Combine output records=23360977
  Reduce input groups=2468614
  Reduce shuffle bytes=142078448
  Reduce input records=9100969
  Reduce output records=2468614
  Spilled Records=32461946
  Shuffled Maps =50
  Failed Shuffles=0
  Merged Map outputs=50
  GC time elapsed (ms)=47124
  CPU time spent (ms)=3777470
  Physical memory (bytes) snapshot=2611266816
  Virtual memory (bytes) snapshot=135858139136
  Total committed heap usage (bytes)=22232956928
  Peak Map Physical memory (bytes)=523751424
  Peak Map Virtual memory (bytes)=2668056576
  Peak Reduce Physical memory (bytes)=519512064
  Peak Reduce Virtual memory (bytes)=2674622464

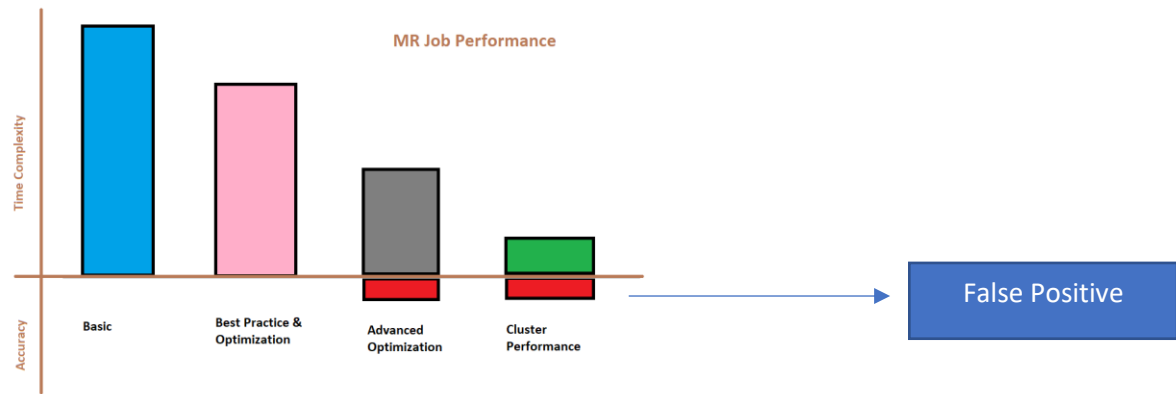
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0

File Input Format Counters
  Bytes Read=6697186130
File Output Format Counters
  Bytes Written=32858684

real    20m37.524s
user    0m21.181s
sys     0m2.800s
(base) pramodnagare@pramodnagare:~$
```

Task 7: Conclusion

- Parallel processes for file download are limited by internet speed
- Parallel processes for file extraction are limited by disk I/O
- MR Job Performance observation



Best Practices:

1. Use compact / optimize data type in MR jobs
2. Use of combiners
3. Use file compressions format (ex. LZO)
4. Distributed caching
5. Multiple mapper and reducer

Advanced Optimization:

1. Bloom filter