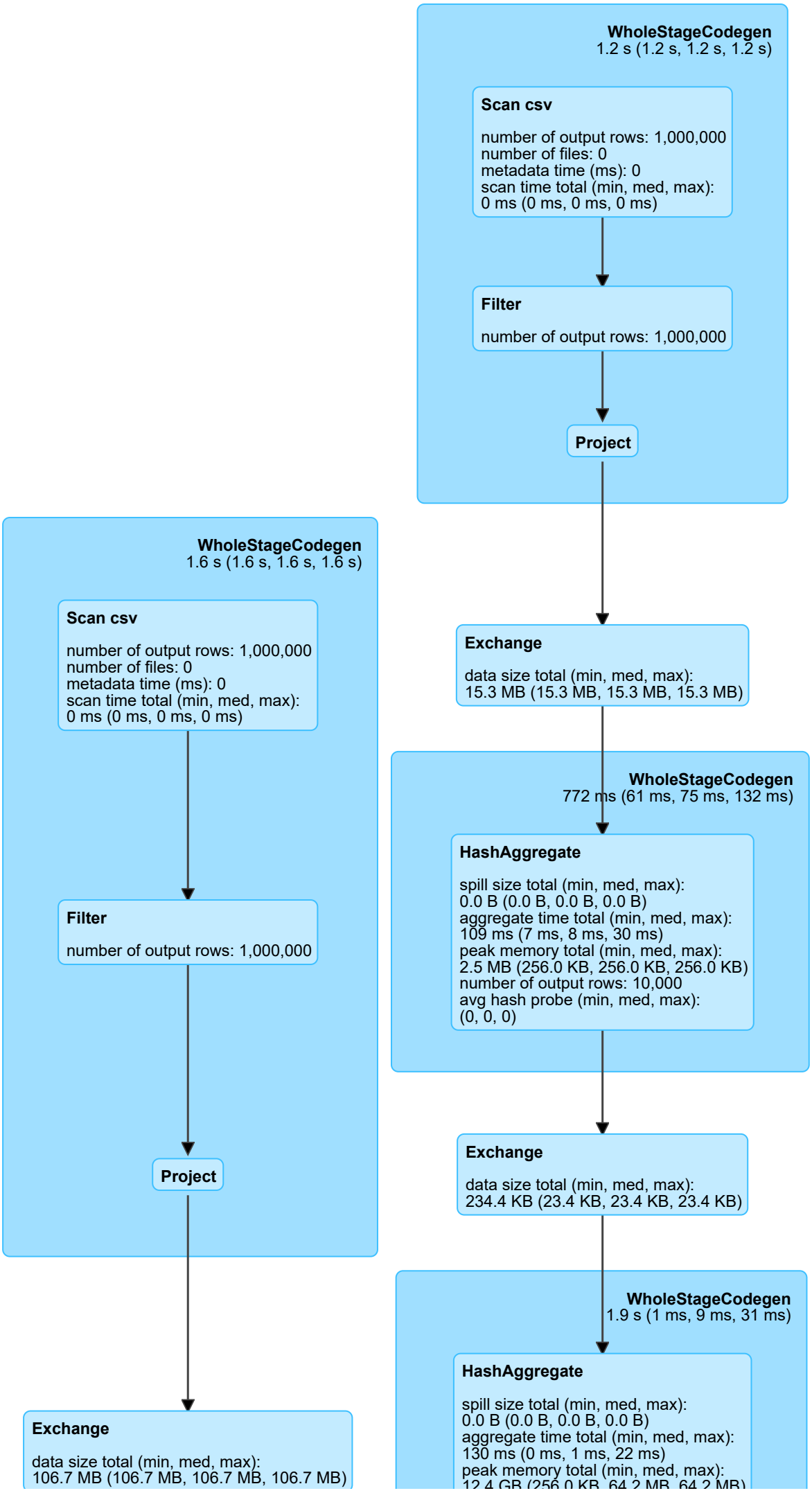
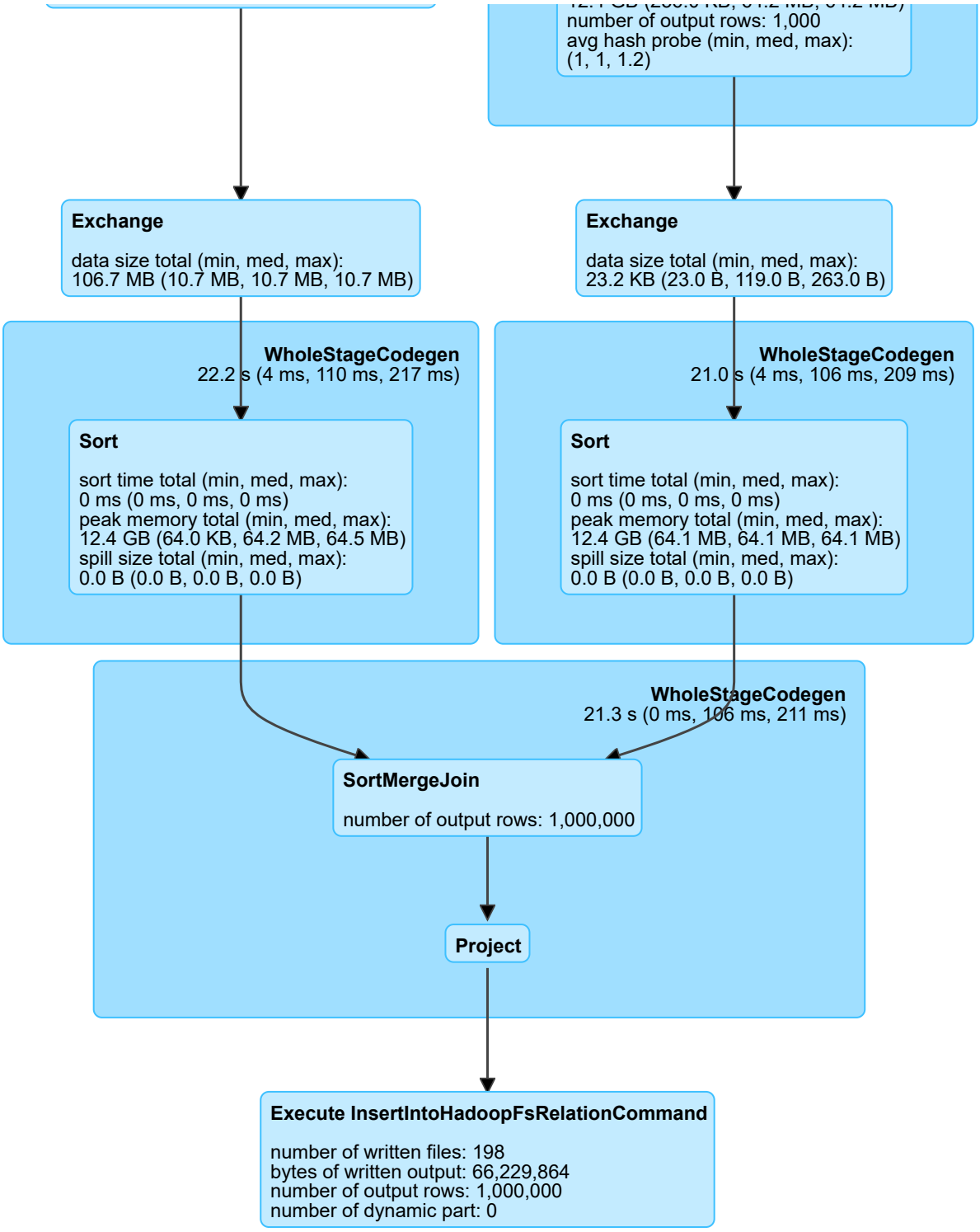
Details for Query 2

Submitted Time: 2023/11/18 20:21:31
Duration: 51 s
Succeeded Jobs: 3 (/jobs/job?id=3)


WholeStageCodegen
1.2 s (1.2 s, 1.2 s, 1.2 s)
Scan csv
number of output rows: 1,000,000
number of files: 0
metadata time (ms): 0
scan time total (min, med, max):
0 ms (0 ms, 0 ms, 0 ms)
Filter
number of output rows: 1,000,000
Project
WholeStageCodegen
1.6 s (1.6 s, 1.6 s, 1.6 s)
Scan csv
number of output rows: 1,000,000
number of files: 0
metadata time (ms): 0
scan time total (min, med, max):
0 ms (0 ms, 0 ms, 0 ms)
Filter
number of output rows: 1,000,000
Project
Exchange
data size total (min, med, max):
106.7 MB (106.7 MB, 106.7 MB, 106.7 MB)
Exchange
data size total (min, med, max):
15.3 MB (15.3 MB, 15.3 MB, 15.3 MB)
WholeStageCodegen
772 ms (61 ms, 75 ms, 132 ms)
HashAggregate
spill size total (min, med, max):
0.0 B (0.0 B, 0.0 B, 0.0 B)
aggregate time total (min, med, max):
109 ms (7 ms, 8 ms, 30 ms)
peak memory total (min, med, max):
2.5 MB (256.0 KB, 256.0 KB, 256.0 KB)
number of output rows: 10,000
avg hash probe (min, med, max):
(0, 0, 0)
Exchange
data size total (min, med, max):
234.4 KB (23.4 KB, 23.4 KB, 23.4 KB)
WholeStageCodegen
1.9 s (1 ms, 9 ms, 31 ms)
HashAggregate
spill size total (min, med, max):
0.0 B (0.0 B, 0.0 B, 0.0 B)
aggregate time total (min, med, max):
130 ms (0 ms, 1 ms, 22 ms)
peak memory total (min, med, max):


number of output rows: 1,000
avg hash probe (min, med, max):
(1, 1, 1.2)
Exchange
data size total (min, med, max):
106.7 MB (10.7 MB, 10.7 MB, 10.7 MB)
Exchange
data size total (min, med, max):
23.2 KB (23.0 B, 119.0 B, 263.0 B)
WholeStageCodegen
22.2 s (4 ms, 110 ms, 217 ms)
WholeStageCodegen
21.0 s (4 ms, 106 ms, 209 ms)
Sort
sort time total (min, med, max):
0 ms (0 ms, 0 ms, 0 ms)
peak memory total (min, med, max):
12.4 GB (64.0 KB, 64.2 MB, 64.5 MB)
spill size total (min, med, max):
0.0 B (0.0 B, 0.0 B, 0.0 B)
Sort
sort time total (min, med, max):
0 ms (0 ms, 0 ms, 0 ms)
peak memory total (min, med, max):
12.4 GB (64.1 MB, 64.1 MB, 64.1 MB)
spill size total (min, med, max):
0.0 B (0.0 B, 0.0 B, 0.0 B)
WholeStageCodegen
21.3 s (0 ms, 106 ms, 211 ms)
SortMergeJoin
number of output rows: 1,000,000
Project
Execute InsertIntoHadoopFsRelationCommand
number of written files: 198
bytes of written output: 66,229,864
number of output rows: 1,000,000
number of dynamic part: 0

▼ Details

```
== Parsed Logical Plan ==
InsertIntoHadoopFsRelationCommand file:/c:/temp/simple.csv, false, CSV, Map(header -> true, path ->
c:/temp/simple.csv), Overwrite, [id#10, parentid#11, data#12, date#13, url#14, count#49L]
+- AnalysisBarrier
      +- Project [id#10, parentid#11, data#12, date#13, url#14, count#49L]
         +- Join Inner, (parentId#11 = pid#52)
            :- Repartition 10, true
            :  +- Relation[id#10,parentid#11,data#12,date#13,url#14] csv
            +- Project [parentid#11 AS pid#52, count#49L]
               +- Aggregate [parentid#11], [parentid#11, count(1) AS count#49L]
                  +- Repartition 10, true
                     +- Relation[id#10,parentid#11,data#12,date#13,url#14] csv

== Analyzed Logical Plan ==
InsertIntoHadoopFsRelationCommand file:/c:/temp/simple.csv, false, CSV, Map(header -> true, path ->
c:/temp/simple.csv), Overwrite, [id#10, parentid#11, data#12, date#13, url#14, count#49L]
+- Project [id#10, parentid#11, data#12, date#13, url#14, count#49L]
   +- Join Inner, (parentId#11 = pid#52)
      :- Repartition 10, true
      :  +- Relation[id#10,parentid#11,data#12,date#13,url#14] csv
      +- Project [parentid#11 AS pid#52, count#49L]
         +- Aggregate [parentid#11], [parentid#11, count(1) AS count#49L]
            +- Repartition 10, true
               +- Relation[id#10,parentid#11,data#12,date#13,url#14] csv

== Optimized Logical Plan ==
InsertIntoHadoopFsRelationCommand file:/c:/temp/simple.csv, false, CSV, Map(header -> true, path ->
c:/temp/simple.csv), Overwrite, [id#10, parentid#11, data#12, date#13, url#14, count#49L]
+- Project [id#10, parentid#11, data#12, date#13, url#14, count#49L]
   +- Join Inner, (parentId#11 = pid#52)
      :- Repartition 10, true
      :  +- Filter isnotnull(parentId#11)
      :     +- Relation[id#10,parentid#11,data#12,date#13,url#14] csv
      +- Aggregate [parentid#11], [parentid#11 AS pid#52, count(1) AS count#49L]
         +- Repartition 10, true
            +- Project [parentid#11]
               +- Filter isnotnull(parentid#11)
                  +- Relation[id#10,parentid#11,data#12,date#13,url#14] csv

== Physical Plan ==
Execute InsertIntoHadoopFsRelationCommand InsertIntoHadoopFsRelationCommand file:/c:/temp/simple.csv,
false, CSV, Map(header -> true, path -> c:/temp/simple.csv), Overwrite, [id#10, parentid#11, data#12,
date#13, url#14, count#49L]
+- *(7) Project [id#10, parentid#11, data#12, date#13, url#14, count#49L]
   +- *(7) SortMergeJoin [parentId#11], [pid#52], Inner
      :- *(2) Sort [parentId#11 ASC NULLS FIRST], false, 0
      :  +- Exchange hashpartitioning(parentId#11, 200)
      :     +- Exchange RoundRobinPartitioning(10)
      :        +- *(1) Project [id#10, parentid#11, data#12, date#13, url#14]
      :           +- *(1) Filter isnotnull(parentId#11)
      :              +- *(1) FileScan csv [id#10,parentid#11,data#12,date#13,url#14] Batched: false,
Format: CSV, Location: InMemoryFileIndex[file:/C:/temp/data.csv], PartitionFilters: [], PushedFilters:
[IsNotNull(parentid)], ReadSchema: struct<id:int,parentid:int,data:string,date:string,url:string>
      +- *(6) Sort [pid#52 ASC NULLS FIRST], false, 0
         +- Exchange hashpartitioning(pid#52, 200)
            +- *(5) HashAggregate(keys=[parentid#11], functions=[count(1)], output=[pid#52, count#49L])
               +- Exchange hashpartitioning(parentid#11, 200)
                  +- *(4) HashAggregate(keys=[parentid#11], functions=[partial_count(1)], output=
[parentid#11, count#89L])
                     +- Exchange RoundRobinPartitioning(10)
```

```
                        +- *(3) Project [parentid#11]
                           +- *(3) Filter isnotnull(parentid#11)
                              +- *(3) FileScan csv [parentid#11] Batched: false, Format: CSV, Location:
InMemoryFileIndex[file:/C:/temp/data.csv], PartitionFilters: [], PushedFilters: [IsNotNull(parentid)],
ReadSchema: struct<parentid:int>
```