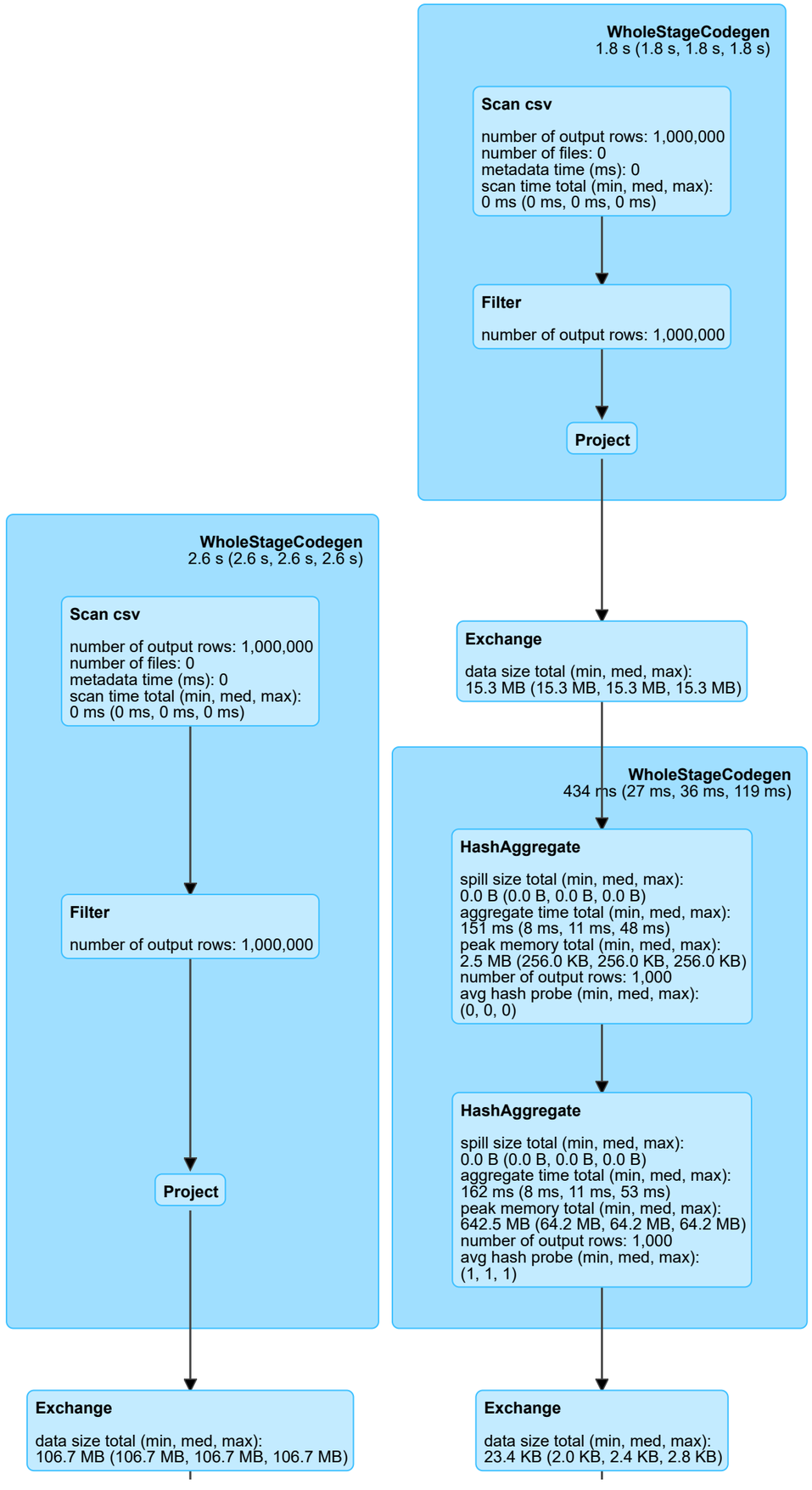
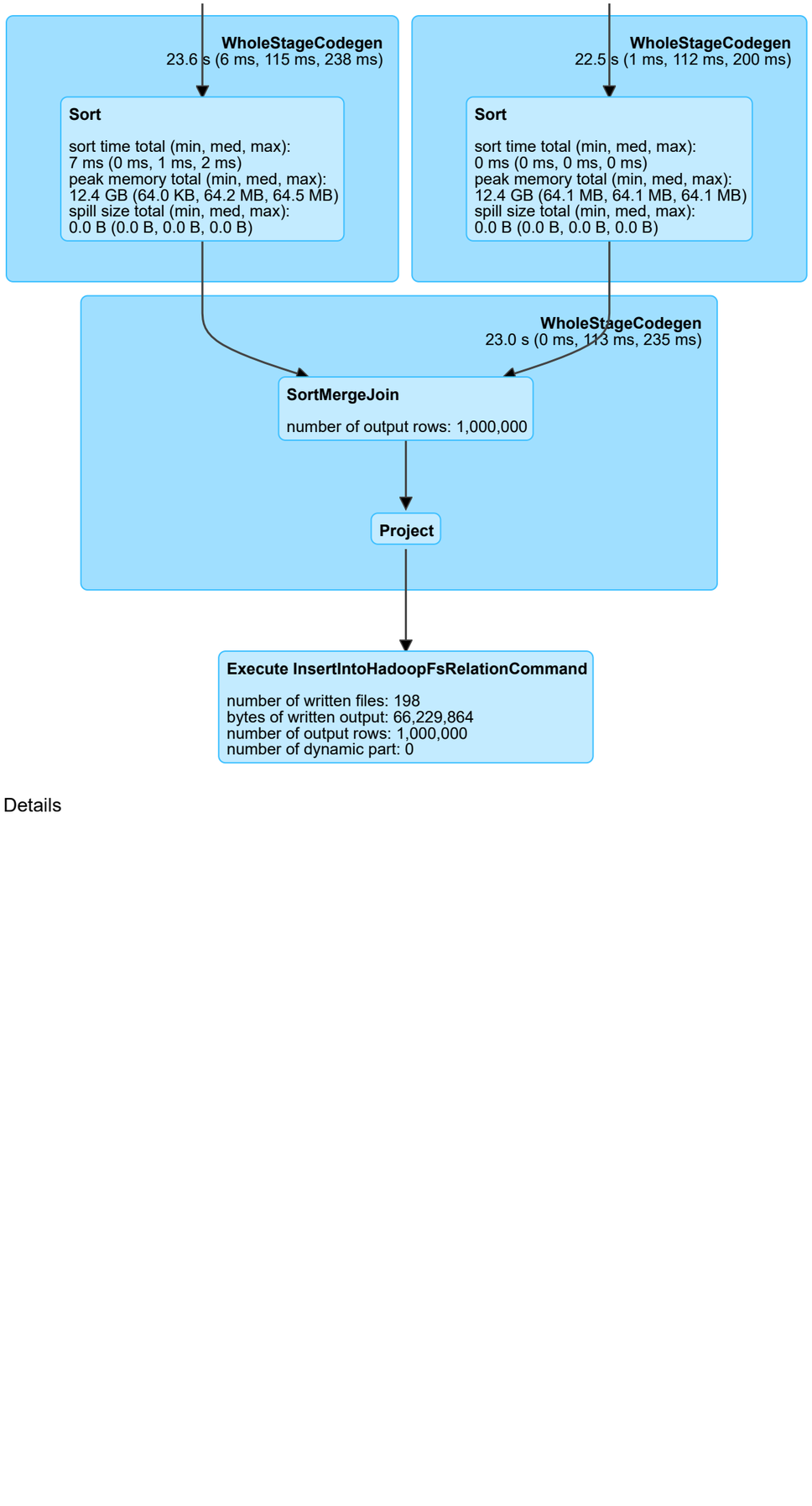
Details for Query 2

Submitted Time: 2023/11/19 11:42:13
Duration: 36 s
Succeeded Jobs: 3 (/jobs/job?id=3)


WholeStageCodegen
1.8 s (1.8 s, 1.8 s, 1.8 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
2.6 s (2.6 s, 2.6 s, 2.6 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
434 ms (27 ms, 36 ms, 119 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):
151 ms (8 ms, 11 ms, 48 ms)
peak memory total (min, med, max):
2.5 MB (256.0 KB, 256.0 KB, 256.0 KB)
number of output rows: 1,000
avg hash probe (min, med, max):
(0, 0, 0)
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):
162 ms (8 ms, 11 ms, 53 ms)
peak memory total (min, med, max):
642.5 MB (64.2 MB, 64.2 MB, 64.2 MB)
number of output rows: 1,000
avg hash probe (min, med, max):
(1, 1, 1)
Exchange
data size total (min, med, max):
23.4 KB (2.0 KB, 2.4 KB, 2.8 KB)


WholeStageCodegen
23.6 s (6 ms, 115 ms, 238 ms)
Sort
sort time total (min, med, max):
7 ms (0 ms, 1 ms, 2 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)
WholeStageCodegen
22.5 s (1 ms, 112 ms, 200 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 (64.1 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
23.0 s (0 ms, 113 ms, 235 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)
            :- RepartitionByExpression [parentid#11], 10
            :  +- 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]
                  +- RepartitionByExpression [parentid#11], 10
                     +- 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)
      :- RepartitionByExpression [parentid#11], 10
      :  +- 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]
            +- RepartitionByExpression [parentid#11], 10
               +- 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)
      :- RepartitionByExpression [parentid#11], 10
      :  +- 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]
         +- RepartitionByExpression [parentid#11], 10
            +- 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]
+- *(6) Project [id#10, parentid#11, data#12, date#13, url#14, count#49L]
   +- *(6) SortMergeJoin [parentId#11], [pid#52], Inner
      :- *(2) Sort [parentId#11 ASC NULLS FIRST], false, 0
      :  +- Exchange hashpartitioning(parentId#11, 200)
      :     +- *(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>
      +- *(5) Sort [pid#52 ASC NULLS FIRST], false, 0
         +- Exchange hashpartitioning(pid#52, 200)
            +- *(4) HashAggregate(keys=[parentid#11], functions=[count(1)], output=[pid#52, count#49L])
               +- *(4) HashAggregate(keys=[parentid#11], functions=[partial_count(1)], output=[parentid#11, count#89L])
                  +- Exchange hashpartitioning(parentid#11, 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>
```