Whats cooking in 0.11?

Ashutosh Chauhan
hashutosh@apache.org
Future : Have you seen it?

Interviewer : Whats coming next in Linux ?

Linus : Whatever they are going to do.
Future: Have you seen it?

Interviewer: What’s coming next in Linux?

Linus: Whatever they are going to do.
They = You
Optimizations

• Numerous different possibilities:
  - Better Query planning
  - Better infrastructure runtime
  - Better application runtime
Better query planning

• Most optimal plan
  - Minimum needed I/O
    - Smallest number of MR jobs
    - Each MR job reading and writing least possible I/O
  - Minimum needed CPU
    - Smallest amount of CPU cycles
Smallest number of MR jobs

Select col5, avg(col6)

From fact_table
  join dim1 on (fact_table.col1 = dim1.col1)
  join dim2 on (fact_table.col2 = dim2.col1)
  join dim3 on (fact_table.col3 = dim3.col1)
  join dim4 on (fact_table.col4 = dim4.col1)

  group by col5
  order by col5
  limit 100;
Select col5, avg(col6)
From fact_table
    join dim1 on (fact_table.col1 = dim1.col1)                  1 MR job
    join dim2 on (fact_table.col2 = dim2.col1)                  1 MR job
    join dim3 on (fact_table.col3 = dim3.col1)                  1 MR job
    join dim4 on (fact_table.col4 = dim4.col1)                  1 MR job
group by col5                                                                 1 MR job
order by col5                                                                1 MR job
limit 100;

On 0.10, total 6 MR jobs
HIVE-3784

Select col5, avg(col6)
From fact_table
  join dim1 on (fact_table.col1 = dim1.col1)
  join dim2 on (fact_table.col2 = dim2.col1)
  join dim3 on (fact_table.col3 = dim3.col1)
  join dim4 on (fact_table.col4 = dim4.col1)
  1 MR job
  group by col5
  order by col5
  1 MR job
  limit 100;
  1 MR job

Note
  - no map-join hints
  - Joins are on different keys
HIVE-2340

Select col5, avg(col6)
From fact_table
    join dim1 on (fact_table.col1 = dim1.col1)
    join dim2 on (fact_table.col2 = dim2.col1)
    join dim3 on (fact_table.col3 = dim3.col1)
    join dim4 on (fact_table.col4 = dim4.col1)
group by col5
order by col5
limit 100;

Located in 

total 2 MR jobs

Note
- Order by and Group by is on same column.
Select col5, avg(col6)
From fact_table
    join dim1 on (fact_table.col1 = dim1.col1)
    join dim2 on (fact_table.col2 = dim2.col1)
    join dim3 on (fact_table.col3 = dim3.col1)
    join dim4 on (fact_table.col4 = dim4.col1)
group by col5
order by col5
limit 100;

• Note
  - Joins are in map-phase. Grouping and Ordering in reduce-phase.
Optimizer should be smarter

- Shouldn’t ask user to provide hints to run query faster.
  - Hive-3784: No need to provide hint if join can be converted into non-bucketed map-join
  - HIVE-3403: No need to provide hint if join can be converted into sort-merge bucketed join.

- Only time you will need to provide hints is
  - if your tables are bucketed and
  - you are joining on bucketed columns and
  - smaller table is not small enough to fit in memory
Order by improvements:

Top-k optimization : HIVE-3562
select * from bigTable order by col1 limit 10;

Use multiple reducers for order-by : HIVE-3972

Use sampling for total order partitioning: HIVE-3841
Even more optimizations

HIVE-948: Shorten the operator pipeline by deduplicating select and filter operators

HIVE-2340: Merge MR jobs if you are doing clustering and grouping on same key

HIVE-2340: Merge MR jobs if you are doing clustering and ordering on same key

HIVE-3891: Auto convert SMBJ into map-join
Lets make Hive faster!

hashutosh@apache.org