

Support more subexpression elimination cases

Author: Kun Wan

1. Background and Motivation

Now Spark only supports subexpression elimination in ProjectExec [SPARK-33092](#) and AggregateExec [SPARK-14951](#). But there are still some cases where we can do common subexpression elimination in projectExec such as IF and CASE_WHEN conditional expressions.

And we can also support subexpression elimination in some other operators, such as FilterExec and JoinExec.

2. Goals

- Provide a more general subexpression elimination framework for all physical operators.
- Support for more cases of subexpression elimination in ProjectExec and AggregateExec.
- Support subexpression elimination in FilterExec and JoinExec.

3. Non-Goals

- Evaluation ordering of predicates expressions.
- Support subexpression elimination for PlanExpression

4. Design Sketch

- Get all common expressions from input expressions. Recursively visits all subexpressions regardless of whether the current expression is a conditional expression.
- For each common expression:
 - Add a new boolean variable **subExprInit_n** to indicate whether we have already evaluated the common expression, and reset it to **false** at the start of `operator.consume()`
 - Add a new wrapper `subExpr` function for common subexpression.

```
private void subExpr_n(${argList}) {  
    if (!subExprInit_n) {
```

```

    ${eval.code}
    subExprInit_n = true;
    subExprIsNull_n = ${eval.isNull};
    subExprValue_n = ${eval.value};
  }
}

```

- Replace all the common subexpression with the wrapper function `subExpr_n(argList)`.

5. New support subexpression elimination patterns

5.1. Support subexpression elimination with conditional expressions

```

SELECT case when v + 2 > 1 then 1
         when v + 1 > 2 then 2
         when v + 1 > 3 then 3 END vv
FROM values(1) as t2(v)

```

We can reuse the result of expression `v + 1`

```

SELECT a, max(if(a > 0, b + c, null)) max_bc, min(if(a > 1, b + c, null))
min_bc
FROM values(1, 1, 1) as t(a, b, c)
GROUP BY a

```

We can reuse the result of expression `b + c`

5.2. Support subexpression elimination in FilterExec

```
SELECT * FROM (  
  SELECT v * v + 1 v1 from values(1) as t2(v)  
) t  
where v1 > 5 and v1 < 10
```

We can reuse the result of expression $v * v + 1$

5.3. Support subexpression elimination in JoinExec

```
SELECT *  
FROM values(1, 1) as t1(a, b)  
join values(1, 2) as t2(x, y)  
ON b * y between 2 and 3
```

We can reuse the result of expression $b * y$

5.4. Support subexpression elimination in ExpandExec

```
SELECT a, count(b),  
       count(distinct case when b > 1 then b + c else null end) as count_bc_1,  
       count(distinct case when b < 0 then b + c else null end) as count_bc_2  
FROM values(1, 1, 1) as t(a, b, c)  
GROUP BY a
```

We can reuse the result of expression $b + c$