CS422 Project1 Task 3

Peilin Kang

1. Experiment result

1.1 Different queries

Query1. select and project

SELECT L_ORDERKEY, L_SHIPDATE

FROM lineitem L

WHERE L_ORDERKEY > 666666

Query2. join and project

SELECT L_PARTKEY, O_CUSTKEY

FROM lineitem L, order O

WHERE (L_ORDERKEY=O_ORDERKEY)

Query3. projectAggregate

SELECT AVG(L_DISCOUNT)

FROM lineitem

1.2 execute time on different layout and execution model

Data Layout		NSM	PAX	DSM	DSM	DSM
Execution Model		Volcano	Volcano	Column-early	Column-late	Vector
Execution time	Qurey1	8.565s	7.889s	8.001s	7.746s	8.257s
	Query2	5.666s	5.051s	6.119s	4.781s	5.663s
	Query3	5.377s	4.309s	4.164s	4.112s	5.355s

Note: For PAX, tuples per page is 10000; for vector at a time, the vector size is 10000

1.3 best result

The best result is **DSM with column at a time and late materialization**. Compared to volcano, column at a time loads the whole table at once, doesn't have next() call overhead. Compared to early materialization, late materialization doesn't have materialization overhead. It seems reasonable that DSM with column at a time late materialization has shortest execution time.

2. Analysis

2.1 NSM/volcano vs DSM/columnar

Volcano execution model has many function calls for each tuple and has many context switches which is a huge cost. While the column at a time execution model processes its input all at once and emits its output all at once. But column at a time needs to materialize its output as a single result, which is also costly. From the experiment result we can see the performance of NSM/volcano and DSM/columnar with early materialization doesn't have much difference. But still DSM/columnar is a little bit better.

2.2 DSM/early materialization vs DSM/late materialization

Early materialization needs to do materialization right after each operation, but late materialization only uses ids of the column, thus does not have materialization overhead. Thus, with the same DSM data layout, late materialization has much shorter execute time than early materialization.

2.3 DSM/columnar vs DSM/vector (both early materialization)

For DSM/vector execution model, each operator emits a vector of tuples instead of a single tuple which greatly reduces the number of invocations per operator. But compared to DSM/columnar, which loads all data at once, DSM/vector still perform a little bit worse.

2.4 NSM/volcano vs PAX/volcano

I suppose NSM data layout will have better performance than PAX data layout. Because in order to get a tuple from PAX storage, we need to do data materialization which is an extra cost compared to NSM data layout. So the experiment result seems a little bit weird for me that PAX has shorter execution time than NSM data layout.