Project I - Task 3

I Made Sanadhi Sutandi

March 24, 2018

Notes:

- All results that are obtained do not include load time.
- For each query, all combination of execution model and data layout produce the exact same results.
- For each query, selections and projections are performed as early as possible (heuristic-based optimization).
- All joins are performed using Hash-Join algorithm.
- For PAX data layout, the number of tuple per page is fixed into *100* (**see section 6**).
- For Vectorized execution model, the size of vector is fixed into *100000* (**see section 6**).
- The measurements were conducted on Mac OSX (10.12.6), with 1.6 GHz Intel Core i5 CPU and 8GB memory.

1 Query I - Basic Selection and Projection

```
SELECT L.L_SUPPKEY, L.L_RETURNFLAG, L.L_LINESTATUS
FROM lineitem L
WHERE L.L_SUPPKEY>3000
```

1.1 Result

Execution time of Query I over different execution model-data layout				
Data Layout	NSM	PAX	DSM	DSM
Execution Model	Tuple-at-a-time	Tuple-at-a-time	Block-at-a-time	Vector-at-a-time
Execution time	15.265 s	16.757 s	13.224 s	14.680 s

1.2 Observation

The **DSM with Block-at-a-time** engine performs best since all data are fit into memory and thus it does not has next() function calls overhead.

2 Query II - Basic Join

```
SELECT L.L_COMMENT, O.O_SHIPPRIORITY
FROM orders O, lineitem L
WHERE O.O_CUSTKEY >= 100000 AND
O.O_ORDERKEY = L.L_ORDERKEY
```

2.1 Result

Execution time of Query II over different execution model-data layout				
Data Layout	NSM	PAX	DSM	DSM
Execution Model	Tuple-at-a-time	Tuple-at-a-time	Block-at-a-time	Vector-at-a-time
Execution time	15.636 s	19.697 s	14.117 s	16.977 s

2.2 Observation

The **DSM with Block-at-a-time** engine performs best since all data are fit into memory and thus it does not has next() function calls overhead.

3 Query III - Basic Aggregation

```
SELECT AVG(L_DISCOUNT)
FROM lineitem L
WHERE L.L_QUANTITY<30
```

3.1 Result

Execution time of Query III over different execution model-data layout				
Data Layout	NSM	PAX	DSM	DSM
Execution Model	Tuple-at-a-time	Tuple-at-a-time	Block-at-a-time	Vector-at-a-time
Execution time	15.162 s	17.328 s	12.924 s	14.753 s

3.2 Observation

The **DSM with Block-at-a-time** engine performs best since all data are fit into memory and thus it does not has next() function calls overhead.

4 Query IV - Join with Aggregation

```
SELECT COUNT(O.O_CUSTKEY)
FROM orders O, lineitem L
WHERE O.O_CUSTKEY < 50000 AND
O.O_ORDERKEY = L.L_ORDERKEY
```

4.1 Result

Execution time of Query IV over different execution model-data layout				
Data Layout	NSM	PAX	DSM	DSM
Execution Model	Tuple-at-a-time	Tuple-at-a-time	Block-at-a-time	Vector-at-a-time
Execution time	15.725 s	18.185 s	15.076 s	18.560 s

4.2 Observation

The **DSM with Block-at-a-time** engine performs best since all data are fit into memory and thus it does not has next() function calls overhead.

5 Conclusions

5.1 Volcano: NSM vs PAX

For every scenario, PAX is slower than NSM because of additional overhead for materialization. Every next() call from parent requires PAX to form DBTuple from minipages. As we can see in **section 6**, both decreasing page size or increasing page size (too big) will make the execution slower since both lead to increase in search time for select operator (too many pages and too many tuples in one page). If we set page size into 1, PAX will basically become NSM but with slower performance.

5.2 NSM(Volcano) vs DSM(Block)

For join operations (Query II and Query IV), there is no significant performance between NSM with tuple-at-a-time and DSM with block-at-a-time. This is caused by relatively more complex materialization of DSM layout.

5.3 DSM: Block-at-a-time vs Vector-at-a-time

Note that in **section 6**, if we increase the vector size, the performance of vector-at-a-time execution model will approach block-at-a-time execution model, both for DSM layout. As fewer access methods from parents to children reduces function calls and context switches, thus increasing vector size decreases them and makes execution time faster. As we have already known, setting vector size equal to block size changes vector-at-a-time execution mainly to block-at-a-time.

6 Additional - Tuning Page Size and Vector size

6.1 Variation of Page size for PAX Layout

Execution time over PAX page size				
Page Size	10	100	1000	10000
Query I	18.913 s	16.757 s	53.465 s	$>$ 10 m
Query II	22.286 s	19.697 s	62.285 s	$>$ 10 m
Query III	18.490 s	17.328 s	52.526 s	$>$ 10 m
Query IV	19.593 s	18.185 s	62.088 s	$>$ 10 m

6.2 Variation of Vector Size for Vectorized Engine

Execution time over vector size				
Vector Size	100	1000	10000	100000
Query I	$>$ 10 m	99.262 s	21.517 s	14.680 s
Query II	$>$ 10 m	147.261 s	26.890 s	16.977 s
Query III	$>$ 10 m	100.507 s	22.327 s	14.753 s
Query IV	$>$ 10 m	112.477 s	24.837 s	18.560 s