Report Query execution models

Task1 & Task3 - Analysis

query 0: SELECT SUM(x) FROM rel WHERE x <> 77 AND x <> 30 AND x <> 99 AND x <> 42 AND x <> 11

type	runtime	instructions	cache refference	cache misses	L1 cache misses
volcano	6034.5	41,336,190,725	71,925,786	56,373,841	31,264,620
operator-at-a-time	9035.2	34,185,534,046	990,547,835	621,752,266	433,210,188
vector-at-a-time(vector size=64)	3981.6	27,910,441,641	97,320,434	69,540,004	45,834,243

query 1: SELECT x FROM rel WHERE x == 11

type	runtime	instructions	cache refference	cache misses	L1 cache misses
volcano	2141.9	12,210,941,001	109,623,408	71,454,526	51,562,183
operator-at-a-time	5597.8	13,375,073,598	458,647,219	231,354,269	190,486,429
vector-at-a-time(vector size=64)	1508.1	6,874,376,519	130,753,990	78,978,249	66,682,540

query 2: SELECT SUM(x) FROM rel

type	runtime	instructions	cache refference	cache misses	L1 cache misses
volcano	1544.6	10,613,098,685	62,459,172	46,780,451	29,903,215
operator-at-a-time	3909.5	10,253,048,833	320,494,396	176,553,277	138,763,381
vector-at-a-time(vector size=64)	1272,7	6,206,439,123	96,445,183	64,274,327	50,070,899

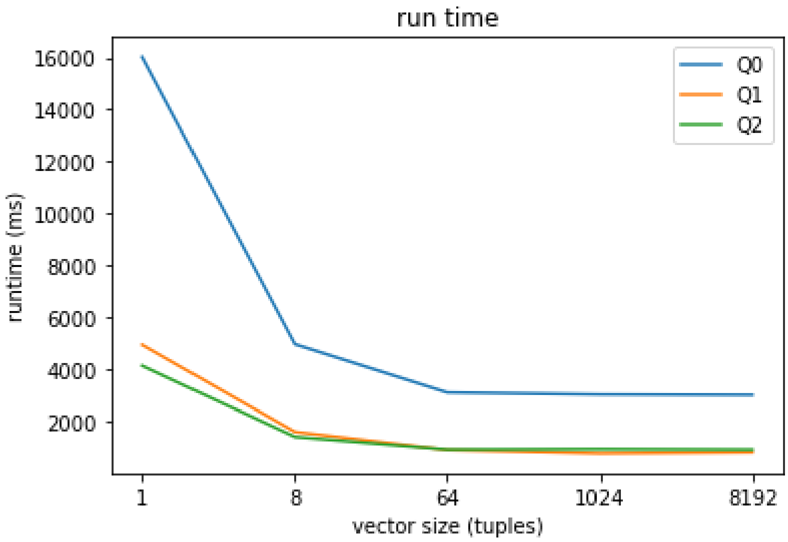
- As the experiments results show, and also because too small cache size of my laptop, materialization model always has a worse performence compared with the volcano model. It tries to consume operators on the full column, so the materializaiton model(operator-at-a-time) will repeat scanning, fetching data from memory to cache over and over. This will cause much more (almost 5 times) cache references and cache misses than the volcano model.
- For system's resources usage, all these three models have a CPU utilization over 95%, I think it just because I have one out-dated, super slow, and old laptop...
- And vectorization model (vector-at-a-time)always has the best performance. As we learned from the lecture, it use the best of both worlds, it can reduce the number of invocations per operator. It's much more efficient than other two models, vector-at-a-time allows for opeartors to use vectorized instructions to process batches of tuples.

Task3 - Influence of vector size on vector-at-a-time

query 0(Q0): SELECT SUM(x) FROM rel WHERE x <> 77 AND x <> 30 AND x <> 99 AND x <> 42 AND x <> 11

query 1(Q1): SELECT x FROM rel WHERE x == 11

query 2(Q2): SELECT SUM(x) FROM rel



- As we can see from the picture, vectorized execution can quickly compensate for iteration overhead.
- For the system I'm using, 64 tuples should conveniently fir into caches.