

Task 2

All tests are performed on next machine(M1):

- Processor - Intel i7 7700HQ
- Memory - 16GB DDR4
- Storage - 512GB NVMe SSD
- Ubuntu 18.04(Kernel: x86_64 Linux 4.15.0-91-generic)

Combined response time for all 540 tests on M1 and CI / CD were 35s and 53s respectively.

In following part of this task only times performed on M1 will be used, each time represent combined time for performing all 45 queries at one combination of execution model and data layout. Response times can vary for different execution and all times are based on only one execution.

Also, it is important to mention that Hash Join is used in all implementations.

Early materialization

Vulcano

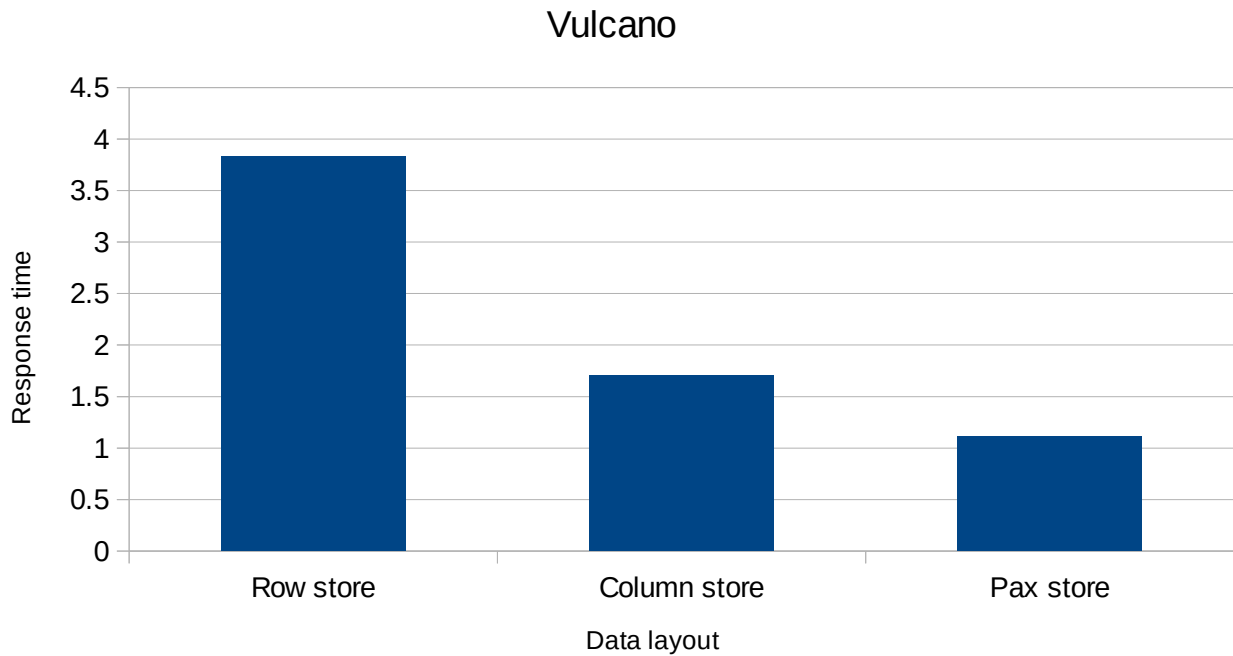


Illustration 1: Vulcano response times using different data layouts

As Illustration 1 presents Pax store performs the best, followed by Column store and slowest one is Row store. This result is unexpected since it is logical that Row store performs better than others because Vulcano executes tuple at a time. I assume that this behavior is due to the order of calls of different order of execution models in the query test file, because Row store is called first. Which I tested using different order of execution methods and in that case Row store indeed outperforms others. Also it is expected that Pax performs better than Column store since the next required tuple is probably on the same Pax page which is cached.

Operator at a time

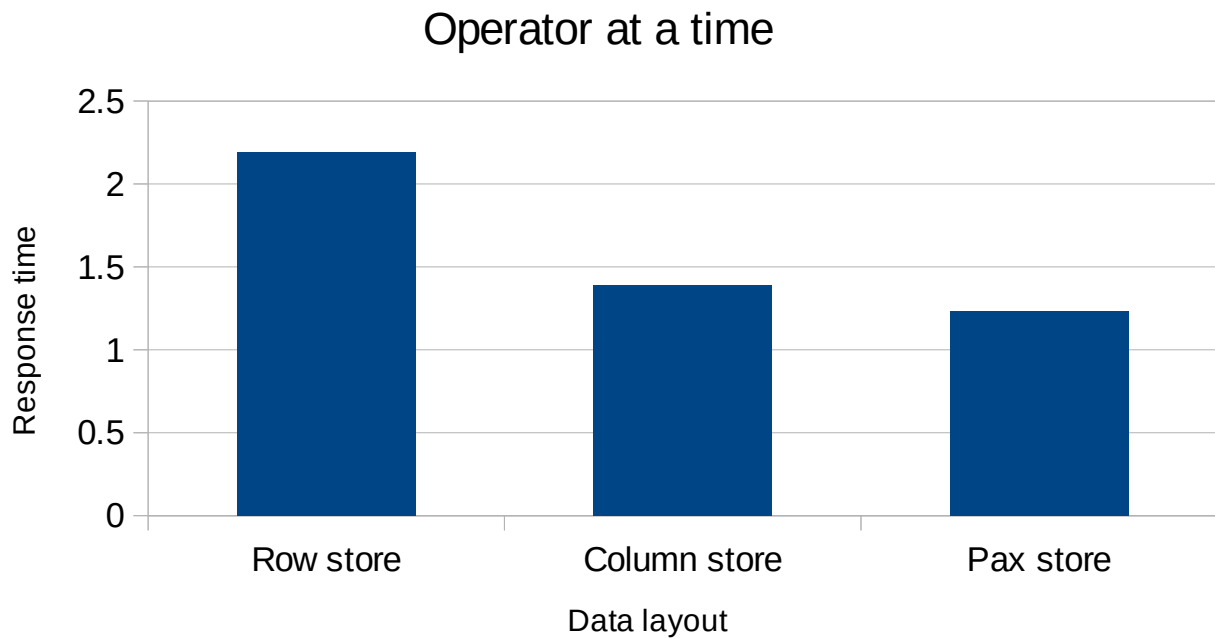


Illustration 2: Operator at a time response times using different data layouts

As Illustration 2 presents Pax store and Column performs almost the same, and Row store performs significantly worse than others. Pax store performs slightly better than Column store, which is expected since it combines best characteristics of other two. Reason why Row store behave the worst is because it does not use locality in this execution model.

Block-at-a-time

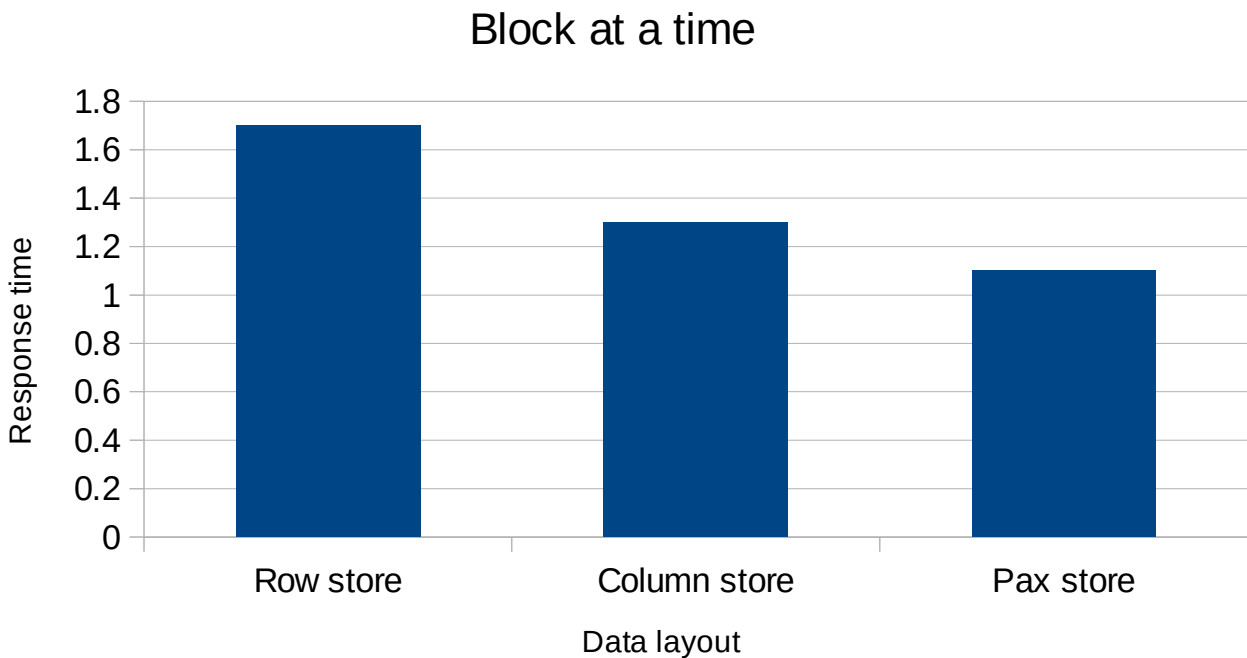


Illustration 3: Block at a time response times using different storage types

As Illustration 3 presents Pax is best choice when Block at a time is used especially when size of block is the same as size of Pax mini page as set in current implementation, because in that case one Pax page contains all information need for one block. Column store performed better than Row store which is not expected, but small difference between them can be justified by different executions, since results between executions can vary.

To conclude for all early materialization execution models Pax store performed the best it is reasonable because it is hybrid of other two and it combines the best of them.

Late materialization

Operator at a time

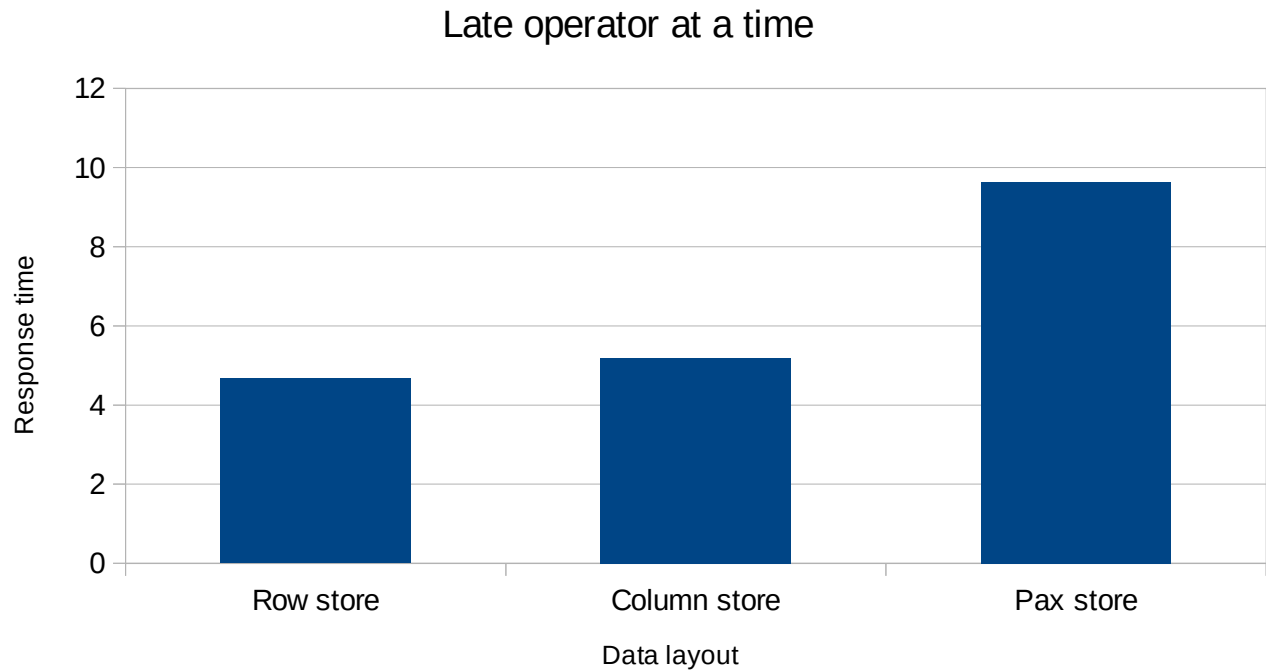


Illustration 4: Late operator at a time response times with different data layouts

As Illustration 4 shows Row store and Column store performed almost identically, and Pax store was the worst one. In this execution model Column store should perform the best, but difference is not that significant and it can be due to implementation.