Отчет по тестированию Яндекс КликХаус и Apache Cassandra

условия тестирования

- делалась партия по **2000** записей
- (пробная что бы долго не ждать, хотя вынесено в настройки)
- варьировалось только ширина записей при N=4 и 8000 столбцов (числа float) + служебные 1-2-3 столбца
- все в один поток – *на больше не хватало памяти (или утечка памяти или мало моих 16гиг озу)*

технические данные

1. язык программы GoLang (выбор пал потому, что утилиту нашел на этом языке)
2. на базе ПО - https://github.com/timescale/tsbs
3. ветка моей программы - https://gitlab.dds.lanit.ru/mmk_niokr/tools/-/tree/test_bigdata_golang (осталось readme написать)

сам отчет

кассанда 4 columns

loaded 2000 rows in 0.394sec with 1 workers (mean rate **5081.59 rows/sec**)

кассанда 8000 columns

loaded 2000 rows in 94.353sec with 1 workers (mean rate 21.20 rows/sec)

яндекс клик-хаус 4 columns

loaded 2000 rows in 7.334sec with 1 workers (mean rate **272.71 rows/sec**)

яндекс клик-хаус *8000 columns*

loaded 2000 rows in 528.171sec with 1 workers (mean rate **3.79 rows/sec**)

вывод

как я и говорил если мало нужно, то кассанда лучше

(на коне - яндекс, если счет 100 тыс записей за раз)
по размеру данных на HDD - не смотрел, но смысла нет на 2000 записей а на большее нужно сутками держать ПК включенным — но, и там, и там есть режим сжатия LZ4 и другие)

при этом 8 тыс столбовая таблица быстрее, чем узкая на 4 столбца, так как издержки на транзакции и пересчет индексов и т .п

при этом, если поток не один 1 а 8 на 4х ядерном процессоре моем но прирост скорости копеечный — смысла распарить нет

моц ПК

4х разрядный CPU Intel i3 4ого поколения 3.8GHz (2 реальных ядра + 2 виртуальных)
SSD для ОС и HDD для базы данных
память DDR3-1600 16Гбайт

Замечание
клик-хаус был на сервере удаленным, а кассандра локально –так как пока временные проблемы с сервером кассанда – при этом кассандра все равно «выиграла» (так как клиент и база ела мои ресурсы а яндекс-бд как бы на 2 ПК был)