

Выпускная квалификационная работа

Продвинутые методы ускорения сортировки в ClickHouse

Евгений Правда, гр. 155

Научный руководитель:
Алексей Миловидов,

Руководитель группы разработки ClickHouse



Введение в предметную область

- ClickHouse – поколоночная база данных
- Изначально разработана в Яндексе для сервиса веб-аналитики Яндекс.Метрика
- Основные конкуренты - Vertica, Infobright, InfiniDB and MonetDB
- Работает в 100-1000 раз быстрее чем традиционные базы данных, производительность выше чем у аналогов
- Использует все доступные аппаратные ресурсы




Актуальность задачи

- Аналитики используют базу данных для выполнения аналитических запросов
- Во многих случаях требуется дожидаться завершения запроса для продолжения работы
- Таким образом, низкая скорость базы данных может негативно сказаться на производительности работы аналитиков


Актуальность задачи

- Сортировка вызывается для каждого запроса с ORDER BY
- Многие запросы используют упорядочивание, часто с LIMIT
- В бенчмарке ClickHouse 32 из 43 запросов используют ORDER BY
- Алгоритм сортировки может существенно влиять на производительность базы данных



Цель и задачи дипломной работы


- Ускорить сортировку
- Реализовать частичную сортировку на основе msd radix sort
- Произвести тестирование производительности



Обзор существующих методов

Основанные на сравнениях:

- Quicksort
 - Heapsort
 - Mergesort
 - Гибридные (Timsort, Introsort, etc)
-
- Альтернативные:
 - Radix sort
 - Counting sort



Обзор существующих методов

Radix sort

- Имеет несколько разных модификаций
- Наиболее известные и рассматриваемые в данной работе: MSD и LSD, есть более экзотические разновидности

Предложенный метод (1)

pdqsort

- По результатам сравнения нескольких алгоритмов сортировки pdqsort был выбран как оптимальный comparison-based алгоритм, и использован в качестве замены `std::sort`

Предложенный метод (2)

Radix sort

- Существующий метод LSD radix sort модифицирован для возможности использования при получении перестановки строк (getPermutation)

Предложенный метод (3)

Partial most significant digit radix sort

- Реализация частичной сортировки на основе kxsort
- kxsort – эффективная in-place реализация алгоритма MSD radix sort
- В данной работе произведена его модификация для частичной сортировки

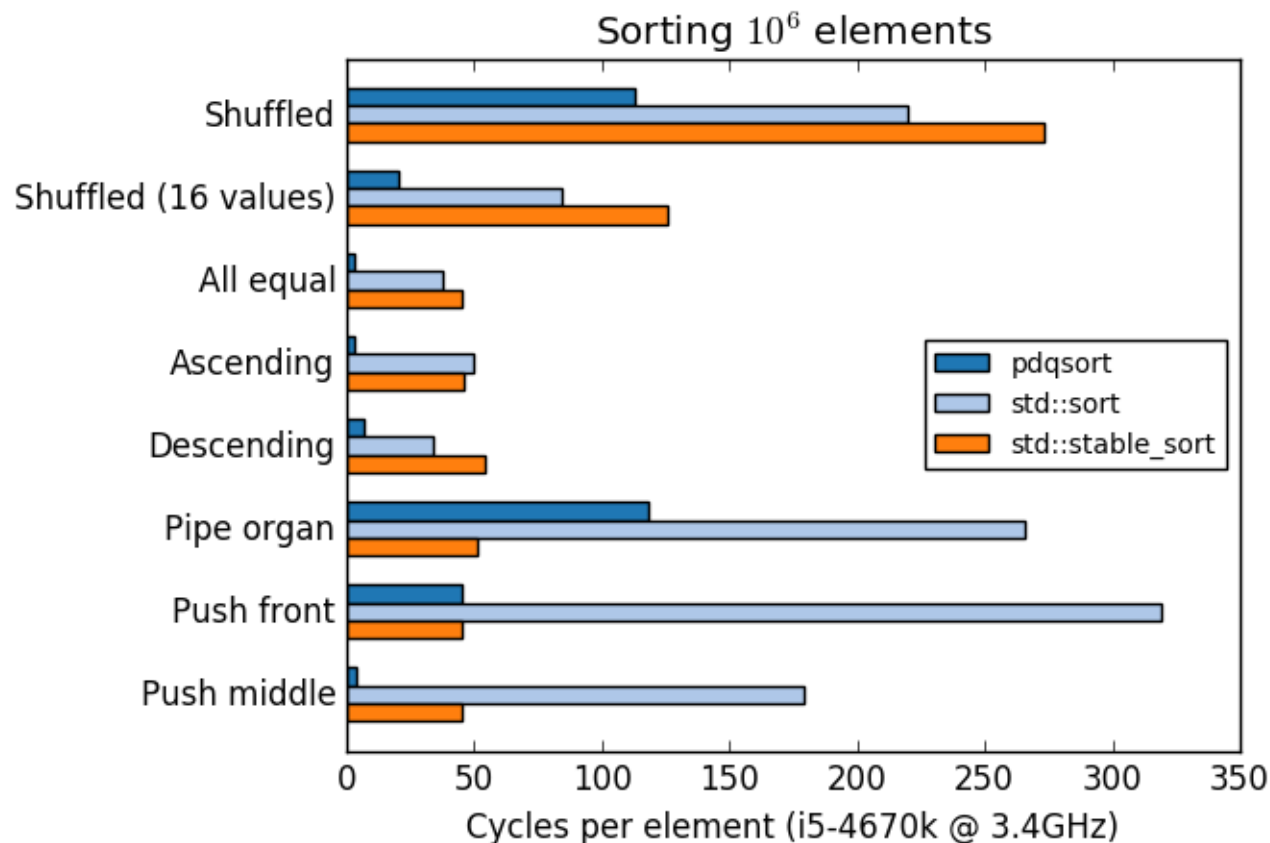


Тест

Написан performance-тест, измеряющий производительность сортировки для массивов чисел разной длины (1 млн, 10 млн), типа (8, 16, 32, 64 бита) и пространства ключей (1000, 100000 и неограниченного)

Оценка производительности pdqsort

Время работы самого алгоритма
(<https://github.com/orlp/pdqsort>)



Экспериментальная оценка (pdqsort)

	std::sort	pdqsort
10^3	0.819	0.747
10^5	0.838	0.826
2^{64}	1.142	1.057

- Таблица для данных на 10 миллионов 64-битных чисел
- По вертикальной оси – размер пространства ключей
- Прирост производительности 5-10%, для других входов данных до 20%

Экспериментальная оценка (radix sort)

	pdqsort	radix sort
10^3	0.95	0.89
10^5	0.98	0.83
2^{64}	1.29	1.04

- Таблица для данных на 10 миллионов 64-битных чисел
- По вертикальной оси – размер пространства ключей
- Прирост производительности 7-20%, для других входов данных до 20%
- Тестировалось на другой машине



Развитие проекта

- Есть возможность дальнейшей оптимизации написанных алгоритмов
- Одно из возможных расширений — поддержка сортировки кортежей, т. е. возможность указания в запросе ORDER BY с несколькими столбцами.

Программная реализация

- Программная часть работы написана на языке C++ стандарта C++17
- Объём кода – 4 пулл-реквеста, в сумме +552 -172
- Добавление внешних библиотек происходило с помощью git submodules и CMake
- Замеры производительности – написание теста и его запуск вручную и автоматически (коммитно)

Результаты дипломной работы

В результате работы ускорена в ClickHouse были добавлены новые алгоритмы сортировки, что позволило повысить эффективность запросов к базе данных, использующих ORDER BY

- Таким образом, в ClickHouse теперь доступны следующие алгоритмы: `std::sort`, `std::partial_sort`, `pdqsort`, LSD radix sort, MSD radix sort
- В зависимости от требований конкретной ситуации может использоваться один из них, наиболее быстрый, и при этом имеющий необходимые особенности