



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Распараллеливание парсинга форматов данных

Курсовая работа

Олег Ершов, 3 курс ВШЭ БПМИ

Алексей Миловидов, руководитель группы разработки СУБД ClickHouse,
Яндекс

Факультет компьютерных наук
01.03.02 «Прикладная математика и информатика»

6 июня 2019 г.



Задача состоит в распараллеливании парсинга форматов данных.

Актуальность задачи

Парсинг форматов данных

- ClickHouse - высокопроизводительная СУБД, в данный момент парсинг форматов максимально эффективен, но ограничен скоростью одного ядра.
- Что в некоторых случаях является узким местом и можем значительно его ускорить.

Цель: улучшить время, затрачиваемое на разбор данных для форматов TSV, JSONEachRow, CSV, Values.

Задачи:

- Изучить архитектуру ClickHouse, которая относится к форматам данных
- Разработать метод для быстрого получения куска файла для обработки.
- Разработать метод для потокобезопасного чтения из входного буфера.
- Реализовать их и оценить полученный результат.

`ReadBuffer` - абстрактный класс для буферизированного чтения из какого-то места. Предоставляет доступ к внутреннему буфер. Его наследники переопределяют функцию `nextImpl()`, которая заполняет буфер.

`IRowInputStream` - интерфейс потока, который позволяет читать данные по строкам.

`IBlockInputStream` - интерфейс потока, который позволяет читать блоки данных.

`UnionBlockInputStream` - поток, который позволяет параллельно читать данные из других

`IBlockInputStream`

`FormatFactory` - класс, который имеет метод `getInput`, который создает и возвращает

`IBlockInputStream` для чтения или вывода данных по имени формата.

В `FormatFactory` создается `IBlockInputStream` по имени формата, который из переданного в конструкторе `ReadBuffer` читает и возвращает данные, исходя из спецификации формата.

- Для каждого формата добавляется `FileSegmentationEngine` - метод, позволяющий получить кусок файла для чтения.
- Для каждого потока создаем свой `SharedReadBuffer`, реализация `ReadBuffer`, которая отвечает за безопасное чтение.
- В методе `getInput` создаем несколько `IBlockInputStream` данного формата, каждый на сход получает свой `ReadBuffer`.
- Возвращаем `UnionBlockInputStream` от полученных потоков.

Класс, который под мьютексом читает данные в собственную память из родительского буфера, используя `FileSegmentationEngine`.

Параметры:

- `ReadBuffer` - родительский буфер из которого идет чтение
- `shared_ptr<mutex>` - указатель на общий mutex
- `FileSegmentationEngine` - функция для получения следующего чанка
- `min_chunk_size` - минимальный размер чанка



Метод, который быстро читает из входного `ReadBuffer` байты, следя только за разделителями. Это делается с помощью функции `find_first_symbols`. Читает минимум `min_chunk_size` байт и ещё сколько-то до конца следующего сообщения и копирует это в память `SharedReadBuffer`.

`JSONEachRow` : сообщение - валидный JSON, разделены запятыми. Следим за символами `\`, `{`, `}`, `"`.

`TSV` : сообщение - отдельная строка, значения разделены табуляциями. Следим за `\`, `\r`, `\n`.

`Values` : сообщение заключены в круглых скобках. Сообщения и значения внутри них разделяются запятыми. Следим за `\`, `'`, `)`, `(`

`CSV` : сообщение - отдельная строка, значения разделяются символом-разделителем. Следим за `\r`, `\n`

Format	1 core	2 cores	4 cores
JSONEachRow	1.391	1.036	0.764
TSKV	1.115	0.676	0.473
Values	0.598	0.383	0.215
TSV	0.619	0.398	0.332
CSV	0.870	0.687	0.589

- Изучена часть архитектуры ClickHouse, которая относится к обработке данных
- Придуманы и реализованы методы, позволяющие быстро получать кусок данных и потокобезопасно читать их
- Был произведен тест на производительность и получены положительные результаты

Распараллеливание парсинга форматов данных

Олег Ершов, 3 курс ВШЭ БПМИ

Алексей Миловидов, руководитель группы разработки СУБД ClickHouse, Яндекс

Факультет компьютерных наук
01.03.02 «Прикладная математика и информатика»

6 июня 2019 г.