Профилирование и трассировка на уровне отдельных запросов в ClickHouse

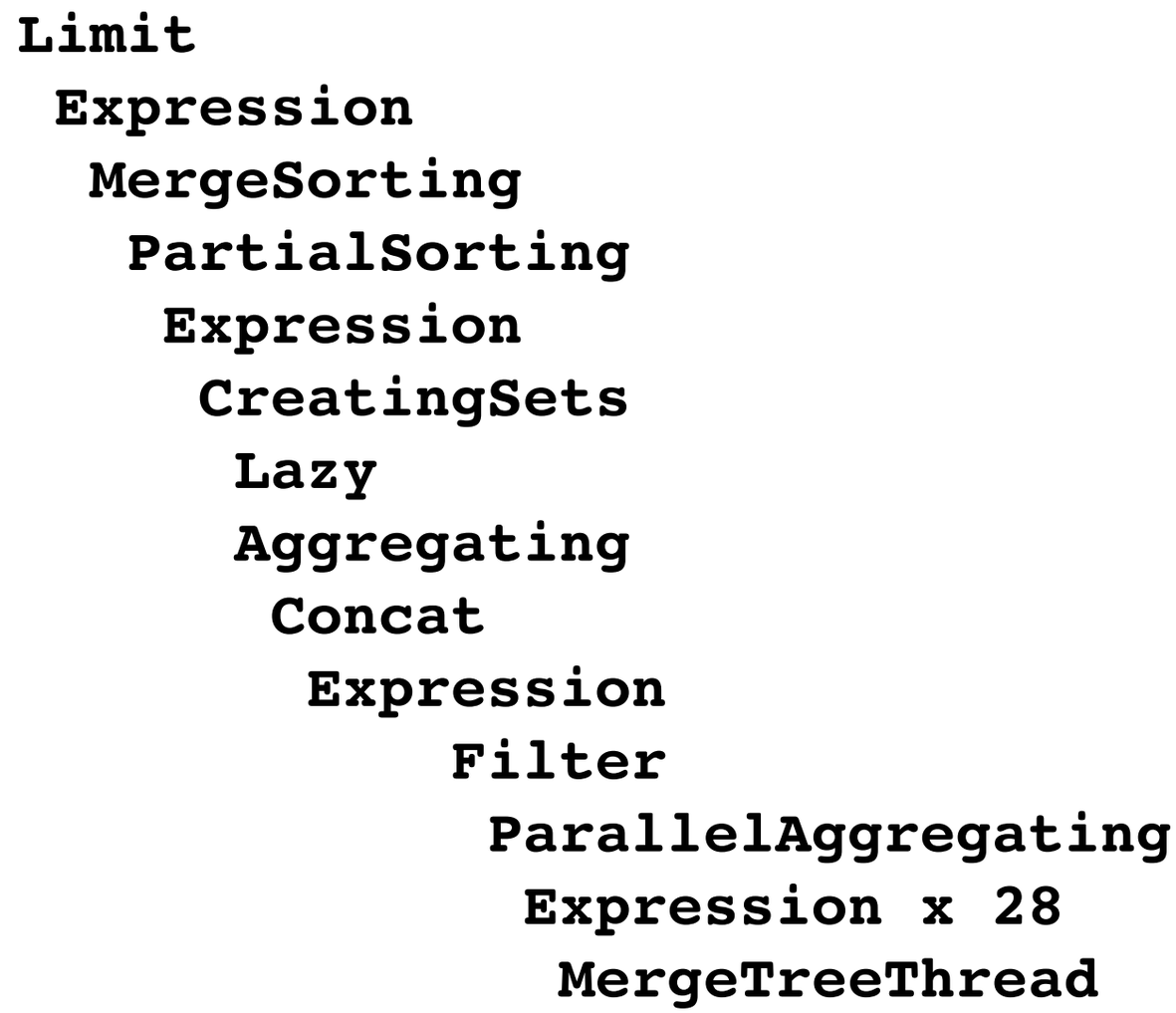
Лапков Никита Алексеевич, БПМИ165

Научный руководитель:
руководитель группы разработки ClickHouse, Миловидов Алексей Николаевич

- СУБД
- Поколоночная
- SQL-like запросы
- Фокус на скорости

Что тормозит на критическом пути?

```
SELECT
    DestCityName,
    any(total),
    avg(abs((monthly * 12) - total) / total) AS
avg_month_diff
FROM
(
    SELECT
        DestCityName,
        count() AS total
    FROM ontime
    GROUP BY DestCityName
    HAVING total > 100000
)
ALL INNER JOIN
(
    ...
```


Limit
Expression
MergeSorting
PartialSorting
Expression
CreatingSets
Lazy
Aggregating
Concat
Expression
Filter
ParallelAggregating
Expression x 28
MergeTreeThread

```
Limit
 Expression
  MergeSorting
   PartialSorting
    Expression
     CreatingSets
      Lazy
       Aggregating
        Concat
         Expression
              Filter
               ParallelAggregating
                Expression x 28
                 MergeTreeThread
```

1. Preallocating buffer
2. Split to N threads
3. ...

```
Limit
 Expression
  MergeSorting
   PartialSorting
    Expression
     CreatingSets
      Lazy
Aggregating ←——— Interpreters/Aggregator.cpp:719
 Concat
  Expression
      Filter
       ParallelAggregating
        Expression x 28
         MergeTreeThread
```

Актуальность

- Очевидные оптимизации проведены
- Сложно анализировать самостоятельно
- Сложно учитывать потребности пользователей

Цель и задачи

- Цель: создать инструментарий для профилирования и трассировки кода на пути выполнения единичного запроса
- Задачи:
    1. Встроить позапросный профайлер в ClickHouse
    2. Провести эксперименты по добавлению трассировки

Формальная постановка

- **Отчет профайлера** - это список функций программы упорядоченный по их суммарному времени исполнения

- **Приближение отчета** - отчет в котором нарушен порядок или представлены не все функции

- **Профиль** - это внутреннее представление профайлера, после обработки которого может быть получено приближение отчета

- Вход: запрос пользователя

- Выход: профиль в контексте запроса

Профайлер бедного человека

- Сэмплирующий профайлер
- Работает с многопоточными приложениями
- Может учитывать не только CPU, но и реальное время
- Почти нет накладных расходов
- Плохо подходит для профилирования процессов с коротким жизненным циклом

Базовый алгоритм

1. Остановка процесса
2. Получение stack trace всех тредов с помощью gdb
3. Возобновление выполнения
4. Ожидание
5. Перейти к пункту 1.

poormansprofiler.org

Проблемы

- Нет привязки к запросу
- Нет возможности отключить или изменить период сэмплирования для отдельных запросов
- CPU часы другого процесса недоступны
- Большие накладные расходы
- Потеря герметичности

Профайлер человека среднего класса

1. Остановка треда исполняющего запрос сигналом
2. Раскрутка текущего stack trace
3. Запись stack trace в системную таблицу
4. Возобновление выполнения треда
5. Ожидание (по CPU или реальным часам)
6. Перейти к пункту 1.

Пример работы

```
SELECT
    timer_type,
    count()
FROM system.trace_log
GROUP BY timer_type
```

timer_type	count()
Real	191358
CPU	81511

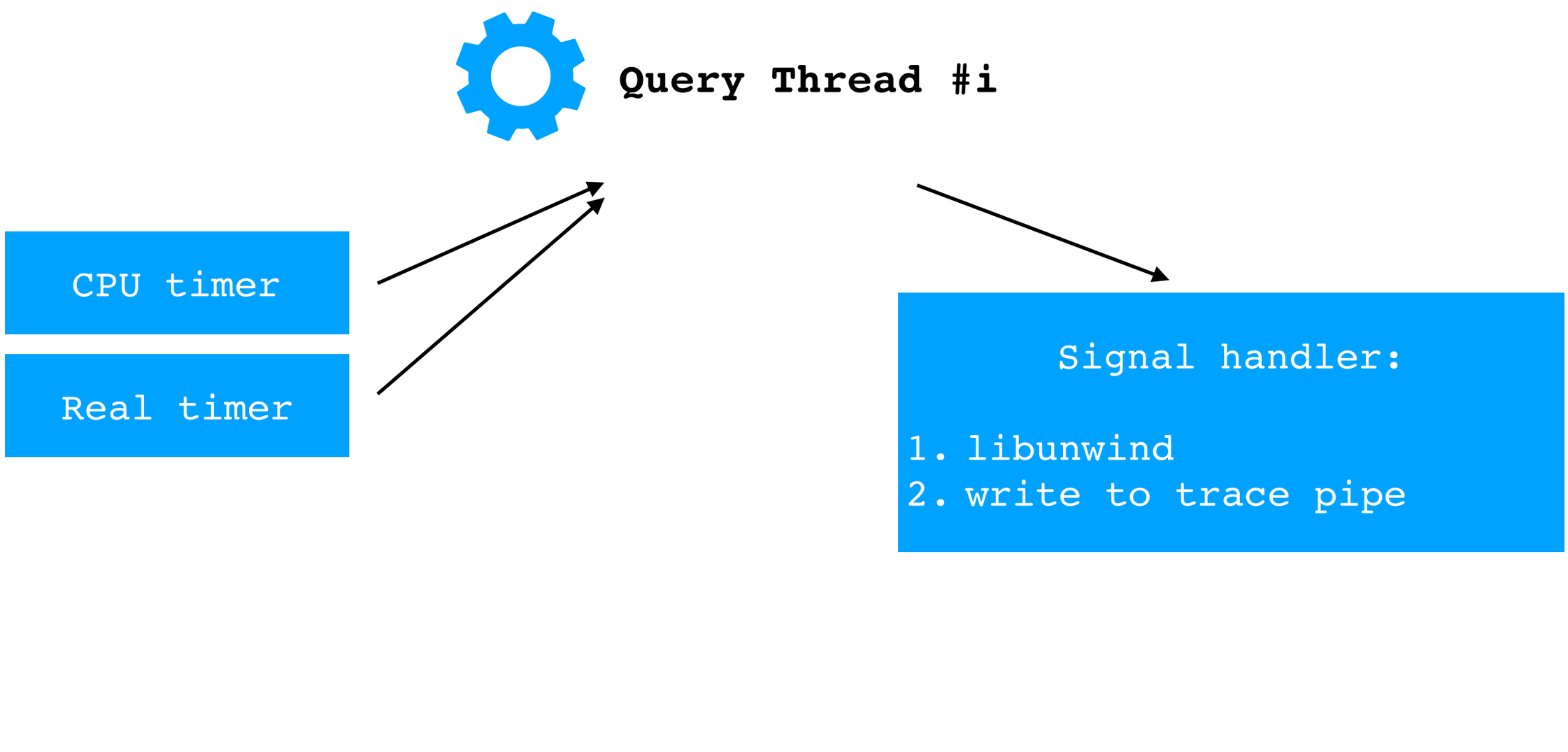
```
SELECT
    length(trace),
    bar(count(), 0, 200000, 20)
FROM system.trace_log
GROUP BY length(trace)
ORDER BY count() DESC
```

length(trace)	bar(count(), 0, 200000, 20)
22	
12	
18	
32	
29	
27	
25	

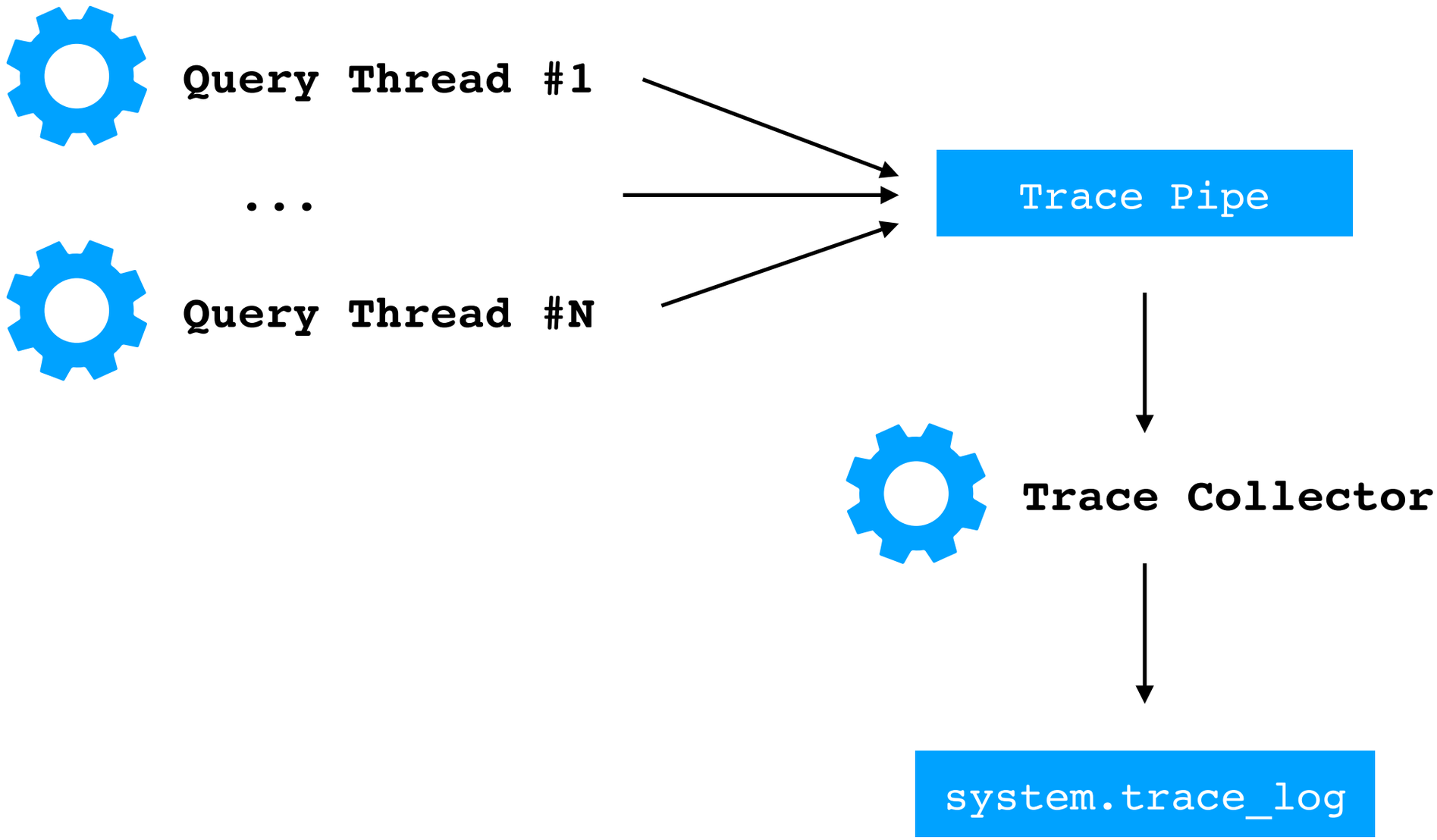
```
SELECT
    query_id,
    timer_type,
    trace,
    symbolizeTrace(trace)
FROM system.trace_log
LIMIT 1
FORMAT Vertical

Row 1:
──────
query_id:                892a2826-bb6b-4fac-90f3-ca72ebf7d4d4
timer_type:              Real
trace:                   [169078224,154929907,154934188,53294340,53501360,28703
01,160183246,151334801,160178626,151334801,160178626,151334801,156739564,1513
8960,168751381,168762104,168740757,168755421,53268307,2870281914,2864174109]
symbolizeTrace(trace): 0. ./clickhouse-server() [0xa13edd0]
1. ./clickhouse-server(auto DB::FunctionBinaryArithmetic<DB::DivideFloatingIm
r<unsigned long> > const&, unsigned long, unsigned long)::'lambda'(auto const
 >(auto const&, auto const&) const+0x653) [0x93c0af3]
2. ./clickhouse-server(auto DB::FunctionBinaryArithmetic<DB::DivideFloatingIm
r<unsigned long> > const&, unsigned long, unsigned long)::'lambda'(auto const
> >(auto const&, auto const&) const+0x25c) [0x93c1bac]
3. ./clickhouse-server() [0x32d3504]
4. ./clickhouse-server() [0x3305db0]
5. [0xab15a390]
6. [0xab15635e]
7. ./clickhouse-server() [0x33367b5]
8. ./clickhouse-server() [0x32cd954]
```

Реализация: снизу



Реализация: сверху



Результаты

- В ClickHouse встроен сэмплирующий позапросный профайлер для двух таймеров (CPU и реальное время)
- Профили доступны для анализа из языка запросов
- Язык запросов расширен для анализа профиля
- Эксперименты с трассировкой не успели из-за большого количества технических проблем с профайлером

Технические трудности

- Сборка под sanitizers
- Замена nongnu libunwind на LLVM libunwind
- Не все syscalls перезапускаются после сигнала
- 17ый регистр в x86_64
- Data races при read-after-close
- Совместимость с Ubuntu 12.04