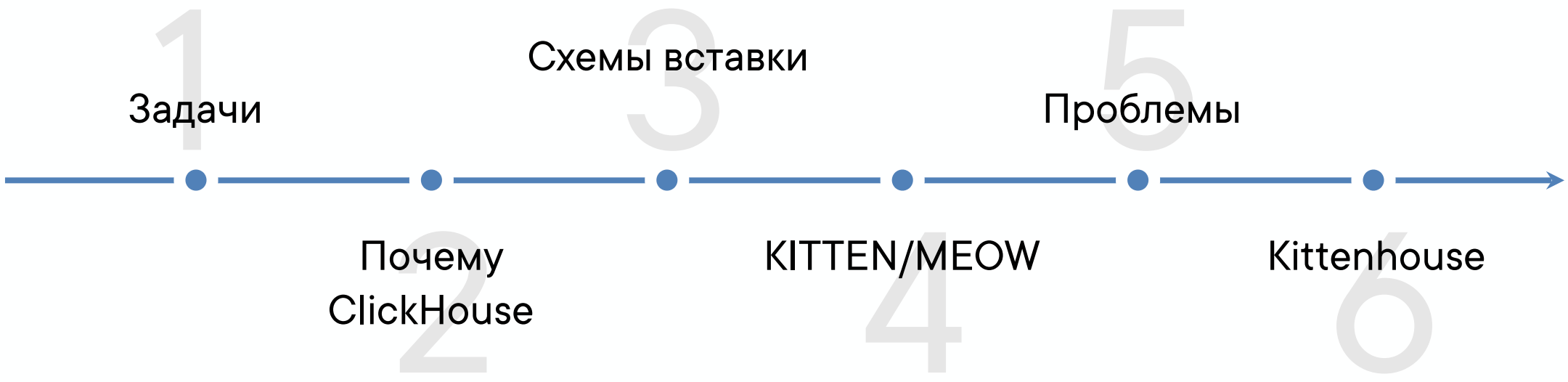
Как VK вставляет данные в ClickHouse с десятков тысяч серверов

Юрий Насретдинов

План


1
Задачи
2
Почему ClickHouse
3
Схемы вставки
4
KITTEN/MEOW
5
Проблемы
6
Kittenhouse

Задачи

Хранить, собирать
и смотреть
дебаг-логи (100+ Тб)

Собирать
статистику

...с десятков
тысяч серверов

Возможные варианты

+ файлы

LSD

+ файлы

Или свой смешной вариант

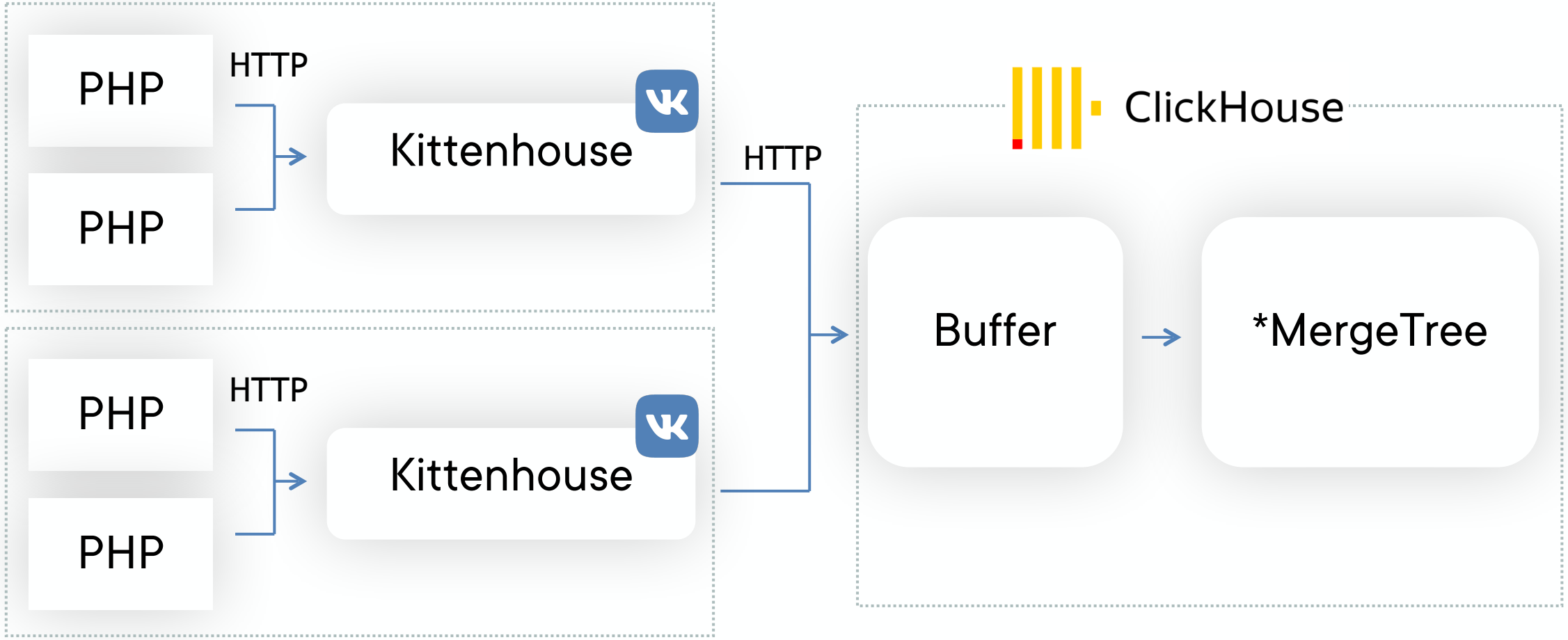
Logs Engine

1. Простота эксплуатации
2. Высокая скорость записи и чтения
3. Долговременное хранение (месяцы, годы)
4. Сжатие данных
5. Запись длинных строк (>4 Кб)
6. Очень много серверов (UDP)

ClickHouse

1. Простота эксплуатации
2. Высокая скорость записи и чтения
3. Долговременное хранение (месяцы, годы)
4. Сжатие данных
5. Запись длинных строк (>4 Кб)
6. Очень много серверов (UDP)

Схема v2


PHP
PHP
HTTP
Kittenhouse
PHP
PHP
HTTP
Kittenhouse
HTTP
ClickHouse
Buffer
*MergeTree

Kittenhouse v1

1. Локальный прокси на Go
2. Одно TCP исходящее соединение с сервера
3. Flood control
4. (Опционально) надежная доставка
5. Формат VALUES (обычный SQL)
6. Сброс локального буфера раз в 2 сек

Накопили буфер

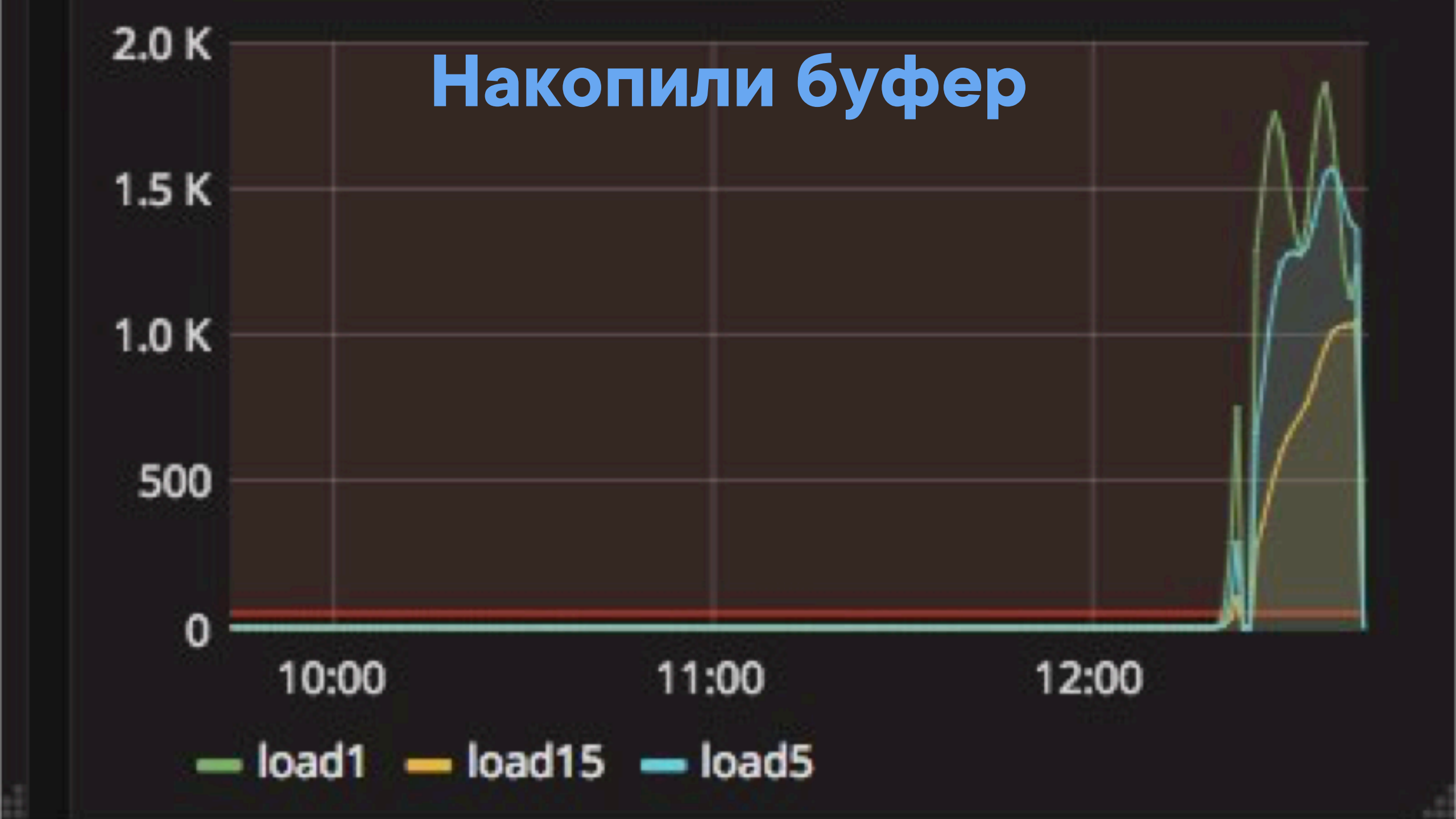
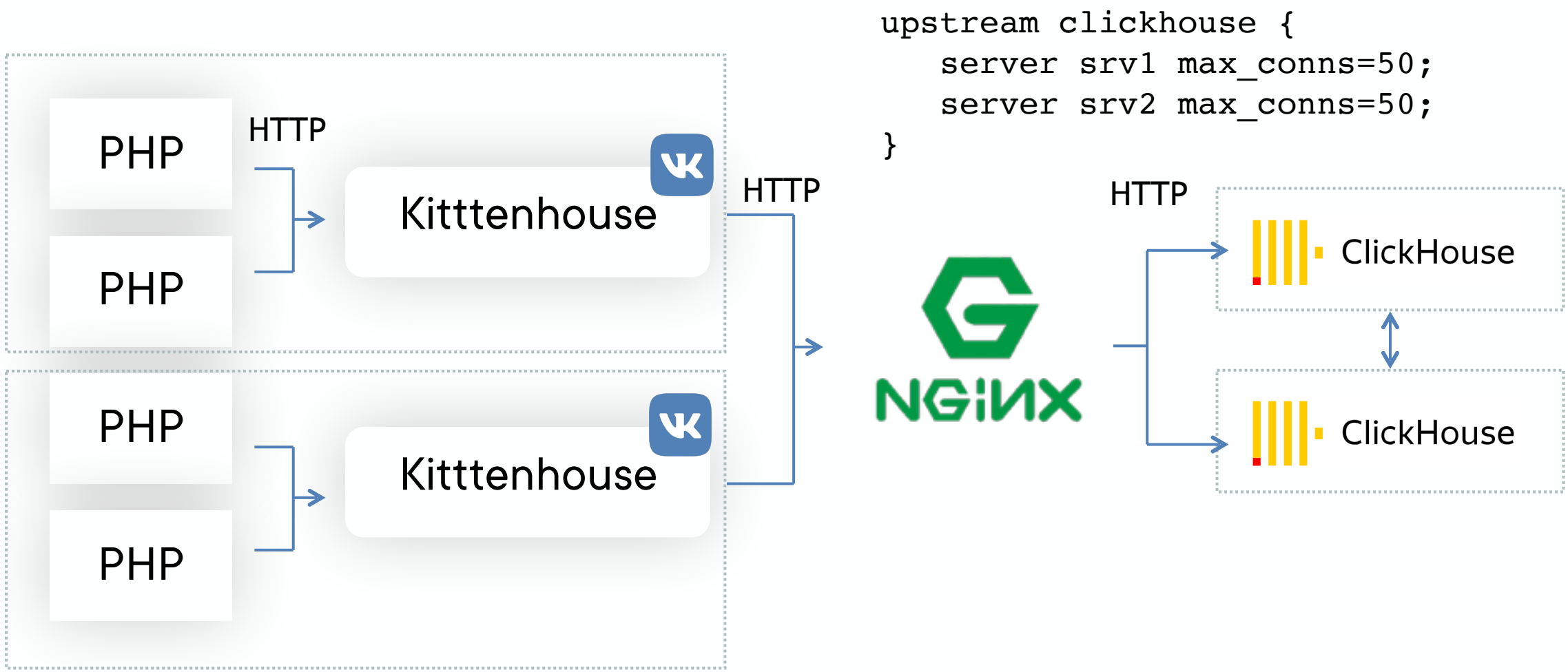


Схема v3




```
upstream clickhouse {
    server srv1 max_conns=50;
    server srv2 max_conns=50;
}
```

Нюансы

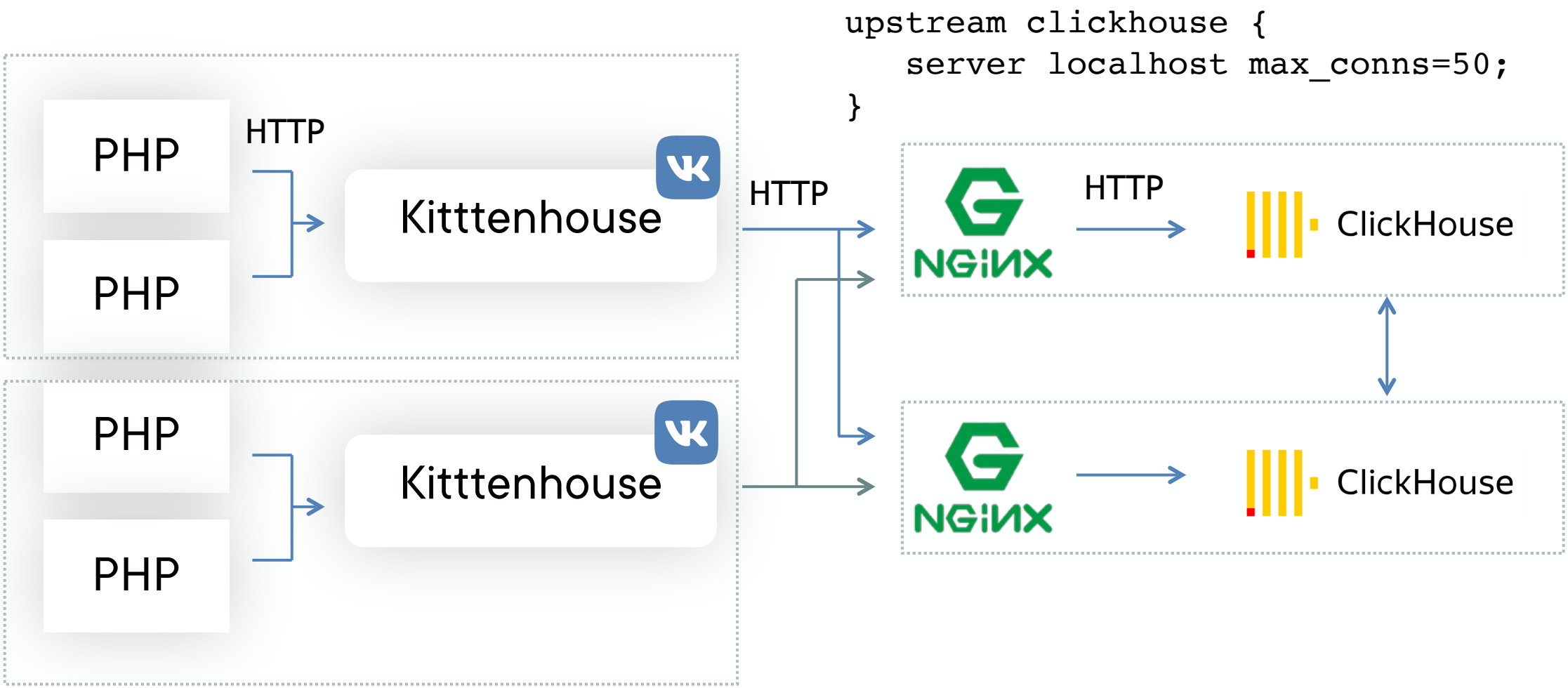
1. Нужны health checks
2. Вставка пачками в ~256Кб фейлится
3. В DateTime можно вставлять UNIX Time...
4. Но только строкой!
5. https://github.com/yandex/ClickHouse/issues/1850
6. input_format_values_interpret_expressions=0

Kittenhouse v2

1. Свои логи пишет тоже в ClickHouse
2. Форсим потоковый SQL Parser
3. Балансировка нагрузки (взвешенный round-robin)
4. Health checks (раз в 30 секунд)
5. Роутинг на разные кластера
6. UDP на localhost

Схема v4

```
upstream clickhouse {
    server localhost max_conns=50;
}
```



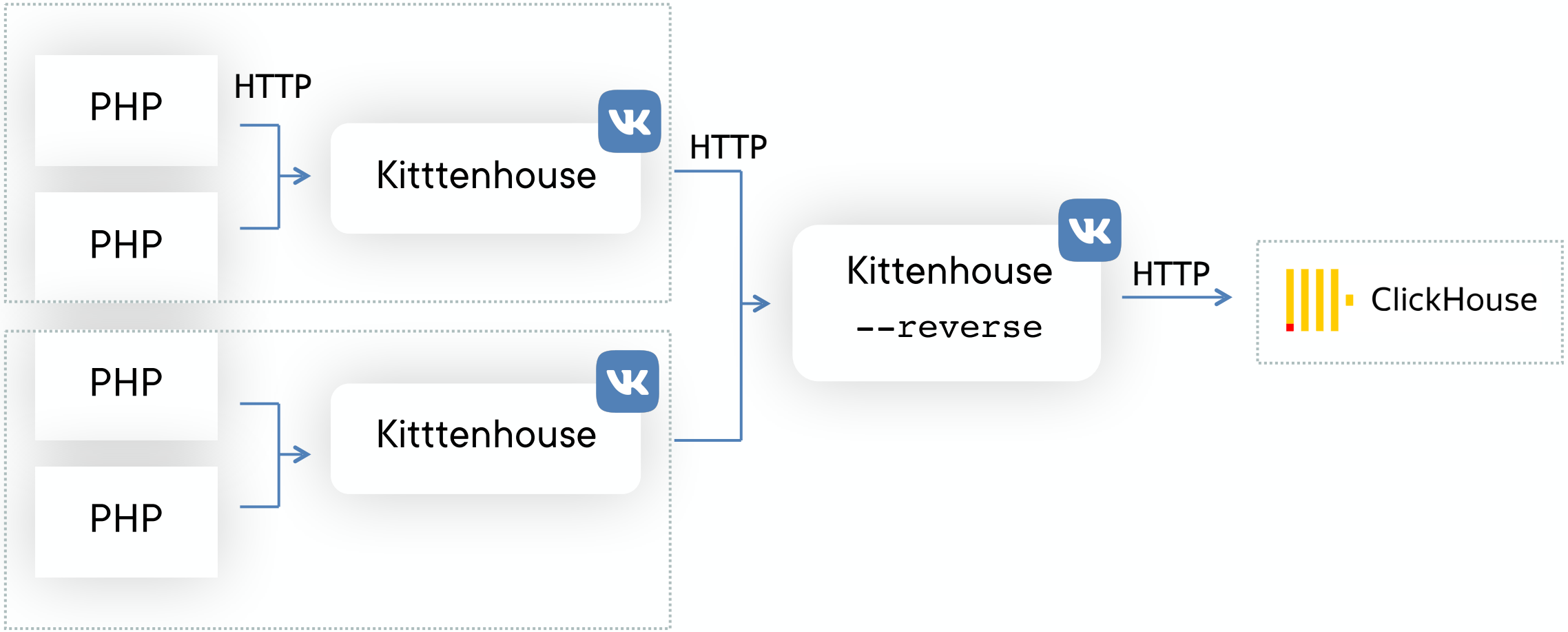
502 Bad Gateway

0,5%

запросов
отдавали 502

- Заканчивались коннекты к upstream в nginx
- 20 таблиц, буферные таблицы с кучей кусков
- Много мержей (2–3 параллельно)
- ioutil 80%

Схема v5


PHP
PHP
HTTP
Kitttenhouse
PHP
PHP
Kitttenhouse
HTTP
Kittenhouse
--reverse
HTTP
ClickHouse

Kittenhouse --reverse

1. Замена nginx для работы с ClickHouse
2. Раздельные пулы для SELECT и INSERT
3. Агрегирует вставку в Buffer-таблицы
4. Маленький буфер: 25мс или 1 Мб
5. Синхронная вставка
6. Использует fasthttp — легко держит 100K+ QPS
7. Можно вставлять по 1 строке

Схема v5

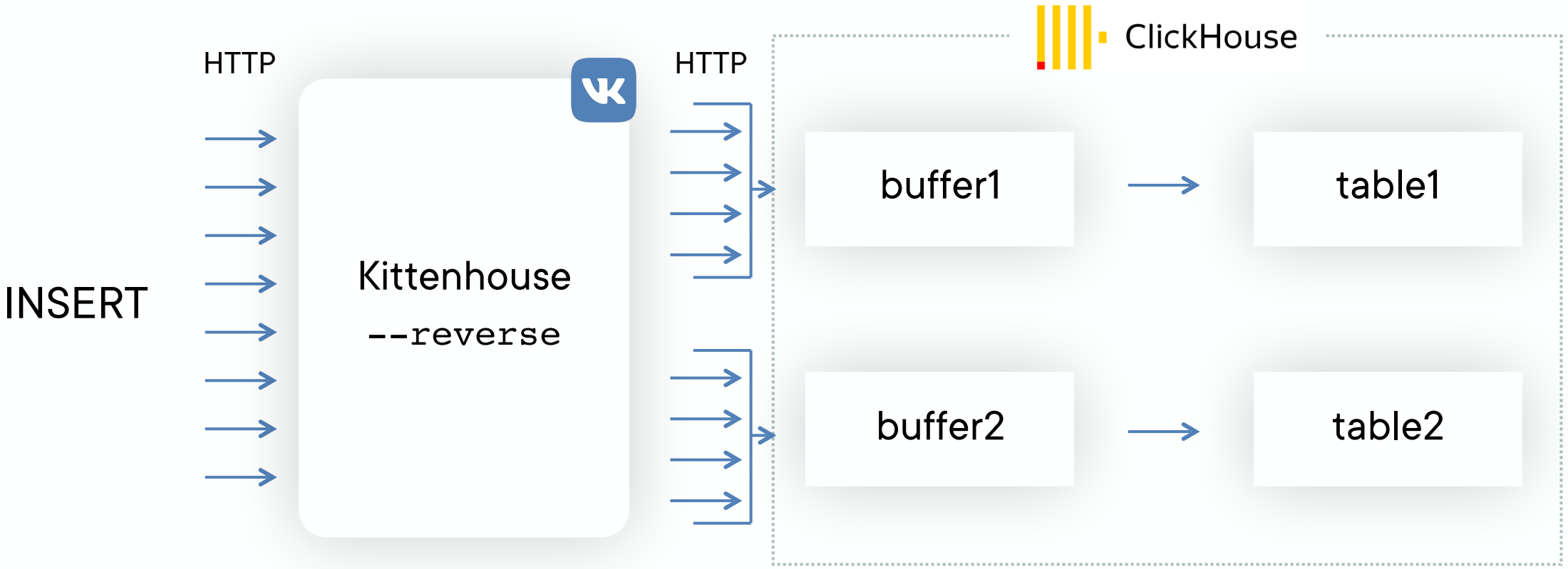
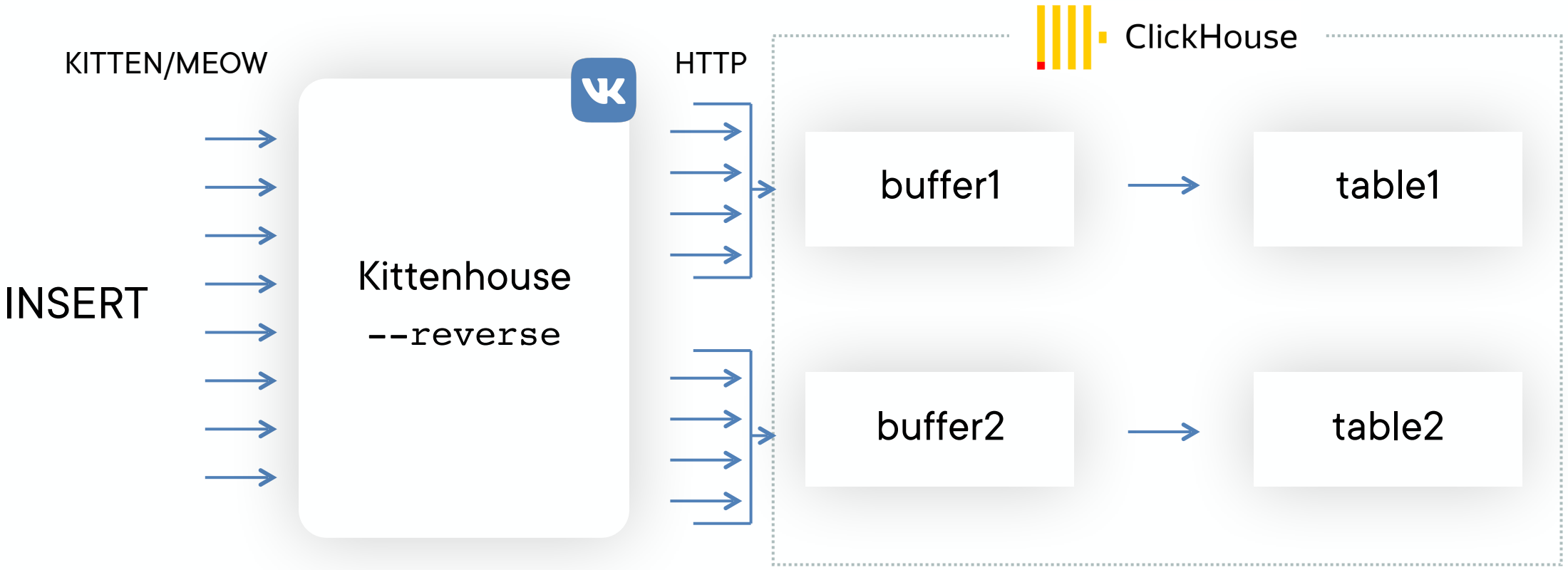

ClickHouse
HTTP
HTTP
INSERT
Kittenhouse
--reverse
buffer1
table1
buffer2
table2

Схема v6


KITTEN/MEOW
INSERT
Kittenhouse
--reverse
HTTP
ClickHouse
buffer1
table1
buffer2
table2

KITTEN/MEOW

1. Надстройка над HTTP
2. Ожидает MEOW в ответ на метод KITTEN
3. Читаем максимум из 50 клиентов за раз
4. Потребление памяти в 20 раз меньше
5. Бинарный протокол
6. Нет оверхеда на HTTP заголовки

Выводы

1. ClickHouse подходит для логов
2. Буферные таблицы имеют много кусков
3. KittenHouse будет выложен в Open Source

ВКонтакте
с Вами!

Юрий Насретдинов
y.nasretdinov@corp.vk.com
vk.com/ynasretdinov

VK Backend: vk.com/vkbackend
VK Github: github.com/vkcom