

666 PR спустя

Как мы делаем userver

Антон Полухин,
эксперт разработчик C++, Яндекс

YANDEX OPEN SOURCE JAM

 Open Source

Содержание

1. Что такое userver
2. Документация
3. Как мы работаем с внешними PR
4. Автоматика по принятию PR
5. Issues, и что мы с ними делаем
6. Пример с Websockets
7. Релизы
8. Смежники
9. Студенты
10. Итоги

Что такое userver?

Давным-давно, когда Go ещё был просто Taxi...

Нужно выбрать фреймворк, который:

Нужно выбрать фреймворк, который:

1

Бесшовно может
переиспользовать
имеющуюся большую
C++ кодовую базу

Нужно выбрать фреймворк, который:

1

Бесшовно может
переиспользовать
имеющуюся большую
C++ кодовую базу

2

Держит большие
нагрузки и не под-
вержен проблеме
10K соединений

Нужно выбрать фреймворк, который:

1

Бесшовно может
переиспользовать
имеющуюся большую
C++ кодовую базу

2

Держит большие
нагрузки и не под-
вержен проблеме
10K соединений

3

Прост и удобен
в использовании

Другие фреймворки

```
void View::Handle(Request&& request, const Dependencies& dependencies) {
    dependencies.pg->GetCluster([request = std::move(request)](auto cluster) {

        cluster->Begin(storages::postgres::ClusterHostType::kMaster, [request = std::move(request)]
(auto trx) {
            const std::string statement = "SELECT ok, baz FROM some WHERE id = $1 LIMIT 1";

            psql::Execute(trx, statement, request.id, [request = std::move(request), trx =
std::move(trx)](auto row_res) {
                auto row = row_res[0];
                if (!row["ok"].As<bool>()) {
                    GetSomeInfoFromDb([request = std::move(request)] (auto data) {
                        LOG_DEBUG() << request.id << " is not OK for " << data; // Не эффективно
                        request.SendResponse(Response400(), []{});
                    });
                }
                return;
            });

            psql::Execute(trx, queries::kUpdateRules, request.foo, request.bar,
[request=std::move(request), trx=std::move(trx)](auto){
                trx.Commit();
                request.SendResponse(Response200{row["baz"].As<std::string>()}, []{});
            });
        });
    });
}
```


Другие фреймворки

```
void View::Handle(Request&& request, const Dependencies& dependencies) {
    dependencies.pg->GetCluster([request = std::move(request)](auto cluster) {

        cluster->Begin(storages::postgres::ClusterHostType::kMaster, [request = std::move(request)]
(auto trx) {
            const std::string statement = "SELECT ok, baz FROM some WHERE id = $1 LIMIT 1";

            psql::Execute(trx, statement, request.id, [request = std::move(request), trx =
std::move(trx)](auto row_res) {
                auto row = row_res[0];
                if (!row["ok"].As<bool>()) {
                    GetSomeInfoFromDb([request = std::move(request)] (auto data) {
                        LOG_DEBUG() << request.id << " is not OK for " << data; // Не эффективно
                        request.SendResponse(Response400(), []{});
                    });
                }
                return;
            });

            psql::Execute(trx, queries::kUpdateRules, request.foo, request.bar,
[request=std::move(request), trx=std::move(trx)](auto){
                trx.Commit();
                request.SendResponse(Response200{row["baz"].As<std::string>()}, []{});
            });
        });
    });
}
```

Userver

```
Response View::Handle(Request&& request, const Dependencies& dependencies) {  
    auto cluster = dependencies.pg->GetCluster();  
    auto trx = cluster->Begin(storages::postgres::ClusterHostType::kMaster);  
  
    const std::string statement = "SELECT ok, baz FROM some WHERE id = $1 LIMIT 1";  
    auto row = psql::Execute(trx, statement, request.id)[0];  
    if (!row["ok"].As<bool>()) {  
        LOG_DEBUG() << request.id << " is not OK for " << GetSomeInfoFromDb();  
        return Response400();  
    }  
  
    psql::Execute(trx, queries::kUpdateRules, request.foo, request.bar);  
    trx.Commit();  
  
    return Response200{row["baz"].As<std::string>()};  
}
```

ФиЧи

Фичи

PostgreSQL, Mongo,
Redis, Kafka,
RabbitMQ, ClickHouse[®],
MySQL, YDB

ФИЧИ

PostgreSQL, Mongo,
Redis, Kafka,
RabbitMQ, ClickHouse[®],
MySQL, YDB

TCP, UDP, TLS, gRPC,
HTTP 1.0, HTTP 1.1,
HTTP 2

ФиЧИ

PostgreSQL, Mongo,
Redis, Kafka,
RabbitMQ, ClickHouse®,
MySQL, YDB

TCP, UDP, TLS, gRPC,
HTTP 1.0, HTTP 1.1,
HTTP 2

Метрики, логи, трейсы

Функциональное
и юнит тестирование

Deadline Propagation,
Timeouts, Retries,
отмены

Производительность

Безопасность

ФиЧИ

PostgreSQL, Mongo,
Redis, Kafka,
RabbitMQ, ClickHouse[®],
MySQL, YDB

TCP, UDP, TLS, gRPC,
HTTP 1.0, HTTP 1.1,
HTTP 2

Метрики, логи, трейсы

Функциональное
и юнит тестирование

Deadline Propagation,
Timeouts, Retries,
отмены

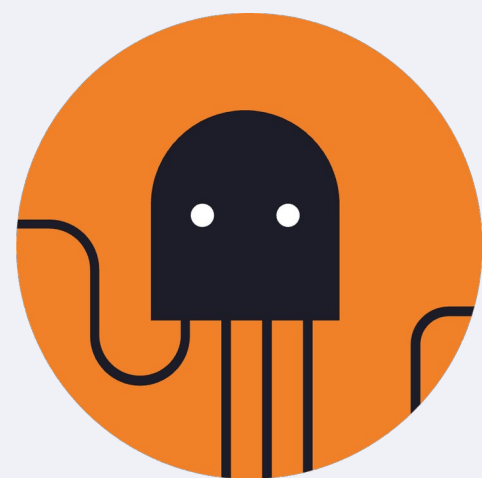
Производительность

Безопасность

[NonFiFO]MpMc Queue,
[NonFiFO]MpScQueue,
[NonFiFO]SpSc Queue,
RCU, Subscriptions,
Mutex, Semaphore,
Condition Variable,...

PeriodicTasks, Cache,
Cache dumps, LRU
caches, Expirable LRU
caches,
Dist Locks,...

JSON, BSON, YAML,...



userver.tech



Документация

1. Отрицание
2. Гнев
3. Торг
4. Депрессия
5. Чтение документации
6. Гнев

Чтобы документация меньше бесила

Чтобы документация меньше бесила

1

Теоретическое
введение

Чтобы документация меньше бесила

1

Теоретическое
введение

2

Примеры
и tutorиалы

Чтобы документация меньше бесила

1

Теоретическое
введение

2

Примеры
и tutorиалы

3

Reference

Внешние PR

...или как не погрязнуть в чтении
сырых PR и при этом не сломать прод

CI

На все PR должны запускаться тесты, чтобы автор PR замечал проблемы без вмешательства команды



CI

На все PR должны запускаться тесты, чтобы автор PR замечал проблемы без вмешательства команды



Защищённое программирование

Все инварианты ваших классов должны валидироваться и выдаваться не игнорируемая диагностика при их нарушении



CI

На все PR должны запускаться тесты, чтобы автор PR замечал проблемы без вмешательства команды



Тесты

Весь функционал должен быть обложен тестами и тесты должны быть публичной частью проекта



Защищённое программирование

Все инварианты ваших классов должны валидироваться и выдаваться не игнорируемая диагностика при их нарушении



CI

На все PR должны запускаться тесты, чтобы автор PR замечал проблемы без вмешательства команды



Тесты

Весь функционал должен быть обложен тестами и тесты должны быть публичной частью проекта



Защищённое программирование

Все инварианты ваших классов должны валидироваться и выдаваться не игнорируемая диагностика при их нарушении



Автоформатирование

Форматированием кода должна заниматься автоматика



Ревью

Автоматика не всё ловит!
Нужны люди, чтобы читать код.
Например — дежурный



Тикеты

Можно расписать
проблему/задачу и сразу
наметить путь её решения



Ревью

Автоматика не всё ловит!
Нужны люди, чтобы читать код.
Например — дежурный



Ревью

Автоматика не всё ловит!
Нужны люди, чтобы читать код.
Например — дежурный



Тикеты

Можно расписать
проблему/задачу и сразу
наметить путь её решения



Документация

Инструкция для разработчика
по сборке и тестированию



Чаты поддержки

Чтобы основное направление решения можно было обсудить ещё до его написания



Ревью

Автоматика не всё ловит!
Нужны люди, чтобы читать код.
Например — дежурный



Тикеты

Можно расписать проблему/задачу и сразу наметить путь её решения



Документация

Инструкция для разработчика по сборке и тестированию



Автоматика по принятию PR

...боремся с человеческим фактором

Итак, нам
принесли PR

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы
- Автор правит, улучшает, получается отличный результат

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы
- Автор правит, улучшает, получается отличный результат
- Дежурный запускает скрипт мержа:

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы
- Автор правит, улучшает, получается отличный результат
- Дежурный запускает скрипт мержа:
Формируется PR во внутреннюю репу с сохранением авторства и описания

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы
- Автор правит, улучшает, получается отличный результат
- Дежурный запускает скрипт мержа:
Формируется PR во внутреннюю репу
с сохранением авторства и описания
Код форматируется

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы
- Автор правит, улучшает, получается отличный результат
- Дежурный запускает скрипт мержа:
Формируется PR во внутреннюю репу
с сохранением авторства и описания
Код форматируется
Запускаются внутренние проверки
на всех сервисах и библиотеках Яндекса

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы
- Автор правит, улучшает, получается отличный результат
- Дежурный запускает скрипт мержа:
Формируется PR во внутреннюю репу
с сохранением авторства и описания
Код форматируется
Запускаются внутренние проверки
на всех сервисах и библиотеках Яндекса
Merge

Итак, нам принесли PR

- CI запускается и подсвечивает проблемы, автор правит и улучшает
- Приходит ревьюер из команды и подсвечивает другие проблемы
- Автор правит, улучшает, получается отличный результат
- Дежурный запускает скрипт мержа:
Формируется PR во внутреннюю репу с сохранением авторства и описания
Код форматируется
Запускаются внутренние проверки на всех сервисах и библиотеках Яндекса
Merge
Автоматика синхронизирует изменения наружу и закрывает изначальный PR

Issue

...когда задач больше чем людей в компании

Всё сложно...

Всё сложно...

Bugs:

Исправляются
оперативно, по мере
сил и возможностей



Всё сложно...

Внешние тикеты для энтузиастов:

Размечаются
как help-wanted и могут
длительное время
быть открытыми



Bugs:

Исправляются
оперативно, по мере
сил и возможностей



Всё сложно...

Внешние тикеты для энтузиастов:

Размечаются
как help-wanted и могут
длительное время
быть открытыми



Bugs:

Исправляются
оперативно, по мере
сил и возможностей



Feature requests:

Всё зависит
от сложности фичи
и её востребованности



На примере Websockets

...когда всё сложилось отлично

Websockets

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
- Появился контрибьютор и принёс PR

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
- Появился контрибьютор и принёс PR
- Поняли что не так уж и страшно, творчески доработали и вмержили

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
- Появился контрибьютор и принёс PR
- Поняли что не так уж и страшно, творчески доработали и вмержили
- Появился Bugreport + PR: быстро вмержили, слегка доработав для производительности

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
- Появился контрибьютор и принёс PR
- Поняли что не так уж и страшно, творчески доработали и вмержили
- Появился Bugreport + PR: быстро вмержили, слегка доработав для производительности
- Появился Feature request на новый способ использования + PR

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
- Появился контрибьютор и принёс PR
- Поняли что не так уж и страшно, творчески доработали и вмержили
- Появился Bugreport + PR: быстро вмержили, слегка доработав для производительности
- Появился Feature request на новый способ использования + PR

Выдали фидбек, запросили улучшения PR

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
- Появился контрибьютор и принёс PR
- Поняли что не так уж и страшно, творчески доработали и вмержили
- Появился Bugreport + PR: быстро вмержили, слегка доработав для производительности
- Появился Feature request на новый способ использования + PR

Выдали фидбек, запросили улучшения PR

Автор поправил PR

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
- Появился контрибьютор и принёс PR
- Поняли что не так уж и страшно, творчески доработали и вмержили
- Появился Bugreport + PR: быстро вмержили, слегка доработав для производительности
- Появился Feature request на новый способ использования + PR

Выдали фидбек, запросили улучшения PR

Автор поправил PR

Вмержили, всем на радость

Websockets

- Изначальный Issue выглядел как большой и сложный Feature request
 - Появился контрибьютор и принёс PR
 - Поняли что не так уж и страшно, творчески доработали и вмержили
 - Появился Bugreport + PR: быстро вмержили, слегка доработав для производительности
 - Появился Feature request на новый способ использования + PR
- Выдали фидбек, запросили улучшения PR
- Автор поправил PR
- Вмержили, всем на радость
- Завели issue на доработки с Help-wanted

Релизы

...когда всё зелёное

Когда
и как релизиться

Когда и как релизиться

- Пробовали релизиться пару раз в год — получилось так себе

Когда и как релизиться

- Пробовали релизиться пару раз в год — получилось так себе
- Сделали релизы раз в 1–2 месяца + большой релиз пару раз в год — сильно лучше

А что во время
релиза происходит?

А что во время релиза происходит?

- Пишем Changelog

А что во время релиза происходит?

- Пишем Changelog
- Инкрементим версию

А что во время релиза происходит?

- Пишем Changelog
- Инкрементим версию
- Оформляем release на GitHub

А что во время релиза происходит?

- Пишем Changelog
- Инкрементим версию
- Оформляем release на GitHub
 - Автоматика собирает Docker-контейнеры
 - Собираем Deb-пакет
 - Публикуем Deb-пакет на GitHub
 - Делаем анонсы в чаты

А что во время релиза происходит?

- Пишем Changelog
- Инкрементим версию
- Оформляем release на GitHub
 - Автоматика собирает Docker-контейнеры
 - Собираем Deb-пакет
 - Публикуем Deb-пакет на GitHub
 - Делаем анонсы в чаты
- При большом релизе — то же самое, но ещё пишем статьи о изменениях с прошлого большого релиза

Смежники

...мы все делаем общее дело

~~Смежники~~ Другие проекты с которыми мы взаимодействуем

- GCC и Clang
- Boost
- Doxygen
- OpenSSL, Mongo
- C++ Standard
- Seastar
- YDB
- Protobuf/gRPC

Студенты

...и дипломные по userver

Дипломы —
это хорошо,
но тяжело

Дипломы —
это хорошо,
но тяжело

- Простая задача! =
легко менторить

Дипломы —
это хорошо,
но тяжело

- Простая задача! = легко менторить
- Сложная задача → сложное внедрение в проде

А что в итоге

А в итоге — нам нравится :)

А в итоге — нам нравится :)

Приятно видеть что проект
востребован за пределами
Яндекса



А в итоге — нам нравится :)

Приятно видеть что проект
востребован за пределами
Яндекса



Приятно что проектом
интересуются и предлагают
правки



А в итоге — нам нравится :)

Приятно видеть что проект
востребован за пределами
Яндекса



Приятно что проектом
интересуются и предлагают
правки



Да, ресурсоёмко. Но можно
подсократить затраты с помощью
автоматики и документации



А в итоге — нам нравится :)

Приятно видеть что проект востребован за пределами Яндекса



Приятно что проектом интересуются и предлагают правки



Да, ресурсоёмко. Но можно подсократить затраты с помощью автоматики и документации



Есть ощущение, что участвуем в чём-то большом и всеобъемлющем



А в итоге — нам нравится :)

Приятно видеть что проект востребован за пределами Яндекса



Приятно что проектом интересуются и предлагают правки



Да, ресурсоёмко. Но можно подсократить затраты с помощью автоматики и документации



Есть ощущение, что участвуем в чём-то большом и всеобъемлющем, в **Open Source**



Спасибо за внимание!

Антон Полухин,
эксперт разработчик C++, Яндекс

YANDEX OPEN SOURCE JAM

 Open Source