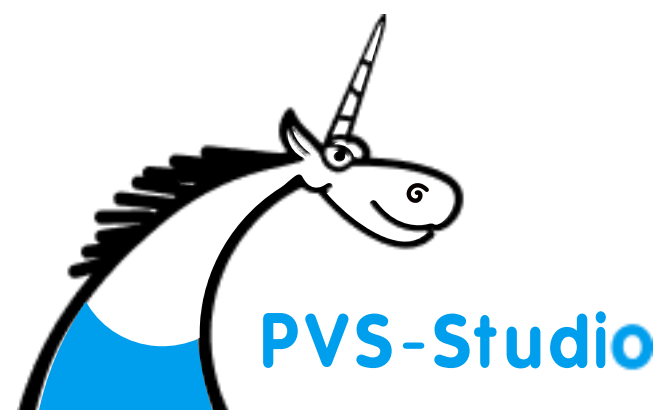


std::string

путешествие туда и обратно



Андрей Карпов
CBDO, ООО «ПВС»



Андрей Карпов

Директор по развитию бизнеса (CBDO)

- Один из основателей проекта PVS-Studio
<https://pvs-studio.ru>
- 18 лет в сфере качества и анализа кода
- Хабр: [@Andrey2008](https://habr.com/ua/user/Andrey2008)
- Telegram канал
«Бестиарий программирования»
t.me/programming_tales



PVS-Studio

Как основатель, я был там

3



- Статический анализатор кода для поиска багов и потенциальных уязвимостей
- Ядро C++ анализатора написано на C++
- Проблематика: **нет отдельных узких мест для оптимизации**
- Время равномерно размазано по разным алгоритмам сбора информации и по сотням детекторов
- Но одно место 15 лет назад было – **строки!**



- Очень много пустых строк
- Очень много коротких строк для хранения типов объектов
 - Тип `int` – `i`
 - Тип указатель на `const unsigned float` – **PCUf**
 - Функция, принимающая два `char` и возвращающая `bool`: **Fcc_b**
- Ещё особенность – эти строки почти никогда не удаляются, т.к. могут требоваться до самого конца анализа дерева

Свой класс строки - это норма

6

- Рано или поздно в любом состоявшемся проекте появляется свой класс строки
- Я ждал, появится ли он в PVS-Studio
- Он появился, и это было обосновано
- Это нормально. Не стесняйтесь это делать



Из моего доклада на C++ Russia 2016

Оптимизация (времен где-то 2010 года)

7

- Здесь уместна SOO
 - Small String Optimization
 - или Short String Optimization

- До неё я не догадался



- Но свой класс строки **vstring** появился (от viva64 **string**)
- По интерфейсу – он почти полный аналог `std::string`

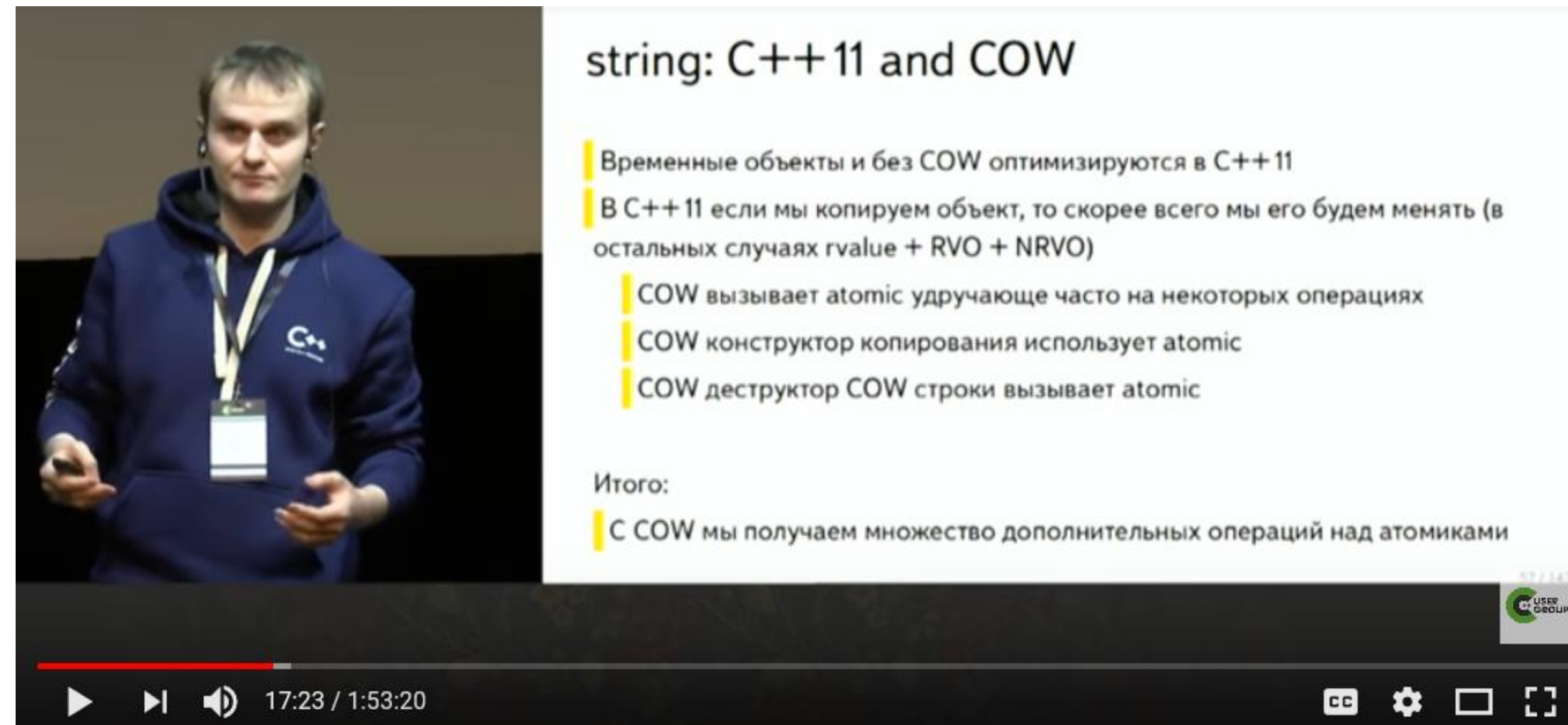
1. Сделал один экземпляр буфера для пустой строки
 2. Сделал свой менеджер выделения памяти для коротких строк
 3. Работает быстро за счёт того, что память не освобождается до конца работы приложения
- **Итог: прирост скорости около 10% !**



В 2017 году я еду на очередную конференцию C++Russia, а там

9

- Антон Полухин, Как делать не надо: C++ велосипедостроение для профессионалов
<https://youtu.be/rJWSSWYL83U>



1. Появилось много новых оптимизаций
 2. Нет смысла создавать свои велосипеды строки
- Я узнаю про **SSO** и главное – понимаю, что **это наш случай**



SSO - Small String Optimization

11

```
class string
{
    size_t capacity;

    union
    {
        struct
        {
            char *ptr;
            size_t size;
        } heapbuf;

        char stackbuf[sizeof(heapbuf)];
    };
};
```



[Подробнее](#)

- **`typedef std::string vstring;`**
 - Ну почти. Конечно, были ещё мелкие правки
1. **Скорость работы не меняется!**
 2. Теоретически меньшее потребление памяти, так как отдельные строки исчезают до работы программы. Но их мало, так что замеры по памяти я даже не делал

Вот такое путешествие туда и обратно

13

- Стандартная библиотека была значительно улучшена и оптимизирована за последнее 10-15 лет
- Велосипеды могут быть уместны
- Важно не забывать их удалять, когда они больше не нужны и только увеличивают «степень легаси»



Остались вопросы



- Предыдущий вебинар «[Оптимизация игр](#)»
- PVS-Studio
 - [об инструменте](#)
 - [скачать и попробовать](#)
 - [совместимость](#) с ГОСТ Р 71207-2024
 - [№9837](#) – запись в едином реестре российского ПО



Карпов Андрей Николаевич

16

- Карпов Андрей Николаевич, 1981
- ООО «ПВС», директор по развитию бизнеса
- Более 18 лет занимается темой статического анализа кода и качества программного обеспечения. Автор большого количества статей, посвящённых написанию качественного кода на языке C++. Один из основателей проекта PVS-Studio. Долгое время являлся СТО компании и занимался разработкой C++ ядра анализатора. Основная деятельность на данный момент — развитие компании, обучение сотрудников и DevRel деятельность
- [Другая информация и контакты](#)

