

Анализатор динамических аллокаций в C++

ВЫПОЛНИЛ: ДОРОНИН ОЛЕГ

РУКОВОДИТЕЛЬ: Д.Ф.М.Н ШПОЛЯНСКИЙ ЮРИЙ
АЛЕКСАНДРОВИЧ

Цель

Разработать проект для анализа динамических аллокаций в C++ с учетом бакетов выделяемых аллокатором, а так же визуализация полученных данных

Tbricks by Itiviti

«Buy and build with apps. Customize to fit. Trade in any style.»

Tbricks by Itiviti is a trading platform designed for today's reality; fusing flexible, scalable and co-located trading with a blazingly fast core system. Banks and trading firms rely on Tbricks to automate sophisticated strategies on diverse venues across the globe.»

Задачи

- Простой анализ памяти (sizeof / jemalloc)
- Рекурсивный анализ динамических аллокаций включая поля классов и структур
- Визуализация полученных данных
- Тестирование

Мотивация

- Расчет необходимой оперативной памяти
- Оптимизации потребления памяти

Разработка приложения

libclang

Плюсы:

- Удобный API
- Используется компилятором clang

Минусы:

- Наличие ошибок в библиотеке
- Проблемы с export в последних версиях

Наличие ошибок в библиотеке

```
CXCursor clang_getTypeDeclaration(CXType CT) {  
...  
// FIXME: Template type parameters!  
...  
}
```

Простой анализ памяти

InstrumentPricing

----- Simple size -----

sizeof = **4.3** KiB (**4408** bytes)

jemalloc = **5.0** KiB (**5120** bytes)

diff = **712** bytes (**712** bytes)

Рекурсивный анализ динамических аллокаций включая поля классов и структур

struct A

```
{  
    int *a;  
};
```

struct B

```
{  
    double *b;  
    A a;  
};
```

Рекурсивный анализ динамических аллокаций включая поля классов и структур

InstrumentPricing

----- Simple size -----

sizeof = 4.3 KiB (4408 bytes)

jemalloc = 5.0 KiB (5120 bytes)

diff = 712 bytes (712 bytes)

----- Size with pointer allocations -----

sizeof = 4.3 KiB + 6.5 KiB = 10.8 KiB (11080 bytes)

jemalloc = 5.0 KiB + 7.4 KiB = 12.4 KiB (12672 bytes)

diff = 1.6 KiB (1592 bytes)

----- Size with dfs pointer allocations -----

sizeof = 4.3 KiB + 667.3 KiB = 671.6 KiB (687745 bytes)

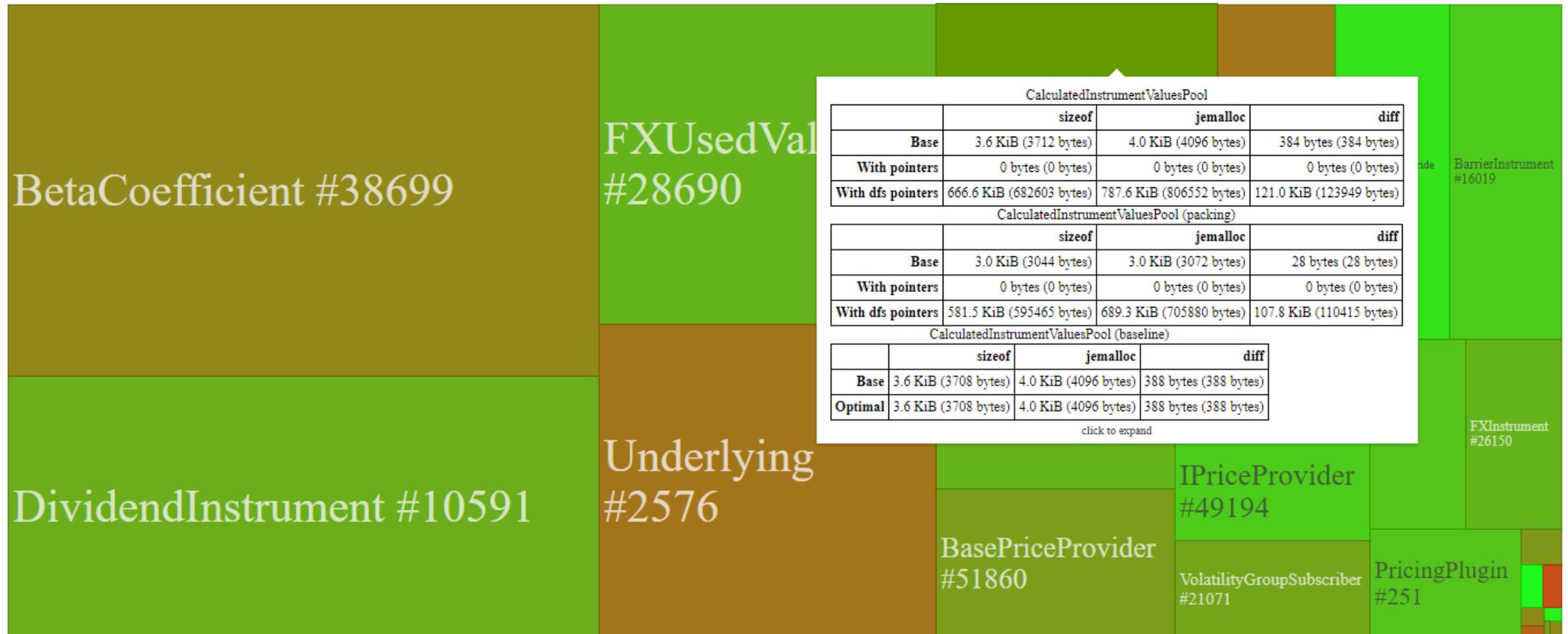
jemalloc = 5.0 KiB + 788.2 KiB = 793.2 KiB (812224 bytes)

diff = 121.6 KiB (124479 bytes)

Визуализация полученных данных Google treemap vs D3plus

< back

InstrumentPricing #1



Конфигурационный файл

Возможности:

- Указывать количества элементов определенного поля
- Учитывать ссылку как указатель
- Задавать количество создаваемых элементов определенного типа вместо явного поля

Тестирование

Тесты для проверки:

- Базовой функциональности
- Работоспособности конфигурационного файла
- Рекурсивной обработки класса или структуры

Заключение

- Создана утилита для простого анализа памяти (sizeof / jemalloc)
- Добавлена функциональность для проведения рекурсивного анализа динамических аллокаций включая поля классов и структур.
- Добавлена возможность для явного указания количества создаваемых объектов
- Реализована визуализация дерева классов и структур
- Разработаны тесты

Спасибо за внимание!