

Эффективная реализация сопрограмм в управляемой среде исполнения

Евгений Пантелеев

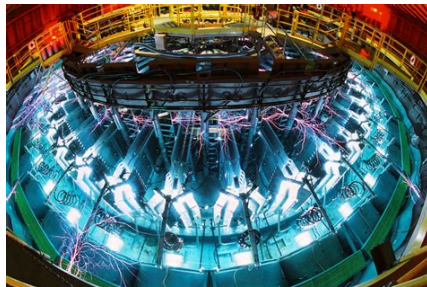
Новосибирский государственный университет

Научный руководитель: Бульонков Михаил Алексеевич, канд. физ-мат наук
ИСИ СО РАН

Новосибирск
2021г.



(a) Серверы.



(b) Ускорители.

Существует множество задач, в которых необходимо обрабатывать много независимых событий.

Сопрограммы

- ▶ **Сопрограмма** (с англ. coroutine) - программный модуль, организованный для обеспечения взаимодействия с другими модулями по принципу кооперативной многозадачности.
- ▶ Сопрограммы способны приостанавливать свое выполнение, сохраняя *контекст* (программный стек и регистры), и передавать управление другой.

Ключевые отличия от потоков ОС

Плюсы сопрограмм

- ▶ Переключение контекста сопрограммы требует меньше накладных расходов, чем потока.
- ▶ Как правило меньший размер стека, а значит, потребление памяти так же меньше.

Минусы

- ▶ Сопрограммы не способны исполняться параллельно.

Поддержка в языках программирования



(a) C++20.



(b) C#.



(c) Go.

Project Loom

Fibers and Continuations



- ▶ Project Loom – проект на базе OpenJDK, целью которого является разработка сопрограмм для языка Java.
- ▶ На данный момент уже доступна ранняя версия проекта.

Цели и задачи.

Цель: реализация прототипа сопрограмм в Java.

Поставленные задачи:

- ▶ Разработать тесты для сравнения производительности потоков и сопрограмм.
- ▶ Реализовать переключение сопрограмм.
- ▶ Поддержать трассировку ссылок объектов на стеках сопрограмм(???)
- ▶ Сравнить производительности сопрограмм и потоков.

Работа проводится на базе Huawei JDK.

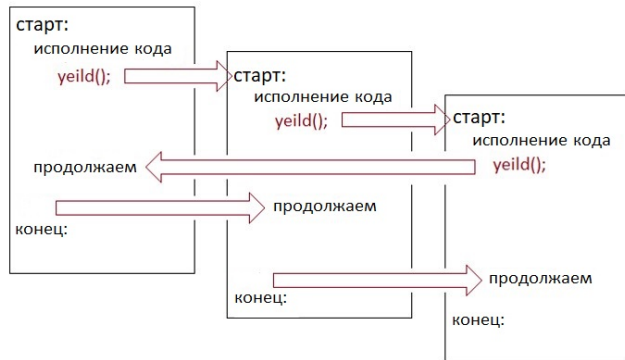
Тесты производительности

Тесты создавались для измерения 2 параметров.

- ▶ Скорость переключения контекста.
- ▶ Потребление памяти.

Репозиторий с тестами: <https://github.com/minium2/coroutines-benchmark>

Переключение сопрограмм.



Подходы к реализации:

- ▶ OpenJDK: копирование стека сопрограммы при переключении.
- ▶ HuaweiJDK : изменение указателя стека.

Сборка мусора

Что-то про сборку мусора

Результаты

- ▶ Создан набор тестов для сравнения производительности потоков и сопрогромаами.
- ▶ Реализовано переключение контекста сопрограмм.
- ▶ Разработана трассировка ссылок объектов на стеках сопрограмм(???).
- ▶ Получены результаты тестов производительности.

Результаты: скорости переключения

Ubuntu, x64, 31 Гб ОЗУ

Каждое значение усреднено по 100 измерениям. HuaweiJDK

Сопрограмм, шт.	Число переключений, 1/сек.	
	Сопрограммы	Потоки
100	0	0 (-/+ 0)
1000	0	0 (-/+ 0)
5000	909'148 (-/+ 59'328)	1'553'872 (-/+ 36'832)
10000	0	0 (-/+ 0)
50000	749'523 (-/+ 6118)	361'088 (-/+ 7'853)

Результаты: потребление памяти

Ubuntu, x64, 31 Гб ОЗУ, HuaweiJDK

Число сопрограмм, шт.	Размер физической памяти.	
	Сопрограммы	Потоки
100	34M	0
1000	38M	0
5000	59M	0
10000	85M	0
50000	107M	0

Применение сопрограмм.

- ▶ Реализация бесконечных списков, итераторов, генераторов.
- ▶ Написание асинхронного и неблокирующего кода(???)
- ▶ Обработка независимых друг от друга событий.

План дальнейших работ

- ▶ Переделать функцию переключения контекста.
- ▶ Реализация возможности миграции сопрограмм с одного потока на другой(???).
- ▶ Синхронизация: поддержка `synchronized` блоков(???).
- ▶ Переключение сопрограммы при вызове ввода вывода.

Спасибо за внимание!