Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

ИСПОЛЬЗОВАНИЕ СОПРОГРАММ ДЛЯ СОЗДАНИЯ АСИНХРОННОГО СОКЕТ СЕРВЕРА НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ KOTLIN

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 «Математическое обеспечение и администрирование информационных систем»
факультета компьютерных наук и информационных технологий
Черногорова Владислава Максимовича

Научный руководитель:
кандидат технических наук, доктор
физико-математических наук,
профессор по кафедре МКиКН Д. К. Андрейченко

Зав. кафедрой
кандидат технических наук, доктор
физико-математических наук,
профессор по кафедре МКиКН Д. К. Андрейченко

Саратов 2018

ВВЕДЕНИЕ

Актуальность темы. Тема данной работы состоит в сравнении текущих устоявшихся практик написания серверной логики используя многопоточность с асинхронным сервером, написанным на корутинах. Эта тема достаточно актуальна на данный момент, хоть и уходит своими корнями в далекие шестидесятые, где термин сопрограмм применялся в программах, написанных еще на ассемблере.

Актуальность заключается в необходимости написания асинхронного кода в случаях, когда в среде не может быть больше одного потока (как например в клиентском браузере), когда потоков становится слишком много для одного процесса или же в случае необходимости производить множество операций ввода/вывода с минимальным ожиданием (например для логирования).

Вышесказанное определило ***цель бакалаврской работы***: …

Поставленная цель определила следующие ***задачи***:

- Изучить библиотеку сопрограмм, написанную на Kotlin.
- Построить асинхронный сокет сервер используя технологию сопрограмм.
- Провести замеры производительности сервера, написанного с и без использования сопрограмм.

Методологические основы разработки серверов на языке Kotlin с использованием сопрограмм представлены в работах М. Ахина [23], А. Собчука [22], и выступлениях на конференциях Р. Элизарова [8] и А. Бреслава [7].

Теоретическая значимость бакалаврской работы. Были проведены исследования стандартной библиотеки сопрограмм и методологий разработки веб-серверов с помощью веб-сокетов. В ходе этих исследований

были выведены основные преимущества сопрограмм, которые стоит рассмотреть в практической части, а также появилось четкое видение архитектуры асинхронного веб-сокет сервера. Также был разобран стиль программирования с помощью продолжений (англ. Continuation-Passing Style), используемый в реализации сопрограмм в языке Kotlin.

Практическая значимость бакалаврской работы. В ходе выполнения практической части было создано веб-приложение с трехслойной архитектурой на языке Kotlin. Оно было протестировано с помощью инструмента для нагрузочного тестирования — Artillery.io. Результаты нескольких тестирований были записаны из логов инструмента, а также с помощью инструмента для мониторинга и профилирования Java приложений — VisualVM. После этого участки кода, содержащие сопрограммы, были заменены на классы из стандартной Java-библиотеки для организации многопоточности, и проведено повторное нагрузочное тестирование с теми же параметрами.

Результаты этих измерений показали, что сопрограммы действительно являются эффективнее в плане потребляемой памяти Java Heap’а и выигрывают в скорости работы у потоков. Программа, написанная с использованием сопрограмм, проходила нагрузочное тестирование на 4-8% быстрее программы без них.

Структура и объём работы. Магистерская работа состоит из введения, двух разделов, заключения, списка использованных источников и одного приложения. Общий объем работы – 69 страниц, из них 59 страницы – основное содержание, включая 9 рисунков, цифровой носитель в качестве приложения, список использованных источников информации – 23 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретическая часть» посвящен разбору отличий синхронной, асинхронной и многопоточной программной модели. Далее будет описываться сама концепция сопрограмм, представленная еще в прошлом веке, а затем конкретная их реализация на языке программирования Kotlin. Будут представлены основные элементы библиотеки сопрограмм с примерами их использования. Разобрав поверхностные классы и функции библиотеки, в разделе 1.4 будет предоставлено описание “внутреннего” устройства сопрограмм, т.е. их реализация на уровне компиляции.

Второй раздел «Практическая часть» посвящен созданию тестового приложения для имитации высоконагруженной браузерной игры с единственным серверным узлом. Цель этого приложения — сравнить производительность кода, написанного на сопрограммах, с кодом, написанным на системных потоках. Не смотря на это, было решено создать минимальную реализацию интерфейса приложения для более удобного ручного тестирования кода во время разработки, а также для усложнения логики запросов.

ЗАКЛЮЧЕНИЕ

Подводя итог данной работы, и опираясь на выводы многочисленных измерений состояния асинхронного сокет сервера, можно сказать, что сопрограммы действительно дают небольшой прирост в производительности. Однако в данной работе был рассмотрен конкретный пример использования сопрограмм, что не гарантирует их глобальное преимущество перед потоками.

Сопрограммы в языке Kotlin, как утверждают сами разработчики языка, лучше всего себя проявляют в зависимых коротких асинхронных вычислениях, нежели в долговременных вычислениях с общеразделяемыми ресурсами. Если вы решили использовать сопрограммы вместе потоков в своем коде, то это вряд ли приведет к росту производительности, а может даже наоборот — ухудшит состояние программы. Такой механизм лучше всего совмещать с многопоточной архитектурой, либо использовать в однопоточных средах, таких как Android приложение или окно браузера.

Также по мере разработки проекта, нельзя не отметить простоту написания сопрограмм, а именно — добавление сопрограммы в синхронный код не требует больших затрат на переделывание контрактов метода или класса, что само собой является существенным плюсом для разработчика. Эта идея также закладывалась в библиотеку корутин, как одна из основных особенностей, ради которой стоит их использовать. Преобразование последовательного на первый взгляд кода в CPS код происходит почти незаметно на этапе компиляции, а промежуточные объекты, создаваемые во время выполнения приложения занимают намного меньше места, чем тяжеловесные системные потоки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Pierre-Yves Saumont. The Joy of Kotlin. // Manning Publications Co, 2018. ISBN-13 978-1617295362. (Яз. англ.)
2. Advanced Kotlin Programming by Hadi Hariri. // [Электронный ресурс] : The world's most comprehensive technology and business learning platform. URL: https://www.safaribooksonline.com/library/view/advanced-kotlin-programming/9781491964149/video283085.html (дата обращения: 09.03.2018) (Яз. англ.)
3. Оф. документация по сопрограммам. // [Электронный ресурс] : Kotlin Programming Language. URL: https://kotlinlang.org/docs/reference/coroutines.html (дата обращения: 09.03.2018) (Яз. англ.)
4. Jemerov D., Isakova S. Kotlin in Action. // Manning Publications Co, 2017. ISBN-13 978-1617293290. (Яз. англ.)
5. Core primitives to work with coroutines. // [Электронный ресурс] : kotlin-coroutines-core. URL: https://kotlin.github.io/kotlinx.coroutines/kotlinx-coroutines-core/index.html (дата обращения: 09.03.2018) (Яз. англ.)
6. Kotlin Coroutines Informal. // [Электронный ресурс] : GitHub. URL: https://github.com/Kotlin/kotlin-coroutines/blob/master/kotlin-coroutines-informal.md (дата обращения: 09.03.2018) (Яз. англ.)
7. Андрей Бреслав — Асинхронно, но понятно. Сопрограммы в Kotlin. // [Электронный ресурс] : YouTube. URL: https://youtu.be/ffIVVWHpups (дата обращения: 09.03.2018) (Яз. рус.)

8. KotlinConf 2017 - Deep Dives into Coroutines on JVM by Roman Elizarov. // [Электронный ресурс] : YouTube. URL: https://youtu.be/YrrUCSi72E8 (дата обращения: 09.03.2018) (Яз. англ.)
9. Approaching Kotlin Coroutines — Concurrent Programming in Kotlin. // [Электронный ресурс] : ProAndroidDev. URL: https://proandroiddev.com/approaching-kotlin-coroutines-an-extensive-feature-concurrent-programming-in-kotlin-eaaa19b003d2 (дата обращения: 09.03.2018) (Яз. англ.)
10. Параллелизм против многопоточности против асинхронного программирования: разъяснение. // [Электронный ресурс] : Хабр. URL: https://habrahabr.ru/post/337528/ (дата обращения: 11.03.2018) (Яз. рус.)
11. Как избавиться от пристрастия к синхронности. // [Электронный ресурс] : Хабр. URL: https://habrahabr.ru/post/97042/ (дата обращения: 17.03.2018) (Яз. рус.)
12. Корутины в Kotlin. // [Электронный ресурс] : Хабр. URL: https://habr.com/company/alfa/blog/336228/ (дата обращения: 17.03.2018) (Яз. рус.)
13. Кнут Д. Искусство программирования, том 1. Основные алгоритмы — 3-е изд. // «Вильямс», 2006. ISBN 0-201-89683-4. (Яз. рус.)
14. Асинхронное программирование с использованием ключевых слов Async и Await (C# и Visual Basic) // [Электронный ресурс] : Learn to Develop with Microsoft Developer Network | MSDN. URL: https://msdn.microsoft.com/ru-ru/library/hh191443(v=vs.120).aspx (дата обращения: 17.03.2018) (Яз. рус.)
15. Programming in Lua : 9.1. // [Электронный ресурс] : The Programming Language Lua. URL: https://www.lua.org/pil/9.1.html (дата обращения: 17.03.2018) (Яз. англ.)

16. 18.5.3 Tasks and Coroutines. // [Электронный ресурс] : Python 3.6.5 documentation. URL: https://docs.python.org/3/library/asyncio-task.html (дата обращения: 17.03.2018) (Яз. англ.)
17. Kotlin Bytecode for the JVM — Bytecode Generation. // [Электронный ресурс] : Kotlin Expertise Blog — Professional Kotlin Development: https://kotlinexpertise.com/kotlin-byte-code-generation/ (дата обращения: 18.03.2018) (Яз. англ.)
18. When to Use Asynchronous Programming. // [Электронный ресурс] : Software Development Tools for Diagnosing App Performance Issues. URL: https://stackify.com/when-to-use-asynchronous-programming/ (дата обращения: 18.03.2018) (Яз. англ.)
19. Kotlin Coroutines, a deeper look. // [Электронный ресурс] : Medium. URL: https://medium.com/@elizarov/kotlin-coroutines-a-deeper-look-180536305c3f (дата обращения: 22.03.2018) (Яз. англ.)
20. Diving deep into Kotlin Coroutines. // [Электронный ресурс] : Kotlin Development. URL: https://www.kotlindevelopment.com/deep-dive-coroutines/ (дата обращения: 22.03.2018) (Яз. англ.)
21. Лекция 8. Корутины, часть 2. // [Электронный ресурс] : Computer Science Center. URL: https://compscicenter.ru/courses/kotlinprogramming/nsk/2018-spring/classes/3933/ (дата обращения: 18.03.2018) (Яз. рус.)
22. Продолжение всемирной паутины. // [Электронный ресурс] : Smalltalk по-русски. URL: http://www.smalltalk.ru/articles/web-continuations.html (дата обращения: 22.03.2018) (Яз. англ.)

23. Kotlin DSL: Теория и Практика. // [Электронный ресурс] : Хабр. URL: https://habr.com/company/haulmont/blog/341402/ (дата обращения: 22.03.2018) (Яз. рус.)