Сравнительный анализ и реализация различных конкурентных паттернов для работы с асинхронным вводом и выводом с использованием kotlin корутин

Компания: JetBrains d.o.o. Beograd

Уровень: бакалаврская

Описание предметной области
(Конкретная информация о текущем состоянии дел в предметной области, в которой выполняется работа: проблематика с точки зрения бизнеса, технологический ландшафт, приоритеты и вызовы в данной области, мотивирующие данную работу)

Проблема C10k, заключающаяся в сложности одновременной обработки сервером большого количества соединений, была обозначена еще в конце 1990-х, но является актуальной и на сегодняшний день. В процессе создания высоконагруженных серверов применялись различные стратегии: от блокирующего подхода до асинхронной работы с серверными сокетами. К примеру, NGINX, разработанный на асинхронной архитектуре, был создан как раз для решения проблемы C10k. Были разработаны и различные конкурентные паттерны, такие как proactor и reactor.
В фреймворке Ktor, для которого и пишется дипломная работа, одной из важных частей является полностью асинхронный движок CIO на основе kotlin корутин, в котором для работы с серверными сокетами используется свой компонент.

Актуальность:
(ответ на вопрос - почему данную работу нужно выполнить прямо сейчас, в текущий момент времени. Важно: обоснование “потому что еще никто не пробовал” не подходит, нужно также раскрыть аспект “почему прямо сейчас от отсутствия решения кто-то страдает”)

В настоящее время у пользователей Ktor усиливается потребность в написании высоконагруженных серверов, способных обрабатывать множество одновременных соединений. Для того чтобы оставаться конкурентоспособными и улучшать пользовательский опыт, нужно развивать и совершенствовать существующие решения и находить новые способы работы с асинхронным вводом и выводом. К тому же одним из фокусов в Ktor является собственный написанный мультиплатформенный движок CIO, для которого и будет создаваться решение. Kotlin сейчас делает большой акцент на мультиплатформенную разработку и поэтому официальные библиотеки во главе с Ktor также ставят это одной из главных своих задач.

Цель работы:
(положительный эффект, который будет достигнут после внедрения результата работы, что планируется улучшить, ускорить, усовершенствовать, оптимизировать)

Реализация и улучшение компонента библиотеки Ktor для работы с асинхронным вводом и выводом. Планируется улучшение отзывчивости серверных приложений, сокращение времени ожидания ответов от сервера и обеспечение высокой производительности в условиях интенсивного ввода-вывода.

Что следует сделать:

- Реализовать сервера с различными паттернами для работы с асинхронным вводом и выводом: блокирующие, неблокирующие, асинхронные
- Разработать несколько сценариев работы клиента и сервера: один клиент и один сервер, 1000 клиентов и один сервер
- Измерить характеристики (пропускная способность, время отклика) серверов в сценариях работы
- Оценить проблемные места и сравнить паттерны используемые в серверах
- Внедрить итоговый паттерн в модуль Ktor

Ожидаемые результаты: внедренный в модуль Ktor компонент для работы с асинхронным вводом и выводом с использованием kotlin корутин

Способ верификации результатов:
(подход к проверке достижения артефактами, перечисленными в “Ожидаемых результатах” целей, перечисленных в “Цели работы” - методики тестирования, тестовые стенды, источники тестовых данных, способы проверки результатов на полноту, методики подтверждения улучшений в предметной области с внедрением решения.)

Нагрузочное тестирование и сравнительный анализ метрик, таких как пропускная способность и время отклика, после реализации каждого паттерна, с учетом различных сценариев клиент-серверного взаимодействия

Источники:
C10k problem http://www.kegel.com/c10k.html
Reactor https://www.dre.vanderbilt.edu/~schmidt/PDF/reactor-siemens.pdf
Ktor https://ktor.io/
Comparing Two High-Performance I/O Design Patterns
https://www.artima.com/articles/comparing-two-high-performance-io-design-patterns
Wrk https://github.com/wg/wrk