

Застосування багатопотокових моделей взірців проектування для створення масштабованого програмного забезпечення

Виконав: Божко В. В., студент 2 р.н., ІПЗ

Керівник: Бублик В. В., кандидат фіз.-мат. наук, доцент

НаУКМА – 2023

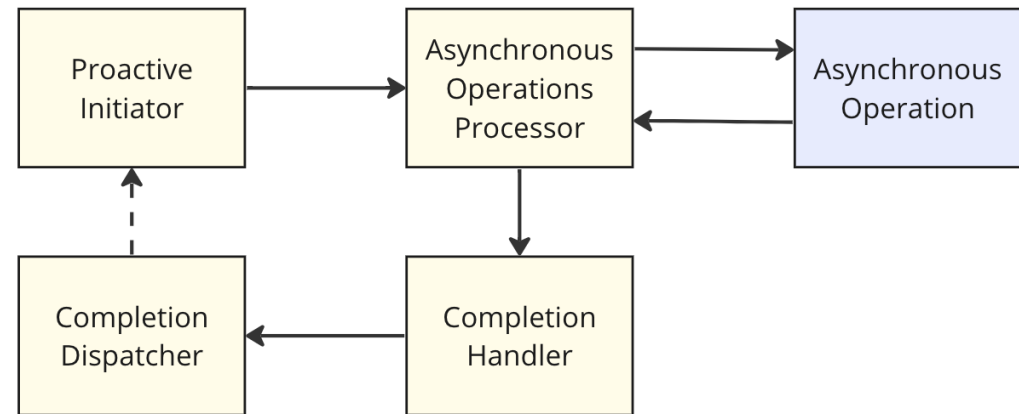


Мета і завдання роботи

- **Мета** роботи – виявити, яким чином асинхронні моделі взірців проектування можуть бути застосовані у масштабованому програмному забезпеченні
- **Завдання** роботи:
 - Дослідити існуючі взірці проектування масштабованого ПЗ
 - Проаналізувати вплив класичних взірців проектування та їх багатопотокових моделей на швидкодію масштабованого ПЗ
 - Розробити власну бібліотеку для комунікації між компонентами масштабованого ПЗ та на її основі змодельовати реалізацію одного з існуючих взірців проектування
 - Порівняти отриману реалізацію із існуючими рішеннями

Взірець «Проактор»

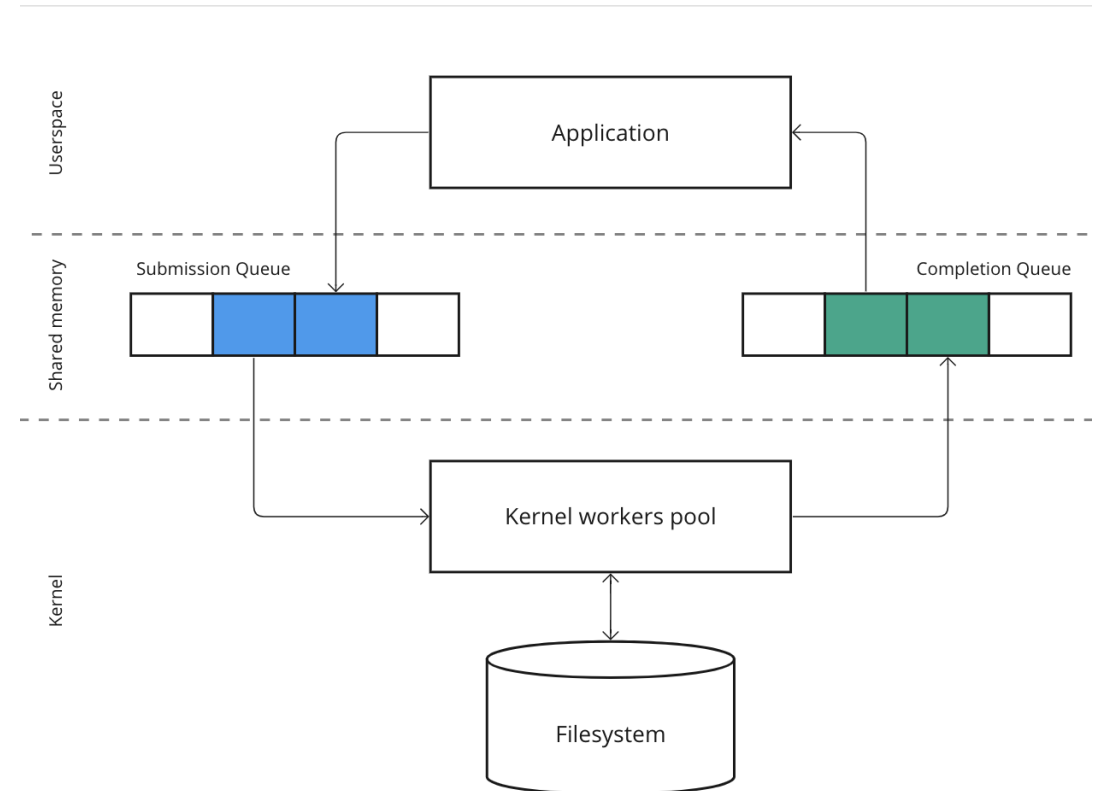
- Візірець проектування для виконання та обробки асинхронних операцій
- Дозволяє структурувати та об'єднати обробку багатьох асинхронних подій



Принципова схема візця «Проактор»

Новий механізм введення-виведення у Linux: io_uring

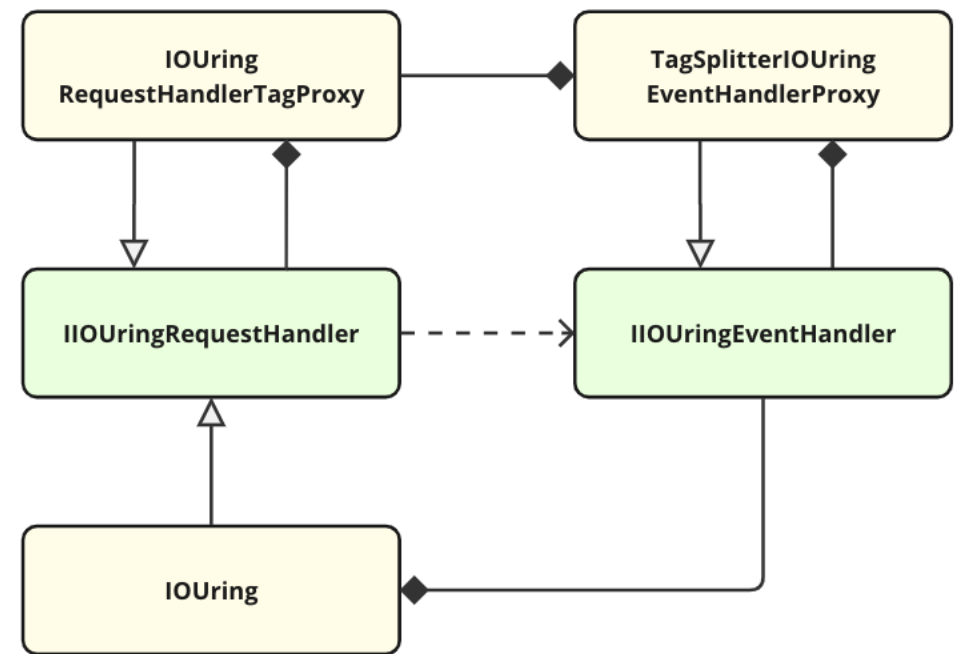
- Включений до складу ядра Linux у 2019 році і досі розробляється
- Мінімізує к-ть системних викликів
- Не потребує додаткових механізмів синхронізації



Принципова схема роботи io_uring

Об'єктно-орієнтована абстракція над `io_uring` – `IOUring`

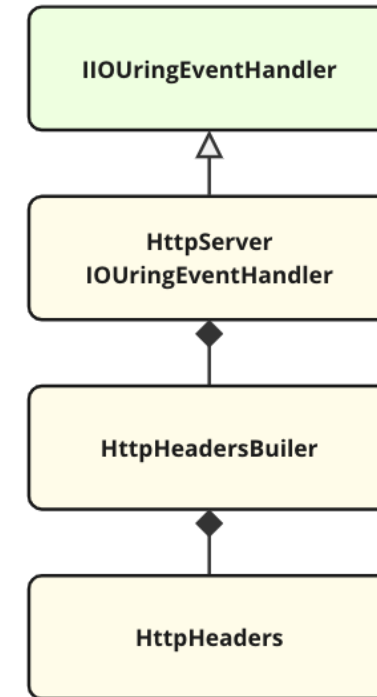
- Приховує від користувача специфіку створення, зберігання та знищення компонентів `io_uring`
- Мінімальний обсяг додаткових обчислень всередині абстракції
- Придатна для застосування у моделі вірця «Проактор»



Діаграма класів компоненту `IOUring`

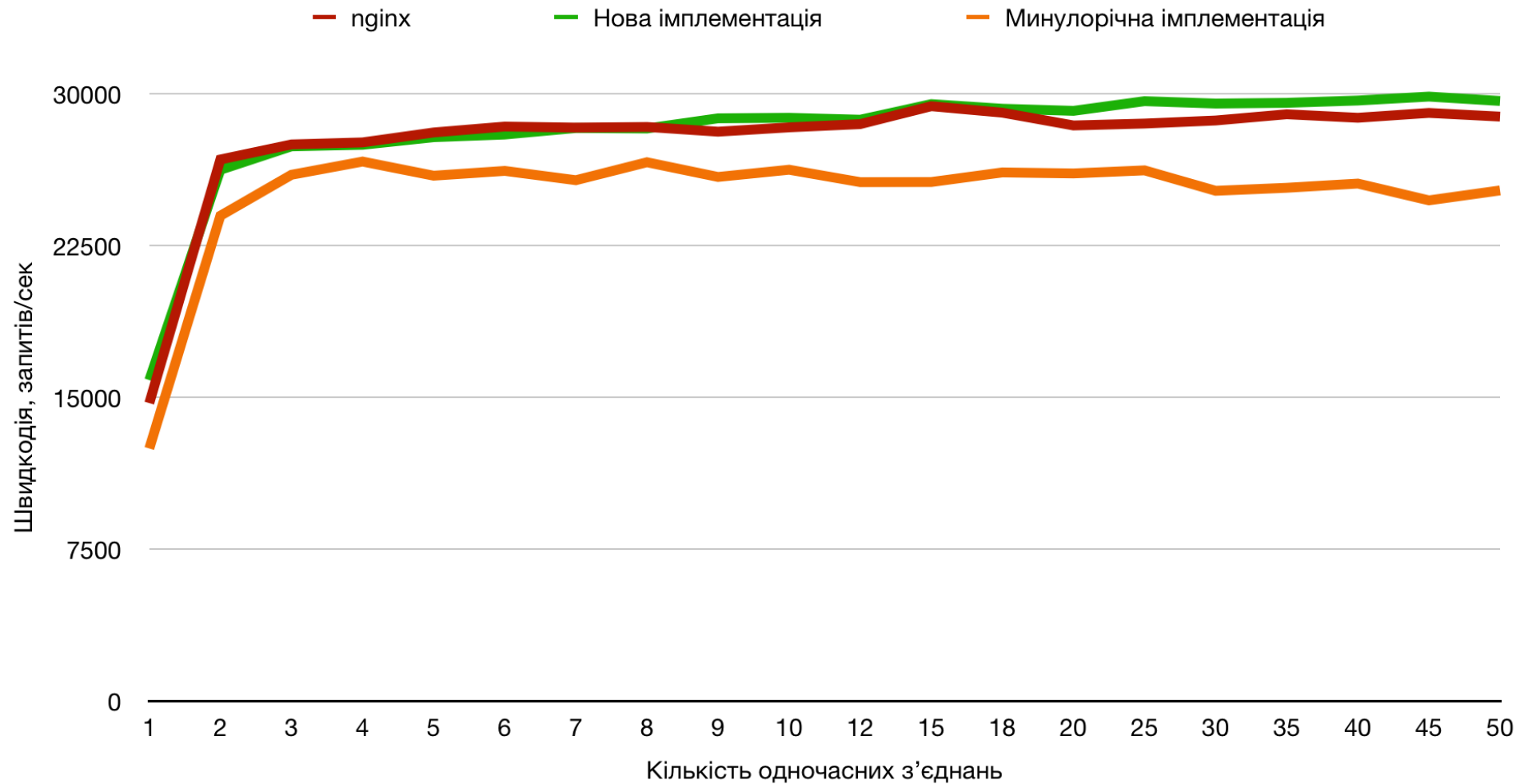
Приклад 1 – HTTP-сервер

- Побудований за допомогою моделі взірця «Проактор»
- Використовує IOUring як для доступу як до файлової системи, так і до мережі
- Перша власна реалізація, що за швидкістю переважає еталонну – nginx



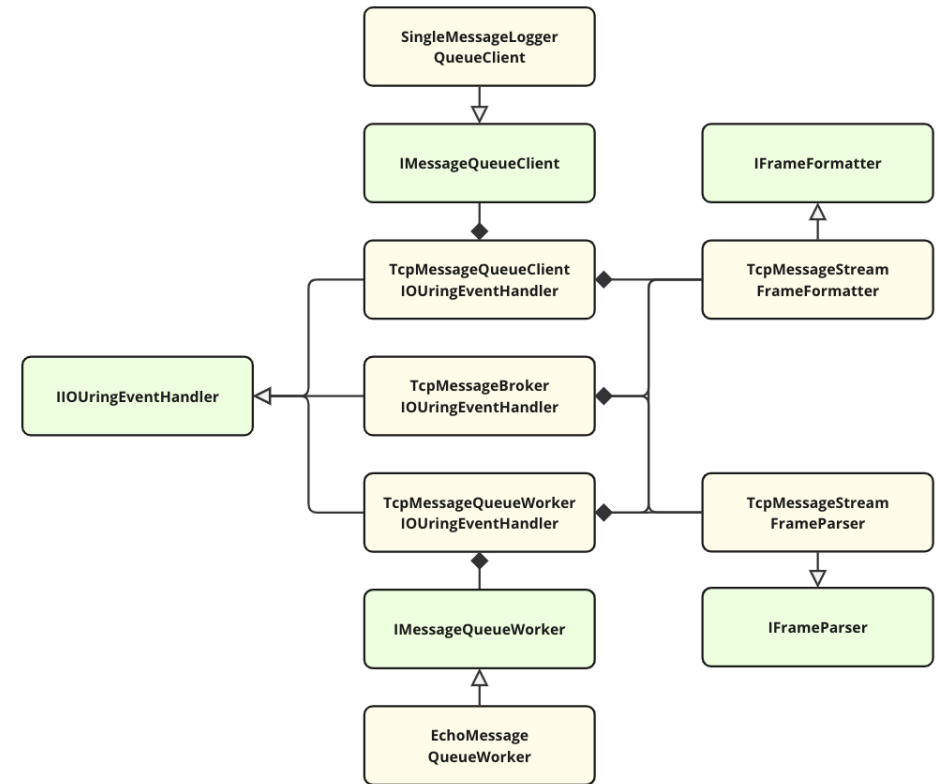
Діаграма класів реалізації HTTP-серверу

Приклад 1 – HTTP-сервер



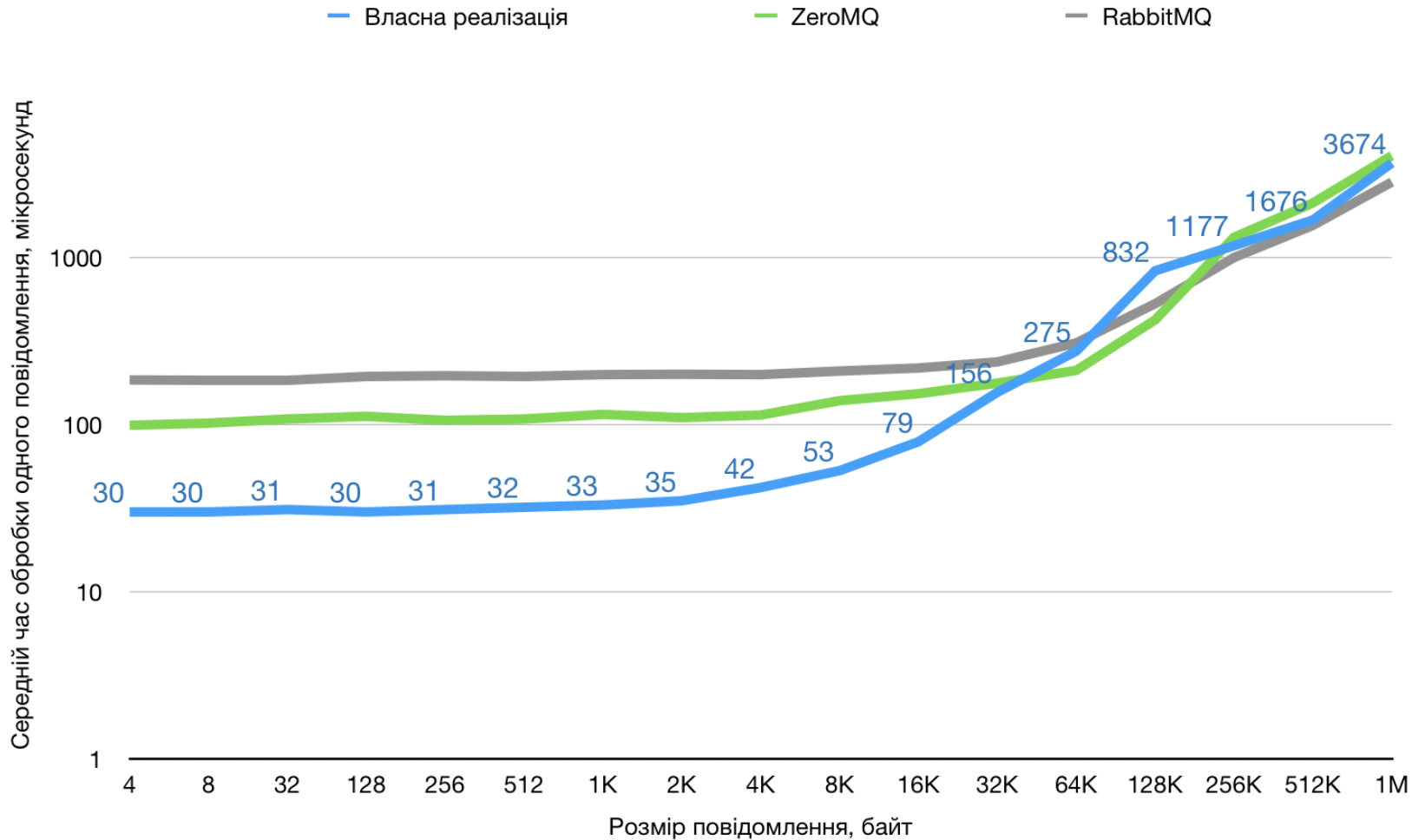
Приклад 2 – Черга повідомлень

- Побудований за допомогою моделі вірця «Проактор»
- Використовує простий протокол передачі повідомлень через TCP-з'єднання
- Впевнено переважає обидві еталонні реалізації на повідомленнях до 16 КБ, демонструє гарну швидкодію на більших розмірах



Діаграма класів реалізації черги повідомлень

Приклад 2 – Черга повідомлень



Результати

- Розглянуто сучасний механізм асинхронного введення-виведення в ОС Linux `io_uring` та адаптовано його до моделі взірця «Проактор»
- На основі цієї моделі було спроектовано архітектуру і реалізовано два варіанти серверів: на основі протоколу HTTP і черг повідомлень
- Аналіз швидкодії запропонованих в роботі рішень продемонстрував їх конкурентну спроможність порівняно з еталонними реалізаціями

Дякую за увагу!