High performance web server
高效能 web 伺服器

學生 : 李祥宇

指導教授 : 王宏鏘

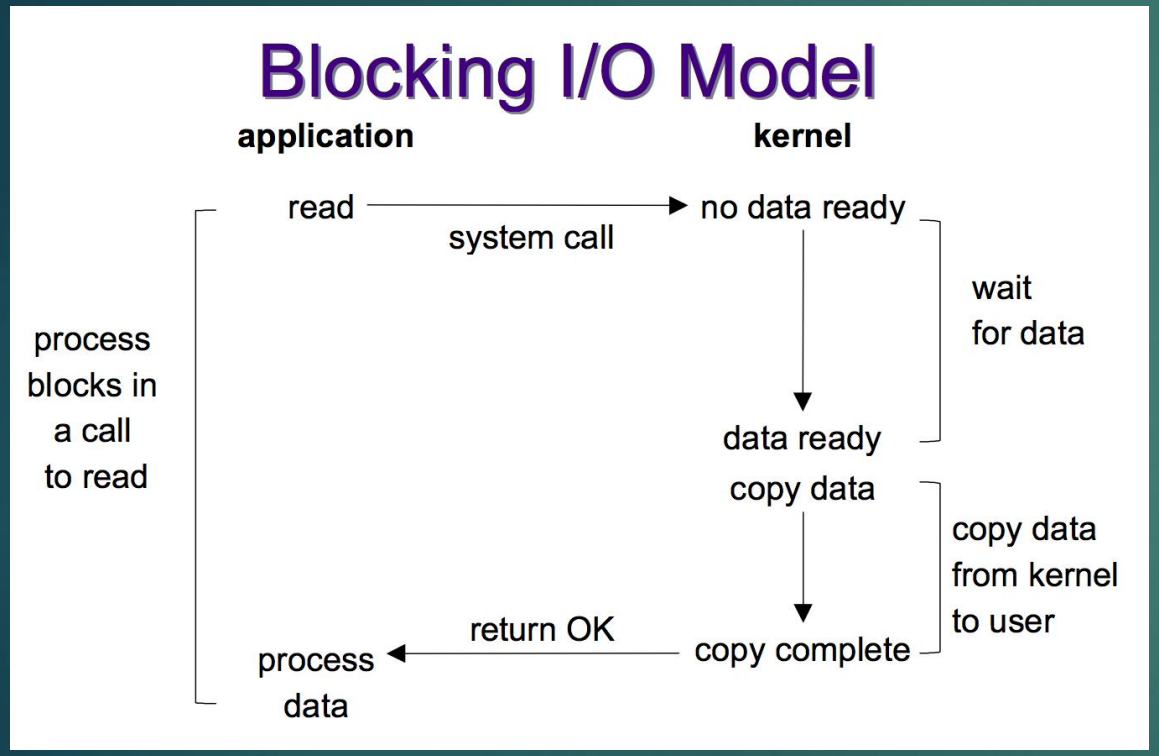
Outline

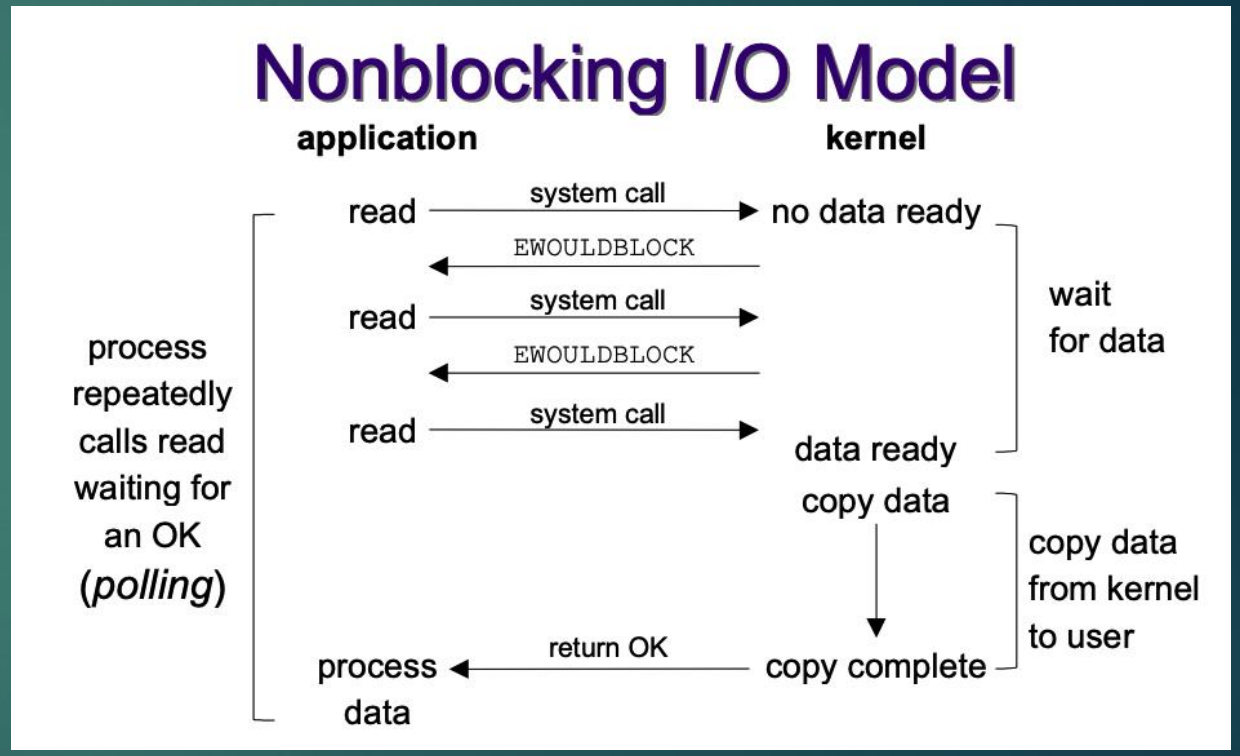
- 成果概述
- IO model 簡介
- Epoll & Io_uring 簡介
- Web 伺服器實作
- 測試結果與 benchmarking

成果概述

- 研究 Linux 上的 I/O model
- 運用近年來嶄新的 io_uring 系統呼叫 ，結合多執行緒，打造高效能 web 伺服器
- 每秒處理約 20 萬個 http requests
- 比較 epoll 和 io_uring 的效能差異

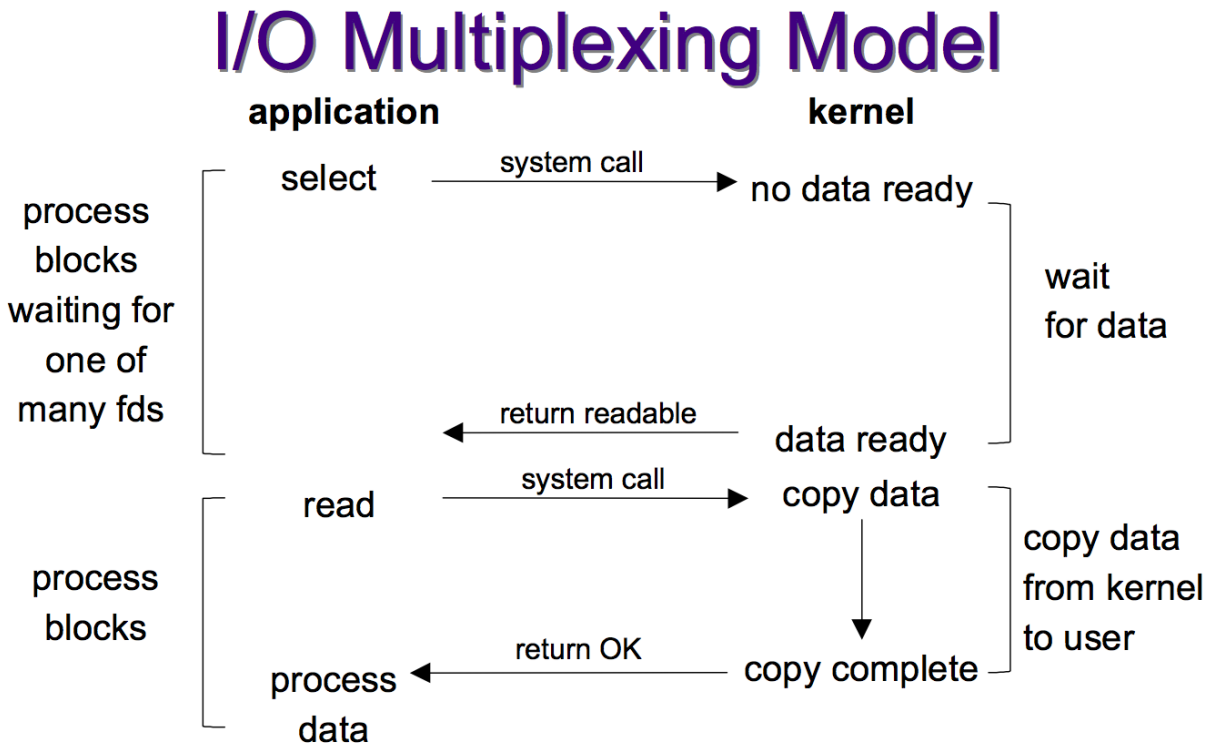
Blocking I/O vs Non-blocking I/O





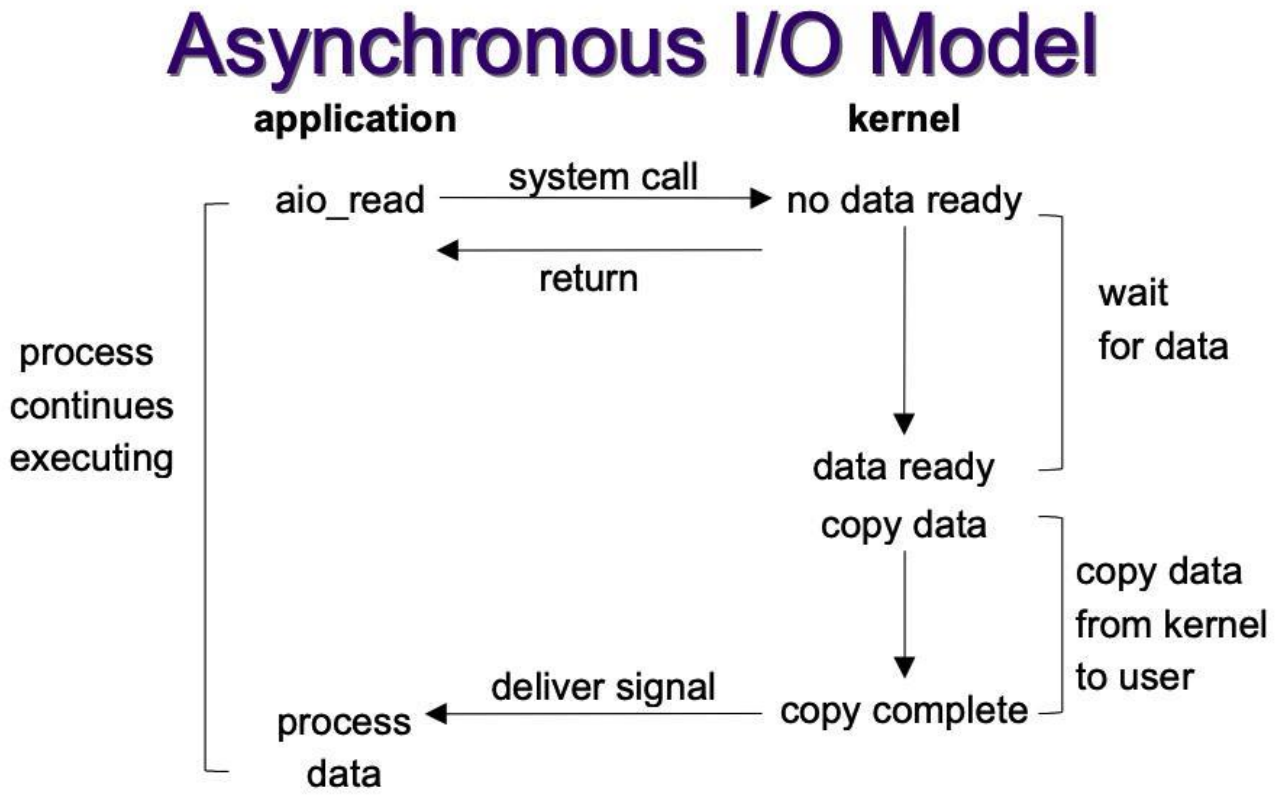
I/O Multiplexing

- I/O multiplexing 其實是 Blocking I/O 的延伸 ，如 select 和 epoll 的行為都是屬於 I/O multiplexing ，差別在於他們能夠同時監事多個 file descriptors 而達到多工。



Asynchronous I/O

- 呼叫 read() 並切換至 kernel mode 之後，不論資料是否就緒，立刻返回 user level ，方才的工作則留在核心，轉交其他 kernel thread 來完成。當完成後，核心會藉由 signal 等方式通知 user level 的 process。

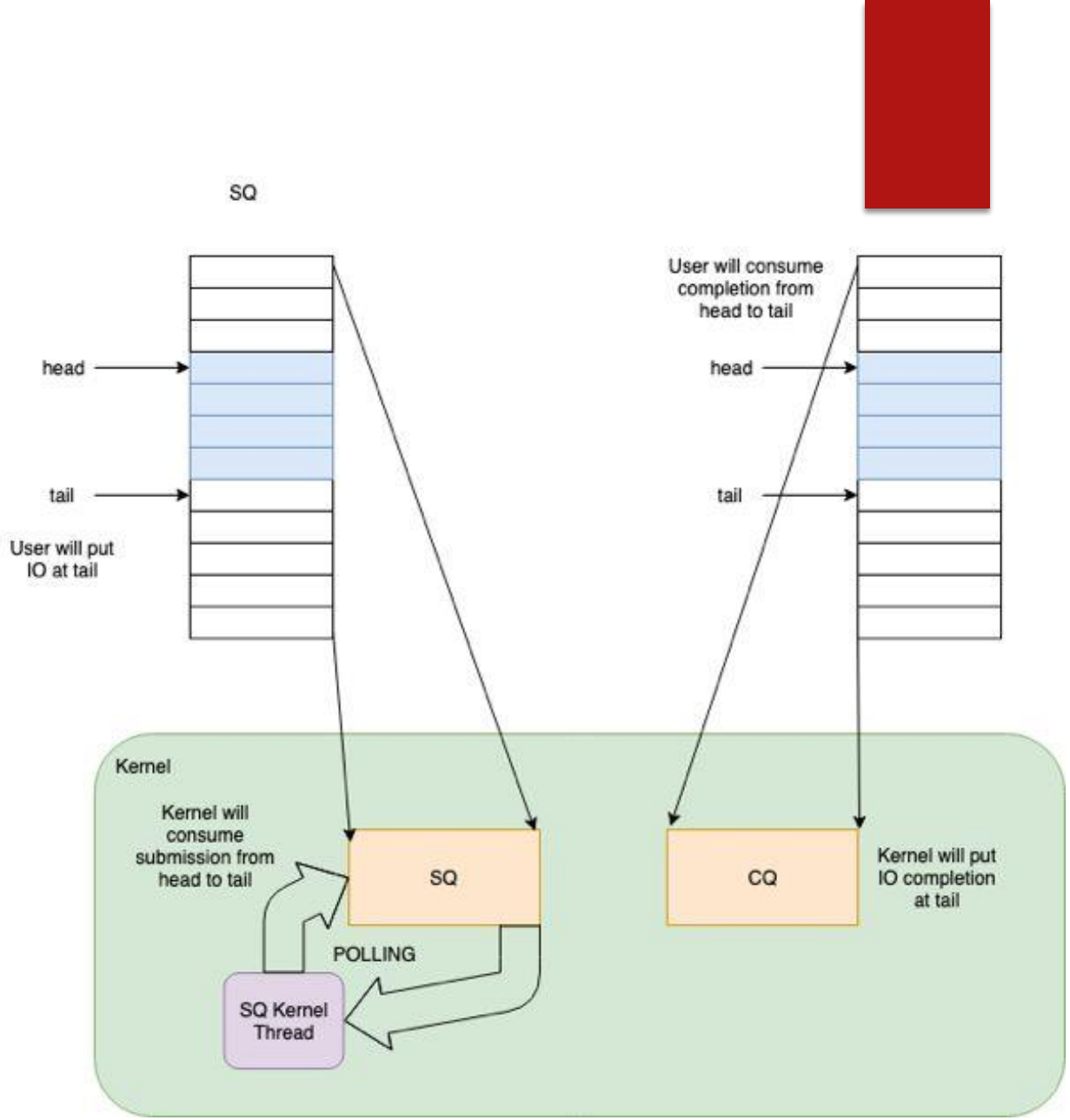


Epoll

- IO multiplexing
- Interest list & ready list(maintained by kernel)
- Search fds in Rbtree
- Faster than select

Io_uring

- 2019, linux 5.1, by Jens Axboe
- A “true” asynchronous I/O
- Shared ring buffer between user space and kernel space.
- Submission queue & Completion queue
- Faster than epoll



Web 伺服器實作 - epoll

- 使用 Epoll 同時監聽多個 client fd
- Thread pool (pthread)
- Memory pool
- 將 request 交給 thread pool 裡的 workers 處理

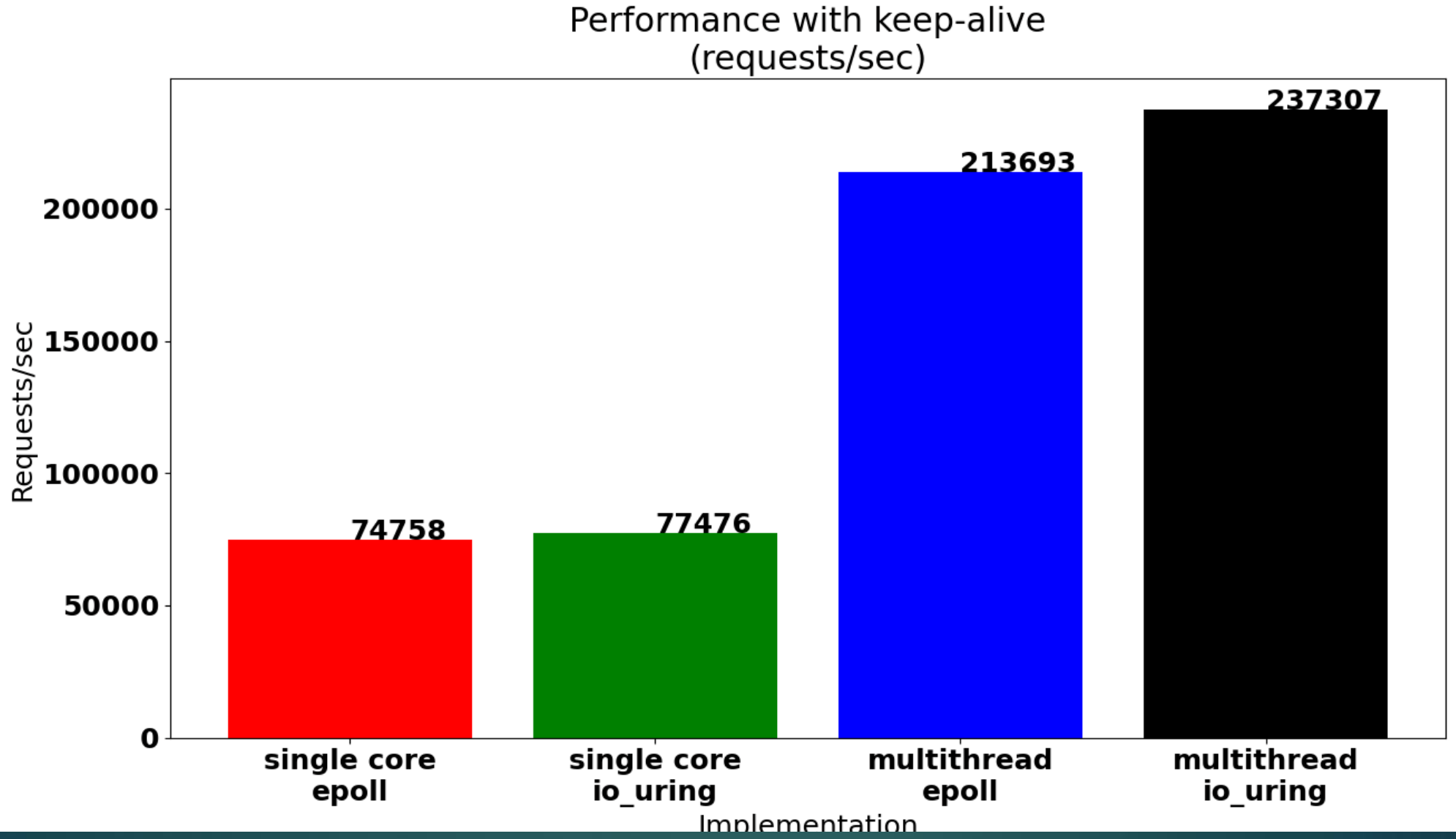
Web 伺服器實作 - io_uring

- Asynchronous I/O
- Multithreaded (pthread)
- One ring buffer per thread
- Each thread handles its own requests
- Memory pool

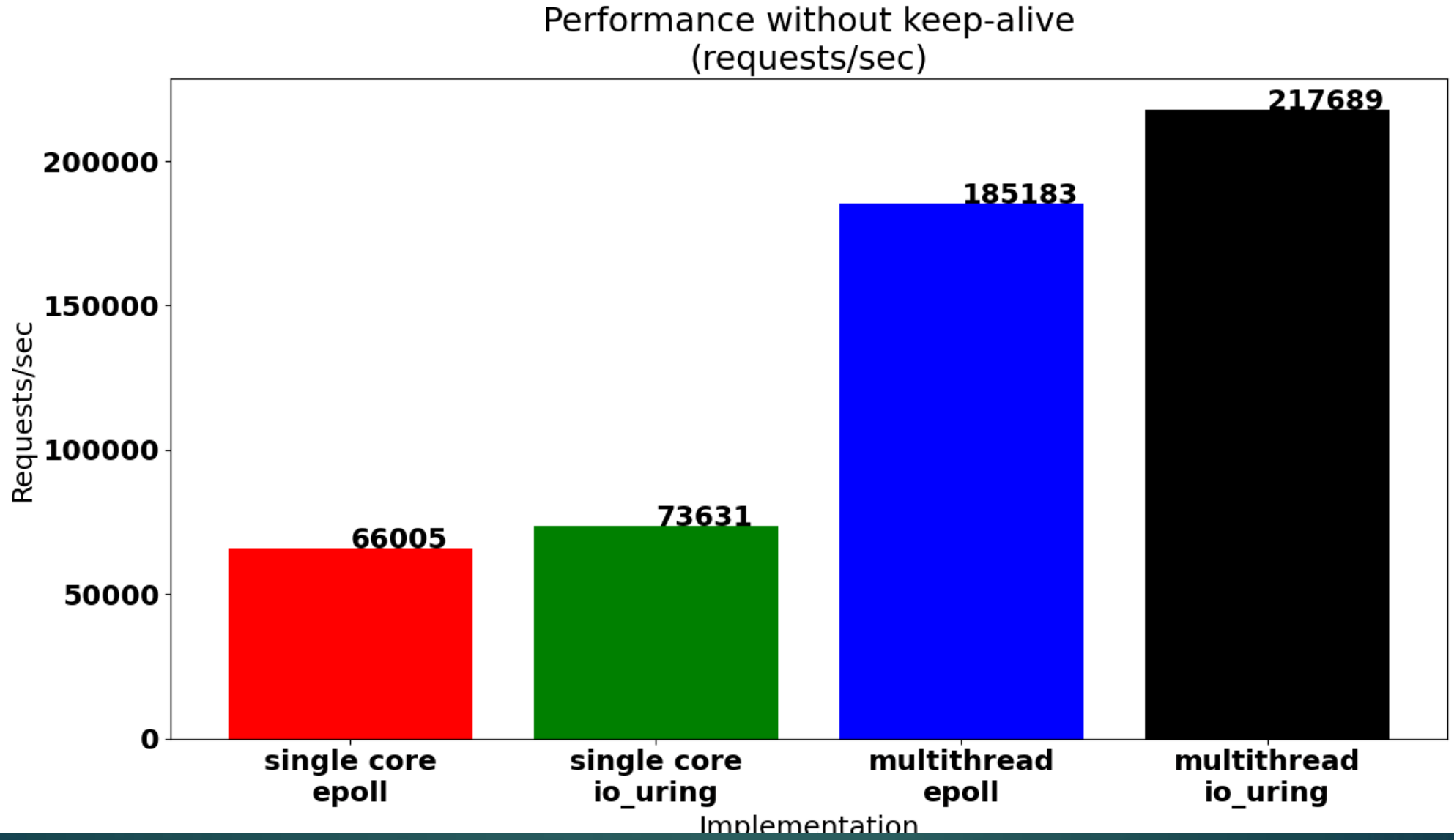
Benchmarking

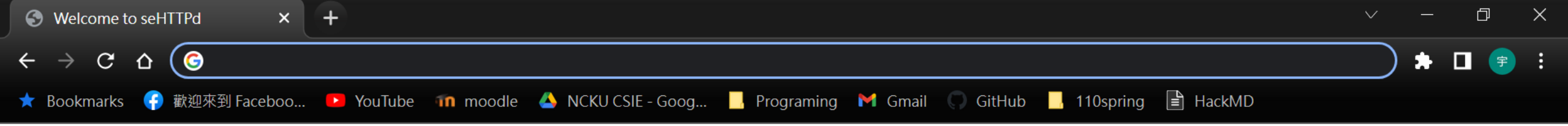
- Tools : wrk, ab, htstress
- 測量 20 次取平均
- Result : io_uring wins!

Tool : Wrk


Performance with keep-alive
(requests/sec)
Requests/sec
0
50000
100000
150000
200000
74758
77476
213693
237307
single core
epoll
single core
io_uring
multithread
epoll
multithread
io_uring
Implementation

Tool : htstress





Welcome!

If you see this page, the seHTTPd web server is successfully working.

DEMO