A Comparative Analysis of Epoll vs. Io_uring

Team Members

S.No.	Name	Roll No.
1.	Tanushree Jatia	202351147
2.	Sanju Rana	202351126
3.	Srushti Chewale	202351140

Objective

This benchmarking report compares the performance of Epoll and Io-uring servers under different client loads. Both systems were tested using identical hardware and software environments to evaluate differences in throughput, latency, and CPU utilization. The goal is to determine which server model provides better scalability and responsiveness for asynchronous I/O operations.

Methodology

- Test Environment – ubuntu
- Benchmarking Tool - wrk
- CPU utilization checker - htop
- Command-Line Tool - curl
- Test Parameters:
  - Threads: Fixed at 4
  - Clients: Varied from 5 to 1000
- Metrics Measured: Requests/sec, Avg Latency, Max Latency, Transfer/sec, CPU Usage

Benchmarking Results

Epoll Server Results

Threads	Clients	Requests/sec	Avg latency	Max latency	Transfer/sec	CPU Usage
4	5	24664.02	0.10862ms	10.81ms	6.56MB	106.9%
4	50	28254.06	1.61ms	14.57ms	7.52MB	108.8%
4	100	27463.89	3.50ms	15.91ms	7.31MB	105.6%
4	500	27303.54	6.21ms	827.70ms	7.26MB	106.4%
4	1000	26779.98	1.16ms	1670ms	7.31MB	106.2%

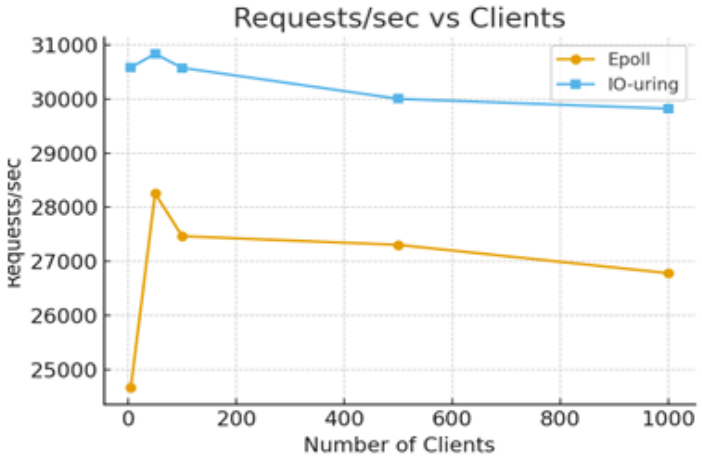
IO-uring Server Results

Threads	Clients	Requests/sec	Avg latency	Max latency	Transfer/sec	CPU Usage
4	5	30586.21	0.0768ms	7.71ms	7.76MB	107.6%

4	50	30836.26	1.48ms	11.54ms	7.82MB	107.1%
4	100	30577.12	3.20ms	9.73ms	7.76MB	107.3%
4	500	30004.32	16.57ms	36.15ms	7.61MB	107.3%
4	1000	29822.67	33.41ms	68.96ms	7.57MB	107.6%

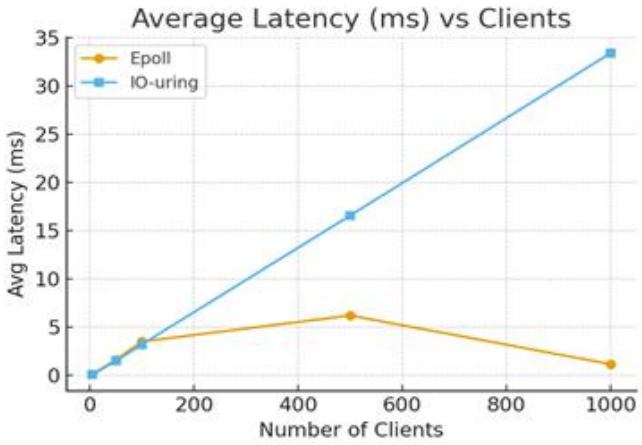
Performance Visualizations

Requests/sec Comparison



- IO-uring consistently delivers higher Requests/sec than Epoll, starting near 30,500 and gradually decreasing, indicating better performance and scaling with an increasing number of clients.
- Epoll's performance is less stable, showing a peak of around 28,200 Requests/sec at a low client count (approx. 50), followed by a noticeable drop-off and a lower stable rate (around 26,800) at 1000 clients.

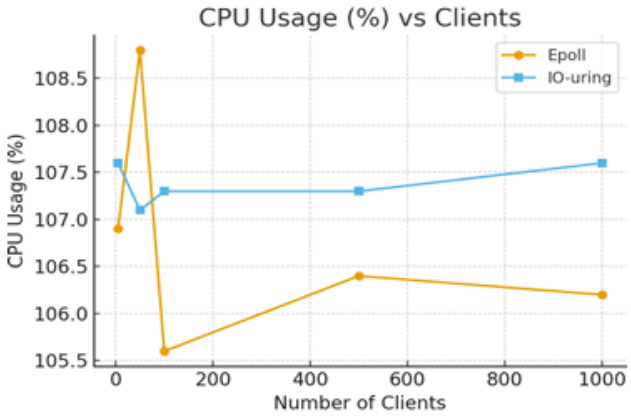
Average Latency Comparison



- Epoll maintains a low average latency throughout the test, peaking slightly around 500 clients (approx. 6.2ms) but then sharply decreasing to a minimal approx 1.5ms at 1000 clients, indicating better latency under high load.

- IO-uring's average latency increases sharply and linearly with the number of clients, starting low but escalating significantly to approximately 33ms at 1000 clients, suggesting it suffers from much higher latency as the load grows.

CPU Usage Comparison



- IO-uring demonstrates a more stable and consistently higher CPU usage compared to Epoll, hovering around 107.2%and showing a slight increase to approximately 107.6% at 1000 clients.
- Epoll's CPU usage is highly variable, peaking at approx108.7% with a low client count (approx. 50), then dropping to a minimum of approx. 105.6% before gradually recovering and settling at a lower rate of approx. 106.2% at 1000 clients.

Comparative Analysis

- **Superior Performance:** IO-uring consistently outperforms Epoll in both request throughput and latency across all tested client loads.
- **Significant Gains at Low Concurrency:** At lower client loads (5–100 clients), IO-uring achieves up to 20% higher request rates and lower latency.
- **Better Predictability Under High Load:** Under heavy load (500–1000 clients), IO-uring maintains more predictable latency, whereas Epoll experiences large latency spikes.
- **Efficiency Source:** Improved I/O Handling: IO-uring's performance gains are attributed to improved asynchronous I/O handling, as both servers exhibit similar CPU usage.
- **Conclusion**: More Efficient Choice: The benchmarking proves IO-uring offers superior performance, especially in high concurrency and low-latency maintenance, making it a more efficient choice for modern high-performance asynchronous I/O servers.

Summary

- Goal: Benchmark Epoll and Io_uring servers under increasing client load.
- Key Finding: Io_uring consistently outperforms Epoll in every key metric.
- Main Advantage: Superior latency control and stability under high concurrency.
- Conclusion: Io_uring is the recommended architecture for high-performance I/O applications.