Performance Comparison: Multi-threaded Client-Server using Select System Call

Summary of Results

Metric	n=50 Clients	n=25 Clients	n=10 Clients
Task Clock (ms)	125.29	Not available	21.01
Context Switches	16	Not available	3
CPU Utilization (%)	0.003	Not available	0.006
Page Faults/sec	526.79	Not available	3.14
CPU Frequency (GHz)	2.942	Not available	3.578
Instructions/sec (G/s)	1.210	Not available	1.422
Branch Misses	550,252	Not available	114,826
TMA Backend Bound (%)	23.6	Not available	25.5
TMA Frontend Bound (%)	34.2	Not available	32.9
TMA Retiring (%)	39.0	Not available	38.0
Total Time Elapsed (s)	44.69	Not available	3.81

Detailed Comparison

1. Task Clock

The task clock (amount of CPU time) for 50 clients is 125.29 ms, significantly higher than the 21.01 ms for 10 clients. This suggests that more CPU time is required to manage a higher number of concurrent clients.

2. Context Switches

With 50 clients, 16 context switches were recorded, compared to only 3 for 10 clients. This indicates that the server has more tasks to manage with a higher number of clients, leading to more frequent task swapping.

3. CPU Utilization

Despite the increase in the number of clients from 10 to 50, CPU utilization remains extremely low (0.003% for 50 clients and 0.006% for 10 clients). This highlights the efficiency of the `select` system call in managing multiple connections.

4. Page Faults

Page faults per second for 50 clients (526.79/sec) are much higher compared to the 10 clients (3.14/sec). With more clients, there is a higher likelihood of the process requiring memory that isn't in physical RAM, leading to more page faults.

5. CPU Frequency

The CPU frequency for 50 clients is 2.942 GHz, lower than 3.578 GHz for 10 clients. This might indicate CPU throttling or reduced demand for cycles with a larger workload.

6. Instructions per Second

The rate of instructions executed per second is higher for 10 clients (1.422 G/sec) than for 50 clients (1.210 G/sec). This suggests that with fewer clients, the server can process more instructions per second, possibly due to reduced overhead.

7. Branch Misses

For 50 clients, branch misses are much higher (550,252) than for 10 clients (114,826). This indicates that managing more clients results in more branch mispredictions, likely due to increased complexity in decision-making.

8. TMA Analysis

- **Backend Bound:** Backend bound is slightly higher with 10 clients (25.5%) compared to 50 clients (23.6%). This indicates that with fewer clients, the server is more often waiting for backend resources.
- **Frontend Bound:** Frontend bound is slightly higher for 50 clients (34.2%) compared to 10 clients (32.9%).
- **Retiring:** The retiring percentage is similar for both setups (39.0% for 50 clients and 38.0% for 10 clients), indicating efficient instruction completion in both cases.

9. Total Time Elapsed

The total time elapsed for 50 clients is 44.69 seconds, while for 10 clients, it is 3.81 seconds. The large difference is expected due to the increased number of clients.

Conclusions

- **Efficiency of `select`:** The `select` system call efficiently handles even a large number of clients (n=50). CPU utilization remains low, despite an increase in context switches and branch misses.
- **Performance Degradation:** As the number of clients increases, performance metrics such as branch misses and instruction execution rate degrade, but the degradation is not dramatic, showing that `select` scales reasonably well.
- **Context Switching:** The low number of context switches demonstrates the efficiency of `select` in avoiding frequent task swapping.
- **Ideal Usage:** For high-concurrency scenarios without significant CPU overhead, `select` is ideal. However, some performance metrics like branch misses and instruction execution may suffer slightly as the number of clients grows.