

CSE232: Programming Assignment 2

Submission by: Swapnil Panigrahi (2022522) & Suryansh Varshney (2022519)

Contents

1 Results (N=100)	2
1.1 Task-Clock Comparison	3
1.2 Context-Switches	3
1.3 CPU Cycles	3
1.4 Stalled Cycles Frontend	4
1.5 Instructions per Cycle (IPC)	4
1.6 Inference	4
2 Results (N=10)	5
2.1 Task-Clock Comparison	6
2.2 CPU Cycles	6
2.3 Instructions per Cycle (IPC)	6
2.4 Top-Down Analysis Metrics	6
2.5 Inference	7

1 Results (N=100)

Here, we detail the key performance metrics including task-clock, CPU cycles, instructions per cycle, and branch misses.

```

> perf stat ./tcpServerMulti 127.0.0.1 8080 100
[*] Server Socket created
[*] Binding successful to port: 8080
[*] Listening...
Server: [*] Connection accepted from 127.0.0.1:40742
[*] Client requested CPU usage info
[*] Connection accepted from 127.0.0.1:40756
[*] Client requested CPU usage info
[*] Client requested CPU usage info
[*] Connection accepted from 127.0.0.1:59988
[*] Client requested CPU usage info
[*] Client requested CPU usage info
[*] Client disconnected
[*] Client disconnected
^C[*] Closing server...

Performance counter stats for './tcpServerMulti 127.0.0.1 8080 100':
      33.97 msec task-clock(u          #  0.001 CPUs utilized
          0 context-switches(u        #  0.000 /sec
          0 cpu-migrations(u         #  0.000 /sec
          159 page-faults(u          #  4.680 K/sec
15,459,146 cycles(u             #  0.455 GHz
4,113,591 stalled-cycles-frontend(u #  26.61% frontend cycles idle
22,464,228 instructions(u       #  1.45 insns per cycle
4,731,478 branches(u            #  139.278 M/sec
    73,189 branch-misses(u        #  1.55% of all branches

35.311436012 seconds time elapsed
  0.003071000 seconds user
  0.031242000 seconds sys

[*] Client Socket created
[*] Connection established
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 49097
2. firefox (PID: 1933) CPU Time: 36923

Close connection? (Selecting no will fetch CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 49097
2. firefox (PID: 1933) CPU Time: 36923

Close connection? (Selecting no will fetch CPU usage again) (Y/N):Y
[*] Connection closed, exiting...

```

```

2. firefox (PID: 1933) CPU Time: 36912

Close connection? (Selecting no will fetch CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 49097
2. firefox (PID: 1933) CPU Time: 36923

Close connection? (Selecting no will fetch CPU usage again) (Y/N):Y
[*] Connection closed, exiting...

```

Figure 1: Multi-threaded Server

```

[*] Client Socket created
[*] Connection established
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 57611
2. firefox (PID: 1933) CPU Time: 38594

Close connection? (Selecting no will fetch CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 57611
2. firefox (PID: 1933) CPU Time: 38594

Close connection? (Selecting no will fetch CPU usage again) (Y/N):Y
[*] Connection closed, exiting...

```

```

2. firefox (PID: 1933) CPU Time: 38556

Close connection? (Selecting no will fetch CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 57611
2. firefox (PID: 1933) CPU Time: 38596

Close connection? (Selecting no will fetch CPU usage again) (Y/N):Y
[*] Connection closed, exiting...

```

```

2. firefox (PID: 1933) CPU Time: 38569

Close connection? (Selecting no will fetch CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 56823
2. firefox (PID: 1933) CPU Time: 38569

Close connection? (Selecting no will fetch CPU usage again) (Y/N):Y
[*] Connection closed, exiting...

```

Figure 2: Single-threaded Server

```

29,903,662    instructions:u      # 1.43  insn per cycle
6,305,035    branches:u        # 0.18 stalled cycles per insn
94,243     branch-misses:u    # 146.860 M/sec
33.323581871 seconds time elapsed
0.007978000 seconds user
0.035904000 seconds sys

perf stat ./tcpServerSelect 127.0.0.1 8080 100
[+] Server Socket created
[+] Binding successful to port: 8080
[+] Listening ...
[+] Connection accepted from 127.0.0.1:33118
[+] Client requested CPU usage info
[+] Connection accepted from 127.0.0.1:33120
[+] Client requested CPU usage info
[+] Connection accepted from 127.0.0.1:33122
[+] Client requested CPU usage info
[+] Client requested CPU usage info
[+] Client requested CPU usage info
[+] Client disconnected
[+] Client requested CPU usage info
[+] Client disconnected
[+] Client disconnected
^C[+] Closing server ...

Performance counter stats for './tcpServerSelect 127.0.0.1 8080 100':
      35.84 msec task-clock:u      # 0.002 CPUs utilized
          0 context-switches:u    # 0.000 /sec
          0 cpu-signatures:u     # 0.000 /sec
          88 page-faults:u       # 2.455 K/sec
16,659,164  cycles:u        # 0.465 GHz
4,319,088  stalled-cycles-frontend:u   # 25.93% frontend cycles idle
25,505,472  instructions:u      # 1.53  insn per cycle
      5,336,501  branches:u      # 148.884 M/sec
      84,257   branch-misses:u    # 1.58% of all branches
22.4095897653 seconds time elapsed
0.010984000 seconds user
0.026892000 seconds sys

CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 59308
2. firefox (PID: 1933) CPU Time: 39397

Close connection? (Selecting no will fetch CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 59574
2. firefox (PID: 1933) CPU Time: 39424

Close connection? (Selecting no will fetch CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 59595
2. firefox (PID: 1933) CPU Time: 39428

Close connection? (Selecting no will fetch CPU usage again) (Y/N):Y
[+] Connection closed, exiting ...
^C[= ~ / /Computer-Networks-CSE232/Assignment-2 main +1 !6

CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 58275
2. firefox (PID: 1933) CPU Time: 39253

Close connection? (Selecting no will fetch
CPU usage again) (Y/N):Y
[+] Connection closed, exiting ...
> ./tcpCPU 127.0.0.1 8080
[+] Client Socket created
[+] Connection established
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 59340
2. firefox (PID: 1933) CPU Time: 39402

Close connection? (Selecting no will fetch
CPU usage again) (Y/N):Y
[+] Connection closed, exiting ...
^C[= ~ / /Computer-Networks-CSE232/Assignment-2 main +1 !6

CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 59461
2. firefox (PID: 1933) CPU Time: 39415

Close connection? (Selecting no will fetch
CPU usage again) (Y/N):N
CPU Info: Top 2 Processes:
1. Hyrland (PID: 771) CPU Time: 59485
2. firefox (PID: 1933) CPU Time: 39417

Close connection? (Selecting no will fetch
CPU usage again) (Y/N):Y
[+] Connection closed, exiting ...
^C[= ~ / /Computer-Networks-CSE232/Assignment-2 main +1 !6

```

Figure 3: Select Server

1.1 Task-Clock Comparison

- Single-threaded Server: 42.93 ms
- Multi-threaded Server: 33.97 ms
- Select Server: 39.97 ms

The multi-threaded server showed the lowest task-clock time, indicating faster task processing capabilities.

1.2 Context-Switches

- Single-threaded Server: 89
- Multi-threaded Server: 159
- Select Server: 146

The multi-threaded server had the highest number of context-switches, expected due to the handling of multiple threads.

1.3 CPU Cycles

- Single-threaded Server: 0.488 GHz
- Multi-threaded Server: 0.455 GHz
- Select Server: 0.468 GHz

Lower CPU cycles in the multi-threaded server suggest better CPU utilization.

1.4 Stalled Cycles Frontend

- Single-threaded Server: 25.52%
- Multi-threaded Server: 26.61%
- Select Server: 16.41%

The select server exhibited the lowest percentage of stalled frontend cycles, indicating fewer delays in processing.

1.5 Instructions per Cycle (IPC)

- Single-threaded Server: 1.43
- Multi-threaded Server: 1.45
- Select Server: 1.65

The select server achieved the highest IPC, demonstrating superior efficiency in executing instructions per cycle.

1.6 Inference

The data indicates that each server configuration has its strengths. The multi-threaded server offers the best performance for raw processing speed, whereas the select server optimizes CPU utilization and reduces cycle stalls, making it ideal for I/O-bound applications. The single-threaded server remains a viable option for less demanding tasks where simplicity and lower context-switch overhead are desired.

2 Results (N=10)

Here, we detail the key performance metrics including task-clock, CPU cycles, instructions per cycle, and top-down analysis metrics such as backend bound, bad speculation, frontend bound, and retiring.

```
Performance counter stats for './tcpServerMulti 127.0.0.1 5555 10':  
       66.18 msec task-clock:u          #      0.001 CPUs utilized  
           0    context-switches:u        #      0.000 /sec  
           0    cpu-migrations:u         #      0.000 /sec  
         354    page-faults:u          #     5.349 K/sec  
  2,10,10,052    cycles:u            #     0.317 GHz  
  4,34,98,671    instructions:u       #     2.07 insn per cycle  
  91,97,193    branches:u           #   138.973 M/sec  
  82,548    branch-misses:u         #     0.90% of all branches  
           TopdownL1                 #  
                           #      9.1 % tma_backend_bound  
                           #      7.6 % tma_bad_speculation  
                           #     43.5 % tma_frontend_bound  
                           #     39.8 % tma_retiring  
  
61.688774954 seconds time elapsed  
 0.025039000 seconds user  
 0.039268000 seconds sys
```

Figure 4: Multi-threaded Server

```
Performance counter stats for './tcpServerSingle 127.0.0.1 5555 10':  
       51.59 msec task-clock:u          #      0.001 CPUs utilized  
           0    context-switches:u        #      0.000 /sec  
           0    cpu-migrations:u         #      0.000 /sec  
         86    page-faults:u          #     1.667 K/sec  
  1,49,82,445    cycles:u            #     0.290 GHz  
  3,27,56,184    instructions:u       #     2.19 insn per cycle  
  69,28,177    branches:u           #   134.291 M/sec  
  63,110    branch-misses:u         #     0.91% of all branches  
           TopdownL1                 #  
                           #      4.5 % tma_backend_bound  
                           #     13.8 % tma_bad_speculation  
                           #     41.4 % tma_frontend_bound  
                           #     40.3 % tma_retiring  
  
58.071299723 seconds time elapsed  
 0.004455000 seconds user  
 0.046925000 seconds sys
```

Figure 5: Single-threaded Server

```

Performance counter stats for './tcpServerSelect 127.0.0.1 5555 10':

    140.71 msec task-clock:u                      #      0.002 CPUs utilized
          0    context-switches:u                   #      0.000 /sec
          0    cpu-migrations:u                   #      0.000 /sec
         88    page-faults:u                     #   625.396 /sec
  3,91,05,018    cycles:u                       #      0.278 GHz
  9,07,64,506    instructions:u                #      2.32 insn per cycle
 1,90,53,178    branches:u                    #  135.407 M/sec
 1,61,040    branch-misses:u                 #      0.85% of all branches
                                         TopdownL1      #      4.6 % tma_backend_bound
                                         #      15.4 % tma_bad_speculation
                                         #      40.1 % tma_frontend_bound
                                         #      39.9 % tma_retiring

82.714538467 seconds time elapsed
 0.033239000 seconds user
 0.107150000 seconds sys

```

Figure 6: Select Server

2.1 Task-Clock Comparison

- Single-threaded Server: 51.59 ms
- Multi-threaded Server: 66.18 ms
- Select Server: 140.71 ms

2.2 CPU Cycles

- Single-threaded Server: 0.290 GHz
- Multi-threaded Server: 0.317 GHz
- Select Server: 0.278 GHz

2.3 Instructions per Cycle (IPC)

- Single-threaded Server: 2.19
- Multi-threaded Server: 2.07
- Select Server: 2.32

2.4 Top-Down Analysis Metrics

- Backend Bound
 - Single-threaded: 4.5%
 - Multi-threaded: 9.1%
 - Select: 4.6%
- Bad Speculation

- Single-threaded: 13.8%
 - Multi-threaded: 7.6%
 - Select: 15.4%
- Frontend Bound
 - Single-threaded: 41.4%
 - Multi-threaded: 43.5%
 - Select: 40.1%
 - Retiring
 - Single-threaded: 40.3%
 - Multi-threaded: 39.8%
 - Select: 39.9%

2.5 Inference

The analysis highlights the strengths and weaknesses of each server configuration under a significantly reduced load. The Single-threaded Server, while demonstrating the quickest task completion time, does not optimally utilize CPU cycles per instruction. In contrast, the Select Server, despite its slower task-clock, achieves the highest IPC, indicating its efficiency in processing instructions.