Design Optimization of Computing Systems

Assignment 3

DOCS-DB (Part-B)

Submitted by:

Bratin Mondal (21CS10016)
Swarnabh Mandal (21CS10068)
Somya Kumar (21CS30050)

Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur

1 Introduction

This section presents the performance metrics for a network application measured under two scenarios: without using DPDK (Data Plane Development Kit) and with DPDK. The measurements were conducted for varying numbers of parallel clients (10, 100, and 1000). For each case, the number of packets sent by each client is varied, and key metrics such as latency, bandwidth, CPU utilization, and memory utilization are recorded.

It is important to note that the latency and bandwidth metrics are reported on a per-client basis. In contrast, the CPU and memory utilization metrics correspond to the server's overall resource usage. These distinctions provide a granular view of the performance characteristics under different scenarios and workloads.

2 Metrics Without DPDK

Table 1: Performance Metrics for 10 Parallel Clients

Number of Packets Sent by each Client	**Total Number of Packets**	**Latency (microseconds)**	**Bandwidth (bps)**	**CPU Utilization (%)**	**Memory Utilization (KB)**
1,000	10,000	1578.447	486033	0.291	1949
10,000	100,000	162.322	4734803	4.231	1977
100,000	1,000,000	154.653	4964883	12.002	1917

Table 2: Performance Metrics for 100 Parallel Clients

Number of Packets Sent by each Client	**Total Number of Packets**	**Latency (microseconds)**	**Bandwidth (bps)**	**CPU Utilization (%)**	**Memory Utilization (KB)**
100	10,000	1154.561	701895	0.195	3241
1,000	100,000	1372.372	567331	3.325	3321
10,000	1,000,000	1233.469	622722	13.235	3569

Table 3: Performance Metrics for 1000 Parallel Clients

Number of Packets Sent by each Client	**Total Number of Packets**	**Latency (microseconds)**	**Bandwidth (bps)**	**CPU Utilization (%)**	**Memory Utilization (KB)**
10	10,000	4853.634	580192	1.517	7372
100	100,000	7971.900	182634	6.488	7005
1,000	1,000,000	12481.949	61716	31.648	6724

3 Metrics With DPDK

Table 4: Performance Metrics for 10 Parallel Clients

Number of Packets Sent by each Client	Total Number of Packets	Latency (microseconds)	Bandwidth (bps)	CPU Utilization (%)	Memory Utilization (KB)
1,000	10,000	189.793	4052276	99.227	42496
10,000	100,000	192.158	3994910	97.225	42496
100,000	1,000,000	173.884	4417136	100.000	42496

Table 5: Performance Metrics for 100 Parallel Clients

Number of Packets Sent by each Client	Total Number of Packets	Latency (microseconds)	Bandwidth (bps)	CPU Utilization (%)	Memory Utilization (KB)
100	10,000	1212.710	647904	100.000	42647
1,000	100,000	1367.602	561697	98.394	42697
10,000	1,000,000	1249.108	614872	97.764	42705

Table 6: Performance Metrics for 1000 Parallel Clients

Number of Packets Sent by each Client	Total Number of Packets	Latency (microseconds)	Bandwidth (bps)	CPU Utilization (%)	Memory Utilization (KB)
10	10,000	2124.149	613000	99.173	42974
100	100,000	8584.248	105523	97.843	44233
1,000	1,000,000	9570.430	80238	97.886	45400

3.1 Discussion

The performance metrics reveal significant variations across the different scenarios and configurations, which are discussed below.

3.1.1 Latency Analysis

Without DPDK, the latency generally decreases as the number of packets sent by each client increases. This trend reflects the inefficiency of traditional packet processing frameworks in handling small packet sizes, where overheads dominate. However, for larger packet counts, these inefficiencies are amortized, leading to lower latency.

With DPDK, the latency is significantly reduced across all configurations due to its optimized packet processing capabilities. The use of polling-based mechanisms and efficient memory management in DPDK eliminates context-switching overheads, resulting in consistently low latency. However, as the number of clients increases, the server experiences slightly higher latencies due to contention for CPU resources.

3.1.2 Bandwidth Trends

Bandwidth, like latency, is affected by the number of packets sent by each client. Without DPDK, the bandwidth increases with larger packet counts, but the increase is relatively modest. This limitation arises from the higher overheads and less efficient utilization of server resources in traditional packet processing frameworks.

In contrast, DPDK achieves significantly higher bandwidth across all configurations. Even with 1000 parallel clients, the bandwidth remains competitive, demonstrating DPDK's ability to scale effectively with increased workloads.

3.1.3 CPU Utilization

Without DPDK, CPU utilization remains low due to the reliance on interrupt-driven processing. However, this low utilization comes at the expense of higher latency and lower bandwidth, as the CPU is not fully utilized for packet processing.

With DPDK, CPU utilization reaches near 100% for most configurations. This is because DPDK employs polling-based techniques that continuously consume CPU cycles to maximize packet throughput. While this approach ensures optimal performance, it also leaves little room for other processes on the server.

3.1.4 Memory Utilization

Memory utilization varies slightly across configurations in both scenarios. Without DPDK, memory usage remains relatively low, reflecting the simpler memory allocation strategies of traditional frameworks. With DPDK, memory usage is higher due to the pre-allocation of large memory pools for packet buffers, a critical feature that supports its high-performance design.

3.1.5 Impact of Client Count

As the number of parallel clients increases, both latency and bandwidth metrics reflect the additional processing burden on the server. Without DPDK, this results in a steep decline in performance metrics, particularly for 1000 clients. With DPDK, while the performance also degrades, the impact is less pronounced, showcasing its superior scalability and efficiency under heavy workloads.

Overall, the results highlight the advantages of using DPDK for high-performance network applications, particularly in scenarios requiring low latency and high throughput. However, the trade-offs in terms of higher CPU and memory usage should be carefully considered when deploying DPDK in production environments.