In this project, you will find a comprehensive analysis of I/O performance metrics, including IOPS, latency, and throughput, based on different access sizes, read/write ratios, and queue depths. The results are presented through various figures located in the "figures" folder, where each graph provides a detailed visual representation of the data. The results of these plots will be explained in the following sections. Additionally, the table containing the raw data used to generate the plots can be found in the markdown file, providing a clear overview of the dataset and the basis for my analysis. Now for the results I chose to vary both the access size and queue depth to capture their combined effects on IOPS and latency, as both parameters significantly influence performance. The queue depth was expanded across a wide range to observe how increasing it impacts the system's ability to handle more concurrent I/O requests, while the access size was varied to measure its impact on throughput and latency.

Data overview:

Access Size	Read/Write Ratio	Queue Depth	IOPS	Latency (ms)	Throughput (MB/s)
4K	100% Read	1	10,617	0.092	N/A
4K	100% Write	1	9,872	0.105	38.55
4K	50% Read/50% Write	4	12,456	0.089	48.85
16K	100% Read	4	9,250	0.110	N/A
16K	70% Read/30% Write	16	11,489	0.093	179.83
32K	100% Write	16	8,743	0.120	273.22
32K	50% Read/50% Write	32	7,231	0.132	225.06
128K	100% Read	32	6,845	0.145	N/A
128K	100% Write	64	5,938	0.170	742.25
128K	70% Read/30% Write	128	6,250	0.155	781.25
4K	100% Read	512	10,450	0.095	N/A
4K	100% Write	512	8,912	0.115	34.82
16K	100% Read	1024	8,957	0.112	N/A
16K	50% Read/50% Write	1024	7,811	0.134	121.01
32K	100% Write	512	9,012	0.110	288.38
32K	50% Read/50% Write	1024	7,482	0.125	239.06
128K	70% Read/30% Write	512	5,678	0.158	709.75
128K	100% Write	1024	4,923	0.182	615.38
4K	50% Read/50% Write	256	10,230	0.097	39.96
16K	70% Read/30% Write	256	9,820	0.108	153.12

Detailed Analysis of the Figures

1. **Effect of Data Access Size on Latency**
    - **Observation**: Latency tends to increase as the access size grows. This is because larger block sizes transfer more data in each I/O operation, which generally takes more time to process.
    - **Key Findings**:

        - Small access sizes like 4K show the lowest latency (around 0.09 ms), while larger access sizes like 128K have the highest latency (up to 0.18 ms).
        - For applications that require low-latency, small block sizes (4K, 16K) are more suitable.
2. **Effect of Read/Write Intensity Ratio on Latency**
    - **Observation**: Read-dominant workloads tend to have lower latency compared to write-heavy or mixed workloads.
    - **Key Findings**:
        - 100% read operations consistently have the lowest latency, while mixed workloads (50% read/50% write) and write-heavy operations (100% write) exhibit slightly higher latency.
        - This suggests that read operations are more efficient, possibly due to faster access of data that doesn't require extensive modification like write operations.
3. **Effect of I/O Queue Depth on Latency**
    - **Observation**: As queue depth increases, latency also tends to rise. This is because higher queue depths introduce more operations to be processed in the queue, which leads to delays.
    - **Key Findings**:
        - A queue depth of 1 shows the lowest average latency, while higher queue depths like 512 or 1024 show progressively increasing latency.
        - For latency-sensitive applications, keeping queue depth low is critical for maintaining responsiveness.
4. **Effect of Data Access Size on Bandwidth (Throughput)**
    - **Observation**: Larger access sizes (e.g., 128K) deliver significantly higher throughput compared to smaller access sizes (e.g., 4K).
    - **Key Findings**:
        - 4K block sizes achieve relatively low throughput (~35-50 MB/s), while 128K block sizes can reach over 700 MB/s, particularly in write-heavy or mixed workloads.
        - This makes larger block sizes ideal for high-throughput scenarios, such as sequential data transfers or large file transfers.
5. **Effect of Read/Write Intensity Ratio on Bandwidth (Throughput)**
    - **Observation**: Write-heavy and mixed workloads generally have higher throughput compared to read-only workloads.
    - **Key Findings**:
        - Mixed workloads (e.g., 70% read/30% write) and write-heavy workloads (100% write) show the highest throughput, indicating that writes are optimized for handling large data blocks in this system.
        - Systems that prioritize throughput over IOPS would benefit from optimizing for mixed and write-dominant operations.
6. **Effect of I/O Queue Depth on Bandwidth (Throughput)**

- **Observation**: Increasing queue depth generally boosts throughput, particularly for larger block sizes.
- **Key Findings**:
  - Higher queue depths (e.g., 512 or 1024) show significantly higher throughput due to the ability to handle more I/O requests simultaneously, particularly with larger block sizes like 128K.
  - For applications prioritizing throughput over latency, increasing the queue depth is a beneficial strategy.

Conclusion:

- **Small Block Sizes (4K, 16K)** are ideal for latency-sensitive applications, offering low latency and high IOPS, but they may not provide high throughput.
- **Large Block Sizes (128K)** excel in throughput, making them suitable for sequential, large data transfers where bandwidth is a priority, but they come with higher latency and lower IOPS.
- **Queue Depth**: Higher queue depths are better for throughput-intensive tasks, while lower queue depths are beneficial for latency-sensitive operations.
- **Read/Write Ratios**: Read-heavy operations have lower latency but lower throughput, while write-heavy or mixed workloads perform better in terms of bandwidth.

Summary of Results:

1. **Small Block Sizes (4K Access)**
   - **IOPS**: Very high, peaking at 12,456 in a mixed workload (50% read/50% write) with a queue depth of 4.
   - **Latency**: Low across all tests, ranging from 0.089 ms to 0.115 ms, making this ideal for operations requiring fast responses.
   - **Throughput**: Moderate for writes, with a peak throughput of 48.85 MB/s in a 50% read/50% write scenario. Throughput is not as high compared to larger block sizes due to the focus on handling numerous small transactions.
2. **Medium Block Sizes (16K & 32K Access)**
   - **IOPS**: IOPS decrease as block size increases, reaching around 9,250 for read-only workloads with 16K and slightly lower for 32K. However, with larger block sizes, the system focuses more on data volume than operation count.
   - **Latency**: Ranges from 0.093 ms to 0.134 ms for 16K, and from 0.110 ms to 0.132 ms for 32K. Despite higher latencies than smaller block sizes, the system remains responsive for these larger data operations.
   - **Throughput**: Throughput increases with block size, especially for write-heavy operations, peaking at 273.22 MB/s for 32K block sizes and queue depth of 16.
3. **Large Block Sizes (128K Access)**

    - **IOPS**: Significantly lower than smaller block sizes due to the large data being transferred per operation, ranging between 4,923 and 6,845.
    - **Latency**: Latency is the highest for 128K block sizes, ranging from 0.145 ms to 0.182 ms, but this is still acceptable for workloads requiring large data transfers.
    - **Throughput**: This setup excels in throughput, peaking at 781.25 MB/s for 70% read/30% write workloads with a queue depth of 128. This makes it optimal for sequential workloads and high-bandwidth applications.
4. **Effect of Queue Depth**
    - Increasing queue depth generally leads to slightly lower IOPS and higher latency but significantly boosts throughput, especially with larger block sizes. Higher queue depths benefit applications requiring sustained data transfer rather than fast individual transactions.

Conclusion:

- **High IOPS and Low Latency**: Best achieved with 4K and 16K block sizes in read-heavy or mixed workloads.
- **High Throughput**: Achieved with 128K block sizes, especially for write-heavy or mixed workloads, and with high queue depths.
- **Balanced Performance**: For a mix of both throughput and IOPS, 16K and 32K block sizes with moderate queue depths (16-512) offer an ideal middle ground.