The trade-offs presented by write-optimized indices in comparison to B+ tree indices are as follows:

Write-Optimized Indices:

Advantages:

- Enhanced write throughput: These indices are specifically designed to efficiently handle large volumes of write operations.
- Buffer writes in memory then flush to disk in sequential manner, minimizing I/O

Disadvantage:

- Read latency/compaction costs: since data may be spread across multiple levels/components, read may require accessing multiple data structures

B+ Trees:

Advantages:

- Consistent and efficient read performance: B+ trees offer predictable read operations, making them suitable for various applications, including online transaction systems.
- Fast lookup times: By maintaining keys in sorted order and facilitating data retrieval through tree traversal, B+ trees ensure swift access to data.

Disadvantage:

- Write overhead: each insert update or delete op may require multiple disk I/O operations