Part C- Analysis

Report: B+Tree Indexing Analysis

Introduction

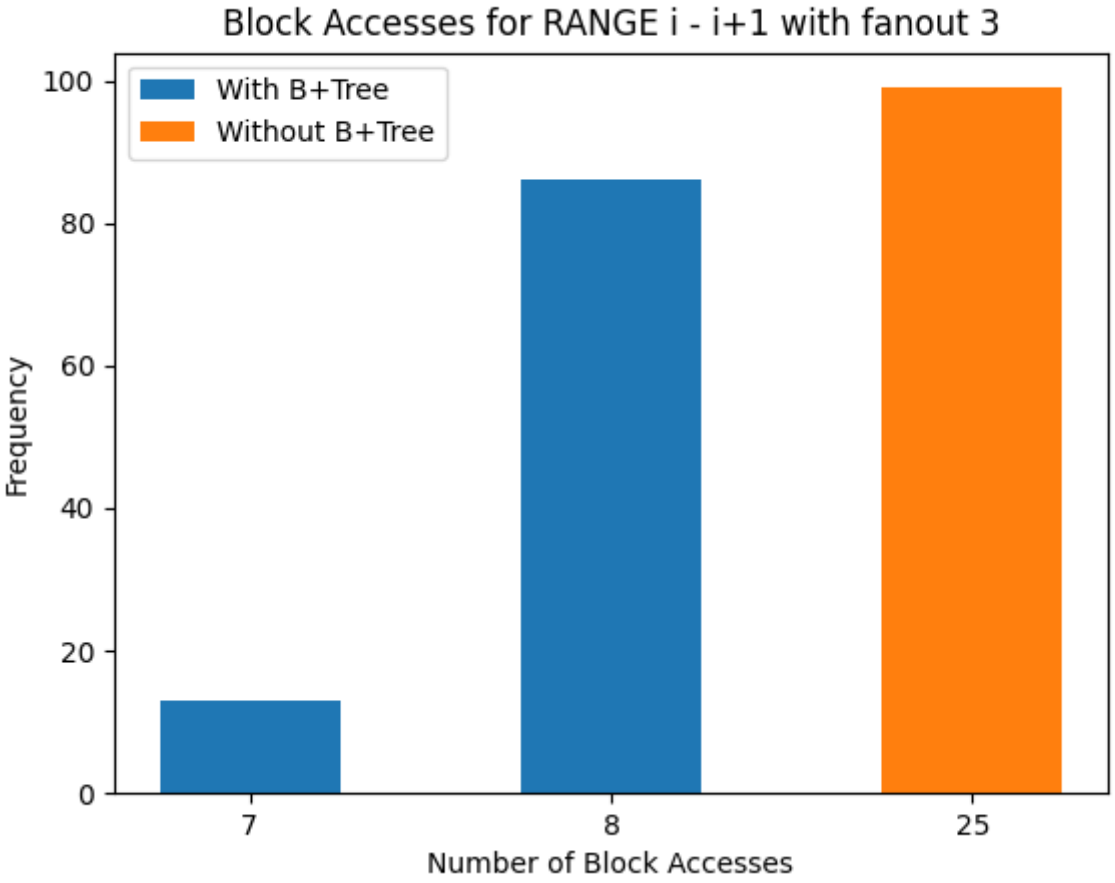
In this report, we analyze the performance of B+Tree indexing with different values of `FANOUT` (fanout factor) to understand its impact on the number of block accesses required for RANGE queries. We conducted experiments with `FANOUT` values of 3, 10, and 15, and the results are presented and discussed in this report.

Experimental Setup

- `FANOUT` Values:
  - FANOUT = 3
  - FANOUT = 10
  - FANOUT = 15
- Number of Queries:
  - We simulated 100 RANGE queries from 1 to 100 for each `FANOUT` value.
- Block Accesses:
  - We recorded the number of block accesses for each query with and without the B+Tree indexing.

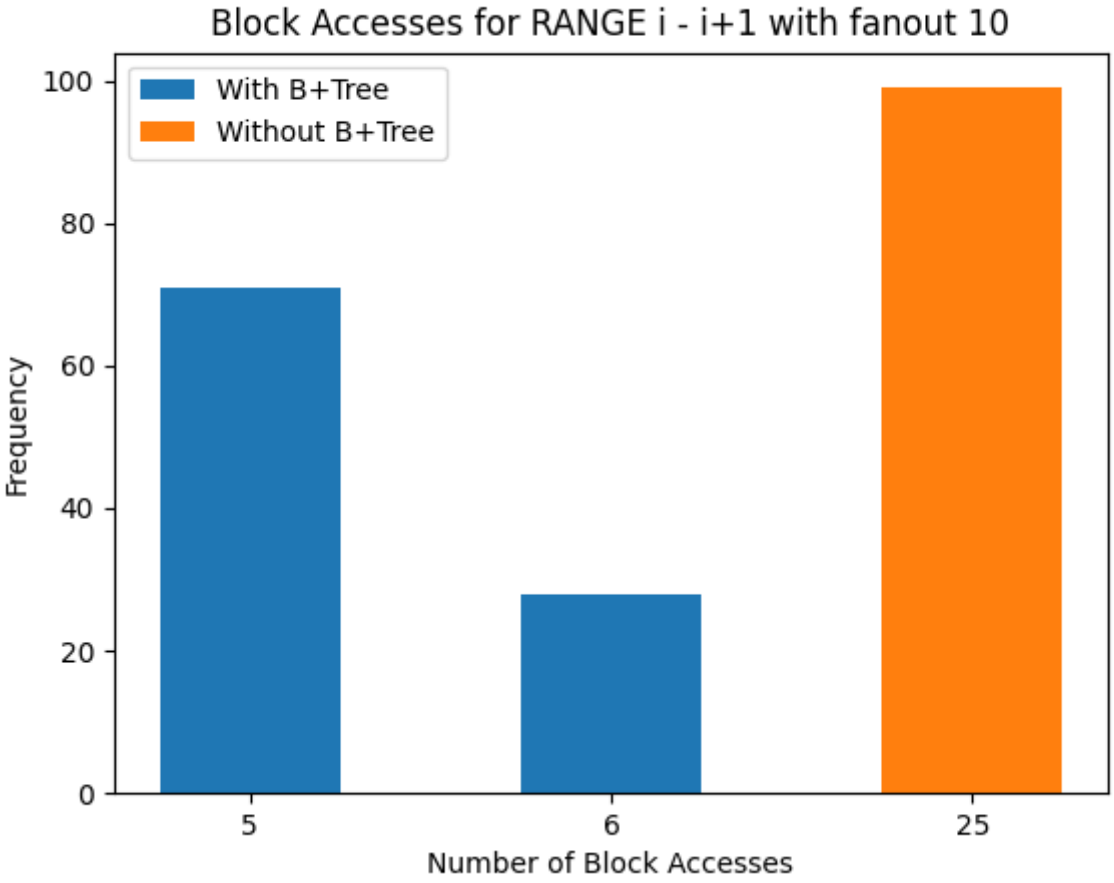
Results and Analysis

FANOUT = 3



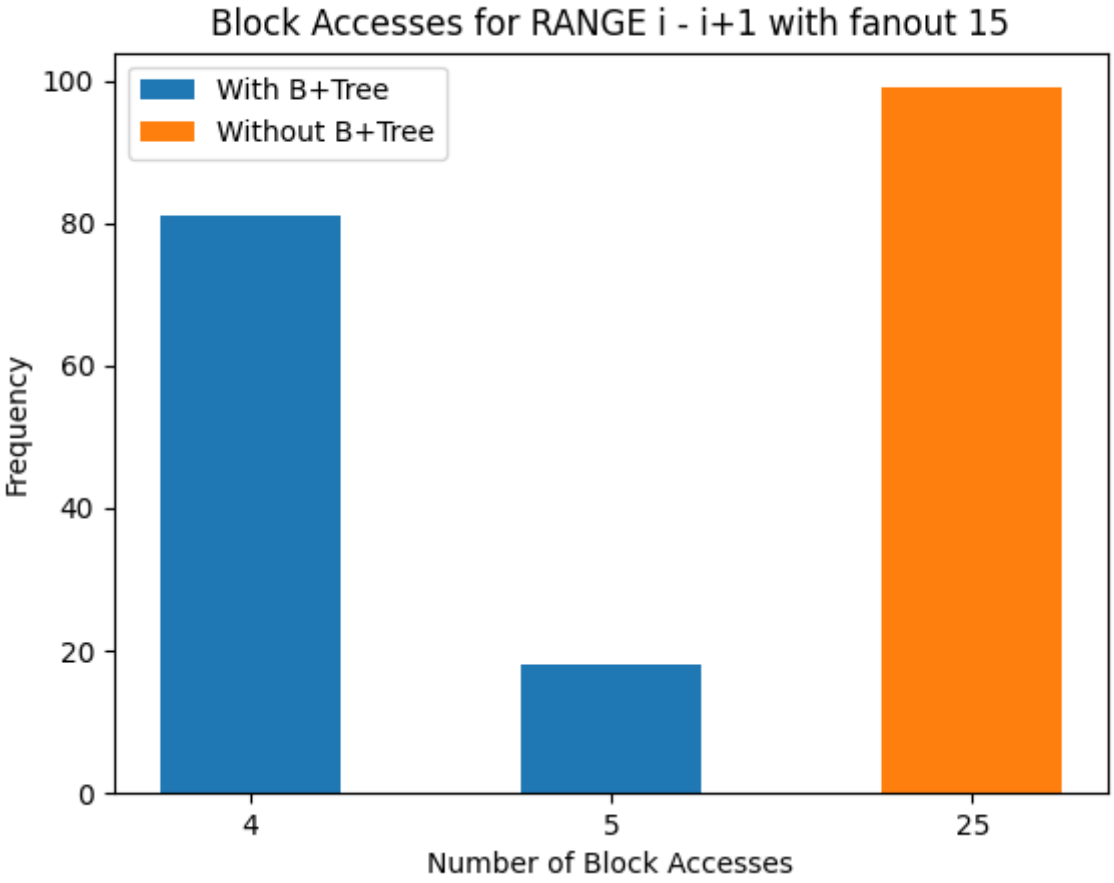
- Observations:
  - With a small `FANOUT` of 3, we observed that the number of block accesses was relatively high in the B+ Tree
  - The B+Tree showed improvement over direct access, but there's lot of room for improvement.
  - The performance benefit of the B+Tree was limited due to the small fanout.

FANOUT = 10



- Observations:
  - With a moderate `FANOUT` of 10, we noticed a noticeable improvement in the number of block accesses for B+Tree.
  - The B+Tree indexing effectively reduced the block accesses, demonstrating its effectiveness in managing data.
  - The reduction in block accesses was more prominent, especially when the data is distributed over a larger fanout.

FANOUT = 15



- Observations:
  - With a higher `FANOUT` of 15, the B+Tree indexing showed remarkable performance.
  - The number of block accesses with the B+Tree was significantly lower compared to direct access.
  - B+Tree indexing with a larger fanout efficiently organized the data, leading to fewer block accesses.

Conclusion

The choice of `FANOUT` in B+Tree indexing plays a crucial role in the performance of data retrieval operations. In our experiments, a larger `FANOUT` value, such as 15, led to a substantial reduction in block accesses, making B+Tree indexing highly effective for range queries. However, with a smaller `FANOUT`, the performance improvement was limited.

It is essential to choose an appropriate `FANOUT` value based on the characteristics of the dataset and the specific use case to harness the full potential of B+Tree indexing.