How to Optimize Slow Queries in PostgreSQL

Optimizing slow queries in PostgreSQL is a critical aspect of improving database performance. Here are several tips and recommendations to help you optimize slow queries effectively:

Step 1) Enable Slow Query Logging in PostgreSQL config file.

```
log_min_duration_statement = 5000 # Log queries that take longer than 5000ms (5 seconds)
log_line_prefix = '%m %p %q%u@%d '
```

Step 2) Analyze Query Execution Plan

```
EXPLAIN ANALYZE SELECT * FROM emp2024 WHERE dno = 10 ORDER BY esal DESC LIMIT 10;
```

Key points to note:

- Index Scan: Indicates that an index is being used, which is good for performance.
- Cost: The planners estimate of the query execution cost.
- Actual Time: The actual time spent on execution.

Step 3) Optimizing Based on the Execution Plan.

```
CREATE INDEX idx_emp2024_department_esal ON emp2024(dno, esal DESC);
```

Step 4) Rewrite the Query

```
SELECT employee_id, name, esal FROM emp2024 WHERE dno = 10 ORDER BY esal DESC LIMIT 10;
```

Step 5) Use LIMIT Early in Subqueries

```
SELECT * FROM (SELECT * FROM emp2024 WHERE dno = 10 LIMIT 100) AS subquery
```

ORDER BY esal DESC LIMIT 10;

Step 6) Vacuum and Analyze the Database

VACUUM ANALYZE;

Step 7) Optimize PostgreSQL Configuration

- work_mem: (Total RAM * 1% to 2% of available RAM) / Max Expected Concurrent Queries
- shared_buffers: 20-35% of available RAM.
- effective_cache_size: 50-75% of RAM.

Step 8) Partitioning large tables to reduce query time.

Step 9) Avoid Locks - Use FOR UPDATE and FOR SHARE to avoid deadlocks and minimize long transactions.

Step 10) Use Connection Pooling to manage DB connections efficiently.

Step 11) Materialized Views - Cache complex query results with materialized views.

Step 12) Monitor slow queries using pg_stat_activity view or pg_stat_statements extension.

Step 13) Leverage Parallel Queries - Enable parallel query execution for large queries.

Step 14) Use read replicas to offload read-heavy queries.

Step 15) Log temp file usage (log_temp_files = 0) for optimization.

By following these tips, you can identify and fix performance issues in PostgreSQL queries effectively.