Performance Benchmarking of Oracle 21C

Step 1: Download Oracle Database 21C

Visit the Oracle Software website (https://www.oracle.com/database/technologies/oracle21c-windows-downloads.html) to download Oracle Database 21C.

Step 2: Install Oracle Database

Follow the official installation guides and reference the Oracle Database 19c documentation (https://www.oracle.com/database/technologies/) for detailed steps.

Step 3: Verify Installation

Ensure the installation is successful by logging into the database as 'sys dba'. Properly configure the 'listener.ora' and 'tnsnames.ora' files to establish database connectivity.

Step 4: Set Up a Database User

Create a new user and grant all necessary privileges and resources to this user. Example commands:

```
CREATE USER my_user IDENTIFIED BY my_password;
GRANT ALL PRIVILEGES TO my_user;
ALTER USER my_user QUOTA UNLIMITED ON USERS;
```

Step 5: Download and Set Up SwingBench

Download the SwingBench tool (https://www.dominicgiles.com/swingbench/#about-swingbench) to simulate workloads and perform benchmarking.

Step 6: Load the Simple Order Entry Schema

Use SwingBench to load the 'Simple Order Entry' schema into your Oracle Database.

Step 7: Optimize the Schema

Index Creation: Enhance query performance by creating indexes. Example:

```
CREATE INDEX idx_logon_id ON LOGON(LOGON_ID);
CREATE INDEX idx_user_id ON LOGON(USER_ID);
CREATE INDEX idx_last_logon ON LOGON(LAST_LOGON);
```

For bulk index creation:

```
BEGIN
  FOR cols IN (SELECT COLUMN_NAME, TABLE_NAME FROM ALL_TAB_COLUMNS WHERE TABLE_NAME IN ('CUSTOMERS', 'ADDRESSES', ...)) LOOP
    EXECUTE IMMEDIATE 'CREATE INDEX idx_' || cols.TABLE_NAME || '_' || cols.COLUMN_NAME || ' ON ' || cols.TABLE_NAME || '(' || cols.COLUMN_NAME || ')';
  END LOOP;
END;
```

Partitioning: Improve query efficiency by partitioning tables:
- Range Partitioning for the orders table based on order_date:

```
CREATE TABLE orders_partitioned (
   order_id NUMBER,
   customer_id NUMBER,
   order_date DATE,
   order_status VARCHAR2(10),
   order_amount NUMBER
)
PARTITION BY RANGE (order_date) (
   PARTITION p1 VALUES LESS THAN (TO_DATE('2022-01-01', 'YYYY-MM-DD')),
   PARTITION p2 VALUES LESS THAN (TO_DATE('2023-01-01', 'YYYY-MM-DD')),
   PARTITION p3 VALUES LESS THAN (TO_DATE('2024-01-01', 'YYYY-MM-DD'))
);
```

- Hash Partitioning for the customers table based on customer_id:

```
CREATE TABLE customers_partitioned (
   customer_id NUMBER,
   customer_name VARCHAR2(100),
   email VARCHAR2(100),
   phone VARCHAR2(15)
)
PARTITION BY HASH (customer_id) PARTITIONS 4;
```

Step 8: Run Benchmarks with SwingBench

Open SwingBench and follow the on-screen instructions to connect to your database using the configured user credentials.

Step 9: Configure Workloads

Experiment with different configurations:
- Set up user concurrency.
- Adjust inter- and intra-think times.
- Simulate varied workload scenarios.

Step 10: Analyze Benchmark Results

Examine key performance metrics such as:
- Transactions Per Second (TPS).
- Response times (average, 50th percentile, 90th percentile).
- Resource utilization (CPU, disk I/O).

Step 11: Visualize Results

Utilize provided scripts in the repository to generate visualizations of benchmark outcomes for easier interpretation.

Step 12: Derive Insights

Analyze the results to identify performance bottlenecks, scalability limits, and areas for optimization. Provide actionable recommendations for improving database performance based on observed trends and metrics.