PostgreSQL Performance Tuning: A Comprehensive Guide

TheDBAdmin Team • January 30, 2025

postgresql performance database-administration optimization tuning

Performance tuning is crucial for maintaining a healthy and responsive PostgreSQL database. This comprehensive guide covers essential techniques and best practices for optimizing your PostgreSQL database performance.

Table of Contents

Query Optimization
Indexing Strategies
Configuration Tuning
Memory Management
Monitoring and Analysis
Maintenance Operations

Query Optimization

EXPLAIN ANALYZE

```
-- Basic EXPLAIN ANALYZE
EXPLAIN ANALYZE
SELECT * FROM orders
WHERE order_date >= '2025-01-01'
```

```
AND customer_id IN (
    SELECT id FROM customers WHERE country = 'USA'
);

-- Show buffers and timing information
EXPLAIN (ANALYZE, BUFFERS, TIMING)
SELECT * FROM orders
WHERE order_date >= '2025-01-01';
```

Query Optimization Techniques

```
-- Use EXISTS instead of IN for better performance
-- Before
SELECT * FROM orders
WHERE customer_id IN (SELECT id FROM customers WHERE country = 'USA');

-- After
SELECT * FROM orders o
WHERE EXISTS (
    SELECT 1 FROM customers c
    WHERE c.id = o.customer_id
    AND c.country = 'USA'
);

-- Use JOIN instead of correlated subqueries
-- Before
SELECT *,
    (SELECT COUNT(*) FROM order_items oi WHERE oi.order_id = o.id)
FROM orders o;

-- After
SELECT o.*, COUNT(oi.id)
FROM orders o
LEFT JOIN order_items oi ON oi.order_id = o.id
GROUP BY o.id;
```

Indexing Strategies

Index Types

```sql
-- B-tree index (default)
CREATE INDEX idx_orders_date ON orders(order_date);

-- Partial index
CREATE INDEX idx_orders_status ON orders(status)
WHERE status IN ('pending', 'processing');

-- Multi-column index
CREATE INDEX idx_orders_customer_date ON orders(customer_id, order_date)

-- Expression index
CREATE INDEX idx_lower_email ON customers(LOWER(email));

-- BRIN index for sequential data
CREATE INDEX idx_orders_date_brin ON orders USING BRIN(order_date);
```

Index Maintenance

```sql
-- Find unused indexes
SELECT
    schemaname || '.' || tablename as table_name,
    indexname,
    idx_scan,
    idx_tup_read,
    idx_tup_fetch
FROM pg_stat_user_indexes
WHERE idx_scan = 0
AND schemaname NOT IN ('pg_catalog', 'pg_toast')
ORDER BY pg_relation_size(indexrelid) DESC;

-- Reindex table
REINDEX TABLE orders;

-- Concurrent reindex (no lock)
CREATE INDEX CONCURRENTLY idx_new_index ON orders(column_name);
DROP INDEX CONCURRENTLY idx_old_index;
```

Configuration Tuning

Memory Settings

```
# postgresql.conf

# Memory Configuration
shared_buffers = 2GB          # 25% of RAM for dedicated servers
work_mem = 16MB              # Depends on max_connections
maintenance_work_mem = 256MB  # For maintenance operations
effective_cache_size = 6GB    # 75% of RAM for dedicated servers

# Query Planning
random_page_cost = 1.1       # For SSD storage
effective_io_concurrency = 200 # For SSD storage
```

Connection Settings

```
# postgresql.conf

# Connection Settings
max_connections = 100
superuser_reserved_connections = 3

# Statement Timeout
statement_timeout = '1min'
lock_timeout = '10s'
idle_in_transaction_session_timeout = '1min'
```

Memory Management

Vacuum Settings

```
# postgresql.conf

# Autovacuum Configuration
autovacuum = on
autovacuum_vacuum_scale_factor = 0.1
autovacuum_analyze_scale_factor = 0.05
```

```
autovacuum_vacuum_cost_delay = 2ms
autovacuum_vacuum_cost_limit = 200
```

Buffer Cache Management

```
-- Check buffer cache hit ratio
SELECT
    sum(heap_blks_read) as heap_read,
    sum(heap_blks_hit)  as heap_hit,
    sum(heap_blks_hit) / (sum(heap_blks_hit) + sum(heap_blks_read))::flo
FROM pg_statio_user_tables;

-- Find tables with low cache hit ratio
SELECT
    schemaname,
    relname,
    heap_blks_read,
    heap_blks_hit,
    heap_blks_hit::float / (heap_blks_read + heap_blks_hit) as hit_ratio
FROM pg_statio_user_tables
WHERE heap_blks_read + heap_blks_hit > 0
ORDER BY hit_ratio ASC;
```

Monitoring and Analysis

Performance Monitoring

```
-- Monitor active queries
SELECT
    pid,
    age(clock_timestamp(), query_start) as duration,
    usename,
    query
FROM pg_stat_activity
WHERE state != 'idle'
AND query NOT ILIKE '%pg_stat_activity%'
ORDER BY duration DESC;

-- Find slow queries
```

```
SELECT
    substring(query, 1, 50) as short_query,
    round(total_time::numeric, 2) as total_time,
    calls,
    round(mean_time::numeric, 2) as mean_time,
    round((100 * total_time / sum(total_time::numeric) over ())::numeric
FROM pg_stat_statements
ORDER BY total_time DESC
LIMIT 10;
```

Table Statistics

```
-- Table size and bloat
SELECT
    schemaname,
    tablename,
    pg_size_pretty(pg_total_relation_size(schemaname || '.' || tablename
    pg_size_pretty(pg_table_size(schemaname || '.' || tablename)) as tab
    pg_size_pretty(pg_indexes_size(schemaname || '.' || tablename)) as i
    pg_size_pretty(
        pg_total_relation_size(schemaname || '.' || tablename) -
        pg_table_size(schemaname || '.' || tablename)
    ) as bloat_size
FROM pg_tables
WHERE schemaname NOT IN ('pg_catalog', 'information_schema')
ORDER BY pg_total_relation_size(schemaname || '.' || tablename) DESC;
```

Maintenance Operations

Regular Maintenance Tasks

```
-- Analyze tables
ANALYZE VERBOSE;

-- Update table statistics
ANALYZE VERBOSE mytable;

-- VACUUM tables
```

```
VACUUM (VERBOSE, ANALYZE) mytable;

-- Reindex database
REINDEX DATABASE mydb;
```

Maintenance Schedule

```
-- Create maintenance function
CREATE OR REPLACE FUNCTION perform_maintenance()
RETURNS void AS $$
BEGIN
    -- Vacuum analyze all tables
    VACUUM (ANALYZE, VERBOSE);

    -- Update statistics
    ANALYZE VERBOSE;

    -- Reindex specific tables if needed
    REINDEX TABLE frequently_updated_table;
END;
$$ LANGUAGE plpgsql;

-- Schedule maintenance (using cron or similar)
SELECT perform_maintenance();
```

Performance Tuning Checklist

Query Optimization

- [ ] Use EXPLAIN ANALYZE
- [ ] Optimize JOIN operations
- [ ] Use appropriate subquery types

Indexing

- [ ] Create necessary indexes
- [ ] Remove unused indexes
- [ ] Use appropriate index types

Configuration

- [ ] Optimize memory settings
- [ ] Configure autovacuum
- [ ] Set appropriate timeouts

Monitoring

- [ ] Track slow queries

- [ ] Monitor cache hit ratios
- [ ] Check for bloat

Maintenance

- [ ] Regular VACUUM
- [ ] Update statistics
- [ ] Reindex when needed

Further Reading

PostgreSQL Official Documentation
TheDBAdmin PostgreSQL Security Guide
50 Essential PostgreSQL Queries
PostgreSQL DBA Course

Remember to test performance changes in a development environment first and monitor the impact of each optimization.

Share this article

Tags

oracle (1) postgresql (13) aws-dms (1) ora2pg (1) database-migration (1) cloud-migration (1) database-administration (11) ai (6) machine-learning (4) dba (2) career-development (2) data-science (2) mlops (1) qwen (1) local-deployment (2) llm (2) ollama (2) deepseek (3) nosql (1) career-guide (1) mongodb (2) cassandra (1) redis (2) docker (2) docker-compose (2) containerization (2) monitoring (1) performance (5) devops (1) backup (1) recovery (1) disaster-recovery (2) caching (1) database (3) high-availability (1) replication (1) failover (1) scalability (1) high-concurrency (1) database-tuning (1) optimization (1) tuning (1) security (1) encryption (1) authentication (1) sql (1) tutorials (1) automation (1) productivity (1) openai (1) claude (1) comparison (1) technology (1) career (1) certification (1) welcome (1) introduction (1)

Recent Posts

Oracle to PostgreSQL Migration: Comprehensive Guide Using AWS DMS & Ora2Pg
Jan 31, 2025

PostgreSQL DBA's Role in AI & ML: A Comprehensive Guide to the Future
Jan 31, 2025

Running Qwen 2.5 Locally: Complete Implementation Guide
Jan 30, 2025

Running DeepSeek Models Locally with Ollama: Complete Guide
Jan 30, 2025

How to Become a NoSQL Database Administrator: Complete Career Guide
Jan 30, 2025

TheDBAdmin
Database Administration Services

Expert database administration and consulting services for modern enterprises.

Services

Database Services

Courses

Blog

Company

About Us

Contact

Support

Documentation

Help Center

Status

Stay Updated

Subscribe to our newsletter for the latest updates and insights.

Enter your email

Subscribe

Privacy Policy Terms of Service Cookie Policy