PostgreSQL Performance Tuning - TTDB7027

Optimize PostgreSQL databases for peak performance through advanced tuning, query optimization, and resource management.

Duration: 3 Days
Skill Level: Intermediate
Available Format: Instructor-Led Online ; On Public Schedule

PostgreSQL is a powerful, open-source object-relational database management system that emphasizes extensibility, data integrity, and high performance.

What You'll Learn

Overview

PostgreSQL is a powerful, open-source object-relational database management system that emphasizes extensibility, data integrity, and high performance. Its robust architecture and flexibility make it a top choice for database administrators and engineers managing data-intensive applications across various industries. PostgreSQL's advanced features, such as full-text search, spatial data support, and customizable data types, allow organizations to design efficient, scalable, and reliable solutions tailored to their specific needs.

This five-day PostgreSQL Performance Tuning course is specifically tailored for database professionals tasked with optimizing PostgreSQL systems for performance. Whether you're managing high-volume transactional workloads (OLTP), analytical workloads (OLAP), or hybrid environments, this course equips participants with the expertise to identify bottlenecks, tune system configurations, optimize query performance, and enhance overall database efficiency.

The course provides a mix of conceptual understanding and practical application, led by an expert instructor. Throughout the event you will delve into PostgreSQL's architecture, explore workload-specific tuning strategies, and gain hands-on experience with tools and techniques to analyze, monitor, and improve performance. Advanced topics, such as parallel query execution, replication, and caching, are covered alongside routine maintenance and best practices for high availability and reliability.

By the end of this course, you will have a deep understanding of PostgreSQL performance tuning principles and will be prepared to implement these techniques in real-world environments to achieve optimal database performance and operational excellence.

Objectives

Working in a hands-on learning environment, guided by an expert instructor, **you will learn to:**

- Understand PostgreSQL Architecture and Performance Concepts: Gain a foundational understanding of PostgreSQL's internal architecture and key performance tuning principles, tailored to both OLTP and OLAP workloads.
- Optimize Database Design: Learn to design efficient databases using advanced indexing strategies, partitioning techniques, and constraint management to minimize performance overhead.
- Improve Query Performance: Master tools like EXPLAIN and EXPLAIN ANALYZE to analyze and optimize queries, identify slow-running operations, and resolve common query performance issues.
- Fine-Tune PostgreSQL Configuration Settings: Adjust memory, disk I/O, connection settings, and autovacuum parameters for maximum efficiency based on workload requirements.
- Apply Advanced Optimization Techniques: Explore advanced topics such as parallel query execution, caching strategies, and replication to enhance database scalability and responsiveness.
- Monitor and Maintain Performance: Use PostgreSQL's built-in monitoring tools and third-party solutions to proactively manage performance, resolve issues, and conduct routine maintenance.
- Implement High Availability and Reliability: Configure replication, load balancing, and failover mechanisms to ensure database availability and resilience under high-demand conditions.

Audience

This course is designed for database administrators, system engineers, and IT professionals responsible for maintaining and optimizing PostgreSQL databases. The course is also relevant for data architects and performance specialists managing mission-critical PostgreSQL environments. Participants should have a foundational understanding of relational databases and SQL. Familiarity with PostgreSQL basics, such as installation, database creation, and query execution, is recommended but not required.

Pre-Requisites

Participants should have a foundational understanding of relational databases and SQL, as these are essential for interpreting and optimizing queries. Familiarity with PostgreSQL basics, such as installation, database creation, query execution, and server configuration, is recommended to fully engage with the course's advanced performance tuning tasks.

TTDB7020	PostgreSQL for Database Developers
TTSQL002	Introduction to SQL Programming Basics

Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We'll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change, and may adjust during live delivery based on audience skill level, interests and participation.

1. Introduction

- **Introduction to PostgreSQL Performance Tuning**
- Overview of PostgreSQL architecture
- Importance of performance tuning
- Key areas of performance tuning: Database design, queries, configuration settings, system resources

2. Understanding PostgreSQL Workload

- Types of workloads: OLTP (Online Transaction Processing) vs. OLAP (Online Analytical Processing)
- Identifying performance bottlenecks: CPU, memory, disk I/O, query execution time
- Tools for monitoring performance: pg_stat_statements, pg_stat_activity, EXPLAIN, EXPLAIN ANALYZE

3. Database Design Optimization

- **Indexes:**

- Types of indexes (B-tree, GIN, GiST, BRIN, etc.)
- Creating appropriate indexes for frequently queried columns
- Impact of excessive indexing
- Monitoring index usage and maintenance (e.g., REINDEX, VACUUM)

- **Partitioning:**
- Partitioning strategies (Range, List, Hash)
- Use cases and benefits of partitioning large tables
- Managing partitions with pg_partman

- **Foreign Keys and Constraints:**
- Impact on query performance
- Optimizing constraint enforcement

4. Query Optimization

- **Using EXPLAIN and EXPLAIN ANALYZE:**
- Analyzing query execution plans
- Identifying slow queries and bottlenecks

- **Common Query Performance Problems:**
- Missing indexes
- Inefficient joins (Nested Loop vs. Hash Join vs. Merge Join)
- Suboptimal WHERE clauses (lack of index utilization)

- **Query Rewriting:**
- Using CTE (Common Table Expressions) vs. subqueries
- Refactoring queries for better performance

- **Vacuuming and Analyzing Tables:**
- Importance of VACUUM and ANALYZE in maintaining performance
- Autovacuum configuration and tuning
- Handling dead tuples and table bloat

5. Configuration Tuning

- **Memory Settings:**
- shared_buffers
- work_mem

- maintenance_work_mem
- effective_cache_size
- Tuning for different workload types (OLTP vs OLAP)

- **Disk I/O:**
- wal_buffers and Write-Ahead Log (WAL) configuration
- checkpoint_segments and checkpoint_completion_target
- fsync settings
- Disk subsystems and RAID configurations for PostgreSQL

- **Connection Settings:**
- max_connections
- Connection pooling (e.g., PgBouncer)
- superuser_reserved_connections

- **Autovacuum and Vacuum Configuration:**
- Autovacuum thresholds: autovacuum_vacuum_scale_factor, autovacuum_analyze_scale_factor
- Autovacuum tuning for large tables

6. Advanced Performance Techniques

- **Parallel Query Execution:**
- Enabling and tuning parallel queries (max_parallel_workers, max_parallel_workers_per_gather)
- Benefits and limitations of parallelism

- **Read and Write Optimization:**
- Use of synchronous vs. asynchronous replication
- Write-optimized storage (e.g., SSD vs. HDD)
- Clustering and load balancing

- **Caching:**
- Query caching with tools like pg_bench or external caching layers (Redis, Memcached)
- Query result caching with pg_cache or custom solutions

7. System Resource Optimization

- **CPU Usage Optimization:**

- Managing CPU resources for PostgreSQL processes
- Optimizing parallel processing

- **Memory Management:**
- PostgreSQL's memory consumption and tuning memory settings
- Managing shared_buffers vs. OS cache

- **Disk I/O Optimization:**
- Tuning disk throughput and latency
- Using fast storage options (SSD vs HDD)

- **Networking:**
- Optimizing network performance for remote clients and replication
- Adjusting tcp_keepalives and connection pooling

8. Replication and High Availability

- **Replication Setup:**
- Streaming replication (master-slave)
- Logical replication for selective data sync

- **High Availability Solutions:**
- Failover mechanisms: Patroni, pgpool-II, repmgr
- Load balancing strategies in replicated environments

- **Replication Tuning:**
- WAL-related tuning (synchronous_commit, wal_level)
- Replication lag and its effects on performance

9. Monitoring and Maintenance

- **Performance Monitoring Tools:**
- pg_stat_activity, pg_stat_statements, pg_stat_user_indexes
- Third-party tools: pgBadger, pgtune, pgAdmin

- **Logging and Alerting:**
- Configuring log_min_duration_statement, log_statement
- Monitoring slow queries and locking issues

- **Routine Maintenance:**
- Scheduled backups and performance-impacting tasks
- Handling PostgreSQL upgrades and performance regression testing

Related Courses

TTDB7020	PostgreSQL for Database Developers
TTDB7024	PostgreSQL Server Administration
TTSQL002	Introduction to SQL Programming Basics

For More Information

Please contact us or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.