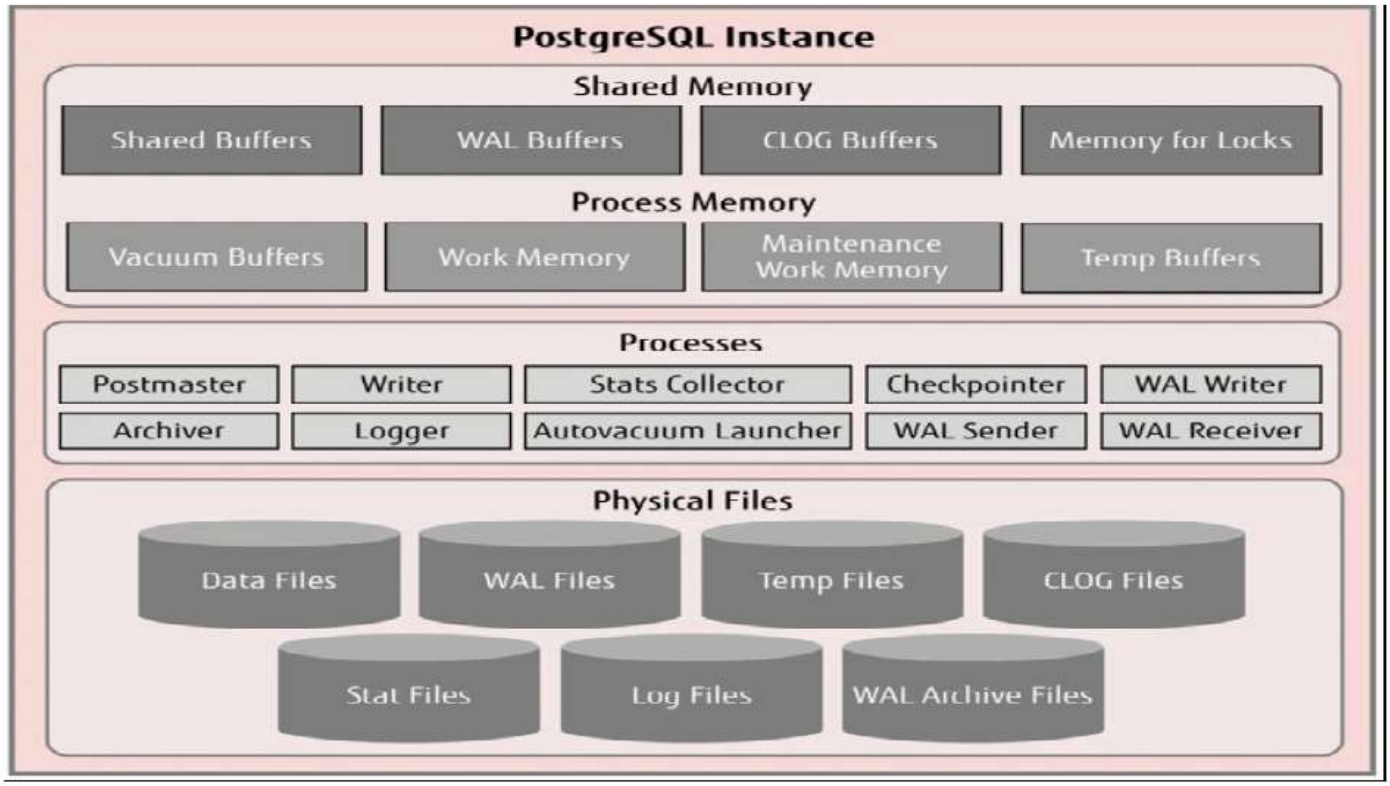

PostgreSQL Server Memory Management

Asad Ali
Database Adiministrator | Database Developer | Database Architect | PostgreSQL | Oracle Database | MySQL | MSSQL | Data Migration | Server/Database Performance | Web Scraping Master | ETL | ELT

Published Jul 18, 2024

Server Memory Allocation

Server memory allocation involves configuring the memory parameters for PostgreSQL to ensure efficient use of system resources and optimal performance. It primarily focuses on settings within PostgreSQL's configuration file (postgresql.conf).

Key Parameters:

1. **shared_buffers:** This parameter defines the amount of memory that PostgreSQL uses for caching data. A larger buffer can reduce disk I/O by keeping frequently accessed data in memory.

Example:

shared_buffers = 4GB

Explanation: If you have 16GB of RAM, setting shared_buffers to 4GB is a good starting point, as it allows PostgreSQL to cache more data in memory, thereby reducing the need for frequent disk access.

2. **work_mem:** This parameter determines the amount of memory allocated for internal sort operations and hash tables before writing to temporary disk files. It's per operation, not per query.

Example:

work_mem = 64MB

Explanation: For complex queries involving sorts, large joins, and aggregations, increasing work_mem can improve performance by keeping operations in memory rather than spilling to disk.

3. **maintenance_work_mem:** This setting specifies the amount of memory used for maintenance tasks such as VACUUM, CREATE INDEX, and ALTER TABLE.

Example:

maintenance_work_mem = 1GB

Explanation: Increasing maintenance_work_mem allows maintenance operations to complete faster, as more data can be processed in memory.

Virtual Memory

Virtual memory allows the operating system to use disk space as an extension of RAM, enabling the system to handle larger workloads than physical memory alone would allow.

Key Concepts:

1. **Swap Space:** Swap space is a portion of the disk set aside to extend RAM. When physical memory is exhausted, the OS moves inactive pages to swap space.

Example: Configuring swap space on Linux:

dd if=/dev/zero of=/swapfile bs=1G count=8

mkswap /swapfile

swapon /swapfile

Explanation: Creating an 8GB swap file provides additional memory space, but swapping is slower than physical RAM and should be minimized for performance-critical applications like PostgreSQL.

2. **vm.overcommit_memory:** This kernel parameter controls the behavior of the virtual memory manager.**Values**:0: Heuristic overcommit handling.1: Always overcommit.2: Never overcommit unless it's safe.

Example:

sysctl -w vm.overcommit_memory=2

Explanation: Setting vm.overcommit_memory to 2 ensures that the system only allows memory allocation if there is sufficient swap space, which helps prevent out-of-memory errors.

OS Cache

OS Cache refers to the use of system memory by the operating system to cache frequently accessed disk blocks, reducing the need for physical disk I/O.

Key Parameters:

1. **effective_cache_size:** This parameter gives PostgreSQL an estimate of the effective size of the disk cache available to the database server.

Example:

effective_cache_size = 12GB

Explanation: For a server with 16GB of RAM, setting effective_cache_size to 12GB informs the query planner that a large portion of the data can be cached, helping it make more informed decisions about query execution plans.

2. **vm.swappiness:** This kernel parameter controls the tendency of the kernel to swap out memory pages.**Values**: Ranges from 0 (least swappable) to 100 (most swappable).

Example:

sysctl -w vm.swappiness=10

Explanation: Lowering vm.swappiness to 10 reduces the tendency of the OS to swap, ensuring that more memory is available for OS cache and PostgreSQL buffers, which improves performance.

Solution Memory

Solution Memory refers to the memory allocated for specific solutions or applications running on the server, ensuring they have sufficient resources for optimal performance.

Key Considerations:

1. **Dedicated Memory Allocation:** Allocate specific memory resources to PostgreSQL to avoid contention with other applications.

Example: On a server running both PostgreSQL and a web server, ensure each has dedicated memory.

Explanation: Isolate memory allocation to ensure PostgreSQL has consistent access to necessary resources, avoiding performance degradation due to resource contention.

2. **NUMA (Non-Uniform Memory Access):Description**: On NUMA systems, ensure PostgreSQL is configured to make optimal use of available memory nodes.

Example:

numactl --interleave=all postgres

Explanation: Using numactl to interleave memory across NUMA nodes can improve memory access patterns, reducing latency and improving performance.

Summary

Efficient memory management is crucial for PostgreSQL performance. By understanding and configuring server memory allocation, virtual memory, OS cache, and solution-specific memory settings, DBA's can ensure optimal performance and stability for their PostgreSQL instances. Proper tuning and isolation of resources are key to handling complex workloads and achieving high performance.

To view or add a comment, **sign in**

More articles by Asad Ali

Dec 14, 2024
PostgreSQL High Availability (Always On & Failover Clustering)
PostgreSQL offers several High Availability (HA) and disaster recovery solutions, but it doesn't have a direct feature...

Nov 6, 2024
How to Set Up Connectivity Between CUBRID and JMeter
1. Installations Install CUBRID Install CUBRID using the official installer.
1

Aug 5, 2024
Authentication Methods in PostgreSQL
PostgreSQL supports several authentication methods to ensure that only authorized users can access the database. Here's...
8

Jun 7, 2024
range_agg: A New Feature in PostgreSQL 14
The range_agg function in PostgreSQL 14 is a new aggregate function that allows you to create ranges from aggregated...
3

Show more

Insights from the community

Data Management
How can you optimize PostgreSQL for read-heavy workloads?

Computer Hardware
What impact do hardware choices have on SQL Server performance?

DBMS
How do you secure and backup your data in MongoDB and Cassandra?

Database Administration

What metrics are crucial for maintaining SQL Server health and why?

Geographic Information Systems (GIS)

How can you migrate your GIS data to Amazon RDS?

Business Intelligence

How does high availability impact SQL Server performance?

Show more

Others also viewed

Tuning Tips to Maximize Your PostgreSQL Performance
Thiago Azadinho - MBA/OCP/OCE/MCSE · 3mo

Run PostgreSQL in Azure Kubernetes Service (AKS) Using KubeDB
AppsCode Inc. · 2y

Deploy and Manage Percona XtraDB in Amazon Elastic Kubernetes Service (Amazon EKS) Using KubeDB
AppsCode Inc. · 2y

[PostgreSQL] Replication with REPMGR
Thiago Azadinho - MBA/OCP/OCE/MCSE · 8mo

Achieving High Availability in PostgreSQL
Avinash Vallarapu (Avi) · 4mo

Active-Active PostgreSQL with Zero Downtime?
Mari C. · 3mo

Show more

Explore topics

Sales

Marketing

IT Services

Business Administration

HR Management

Engineering

Soft Skills

See All