

PostgreSQL System Architecture



Aly Ragab · [Follow](#)

3 min read · Jan 25, 2022



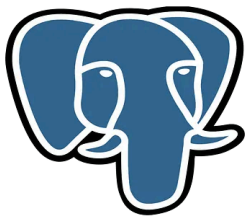
Listen



Share



More



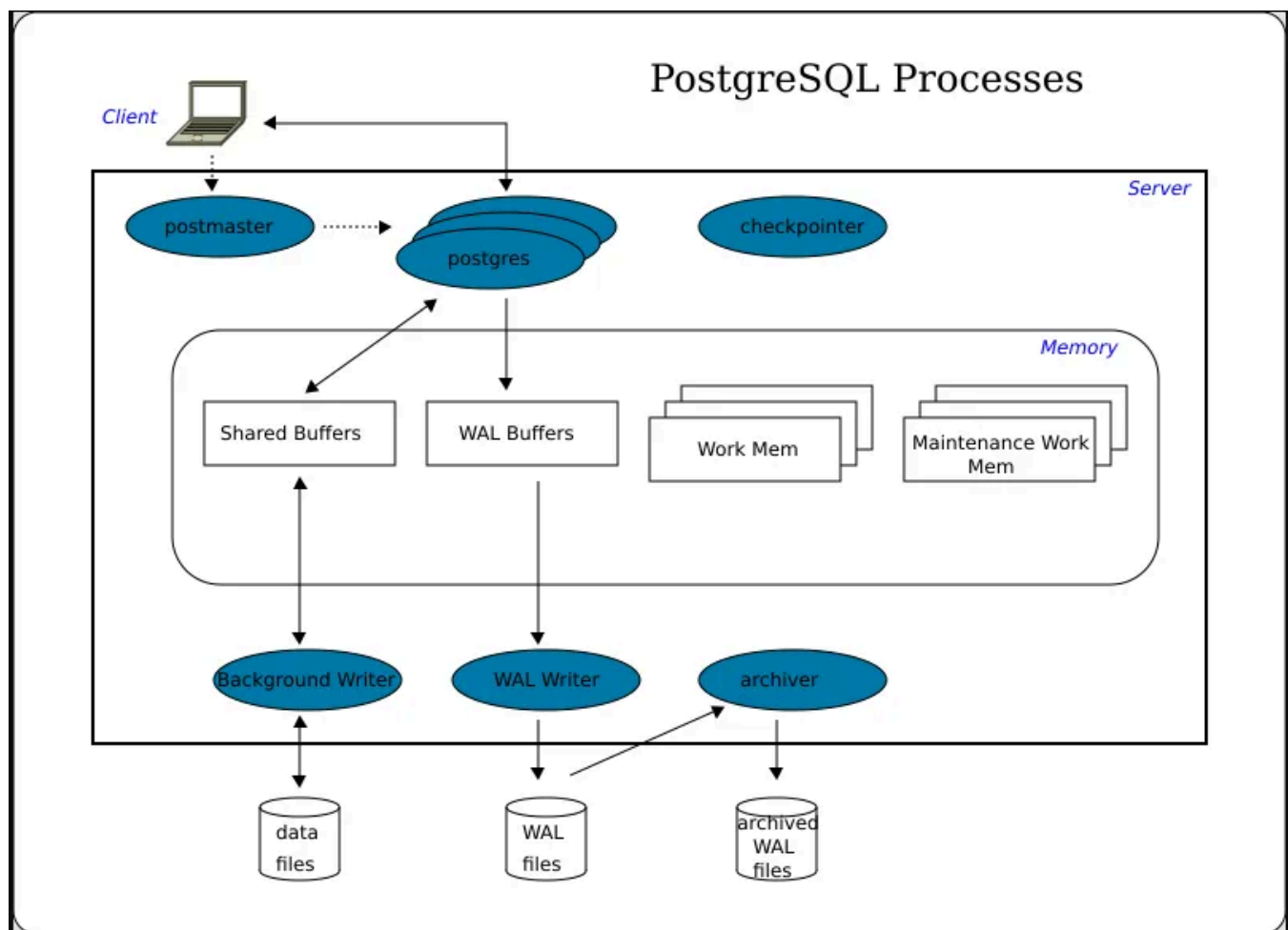
PostgreSQL

Presenting number of articles and writings for PostgreSQL Database Administration and some of its tricks.

This series will be starting with the System Architecture before we deep inside its internals like ACID, Indexing, Some of SQL tricks which helps for daily usage ...etc

Now let's start this series with its System Architecture.

Everything inside PostgreSQL is about a process, A process per connection, A process per request/query.



In order to get started with the Database internals , we have to ask what is the flow once the client is trying to connect and execute a query against the database ?

Client Request Flow:

- The client is initiating connection request using any supported authentication method “Basic Auth or SSL”, The first process in the system that will initially handle the authentication/Authorization request is “**Postmaster**” as shown in the above diagram.
- **Postmaster** will validate the access privilege of the client and then forward the request to “**Postgres**” Process.

- **Postgres** process is actually linked with the “**Shared Buffers**” which will bypass the client request/query to the writers.

So what about the database memory and processes structure ?

Main Memory components:

Postgres has three main kind of memory buffers:

- Shared Buffers
- WAL Buffers
- CLOG Buffers

Shared Buffer:

- The core of the Database as it is accessed by **ALL** Server/Client processes internally.
- It requires some amount of the Server allocated memory in order to do the caching job.
- Any modified Data is actually written in this layer, which is called “**Dirty Data**”, This is before it is being written to the disk.

WAL Buffer:

- It is named as “**Write a head Log**”.
- The **WAL buffer** is a buffer that **temporarily stores changes** to the database.
- This **WAL data** is the metadata information about the changes to the actual data.

CLOG Buffer:

- It is named as “**Committed Buffer Logs**”.
- These logs are the metadata of any transaction being executed in the database.
- It is completely different than the WAL buffer , as it is just containing whether the transaction has been committed (completed) successfully or no.

- It helps on Rollback any in completed transaction as we will discuss later in the ACID concept.

Temp Buffer:

- Used for access to temp tables in a user session during large sort and hash table

Main Processes in Postgres:

Work Memory:

- Is a memory reserved for a single **Sort** or **Hash Table** to just organize the query/data.
- A Sort Operation recognized by “ **ORDER BY** ” or “ **DISTINCT** ” in the client query.
- A Hash Table like “ **hash join**” or “ **IN** ” in the query.

Maintenance Work Memory:

- The max amount of memory allocated for any maintenance operations.
- Maintenance Operations like “ **CREATE INDEX** ” or “ **VACUUM** ”

VACUUM:

- It is considered as the database Garbage Collector.
- It reclaims the storage of any dead tuples, as the deleted data in the database is not 100% removed from the physical storage.
- So It is important that the Vacuum process to be running periodically, So that it can reclaim the space and make it available for re-use again.
- It is doing that by running “ **ANALYZE**” for selected tables.

Auto-Vacuum Launcher:

- The processes to invoke the vacuum process periodically

Checkpoint:

- It is invoked every 5 min (as default) or when `max_wal_size` value is exceeded.
- It syncs all the buffers from the shared buffer to the data files.

- This Sync is to make sure that the state of the **Shared Buffer** memory and the **Data File** are consistent.

Types of Physical files:

- Data File => Presents the Pain Data of the database.
- Wal File => Write a head log file, hosts all transaction before committing it, and mark the transaction whether it is completed or no.
- Log File => All server messages and logging.
- Archived Logs => Data from Wal are written on to the archive log files to be used for recovery.

Next we will explain What is ACID and how PostgreSQL is fully ACID Complaint ?.

Stay Tuned :)



Follow

Written by Aly Ragab

73 Followers · 157 Following

Principal Security Consultant

No responses yet





Gvadakte

What are your thoughts?

More from Aly Ragab



Aly Ragab

Authentication Types and Techniques

API Security is very important to prevent or mitigate attacks on Exposed or Internal APIs.

Aug 8, 2022 🖱️ 39





 Aly Ragab

Linux under the Hood—Memory Management

Memory is the heart of the Operating System, It is focal point between end user / Processes and the CPU, So every thing in Linux OS is...

Sep 13, 2020  64



kubernetes



HashiCorp



Open in app 

Medium



Search



 Aly Ragab

Hashicorp Vault Audit Logging

In a continued sharing knowledge in regards of managing audits in Cloud Native toolings like Kubernetes as we stated in this article...

Jun 10, 2023 🖱 11 💬 1



DynamoDB



Aly Ragab

AWS DynamoDB Deep Dive

A Deep Dive into AWS NoSQL Database “DynamoDB”, with simple and brief theoretical view , I like simplicity :) by the below elements:

Feb 24, 2022 🖱 119 💬 1

[See all from Aly Ragab](#)

Recommended from Medium



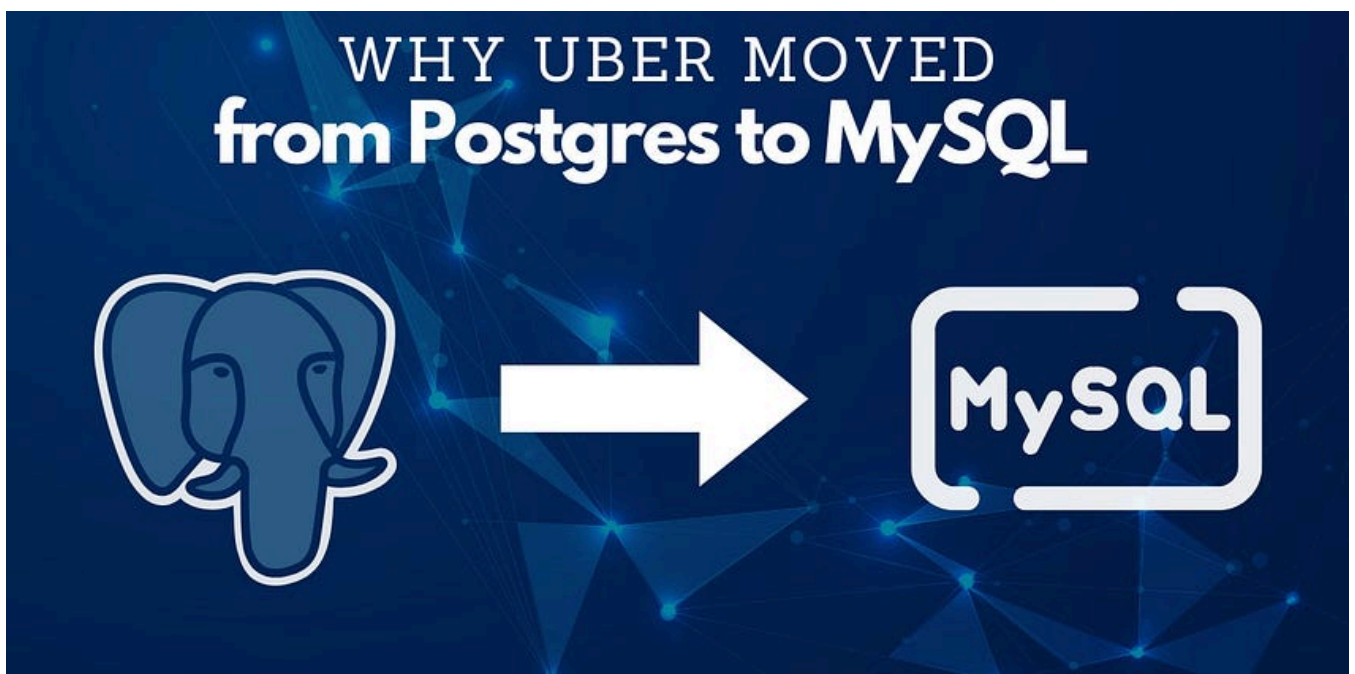
Amir

How PostgreSQL Stores Data: Heaps, Blocks, and ctids Explained 🐘 📦

Ever wondered how PostgreSQL keeps your data organized and accessible behind the scenes?



Nov 17, 2024 🖱️ 6

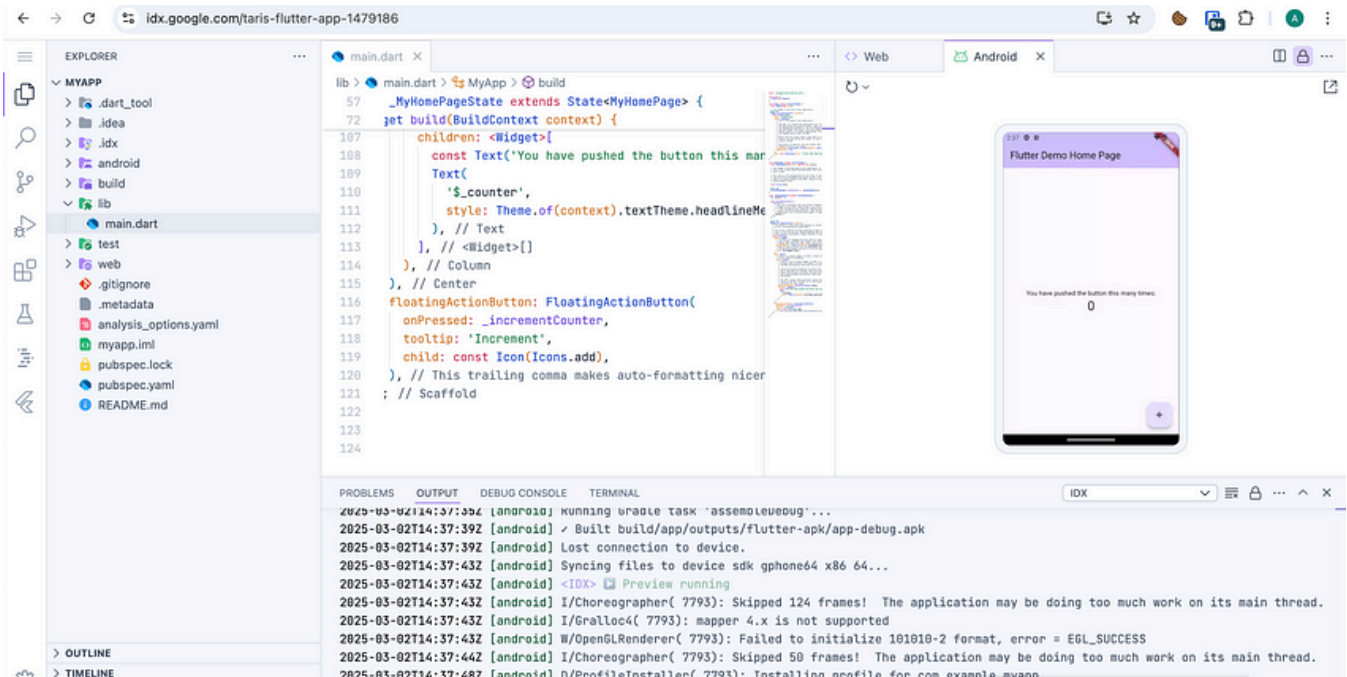


In Databases by Sergey Egorenkov

Why Uber Moved from Postgres to MySQL

How PostgreSQL's architecture clashed with Uber's scale—and why MySQL offered a better path forward

Mar 29 🖱️ 238 💬 8



In Coding Beauty by Tari Ibaba

This new IDE from Google is an absolute game changer

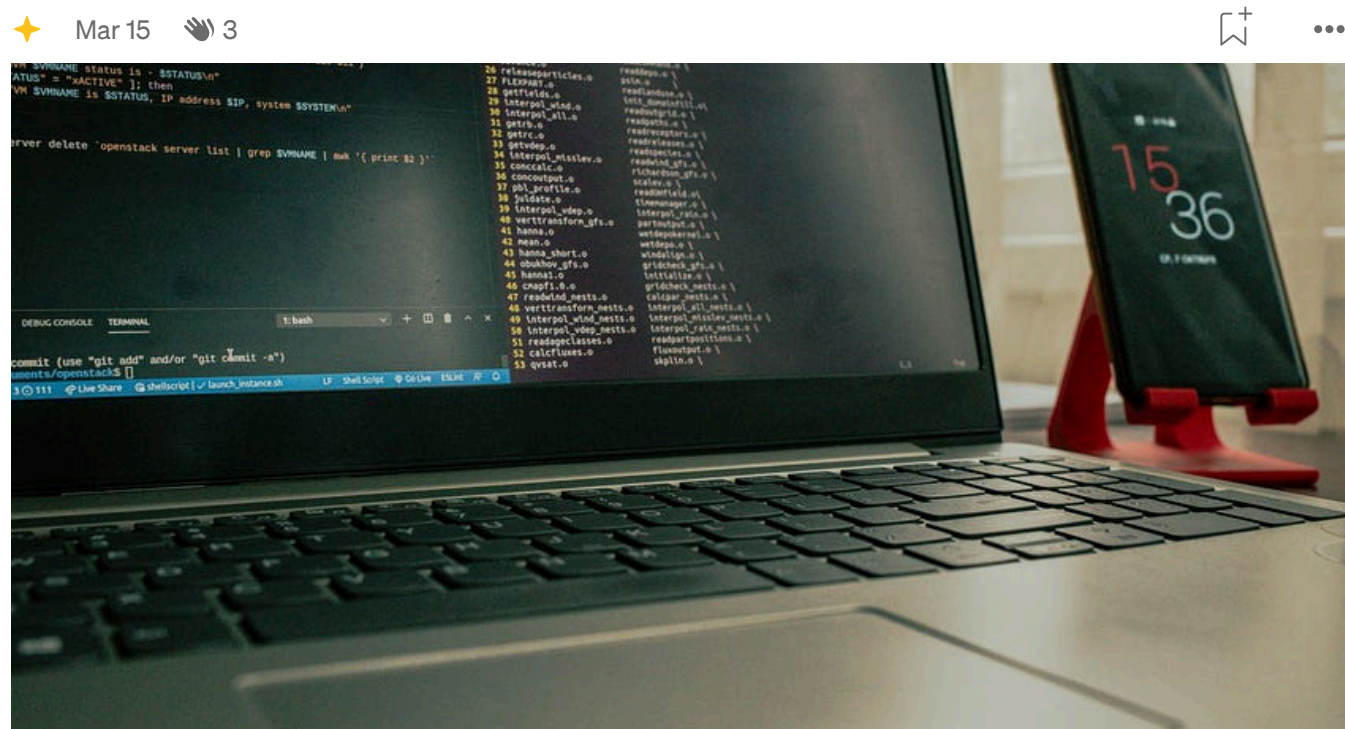
This new IDE from Google is seriously revolutionary.


★ Mar 12 🖱️ 3.7K 💬 201



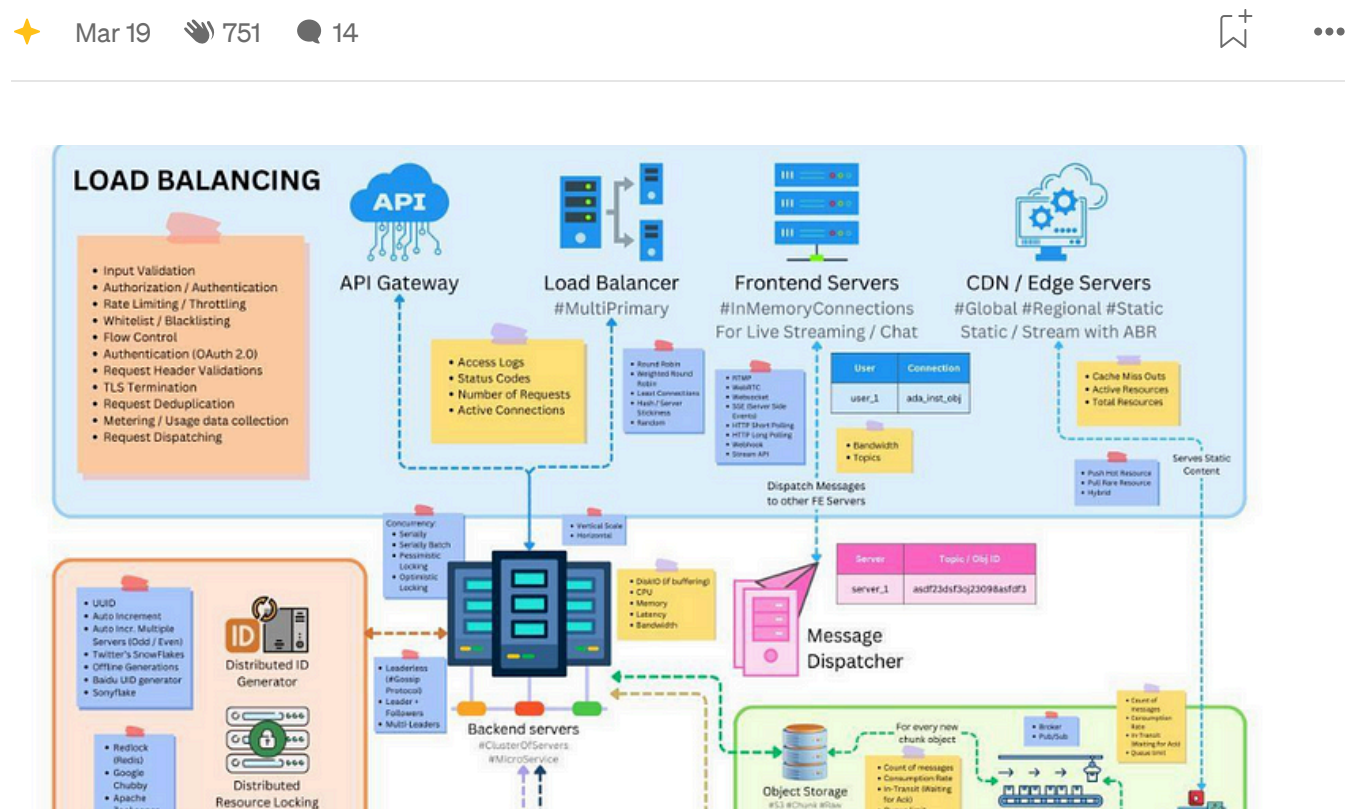
Ajaymaurya

PostgreSQL is a powerful open-source relational database, but ensuring optimal performance requires well-planned indexing strategies. A...



 In Level Up Coding by Renaldi Purwanto

From 100 Requests to 1 Million: My Journey in Scaling a Go Backend





In Javarevisited by Veenarao

System Design CheatSheet for Interview

Dear Readers, I am summarizing the commonly asked concepts in system design interviews. These questions are asked in almost all the system...

Dec 23, 2024



1.4K



26

[See more recommendations](#)