Monitoring PostgreSQL

Navigating the Landscape of Metrics, OS and Hardware Relationships, and Toolsets

Charly Batista

Current openings include:

- Senior Software Engineer (PostgreSQL)
- Support Engineer (PostgreSQL)
- PostgreSQL Evangelist
- ...and more!

Who am I?

Blame ChatGPT for this introduction!

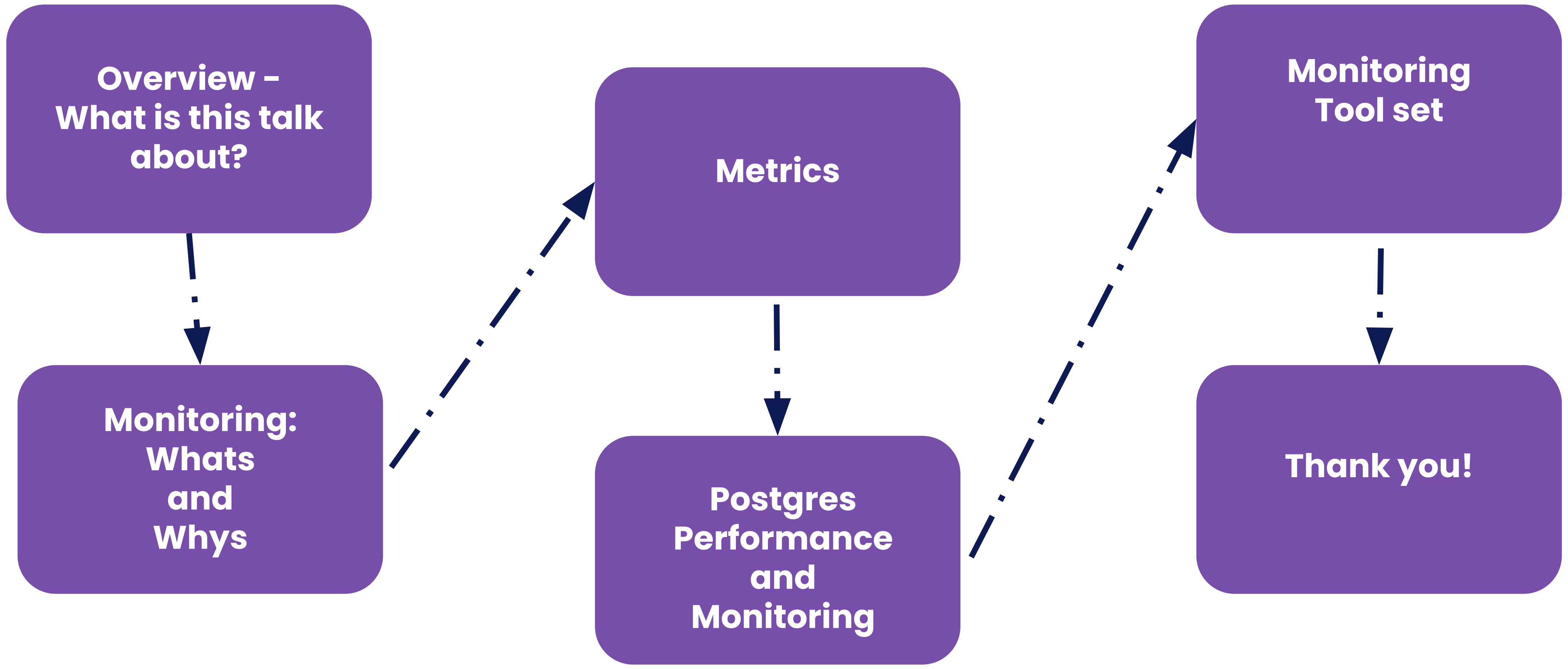
I am Charly Batista

The PostgreSQL Tech Lead at Percona with a knack for turning database queries into poetry! When not crafting SQL magic, you'll find me trading database tips over caipirinhas in Brazil or perfecting their chopstick skills in China. With a love for both the binary world and the great outdoors, I'm equally at home crunching numbers and scaling mountains. Buckle up for a database adventure like no other – this DB guy is ready to merge cultures and conquer queries with a dash of humor!

You can find me at https://www.linkedin.com/in/charlybatista

Agenda

This is what we'll cover today


Overview - What is this talk about?
Monitoring: Whats and Whys
Metrics
Postgres Performance and Monitoring
Monitoring Tool set
Thank you!

What is this talk about?

This is not an

advertising

presentation

What is this talk about?

This is what we'll discuss here today:

- What is Monitoring
- Why do we need monitoring

What is this talk about?

Metrics:

- What matters more? WHAT or WHY to monitor?
- What are them from the Hardware/OS perspective
- What are them from the Postgres perspective
- Sampling
- Applying some statistics

What is this talk about?

Postgres Monitoring:

- What to monitor
- How to monitor
- How to interpret the data
- Alerting

What is this talk about?

Toolset:

- Built-in Tools

- External Tools

Let’s review some concepts

Monitoring: *Whats* **and** *Whys*

What is Monitoring?

- Monitoring is the **continuous observation and collection** of various metrics and parameters to assess the **health**, **performance**, and **behavior of these systems** (ChatGPT)
- Monitoring refers to the **continuous observation and collection** of data about the **health**, **performance**, and **resource utilization of those systems** (Gemini)

What is Monitoring?

A system monitor is a hardware or software component used to monitor system resources and performance, and ensure the availability, performance, reliability, and security of the system

https://www.pickpik.com/ecg-electrocardiogram-medical-heartbeat-heart-frequency-115006

What is Monitoring?

https://freerangestock.com/photos/136411/overhead-view-of-a-doctor-monitoring-blood-pressure.html

Monitoring goes beyond simply collecting metrics, it's about ensuring the system's health and responsiveness from the user's standpoint!

Why do we need Monitoring?

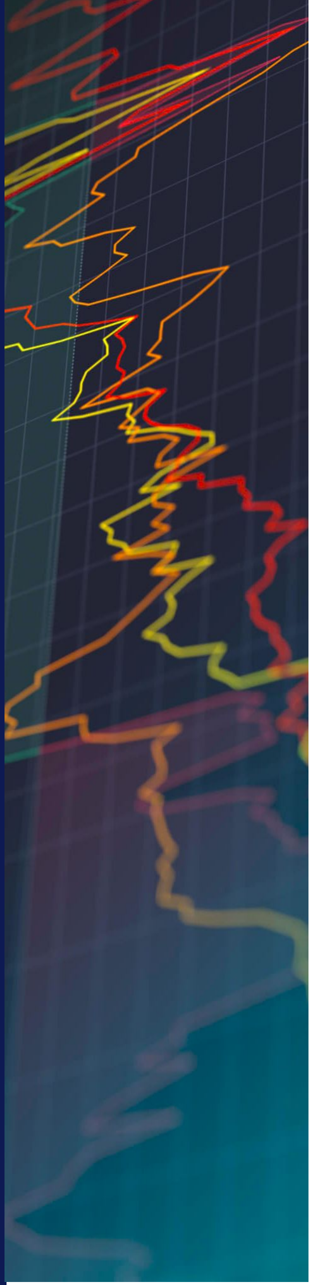
- Service Availability
- Performance Optimization
- Alerting and Incident Response
- Capacity Planning
- Continuous Improvement

https://commons.wikimedia.org/wiki/File:Monitoring_the_Third_Spacewalk.jpg

What matters more? WHAT or WHY to monitor?

- Both are crucial aspects of monitoring for a comprehensive monitoring strategy
- Just knowing the patient's temperature isn't enough, a doctor needs to understand WHY it is high!

https://www.rawpixel.com/image/3306754/free-photo-image-computer-monitors-cc0

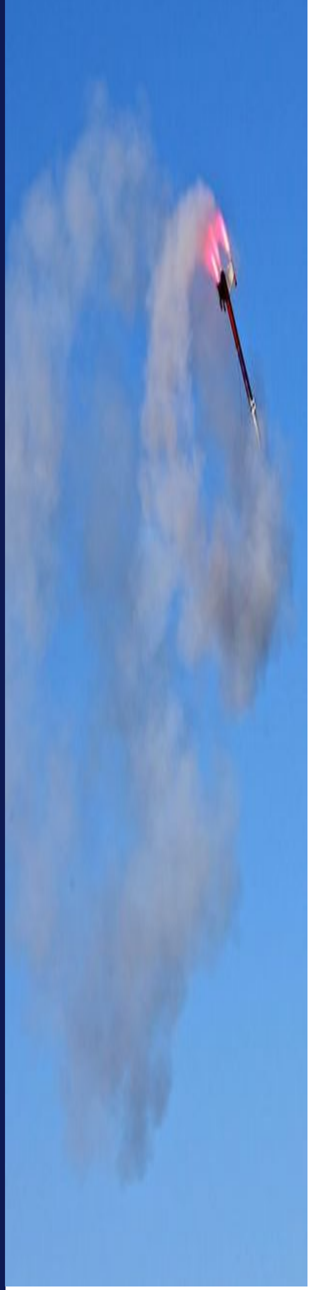


WHAT to monitor?

- Gives the Foundation for Analysis
  - Refers to the specific metrics one needs to collect
  - Are the building blocks of an effective monitoring
- Help Identifying Issues
  - Key metrics can help to identify potential problems
  - Can help to focus monitoring efforts on the critical aspects

https://centralnicregistry.com/about/news/2021-02-05-dns-analytics

PERCONA



https://www.flickr.com/photos/jurvetson/15393495039

WHY to monitor?

- Gives Context and Root Cause
  - Understanding the reasons behind monitoring specific metrics helps to understand the context and identify the root cause
- Helps on Focus and Prioritization
  - Which metrics are the most critical for a specific need?
  - Not everything needs to be monitored!

Metrics

What are them and where they live?

Metrics

- A method of measuring something, or the results obtained from this (Oxford Languages)
- Quantifiable
- Observable
- Relevant
- Granular
- Dynamic
- Contextual

https://www.google.com/search?q=Metrics (https://languages.oup.com/google-dictionary-en/)

Metrics

- Metrics are the cornerstone of system monitoring
- They provide the objective data
- Can be used to :
    - Measure performance
    - Identify trends
    - Detect anomalies
- **Can help to understand the system behavior**

Metrics

- While they are key, they should be used cautiously
- Metrics can also be:
  - Misleading
  - Irrelevant
  - Overly Expensive
  - Privacy-Sensitive
  - Context-Dependent

Sampling

- Collect data at specific intervals rather than every single event
- Helps to reduced overhead
- Can be used for:
    - Noise Reduction
    - Statistical Inference
- Helps to improve scalability
- Focus on relevant events

Utilize Statistical Methods

- A Monitoring System is a time-series system by definition
- Statistical methods can help to:
  - Identifying trends
  - Smooth out fluctuations in metric data over time
  - Identify Outliers, Trends and Deviations
  - Detect subtle changes in system performance
  - **Identify potential problems**

Utilize Statistical Methods

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

σ	= population standard deviation
N	= the size of the population
x_i	= each value from the population
μ	= the population mean

https://medium.com/analytics-vidhya/statistics-mean-median-mode-variance-standard-deviation-47fab926465a

$$\bar{X} = \frac{\sum X}{N}$$

https://graziano-raulin.com/statistics/concepts/central.htm

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}}$$

https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/

$$\bar{a}_{SM} = \frac{1}{M}\sum_{i=0}^{n-1} x_{M-i}$$

https://www.datacamp.com/tutorial/moving-averages-in-pandas

$$z = \frac{\overset{\text{raw score}}{x} - \overset{\text{mean}}{\mu}}{\underset{\text{standard deviation}}{\sigma}}$$

https://www.inchcalculator.com/z-score-calculator/

Utilize Statistical Methods

- Percentiles
- Mean (Average)
- Median
- Moving Average
- Standard Deviation and Z-Score
- Correlation Coefficient
- Many other techniques can be used for anomaly identification

How PostgreSQL stores data?

Heap files

Hardware/OS Monitoring

There are many metrics one can use for HW/OS monitoring.

Keep in mind that this is **NOT** a comprehensive list:

- CPU Metrics
  - CPU Utilization
  - User vs. System CPU Time
  - Context Switching
  - CPU Queue Length
  - CPU Wait Time
  - CPU Throttling

Hardware/OS Monitoring

- Memory Metrics
  - Memory Utilization
  - Memory Buffers and Caches
  - Dirty Memory
  - Swap Usage and In/Out Rates
  - Page Faults
  - Memory Fragmentation
  - Memory Allocation Failures

Hardware/OS Monitoring

- Disk O/I Metrics
  - Disk Throughput
  - Disk Utilization
  - Disk Response Time and Disk Latency Distribution
  - I/O Wait Time
  - IOPS
  - Disk Activity by Process
  - Average Transfer Size
  - Disk Errors and Failures (Disk Health Metrics)

Hardware/OS Monitoring

- Network Metrics
  - Network Inbound/Outbound Traffic
  - Packet Loss
  - Retries/Retransmissions
  - Network Errors (CRC errors, packet drops, frame collisions, etc)
  - Round-Trip Time (RTT)
  - Network Throughput and Congestion
  - Network Traffic by Protocol
  - Jitter

PostgreSQL Metrics

- Number of Active and Idle Connections
- Queries per Second
- Long Running Queries
- Lock Waits
- Buffer Cache Hit Ratio
- Index Usage
- Full Table Scans
- Row/Table Locks
- Deadlocks

PostgreSQL Metrics

- Database Growth
- Autovacuum Activity
- Checkpoint Activity
- Temporary Files
- Background Writer Activity
- Xact Commit/Rollback Rate
- Replication Lag
- Replication Slots

PostgreSQL Metrics

- Wait Events
  - Can reveal if queries are waiting for resources
  - Can reveal potential bottlenecks
  - Can help with capacity planning
  - Correlate with other metrics (CPU, disk I/O, query time, etc)

Tool set

Performance Monitoring

Linux Monitoring tools

- Linux **procfs (/proc)** filesystem:
    - Memory usage
    - Open file descriptors
    - Command-line arguments
    - Environment variables
    - CPU and I/O statistics
    - Loaded modules
    - Partitions and mounted filesystems
    - Network devices and interface statistics

Linux Monitoring tools

- **Sysstat** package, various utilities to monitor system performance:
  - **iostat** reports CPU statistics and input/output statistics for block devices and partitions.
  - **mpstat** reports individual or combined processor related statistics.
  - **pidstat** reports statistics for Linux tasks (processes) : I/O, CPU, memory, etc.
  - **tapestat** reports statistics for tape drives connected to the system.
  - **cifsiostat** reports CIFS statistics.
  - **sar** collects, reports and saves system activity information

https://github.com/sysstat/sysstat

Linux Monitoring tools

- **0x.tools** is a set of open-source utilities for analyzing application performance on Linux
  - **psn**: Show current top thread activity by sampling /proc files
  - **xcapture**: Low-overhead thread state sampler reading /proc files
  - **schedlat**: Show single processes CPU scheduling latency as a % of its runtime
  - **run_xcapture.sh**: A simple “daemon” script for keeping xcapture running
  - **run_xcpu.sh**: Low-frequency continuous stack sampling for threads on CPU (using perf)

https://0x.tools/

Linux Monitoring tools

Here are some of the most used open source Linux monitoring tools

- Ancient ones:
  - Nagios
  - Zabbix
  - Monit
- Collectd
- Prometheus & Grafana

PERCONA

PostgreSQL Monitoring tools

For Postgres I would like to divide into 2 categories:

- **Built-in Tools**: Tools running inside the database either as part of the database core or extension
- **External Tools**: External tools that connect to the database to acquire metrics

PostgreSQL Monitoring tools

- Built-in Tools
  - The Statistics Collector
  - pg_stat_statements
  - pg_stat_monitor
  - pg_stat_plans
  - pgstattuple
  - pg_buffercache

PostgreSQL Monitoring tools

- External Tools (CLI):
  - pg_view
  - pg_activity
  - pg_top
  - pgmetrics
  - pgstats

PostgreSQL Monitoring tools

- External Tools (Web interface):
  - **pgwatch2** (https://github.com/cybertec-postgresql/pgwatch2)
  - **PGObserver** (https://github.com/zalando/PGObserver)
  - **pgCluu** (https://github.com/darold/pgcluu)
  - **pgexporter** (https://pgexporter.github.io/)
  - **Prometheus & Grafana**
  - **Percona PMM** (https://github.com/percona/pmm)

It then comes to the end

Summary

Summary

- Effective monitoring is essential for ensuring the performance, availability, and reliability of PostgreSQL databases.
- Define Key Metrics
- Establish Baselines
- Alerting and Automation
- Regular Review and Tuning

Summary

Monitoring isn't just about gathering data and making good looking graphs. It's about knowing the WHAT and WHY we're monitoring, which metrics matter most, how they relate to each other.

Most importantly, it's not just about spotting problems, everyone can tell the system isn’t responding, but understanding why they're happening in the first place!

Questions?

https://www.linkedin.com/in/charlybatista/

https://github.com/elchinoo/presentations

percona.com

THANK YOU!

percona.com