

Zooming in on OOM: A Deep Dive into Postgres and Linux Memory

Thursday, June 25 at 16:40–17:30



Oleksii Kliukin
Tiger Data



Dimitris Sideris
Tiger Data

Category: **General**

Level: **Intermediate**

Language: **EN**

Room: **3.010**

How much memory does a Postgres server actually request? Why do Postgres and Linux allocation metrics sometimes disagree by multiple gigabytes? And what happens when the OS encounters a memory request it simply cannot fulfill?

We didn't ask those questions out of mere curiosity. For several years, the platform team at Tigerdata successfully relied on a custom library to shield thousands of Postgres services from the Linux Out-Of-Memory (OOM) killer in an environment where memory overcommit is an unavoidable reality. However, as workloads scaled and memory requirements grew, we noticed an uptick in allocations that were bypassing the library limit and leading to OOMs. We found ourselves facing a question: why were the OOM conditions bypassing our safeguards and what could we do about it?

This talk details our deep dive into the intersection of Linux and PostgreSQL memory management that resulted in a sharp decrease in the number of OOM events. We will explore:

- The Root Cause of OOMs: How overcommit works and how to prevent the kernel from sending SIGKILL to Postgres processes in a modern cgroup-based environment.
- Postgres Memory Architecture: A breakdown of the Postgres memory model and the parameters, both per backend and at the server level, that affect how much memory Postgres uses.
- Our journey into finding unaccounted allocations: How we used modern tools like eBPF to analyze OOM kill statistics and identify both 'extra' memory and unexpected malloc behavior.
- The Solution: What we did to establish a reliable cap on the memory usage, relying on Patroni hooks and Postgres extensions to put everything in place.

Beyond the story, attendees will walk away with strategies to minimize Postgres memory overhead, insights into glibc malloc and its alternatives, and a suite of Linux tools for analyzing per-process and system-wide memory usage.