pgstats in PostgreSQL 18

Pgconf.dev
2026/05/21

Michael Paquier (he/him)

Principal Software Engineer
PostgreSQL Contributor Team

The lecturer

- French, based in Tokyo.
- Major contributor.
- Committer since 2018.

Agenda

Architecture

Postgres 18

Extensions

Architecture

Monitoring

- Aggregated view of cluster state
  - Objects: table, index, function..
  - Processes: checkpointer, WAL receiver/sender, bgwriter..
  - OS activity: WAL, I/O.
  - Misc: Transaction, SLRU.
- System views pg_stat_* (mostly, not pg_stat_activity!)
- https://www.postgresql.org/docs/devel/monitoring-stats.html

Shared Memory implementation

- On-disk file - Durability
  - Loaded at startup.
  - Flushed at shutdown.
  - Lost on crash.
- Each stats kind is assigned an ID.
  - Variable-sized.
  - Fixed-sized.
- Commit 5891c7a8ed8f.

Source code

- Core parts
  - pgstat.c
  - pgstat_shmem.c
  - pgstat_internal.h and pgstat.h
- One file for each stats kind:
  - pgstat_database.c
  - pgstat_archiver.c, etc.
- src/backend/utils/activity/

Fixed-sized stats

- Chunk of shared memory
- Memory allocated at startup
- LWLocks included in each stats structure
  - pgstat_internal.h
  - PgStatShared_*
- Counters cover a known size:
  - WAL
  - I/O
  - SLRU

Variable-sized stats - Details

- Hash table in dynamic shared memory (dshash.c, DSA)
- Hash key:
  - Stats kind (4 bytes)
  - Database OID (4 bytes)
  - 8-byte ID (4-byte OID in v15~17).
- Objects: database, table, functions, DDL-related.
- Locking internal, partitions of dshash.
- Maximum number not capped.

Variable-sized stats - Reporting

- Limit shared memory interaction.
- Pending stats
  - Get reference to entry in shmem, report pending activity.
  - Static to each process.
  - Saved in memory context created in TopMemoryContext.
- Flush by pgstat_report_stat()
  - Transaction commit.
  - Hardcoded max interval (60s).
  - Can be forced, waiting on locks.

Postgres 18

Backend statistics

- Hash key based on proc number.
- Variable-sized.
- pgstat_backend.c
- For WAL and I/O, same fields as pg_stat_wal and pg_stat_io
- Functions
  - pg_stat_get_backend_wal(pid), single row
  - pg_stat_get_backend_io(pid), multiple rows
  - pg_stat_reset_backend_stats()
  - Join with pg_stat_activity

pg_stat_io

- Multiple rows:
  - Type of operation (IOOp): Write, read, syncs..
  - Context (IOContext): VACUUM, bulk-read, bulk-write, init (new in 18!)
  - Object: relation, temp relation, WAL (new in 18!)
  - Backend Type: backend, checkpointer, autovacuum..
- Counters
  - In bytes in v18
  - BLCKSZ and number of operations up to v17.
- Data of pg_stat_wal moved to pg_stat_io

pg_stat_database

- Parallel worker activity
- Fields
  - parallel_workers_to_launch, number decided by planner.
  - parallel_workers_launched, number actually launched.
- Tuning of parallel workers.

Extensions

Custom Cumulative Statistics

- Plugins and extensions can use pgstats!
- Similar to custom WAL RMGRs
  - Load with shared_preload_libraries
  - When removed from shared_preload_libraries, same as corrupted.
- Assign fixed ID
- https://www.postgresql.org/docs/devel/xfunc-c.html#XFUNC-ADDIN-CUSTOM-CUMULATIVE-STATISTICS
- Commit 7949d9594582.

Stats kind ID

- Whole range
  - PGSTAT_KIND_MIN 1
  - PGSTAT_KIND_MAX 32
- Invalid = 0
- Built-in: 1 to 12, up to 23 available.
- Custom:
  - PGSTAT_KIND_CUSTOM_MIN = 24
  - PGSTAT_KIND_CUSTOM_MAX = PGSTAT_KIND_MAX
- pgstat_kind.h

Source code - Register

- PGSTAT_KIND_EXPERIMENTAL for development.
- ID overlap => stats file corruption.
- Reserve an ID: https://wiki.postgresql.org/wiki/CustomCumulativeStats

```
extern void pgstat_register_kind(PgStat_Kind kind,
                                 const PgStat_KindInfo *kind_info);
```

Callbacks - Basics

- PgStat_KindInfo in pgstat_internal.h
- Some boolean fields:
  - fixed_amount: variable or fixed size?
  - accessed_across_databases
  - write_to_file
- Some functions
  - flush_pending_cb => Variable-sized
  - flush_static_cb => Fixed-sized
  - init_backend_cb => Initialization action for backends

Callback – Name/Object ID mapping

- For on-disk storage
  - Name => Object ID, when reading.
  - Object ID => name, when flushing.
- to_serialized_name
- from_serialized_name
- Example: replication slot stats

Source code - Templates

- Minimalistic dummy implementation.
- Hash key:
  - Invalid database OID
  - Hash of string.
- src/test/modules/test_custom_stats/
  - test_custom_var_stats.c for variable-sized stats
  - test_custom_fixed_stats.c for fixed-sized stats
- Use of new callbacks in Postgres 19 with DSA.

pg_stat_statements?

- Query text file:
  - Performance bottleneck
  - Move to DSA
- Deallocation and pg_stat_statements.max
  - Track number of entries in hash table of pgstats.
  - Bgworker doing eviction cleanup with soft min/max.
- Do NOT move in core.
- Move to custom pgstats.

Thank you!

Michael Paquier
paquier@amazon.com