



EDB™

Postgres Distress

Spotlight: Support Services

PostgreSQL 101: Flying Solo </> Flying Blind

HOUSEKEEPING



Slides and recording will be available within 24 hours



Questions will be answered at the end

SPEAKERS



Jamie Watt

Senior Vice President of Support Services



William Ivanski

Senior Principal Support Engineer (lead developer on Lasso, LiveCompare, and OmniDB)

OPEN SOURCE WATERS

Note: Database design is complex. Database solution design in the waters of open source is... more complex. Think "open heart surgery on a roller coaster". For the sake of simplicity, we focus here on the journey rather than the shipbuilding experience.

- Database solutions are built for purpose, and configured for need.
 - Ease of use
 - Performance
 - Availability & Resiliency
 - Maintenance



Harbor-Tested

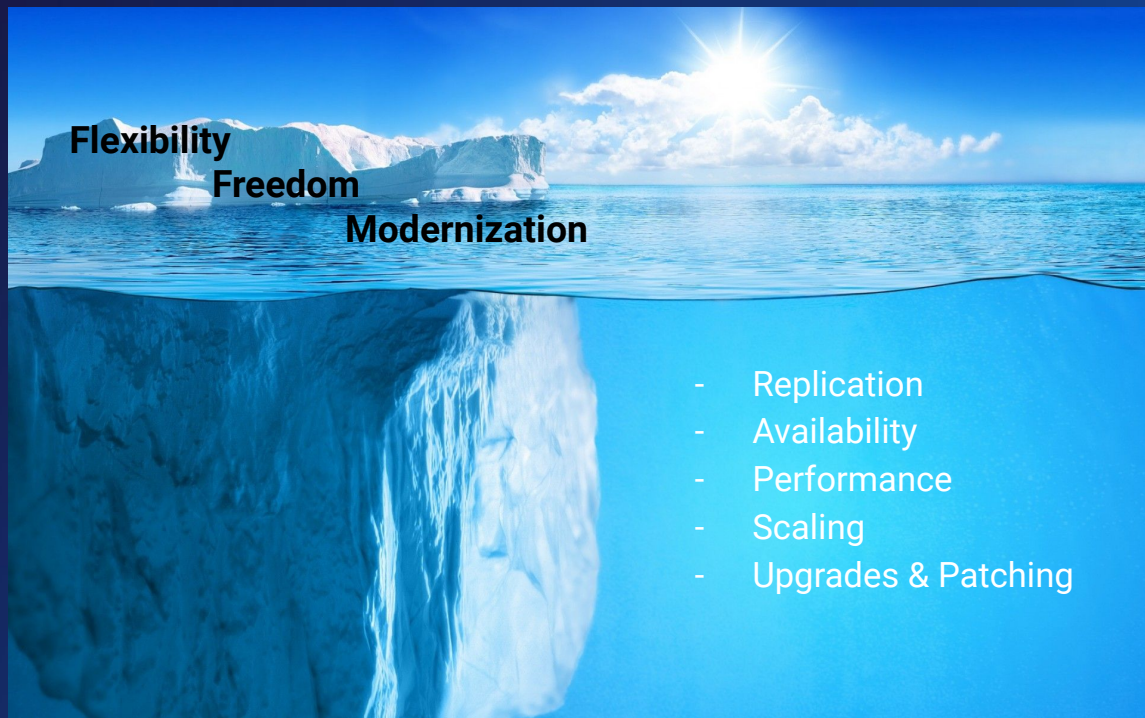
- Query performance
- Connection handling
- Load scaling

Peril-Ready

- Connection failures & saturation
- Failover & availability
- Backups & recovery



THE OPEN SOURCE JOURNEY



Stepping into open source allows a team the flexibility to define both their solution and its ability to grow with the needs of their teams and platforms.

Doing so effectively means looking not only to the horizon, but beneath the surface.

Today, we'll cover some scenarios where the iceberg was a lot bigger than it seemed.

HIGH AVAILABILITY & REPLICATION

Mayday!	Search & Rescue	Remedy and Success
Primary is failing over to a remote replica which isn't able to service the workload.	Found to be caused by network instability; conventional switchback wasn't an option because it was a genuine failover event; primary and standby had consistency and no data loss.	Worked to help fallback to original primary; business needed to reassess order of standbys to prevent recurrence of wrong-fit leader. Configure monitoring to trigger alerts for awareness of failover events.
Error: Canceling statement due to conflict with recovery.	Checked logs for long-running queries and cause - nightly jobs vs consistent & continuous queries can be handled differently	Increase either the max_standby_streaming_delay or max_standby_archive_delay; enable hot_standby feedback or consider a logical standby for reporting.
Sorry, too many clients already.	Analyze pg_stat_activity to determine sources; many idle connections; reviewed current pooling practices	Review connection requirements according to the available resources and application needs; recommendation to use a database-side pooler (pgbouncer, pgpool) and follow best practices for managing sessions & connections.
PgBouncer client / query timeout.	Reviewed max connections in both PostgreSQL & pgbouncer; max_connections set to 1500 on database; max_client_conn set to 10000, default_pool_size set to 20 on pgbouncer; pool size created bottleneck on clients.	Existing configuration created one current problem and several potential tragedies. Recommended reduction of overall max_connections and max_client_connections, as well as pool-specific sizing for best performance.

DURABILITY & PERFORMANCE

Mayday!	Search & Rescue	Remedy and Success
System outage due to pg_xlog / pg_wal causing disk exhaustion.	Reviewed steps taken before outage; verified inactive replication slot tied to a logical replication test performed.	Confirmed suspicion of cause, which was found in the test itself; solution was to drop the slot and await checkpoint to clear WAL. Recommended to catch inactive replication slots and perform test activities on a logical replica of the primary.
Performance degradation on queries against postgres_fdw.	Reviewed example query plans; validated query performance was impacted by foreign tables; reviewed foreign table statistics kept locally	A manual ANALYZE improved the query performance. Recommended to routinely ANALYZE foreign tables. Alternatively, use_remote_estimate can be enabled for complex queries or heavily-written tables on the remote server.
Still waiting for ExclusiveLock on extension of relation.	Review of existing DDL and DML related to affected tables; all events were tied to DDL which included extensions of relations to affected tables.	Heavily-written, append-only or append-most table. Consider Hash (Sub-)Partitioning, which segments the risk and reduces the probability and duration of lock events.

UPGRADES AND PATCHING

Mayday!	Search and Rescue	Remedy and Success
Out-of-order XID insertion in KnownAssignedXids.	Identified error in logs on the standby; ID'ed footprint of open transactions older than the PostgreSQL architectural limitation of 4bn xid's.	Standby was rebuilt; Development authored hotfix in PostgreSQL standby startup code, made available to 10.20, 11.15, 12.10, 13.6 and 14.2.
Upgrade from Postgres 11 to 13 failing on database containing tables with OIDs.	Reviewed pg_upgrade output; discovered application using OIDs, so access to OIDs could not be disrupted during the process.	Guided customer to migrate to a new field with the help of a trigger, and assisted customer in implementing workaround; upgraded via logical replication.

THANK YOU

William Ivanski william.ivanski@enterprisedb.com
Jamie Watt jamie.watt@enterprisedb.com

