Microsoft

PGConf NYC
September 22-23, 2022

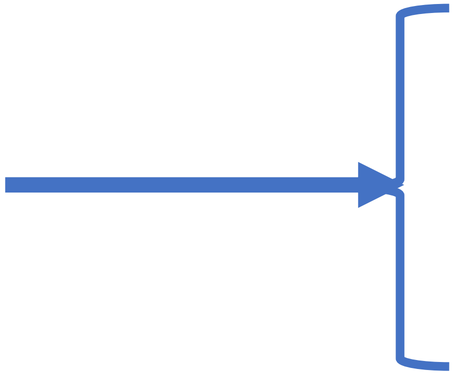
Lessons learned running a managed service for Postgres in the cloud

Sunil Agarwal,
Principal PM Manager,
Azure Database for PostgreSQL

Agenda

Focus Areas

- Application Design
- Server Configurations
- Security
- User Actions



What?

Impact?

Recommendation?

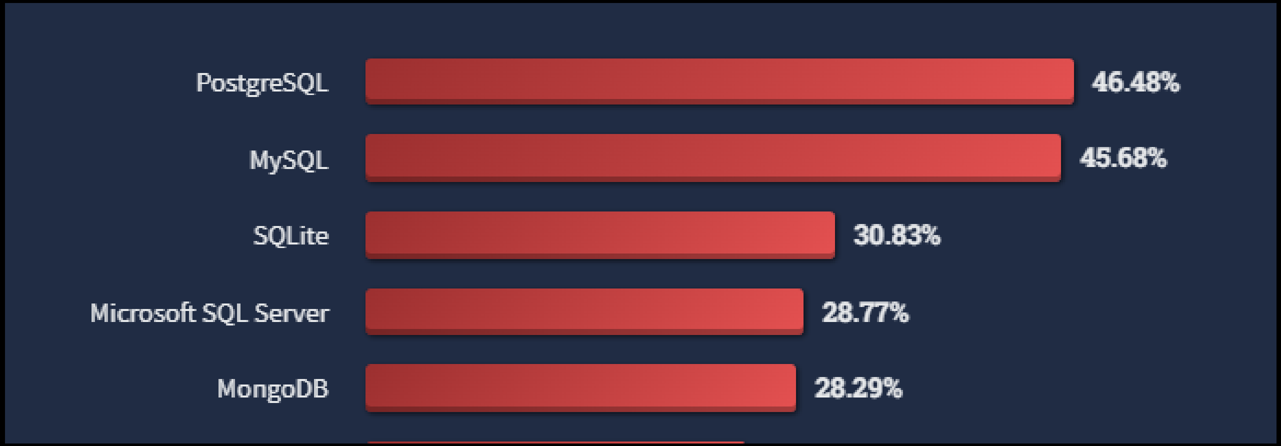
Public cloud learning

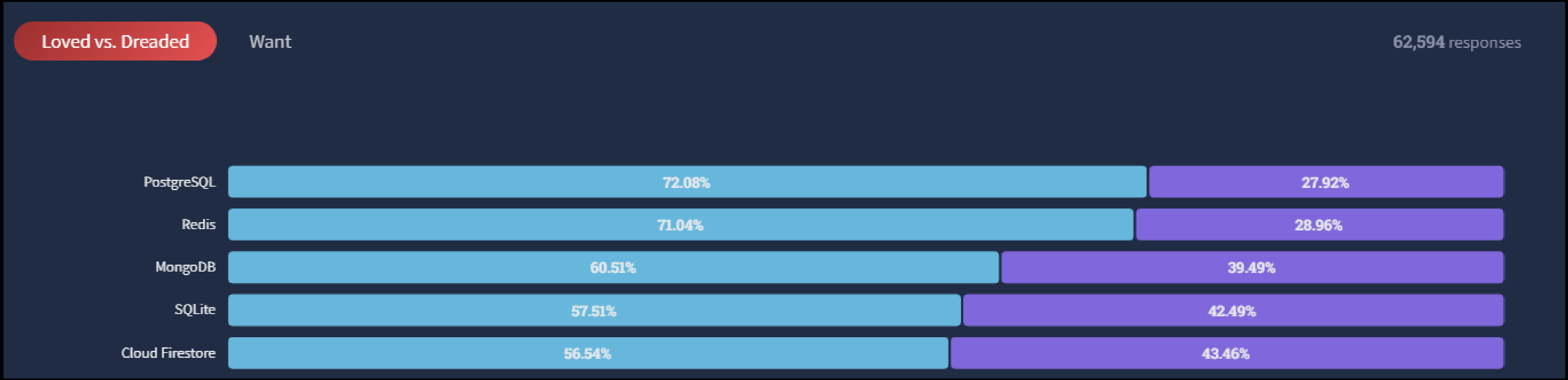
Contributions to Open Source

PostgreSQL is more popular than ever

Professional Developers

Stack Overflow Developer Survey 2022





Application Design: Connection Management

How Applications connect

- Short-lived (< 60 seconds), Normal (last < 1200 seconds) and Large (>1200 seconds)
- Each connection takes 10 MB memory and takes CPU resources to create/drop

Impact

- Short-lived connections significant impact on CPU and latency to the workload
- Idle connections lead to higher memory usage

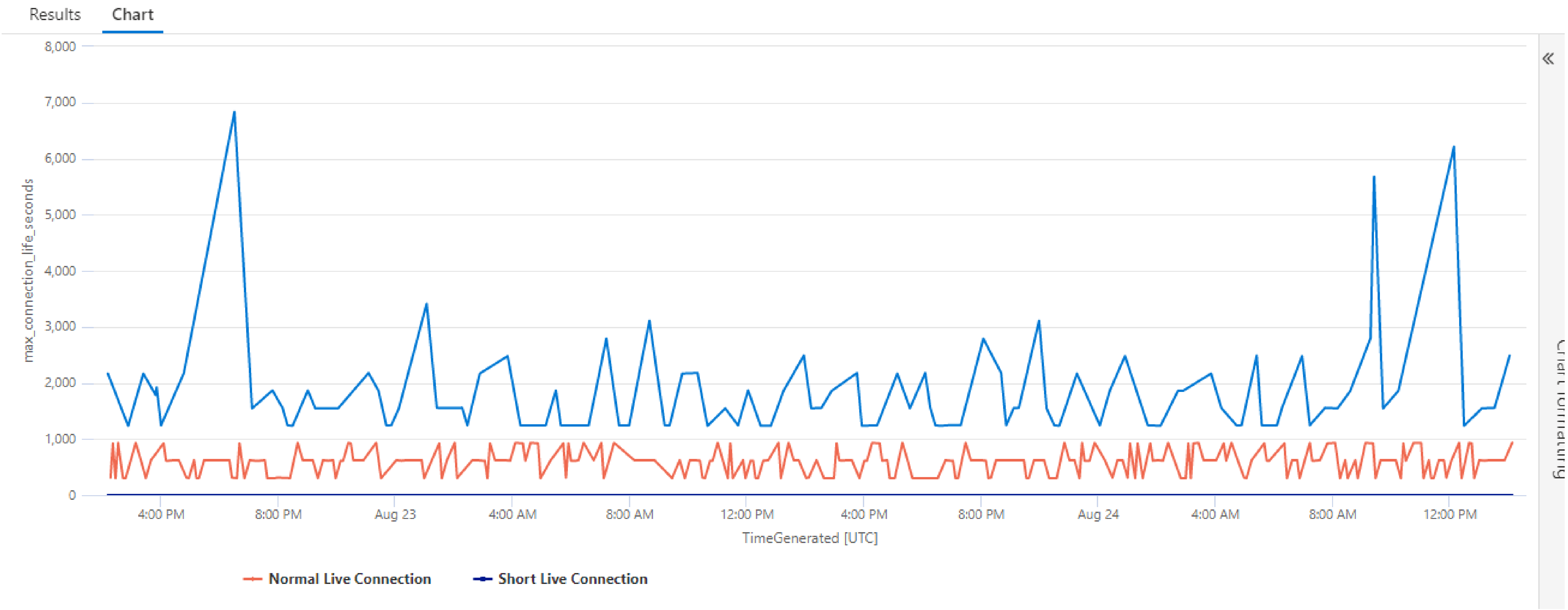
Recommendations:

- Connection pooling to remove short-lived connections and minimize Idle connections

Public Cloud Learnings:

- Provide connection pooling as part of the managed service
- Recommend customers to add connection retry logic

Connection Management: Example



Application Design: Partitioning Scheme

What is Partitioning?

- Ability to break table into multiple smaller chunks
- Commonly used in time-series data
- Simplifies maintenance operations including vacuuming

Impact

- Unpartitioned table or poor partitioning: Can slow down query performance
- On large tables, the maintenance tasks take increasingly longer time

Recommendations:

- Determine if your table indeed needs partitioning.
- Identify partitioning column based on query predicate. Example: timeseries data

Public Cloud Learnings:

- Automatically detect/recommend if partitioning can help

Application Design: Index Management

What is an Index?

- Index used to speed up query performance and enforcing constraints

Impact

- A great tool to improve query performance but adds overhead to DML operations
- Missing index can slow down query performance
- Unused index just adds to the overhead

Recommendations:

- Create missing indexes and drop unused indexes

Public Cloud Learnings:

- Help customers in identifying missing indexes through Intelligent Performance (QueryStore)

Application Design: offloading Reads

What

- Mixing reporting and transactional workloads
- Read-heavy workloads

Impact

- Can slow-down your transactional workload
- Database may not be able to meet the throughput/latency requirements

Recommendations:

- Cache query results (e.g., Redis cache)
- Offload read by setting physical and logical replication

Public Cloud Learnings/Value:

- Multiple local/cross-regional read replicas
- Cross-regional replica for resiliency from regional disasters

Agenda

Focus Area

- Application Design
- Server Configurations
- Security
- User Actions

Contributions to OpenSource

Server Configurations: Work_Mem

What does this parameter do?

- Sets the maximum amount of memory used by per operation within a query
- Used for sort, Hash join operation in queries

Impact:

- If set too low, it will cause spill to temporary disk leading to more IO
- If set too high, it will cause more memory reserved per query. High parallelism will make it worse

Recommendation

- work-mem should be tuned a bit with historical memory profile of the workload
- consider testing and setting this at session level and increase by small values.

Public Cloud Learning

- Network storage has higher latency.
- Improperly IOPS can lead to very slow query execution

Server Configurations: Vacuuming

What Does Vacuuming do?

- Cleans up dead tuples and updates statistics. Multiple configuration settings
- Removes dead rows that cause tables/database to bloat, query slowdown and increased IOs

Impact

- Too little vacuuming leads to bloat, more IOs to get pages in with relevant row versions
- Too much vacuuming - eats up IOs from regular IO
- Reclaiming storage requires running Vacuum with FULL option but locks the table

Recommendations:

- For large table: set at table level
- Check for long running transactions or abandoned replication slots

Public Cloud Learnings:

- Network IO latency and configured IOPS can impact workload performance
- Support extension Pg_repack for reclaiming the space

Server Configuration: Slow Data load

What are common usage

- Initial data load
- Periodic incremental data load
- Loading individual tables (e.g., recovering a dropped table)

Impact

- Improper configuration can lead to slow data-load.

Recommendations:

- Drop indexes and constraints
- Server parameter –
    - Max_WAL_SIZE – 64 GB
    - Checkpoint timeout increase
- Ensure IOPS are aligned with the data-load expectations.

Public Cloud Learnings:

- Provide scaling IOPS independent of provisioned storage.

Agenda

Focus Area

- Application Design
- Server Configurations
- Security
- User Actions

Contributions to OpenSource

Security: Authentication

What

- Authenticating user to PostgreSQL
- Most commonly done using native PostgreSQL userID/PWD

Impact

- Stolen credentials/ID can compromise your data and loss

Recommendations:

- Passwordcheck to enforce password strength
- Use stronger authentication mechanism like LDAP
- Communicate over secure encrypted channel SSL

Public Cloud Learnings/Value:

- Provide built-in Strong authentication : Strong PWD, Expiry, MFA and more (e.g. AAD)
- Secure data from hackers (e.g., Azure Defender)

Agenda

Focus Area

- Application Design
- Server Configurations
- Security
- User Actions

Contributions to OpenSource

User Action: Restarts

What Does Restart do?

- Restarts the server by terminating all connections
- Done to enable changes to ‘static’ server parameters
- Done when server seems hanging

Impact

- Server unavailable for workload processing for the duration
- If restarted at peak workload, it can take much longer. Note, recovery is single-threaded.
- Panic restarts when restart is taking longer

Recommendations:

- Run Restart when transactional activity is low
- Run a manual checkpoint to ensure faster recovery

Public Cloud Learnings:

- Show estimated restart time and progress
- Wait for checkpoint to complete before restart

User Action: Unused Replication Slots

What is a Replication slot?

- Tracks the state of replication between source and target
- Replication slot can become inactive

Impact

- WAL files to be built up
- Prevents clean up of dead/ghost rows,

Recommendations:

- If lag beyond a threshold, Drop the replica and recreate

Public Cloud Learnings:

- We must monitor replication slot and alert customer to drop the replica

User Action: Excessive Logging

What is Error reporting and logging?

- Very useful in debugging and performance troubleshooting

Impact

- Verbose logging can seriously impact performance

Recommendations:

- Keep verbosity to minimum except when debugging
- Remember to reset the verbosity once done

Public Cloud Learnings:

- Many customers keep the verbose logging without realizing impact on the production workload.

Agenda

Focus Area

- Application Design
- Server Configurations
- Security
- User Actions

Contributions to Open Source

PG4 – Key contributions

Manageability

- Allow long-running queries to be canceled if the client disconnects

Performance

- Improve the I/O performance of parallel sequential scans
- Allow a query referencing multiple foreign tables to perform foreign table scans in parallel
- Improve the speed of computing MVCC visibility snapshots on systems with many CPUs and high session counts

PG15 – Key contributions

Development Simplified

- Make it easier to test patches in-development and testing done by cfboot

Performance

- WAL prefetching for faster crash recovery

Monitoring

- Store statistics in shared memory: Reduce IO on stats collection so that we can collect more stats
- Monitoring and visibility improvements: Logical decoding & pg_walinspect

Summary

What we have learnt

- Application Design
- Server Configurations
- Security
- User Actions

Contributions to Open Source

Have a great success with PostgreSQL!!!