WELCOME

A Technical Journey

with :

The Elephant

July 2023

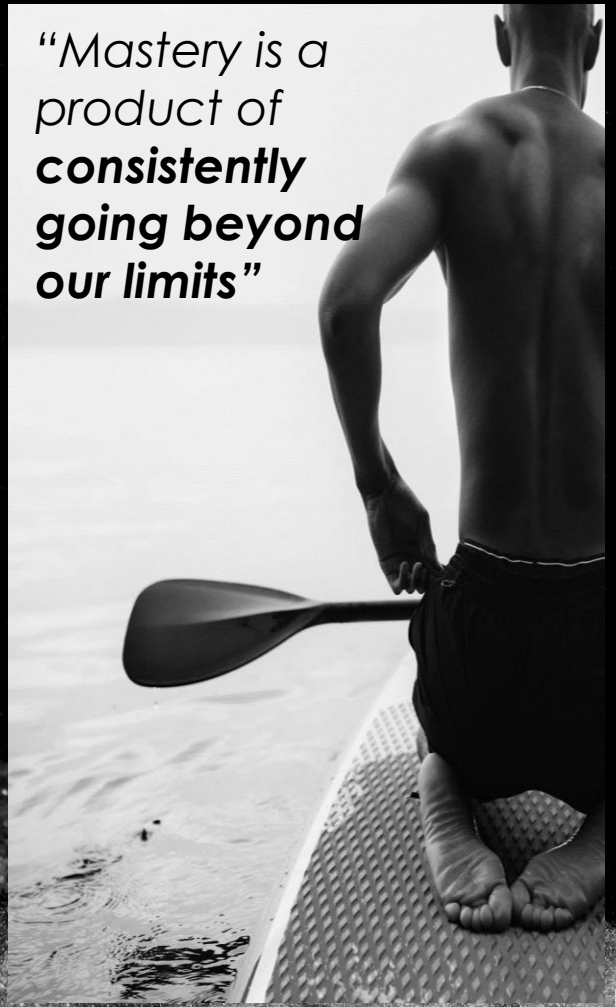
RadixTrie

The Journey

- Reskilling to PostgreSQL
- PostgreSQL Flavours
- Migrations to PostgreSQL
- Monitoring
- *Speed-Up PostgreSQL*

RadixTrie

The Challenge



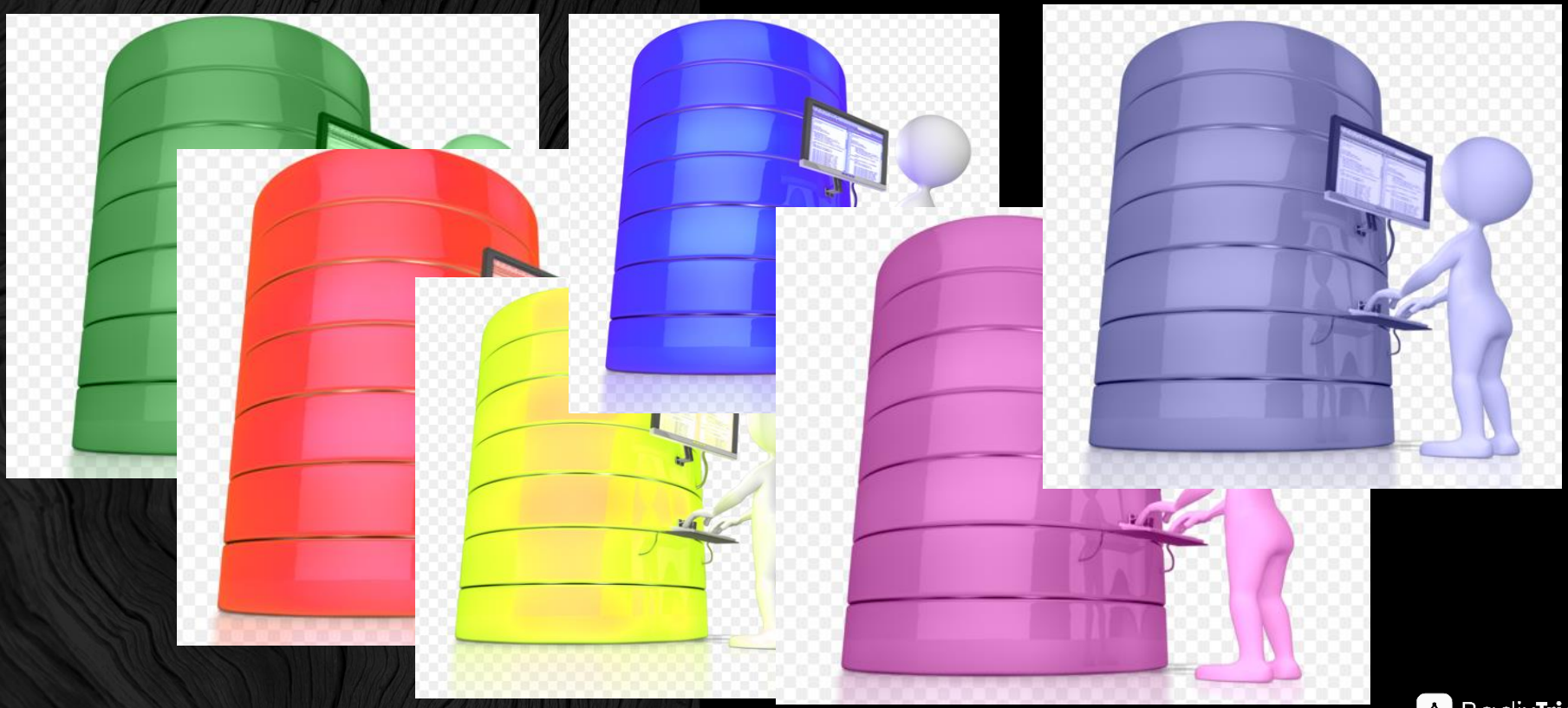
RadixTrie

The Challenge : Reskilling

- **Understanding the differences in the meaning of cluster, database, schema, role and user.**
- **Differences in the use and working of tablespaces and data files.**
- **Getting to know the data dictionary.**
- **No AWR for performance history analysis.**
- **Differences between archive logs and WAL files.**
- **Backup tools.**
- **No Oracle Enterprise Manager.**
- **Extensions. In Oracle almost all features are available with installation.**
- **Security concepts and differences between roles, users and schemas.**
- **No official support in Postgres.**
- **Immaturity of partitioning and limitations in Postgres (no global indexes).**
- **MVCC (not the Mountain View Country Club) .**
- **Tuples and dead tuples.**

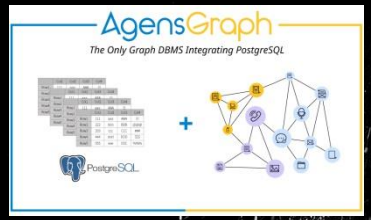
- And more……..

The Bottom Line



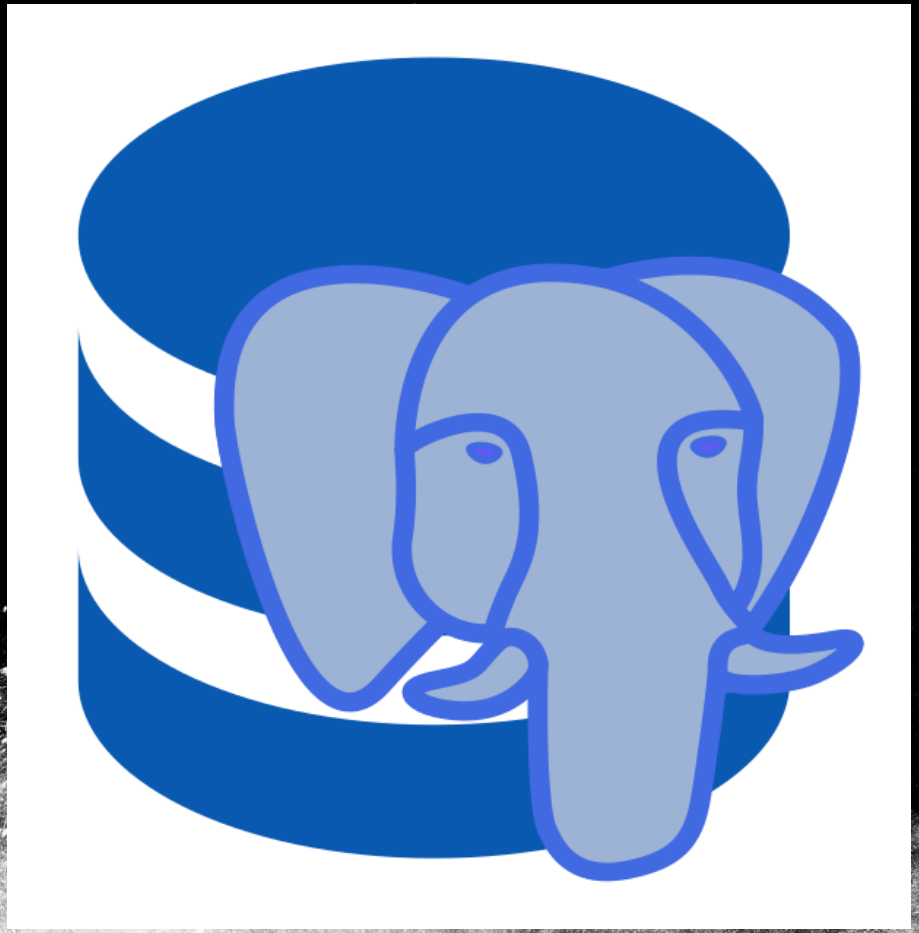
RadixTrie

PostgreSQL - Flavours



RadixTrie

The Bottom Line



Some Insights

- Assess requirements
- Do POC's
- Start with the bare basics
- List pros and cons for each
- Evaluate growth
    - Application
    - Customer base etc
    - Data
    - Resource utilisation
- Compatibility (inbound, outbound, ETL tools, dev tools)
- Cost of ownership (compare)
- Test …. Test …. Test …

Do NOT Assume !

The 3 Magic Questions

- What do you want (really really want)
- What do you need (really really need)
- Will they ever align ?

Good decisions come from experience, and **experience comes from bad decisions.**

PostgreSQL – Migrations

- From * to PostgreSQL
  - Ensure compatibility
  - Background processes
  - Monitoring (current compared to new)
  - Internal structures (tables, columns, types, sequences etc
  - Procedures, functions, triggers (all levels)
  - Hardware requirements (CPU / Memory)
  - Understand the data been migrated

➢ Data in “LONG, RAW, BLOB, GRAPHICS, PDFs” etc
  - Test different options in PostgreSQL
  - Check data after loads
  - Compare sizes

RadixTrie

PostgreSQL – Migrations

- To the Cloud !
  - Ensure compatibility
  - Proper planning and testing
  - Hardware requirements (CPU / memory / IOPS)
  - Background processes
  - Monitoring (current compared to new)
  - CLOUD limitations
  - Performance
  - Scalability and growth (all directions)
  - Security
  - Networking
  - Connectivity
  - ➢ Identifying the right apps and DB's for migration to the CLOUD
  - ➢ Cost (plan and cost for at least 2 year in advance)
  - ➢ Choose the correct CLOUD partner

RadixTrie

MONITORING

ManageEngine

solarwinds

DATADOG

PERCONA
Monitoring and Management

redgate

Quest
Foglight

Nagios® Core™

ClusterControl
severalnines

pgAdmin
Management Tools for PostgreSQL

EDB POSTGRES

sematext

checkmk

APPDYNAMICS

Grafana

RadixTrie

Monitoring

- Server and Operating System
  - CPU Usage
  - Network usage
  - Disk space / disk utilisation
  - RAM usage
  - Disk IOPS
  - Swap space usage
  - Network errors
- Database
  - Buffer cache performance (cache hits vs disk reads)
  - Number of commits
  - Number of connections
  - Number of sessions
  - Locks
  - Replication
  - Long running queries

Monitoring

- Evaluate each tool (open source or licensed)
- Compare with what's in current / planned environment
- Compatibility (Linux, Windows, in the Cloud)
- Cluster setup (HA Proxy, pgBouncer, pgPool, EDB BDR etc)
- Read-only replica's
- Replication lag and connectivity
- Integrated alarms
- Integration with incident and or alarming management systems
- Error logging (Splunk)
- Reporting (month-end , trend analysis, data for planning etc)

- What is missing:
    - How do I monitor this ?
    - How do I add alarms ?
    - How to be proactive ?

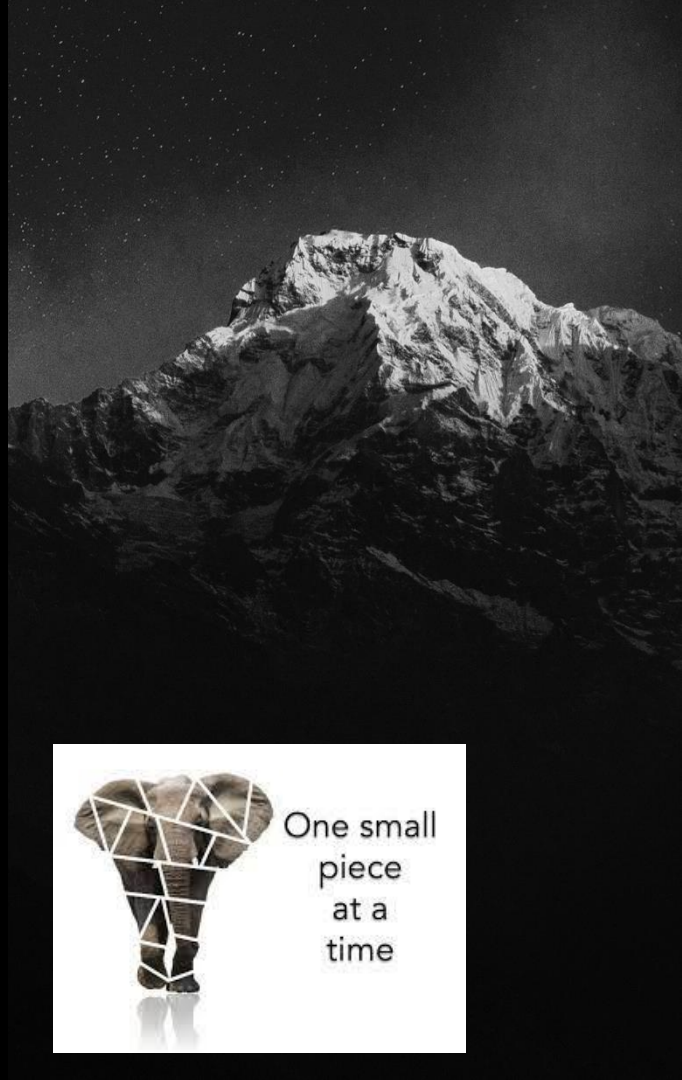
RadixTrie

Speed-Up PostgreSQL

RadixTrie

Everest …

- **Indexing:** Ensure that your database tables have appropriate indexes on columns frequently used for filtering, sorting, and joining data.
  Remove all unused indexes.
- **Foreign Key Constraints:** Foreign key constraints can improve data integrity, but they can also slow down queries.
- **UNIQUE Keys:** Avoid using UNIQUE keys unless you absolutely need them. This can slow down inserts and updates.
- **Connection Pooling:** Implement connection pooling to reduce the overhead of establishing new database connections for each user or request.
- **Configuration Tuning:** Adjust PostgreSQL configuration parameters according to your system's hardware resources and workload characteristics. Parameters like shared_buffers, work_mem, and effective_cache_size can significantly impact performance.
- **Connection Limits:** Limit the maximum number of connections allowed to the database to avoid resource contention.
- **Server Hardware**: Ensure that your server hardware, including CPU, memory, and storage, meet the demands of your workload.
- **Read Replicas:** Implement read replicas to offload read traffic from the primary database.



- **Other (Longer-Term Projects).**
  - Partitioning
  - Archiving
  - Purging

- **The Winner:**
  - Vacuum (**Bronze**)
    - Ensure that tables, especially tables with huge updates and deletes are frequently vacuumed
    - Preferable auto-vacuum (*the friendly one*)
  - Analysing (**Silver**)
    - Ensure that tables are frequently analysed and that statistics are up to date
    - Preferable auto-analyse (*another friendly one*)
  - Query Tuning (**GOLD**)
    - Ensure queries are optimised.
    - Do explain plans.
    - Monitor database activity and queries (***frequently***)
    - Test against recent production data and volumes

PostgreSQL

RadixTrie

RadixTrie