PostgreSQL Performance

Case Study of Critical System – 9.4

March 3, 2017

Aniruddha Deshpande, Prajakta Dhote

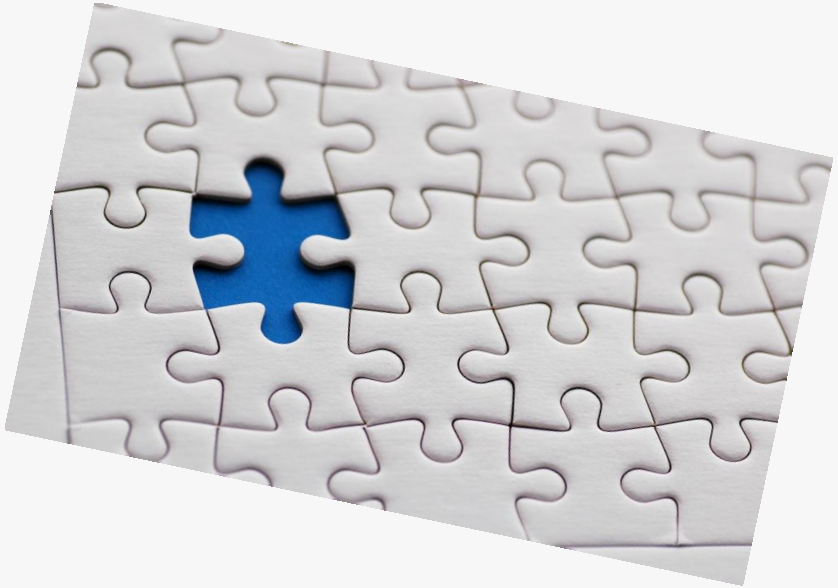
Agenda

- PostgreSQL Performance Proactive and Reactive Proactive Approach
- Reactive Approach

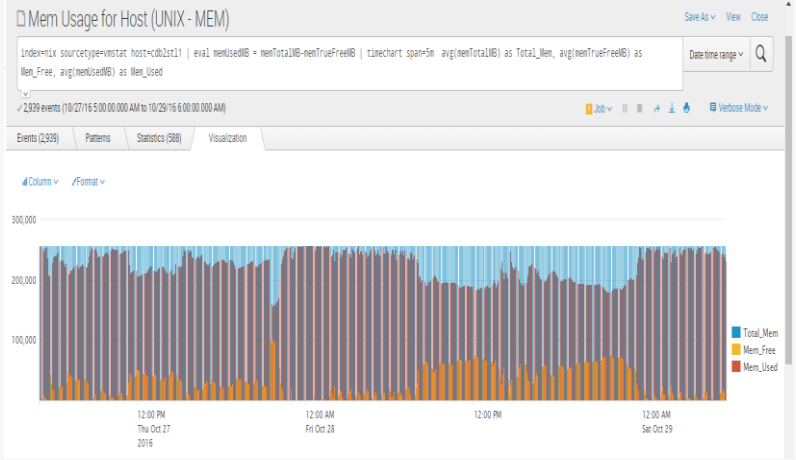
What to follow and we have followed

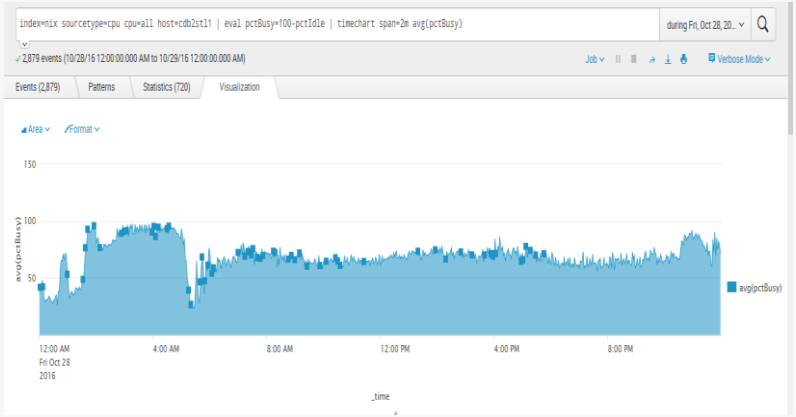
Reactive Approach

- Bring Up Current Pain Points



Reactive Approach





- Technical Analysis of entire systems

  - Gather Disk/Memory info

  - System Anlaysis with Splunk and Dynatrace tools.

  - Splunk tool gathers system stats including CPU, Memory, I/O etc.

  -Dynatrace captures transaction flow of application from end user to application layer.

Reactive Approach

Pull out data from DB end

- Database Findings
  - Top SQL statements
  - Ugly SQL's
  - Peak time hit ratio
  - Partitioning
  - DB growth Pattern
  - Indexing
  - Vacuum

Proactive Approach

Server data tells about your hosts' health

- Memory
- CPU
- Network
- Disk

Proactive approach

What to avoid – Too small to notice but trust us.. They gives results.

- Do not run anything besides PostgreSQL on the host
- If PostgreSQL is in a VM, remember all of the other VMs on the same host
- Disable the Linux OOM killer
- Sessions in the database
- Using the database as a filesystem
- Frequently-locked singleton records
- Very long-running transactions
- Using INSERT instead of COPY for a huge load of data
- Mixing transactional and data warehouse queries on the same database

Proactive Approach

Do periodic checks for

- Unused indexes
- Bloated indexes
- Bloat
- Vacuuming

Proactive Approach

Corruption

- Do you see these errors on your database server –
  - ERROR: could not read block.....
  - ERROR: invalid page header in block.....
  - ERROR: missing chunk number 0 for toast value.....

- This suggests possible corruption

- What impact does it have on my database operation?
  - It may halt your complete database operation
  - Crash
  - Wrong results
  - Does that got into your backups as well?

Proactive Approach

Corruption Part 2

- How corruption can be detected
  - pg_filedump
  - pg_dumpall/pg_dump and pg_restore
  - pg_catcheck – Catalog integrity check utility
  - Checksums

Proactive Approach

PgBadger

- Great tool
- Extract reports and do a detailed analysis even though no one reports problem
- Try to check for running queries. Optimize wherever you can.
- PgBadger can give you great hints about performance.
- Cron periodic pgbadger reports. Share with application teams.

Proactive Approach

Backups

- Backups are very important.
- Can save you from all the bad possibilities after disaster.
- But are my backups really a backup? Or just a bunch of files?
- Ask yourself, when did you tried to restore the backup last time?
- Backups can really help you when they are valid.
- Do periodic restore of your backups on same hardware server.
- This can provide you RTO/RPO values.

Questions