A Postgres Orchestra

by Akshay Gupta

Abstract

Design a postgres cluster that is easy to maintain, fault-tolerant and highly-available

Description

We go over replication strategies, failure scenarios, automatic failovers, clustering and keeping our cluster highly available. Instead of describing the failure scenarios theoretically, we will replay them in real-time on our local machines and test the resiliency of the solutions presented. The majority of this talk will attempt to keep the clustering system self-reliant and allow the application code to be sufficiently isolated from the database system while still providing an accessible mechanism that is easy to monitor and tweak. We will also examine the problem of correctly broadcasting the current master and go over some techniques that will help monitor the health of our cluster.

Outline

Each step below has a tentative time it would take to cover the topic.

Introduction to clustering and replication [5 min]

- Briefly go through synchronous and asynchronous streaming replication

Docker Setup [5 min]

- This will not assume any prior knowledge about docker – we won't go into detail about how it works. We'll just use it to test and showcase our scenarios.
- Setup a postgres cluster with docker.
- Introduce `tc` and blockade[1] to create network latency, arbitrary network partitions, packet drops etc. While still maintaining connection to the host to monitor the health.
- Showcase some quick `pgbench` scripts to simulate slow/fast read and writes

Repmgr [5 min]

- Briefly explain how repmgr works
- Go through various non-obvious gotchas that one should be aware of
- Automating parts of repmgr
  - Scripting various events
  - Electing synchronous standbys
- Compare repmgr with other tools like Patroni[2] or Stolon[3]

Failures (a.k.a disasters) [10 m]

- Go through some high-level failures:
  - Out of disk space
  - Too many connections
  - High network latency
  - Network partitions
  - Partial network partitions
  - Accidental truncates
  - Actual disasters
- Establish some effective scenarios they may cause to our cluster:
  - Mulitple masters or no masters
  - Unreachable nodes
  - Failed failovers

Governance [10 min]

- Solve the problem of discovering and broadcasting the latest master
  - Using Zookeeper / etcd to store the latest master and preventing writing stale masters
  - Split-brain and how witness servers can help in providing a voting majority.
  - Using Agrajag[4] to do all of the above
  - Compare Agrajag with other tools like Patroni and Stolon

Load Balancers [10 m]

- Describe how we can put the clusters behind HAProxy / PGPool / Keepalived and UCARP
- Compare use-cases and scenarios for each mechanism
- Discuss how a Virtual IP mechanism may not be the silver bullet after all
- Contrast the benefits of having a Layer 4 load balancer with Layer 7

Connection Pooling [5 m]

- Compare application-level pooling (like Hikari or BoneCP) with tools such as PgPool

Backup Strategies [5 m]

- Run down a priority list of backup strategies
- Go through Logical Backups, Filsystems Backups (showcase on btrfs and zfs), Standbys
- Explain Barman
- Describe a sample configuration of good practical geographic configurations of all the recovery mechanisms

About

Akshay is modern-day aesthetically-relevant F/OSS hacker. Long time Rubyist, sufficiently long time Clojure programmer and a recently turned DBA. He's contributed to GNOME, Haiku in the past and has worked as a consultant in building energy saving projects and recently a large-scale multi-variate testing platform with tight SLAs and difficult database problems.

He likes to play and generate music in his free time and has recently found the pleasures of drowning in his training in Brazilian Jiu-jitsu.

He's a partner at nilenso: an employee-owned software cooperative based in Bangalore.

Appendix

- This talk is a more hands-on, advanced continuation to the work some of us did at nilenso, previously presented by Srihari Sriraman: https://www.youtube.com/watch?v=OzoyRv_7fEk
- I've spoken at RubyConf and run a few workshops in the past:
  - http://lanyrd.com/2013/rubyconfindia/schdhh/
  - https://metarefresh.talkfunnel.com/2015/1326-a-quick-and-hopefully-painless-ride-through-reactj
  - http://www.meetup.com/Bangalore-Clojure-User-Group/events/202981182/
  - http://www.meetup.com/Bangalore-Ruby-Users-Group/events/174436672/

[1] https://github.com/dcm-oss/blockade
[2] https://github.com/zalando/patroni
[3] https://github.com/sorintlab/stolon

[4] https://github.com/staples-sparx/Agrajag
[5] https://github.com/paunin/postgres-docker-cluster