Ghostferry: the swiss army knife of live data migrations with minimum downtime

Shuhao Wu
Shopify
April 24, 2018
Problems with Existing Tools

- Cloud limitations
  - No access to the filesystem.
  - No direct access to commands like CHANGE MASTER.
- Performance impact of mysqldump.
- Must copy a whole table at a time.
- CHANGE MASTER ...? mysqldump --what?
Ghostferry: The Solution

- Easy: single binary solution to moving data.
- Customizable: a library to implement arbitrary migration flows.
- Proven: used to migrate 70 TiBs of data at Shopify.
- Confident: algorithm modeled and understood with formal methods (TLA+).
- Open source: MIT, https://github.com/Shopify/ghostferry
Ghostferry: the Swiss Army Knife of Live Data Migrations with Minimum Downtime

- General Session
  - Tuesday
  - 4:50 – 5:15 PM
  - Room G
Vitess

High performance, scalable, and available MySQL clustering system for the Cloud

Sugu Sougoumarane
CTO, PlanetScale
@ssougou
Database trends

- Transactional data explosion
- Move to the cloud
- DBAs transitioning to DBEs
Vitess capabilities

- Leverage MySQL
- Take away the pain of sharding
- Make resharding robust and easy
- Pluggable sharding schemes
- Cloud-ready
- Observability
The Community

In production

- YouTube
- slack
- JD.COM
- Square
- HubSpot
- stitchlabs
- Flipkart
- nozzle
- BetterCloud
- Quiz of Kings

Evaluating

- GitHub
- AXON
- AppLift
- Lazada
In conclusion

● Scale out MySQL
● Run in the cloud
● Vitess sessions
  ○ Migrating to Vitess at (Slack) Scale
  ○ Designing and launching the next-generation database system @ Slack: from whiteboard to production
  ○ Observability features of Vitess
List of job postings, popular among startups. 1068 messages (as of Apr 17 2018)

https://news.ycombinator.com/item?id=16735011
Already automated:

- Setup/tune hardware, OS, FS
- Provision Postgres instances
- Create replicas
- High Availability: detect failures and switch to replicas
- Create backups
- Basic monitoring

Little to zero automatization:

- Postgres parameters tuning
- Query analysis and optimization
- Index set optimization
- Detailed monitoring
- Verify optimization ideas
Meet **postgres_dba**

**postgres_dba** – The missing set of useful tools for Postgres [https://github.com/NikolayS/postgres_dba](https://github.com/NikolayS/postgres_dba)

```
test=# :dba
Menu:
  0 - Node & Current DB Information: master/replica, lag, DB size, tmp files, etc
  1 - Databases: Size, Statistics
  2 - Table Sizes
  3 - Load Profile
  a1 - Current Activity: count of current connections grouped by database, user name, state
  b1 - Tables Bloat, rough estimation
  b2 - B-tree Indexes Bloat, rough estimation
  b3 - Tables Bloat, more precise (requires pgstattuple extension; expensive)
  b4 - B-tree Indexes Bloat, more precise (requires pgstattuple extension; expensive)
  b5 - Tables and Columns Without Stats (so bloat cannot be estimated)
  i1 - Unused/Rarely Used Indexes
  i2 - Unused/Redundant Indexes Do & Undo Migration DDL
  i3 - FKs with Missing/Bad Indexes
  p1 - [EXPERIMENTAL] Alignment Padding. How many bytes can be saved if columns are ordered better?
  s1 - Slowest Queries, by Total Time (requires pg_stat_statements extension)
  s2 - Slowest Queries Report (requires pg_stat_statements)
  t1 - Postgres parameters tuning
  v1 - Vacuum: Current Activity
  x - Turn Wide Mode ON (currently OFF): show more details, more columns
  q - Quit

Type your choice and press <Enter>:
```
Back to full-fledged automation

- **Detect** performance bottlenecks
- **Predict** performance bottlenecks
- **Prevent** performance bottlenecks

The ultimate goal of automatization
DIY automated pipeline for DB optimization

How to automate database optimization using ecosystem tools and AWS?

Analyze:
- pg_stat_statements
- auto_explan
- **pgBadger** to parse logs, use JSON output
- **pg_query** to group queries better

Configuration:
- **annotated.conf**
- **pgtune, pgconfigurator, postgresqlco.nf** (wip)
- **ottertune**

Suggested indexes
- (useful: **pgHero, POWA, HypoPG, dexter, plantuner**)

Conduct experiments:
- **pgreplay** to replay logs (different log_line_prefix, you need to handle it)
- EC2 spot instances

Machine learning
- **MADlib**
Meet PostgreSQL.support

Al-based cloud-friendly platform to automate database administration

Steve
AI-based expert in database tuning

Max
AI-based expert in query optimization and Postgres indexes

Nancy
AI-based expert in resource planning. Conducts experiments with benchmarks

Sign up for early access:
http://PostgreSQL.support
Thanks!

Come hear more: Wednesday, 11:00 a.m.

Nikolay Samokhvalov

ru@postgresql.org

twitter: @postgresmen

http://PostgreSQL.support
Andy's Guide on How to Get Tenure in Databases
Hi Andy,

I contact you to let you know, that I have developed an initial prototype of a database, that it could be of your interest.

The prototype is a Key/Value in-memory database, with the ability to perform data access in constant time. The data is grouped into DATASET of ordered data. Each INSERT automatically orders the new element and in Constant time. You can also...
World’s First
"Self-Driving"
Database

Oracle Autonomous Database
No Human Labor - Half the Cost
No Human Error - 100x More Reliable

Copyright © 2019 Oracle. All rights reserved.

Autonomous Database

B. This I know is
that you will love
company together
now if you want to
company.

This is what we
all in

Crazy Emails Received
Emails Per Month
1970s: Self-Adaptive

1990s: Self-Tuning

2010s: Self-Driving

Self-Driving Database Management Systems

Andrew Pavlo, Gustavo Angulo, Jay Aulridge, Habib Lin, Jidun Liu, Lin Ma, Prashanth Manen Todui, C. Mohy, Matthew Peirce, Ian Quinn, Sidinahmar Pillai, Anthony Tomasic, Skyh Tior, Dihan Van Alen, Zong Wang, Yagun Wu, Haiyan Xue, Tingting Zhang

Carnegie Mellon University, National University of Singapore

ABSTRACT

In the past few decades, both commercial and academic research has focused on self-adaptive strategies in database management systems (DBMS). Most of the previous work, however, has been performed in a research laboratory setting, and the DBMS is not designed to address the needs of real-world database systems. In this paper, we present an overview of self-adaptive strategies for DBMSs, and discuss the challenges and opportunities in implementing and deploying self-adaptive DBMSs.

1. INTRODUCTION

The idea of a self-adaptive DBMS is to have the database management system learn from the data it manages and adjust its behavior accordingly. This approach has been shown to be effective in many applications, such as financial trading systems, where the system must adapt to changing market conditions.

Much of the work on self-adaptive systems has focused on monitoring and adjusting the system to improve performance. For example, some work has focused on using the data to improve the database's ability to handle queries, while other work has focused on improving the database's ability to handle transactions.

2. PROBLEM OVERVIEW

The main challenge in implementing a self-adaptive DBMS is to design a system that can learn from the data it manages and adjust its behavior accordingly. This requires a deep understanding of the data and the system itself, as well as the ability to adapt to changing conditions.

This paper is co-published with IEEE Data Eng. Bull. 15(4), 2012; and ICDE Proceedings. (2013).
Self-Driving DBMS

→ What to change?
→ When to change it?
→ Was it helpful?
Today @ 11:30am
Room 203
@andy_pavlo