



KubeCon



CloudNativeCon

Europe 2023

Vitess: Introduction and Real World Usage

Arthur Schreiber, Florent Poinsard



 @vitessio

Arthur Schreiber



Maintainer
Staff Software Engineer

@arthurschreiber

GitHub

Florent Poinsard



Maintainer
Software Engineer

@fouioui

 **PlanetScale**

Agenda

Vitess Overview

Vitess at GitHub

New and Upcoming Features

Q&A and Resources

Overview of Vitess

What is Vitess?

Scalable Distributed
Cloud-Native Database
System Build around MySQL

CNCF graduated project

Massively scalable and highly
available

Vitess in production



Key Adopters

Slack
100% on
Vitess

GitHub

JD.com
10000+
databases

PlanetScale
Database Service



Slack Blog Post (Scaling Datastores at Slack with Vitess):
<https://slack.engineering/scaling-datastores-at-slack-with-vitess/>

 @vitessio

Community

15
maintainers

238
contributors,
109 PR authors

57
companies

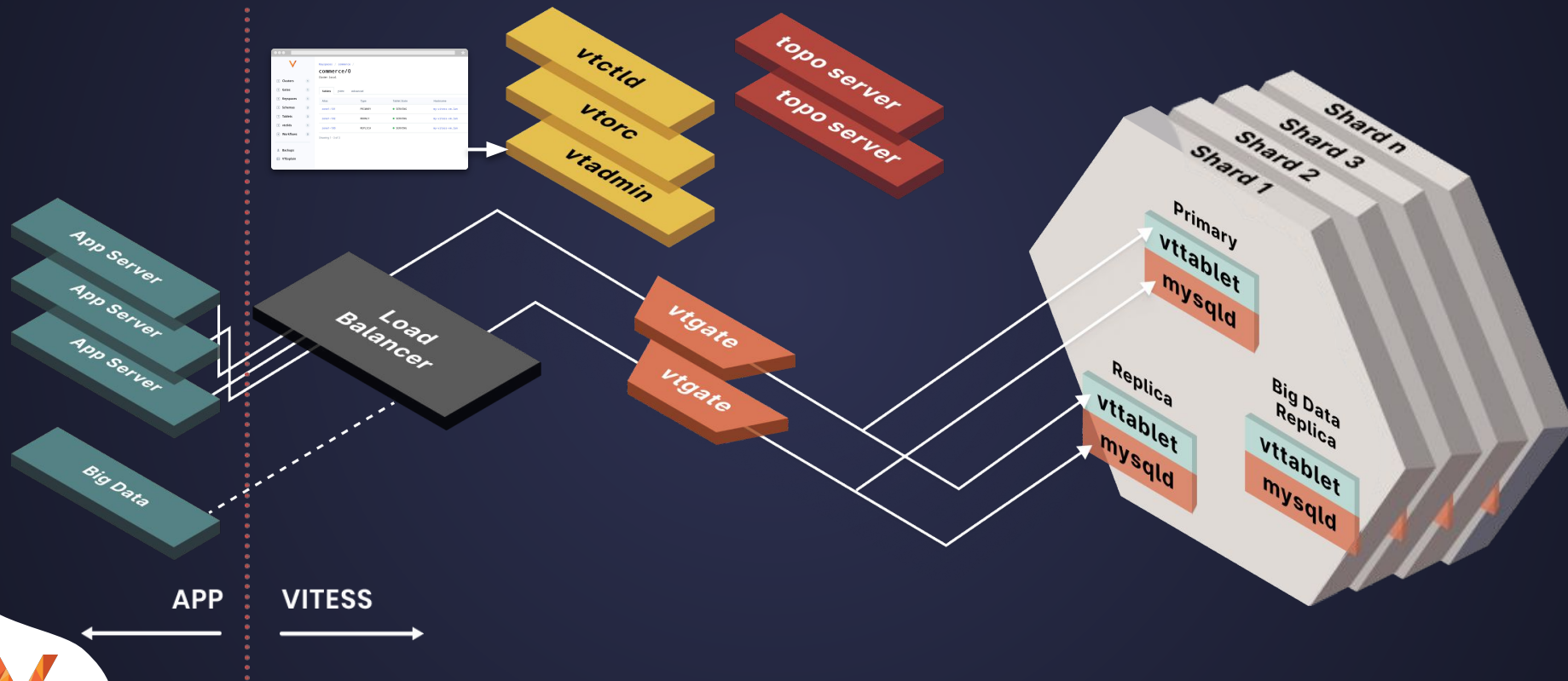
PR authors from
22 companies

Glossary

Keyspace

Shard

Architecture



Features

MySQL Compatibility

Resharding

Materialization

Cluster Management

Online Schema Changes

Seamless Backup/Recovery Operations

Query Consolidation

Automatic Failure Detection and Repair

Vitess at GitHub



MySQL At GitHub

80 clusters

1200+
MySQL
instances

Bare Metal

“Feature” and
“shared”
clusters

MySQL At GitHub

5 MM QPS
(replicas)

500 k QPS
(primaries)

330 TB
(sum of data stored
on primaries)

MySQL Scaling

New
clusters for
new
features

Break up
existing
clusters

Add more
replicas

Larger
machines

MySQL Problems?

Expensive
hardware
upgrades

Unbearable
schema
migration
times

Buffer pool
thrashing

Replication
lag

Why Vitess?

MySQL at
its core

Sharding
model fits our
data model

Acceptable
query
compatibility

Timeline

2019 - First experiments

2020 - Notifications

Early 2022 - Actions / Checks / Statuses

Early 2023 - Issues / Pull Requests

Today

20+
Keyspaces

~150 TB

750k QPS
(at peak)

Issues / Pull Requests

16 shards
16 primaries
48 replicas

~26 TB

30k QPS
(primaries)

220k QPS
(replicas)

Issues / Pull Requests

	Before	After
MySQL Hosts	1 primary 100 replicas	16 primaries 48 replicas
Memory per Host	768 GB	256 GB
Disk read rate (peak, sum across all replicas)	~11 GB/s	~800 MB/s
Disk read rate (peak, sum across all primaries)	~100 MB/s	~20 MB/s
Online schema change duration (largest table)	3 weeks	2 days

Current Architecture

Built on top of our MySQL setup

vttablets + MySQL on bare-metal

vtgates, vtctld, vtadmin in
Kubernetes

Currently running on Vitess v15

Future

Upgrade to newer Vitess versions

Potentially move more clusters to Vitess

Hybrid solution (Vitess and MySQL)

Conclusion

Vitess has enabled GitHub to
scale MySQL further

If you're running a large
MySQL setup, check it out

New and Upcoming Features

New (v15 & v16)

VTOrc

VTAdmin

CLI Flags

Incremental Backup and Point
in Time Recovery

Documentation

Views Support

Upcoming (v17 & beyond)

Foreign Keys Support

Schema Tracking Improvements

Improve MySQL Compatibility

Enhance Arewefastyet UI



Resources

- Website: vitess.io
 - [Docs](#)
 - [Tutorials](#)
 - [Source code](#)
- Slack: vitess.slack.com

Thank You

Q&A

