KubeCon | CloudNativeCon

China 2018

Running Vitess on Kubernetes at Massive Scale - JD.com case study.

About PlanetScale

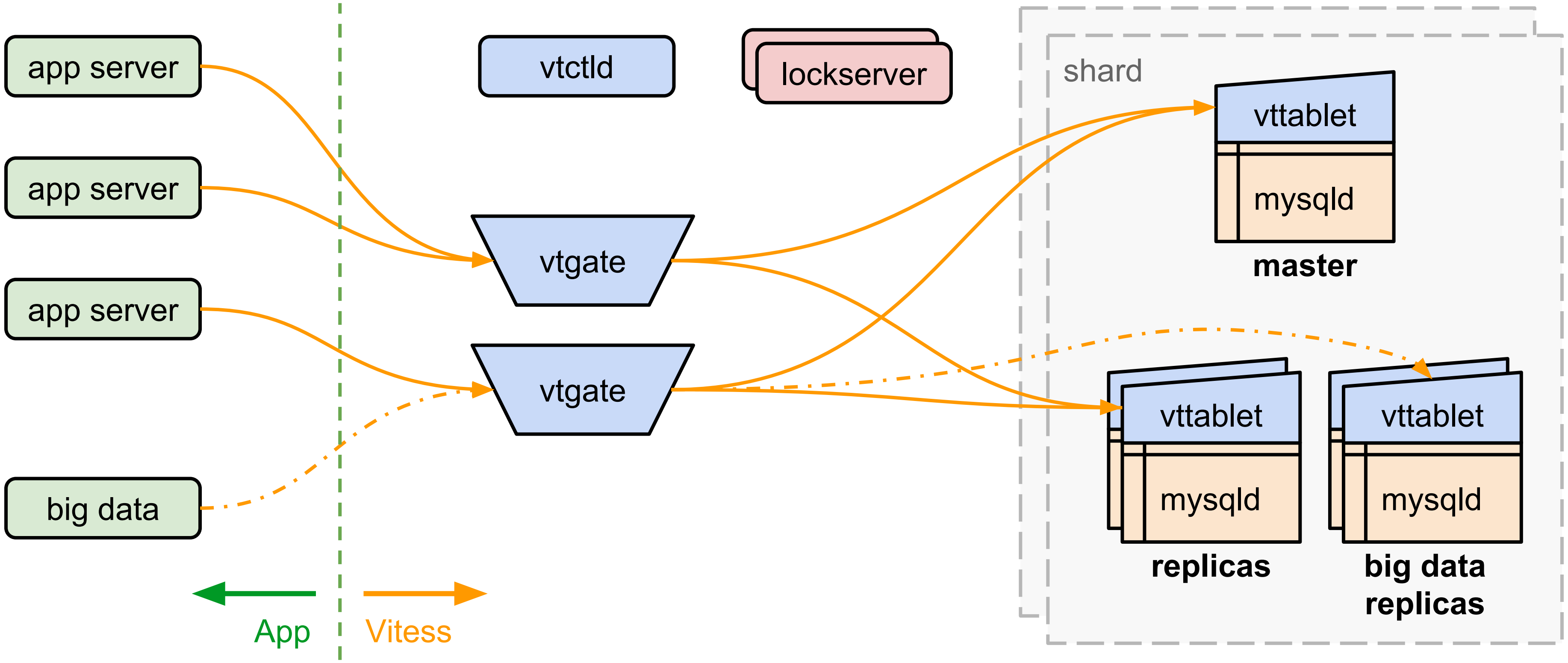
Founded in early 2018 to help operationalize Vitess

- Jiten Vaidya (CEO, Managed teams that operationalized Vitess at Youtube)
- Sugu Sugumaran (CTO, Vitess community leader)

Offerings

- Open Source Vitess Support
- Custom Vitess Development
- Kubernetes Deployment Manager
- Cross-cloud DBaaS

Vitess Architecture



Vtgate in Kubernetes

- Stateless proxy
- Accepts connections as a MySQL compatible server
- Contains GRPC endpoint and Web UI
- Computes target shards
- Sends queries to vttablets for targeted shards
- Receives, collates and serves response to application

- Vtgates can be created as load increase
- Start n vtgates as a Replica Set
- For co-located workloads start one vtgate per node and expose with a ClusterIP

Vtctld in Kubernetes

- Vitess Control Plane
- Serves a Web UI
  - Operational commands
  - Status
  - Topology browser
- Serves an API over GRPC
  - Used by vtctlclient tool
- Supports resharding workflows

- Start one or two vtctld processes per cell
- Start them as a Deployments
- Expose them behind a Service

lockserver (etcd) in Kubernetes

- Knits the Vitess cluster together
- Backing store for metadata
  - Service discovery
  - Topology
  - VSchema
- Not used for query serving
- Needed for any change in topology
  - Add a keyspace
  - Add a shard to keyspace
  - Add a tablet to a shard
  - Change master for a shard

- One global cluster
- One cluster per cell (optional)
- Use etcd-operator to spin out a cluster
- Expose etcd cluster behind a Service

Tablet (vttablet + mysqld) in Kubernetes

- Vitess Tablet is a combination of a mysqld instance and a corresponding vttablet process
- Each tablet requires a unique id in Vitess cluster
- Tablets can be of type: master, replica, rdonly
- Tablets of type “replica” can be promoted to master and should have low replica lag

- 2 containers in the same pod
- Communicate over Unix socket
- Unix socket created in Shared Volume
- Local Persistent Volume for data
- One master, 2 replicas with semi-sync replication enabled for high availability
- Replicas should not be co-located with other members of shard (Anti-Affinity)

Authentication/Secrets management

- What secrets are needed?
  - Application -> Vtgate authentication
  - Vttablet -> mysqld authentication for various roles that Vitess supports (app, dba, replication, filtered replication etc).
  - TLS certs and keys for GRPC traffic over TLS (optional)
  - TLS certs and keys for binary logs over TLS (optional)
  - TLS certs and keys for client authorization and authentication over TLS (optional)
- Use Kubernetes Secrets and mount them in pods

High Availability

- Planned reparent
  - Coordinated via lockserver
  - Existing transactions are allowed to complete
  - New transactions are buffered by vtgate
  - New master is made writable
  - Replicas are made slaves of the new master
  - Query serving is resumed
- Unplanned reparent
  - Orchestrator
  - TabletExternallyReparented
- Resharding
  - No interruption to query traffic during resharding

Supporting multiple cells

- Vitess cell is the equivalent of a failure domain (e.g. AWS availability zones or regions)
- Not necessarily the same as Kubernetes failure domain.
- Choice to use global lockserver cluster OR use one lockserver cluster per cell.
- Global lockserver cluster typically outside of Kubernetes.
- Expose lockserver behind a service definition.
- If using etcd use etcd-operator to start per cell cluster.

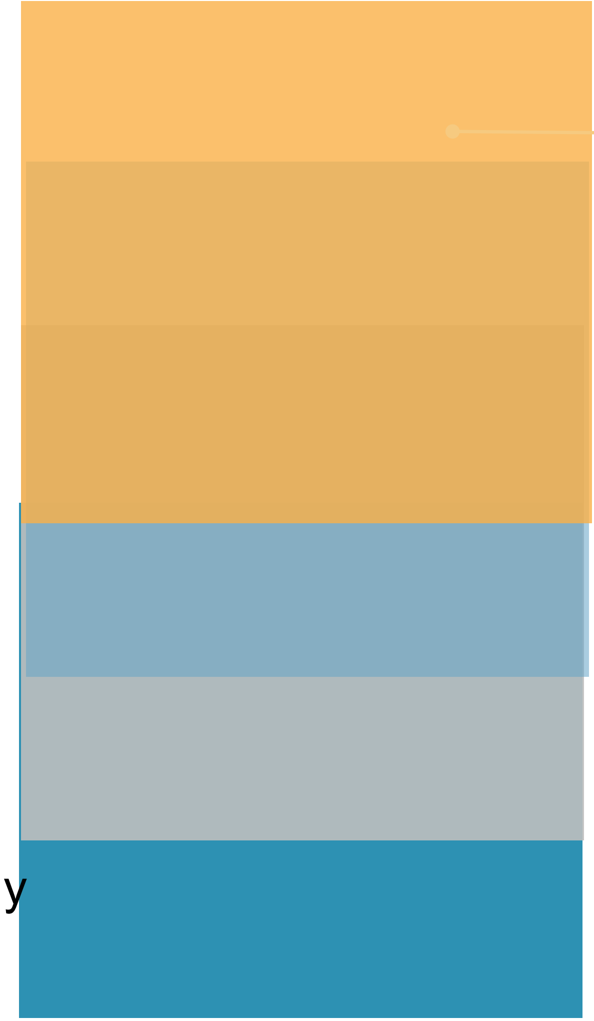
Vitess clusters spanning Kubernetes clusters

- Global lockserver cluster should be outside of Kubernetes
- While creating each cell designate which Kubernetes cluster it resides in
- Must have a non-overlapping ip space and all addresses must be routable.
- pod to pod communication
  - Needed for mysql replication and query serving
  - Achieved by Peering and Routing

About JD.COM

- China’s largest online and offline retailer
  - 300 million active users
- A **Fortune** 200 Company (#181 on the 2017 **Fortune** 500 list)
- Largest e-commerce logistics infrastructure in China
  - Covering 99% of the Consumers
  - Delivering 90% of the orders within 2 days
- Strategic Partnerships
  - Tencent
  - Walmart
  - Google

Database Management Challenges at JD.com



Application

- use multiple mysql clusters
- routing the query
- implement the query across multiple clusters.

Robustness

- No anti-affinity

Resource

- Pre-allocated resources, resource usage is low.

OPS

- Expand cluster manually
- Reshard cluster manually

Why Vitess

Online Split

- Realize the database cluster splitting online with stoping write in seconds

Functions cross shards

- Queries cross shards
- Transparent routing
- Realize the atomicity of transaction with the 2PC model

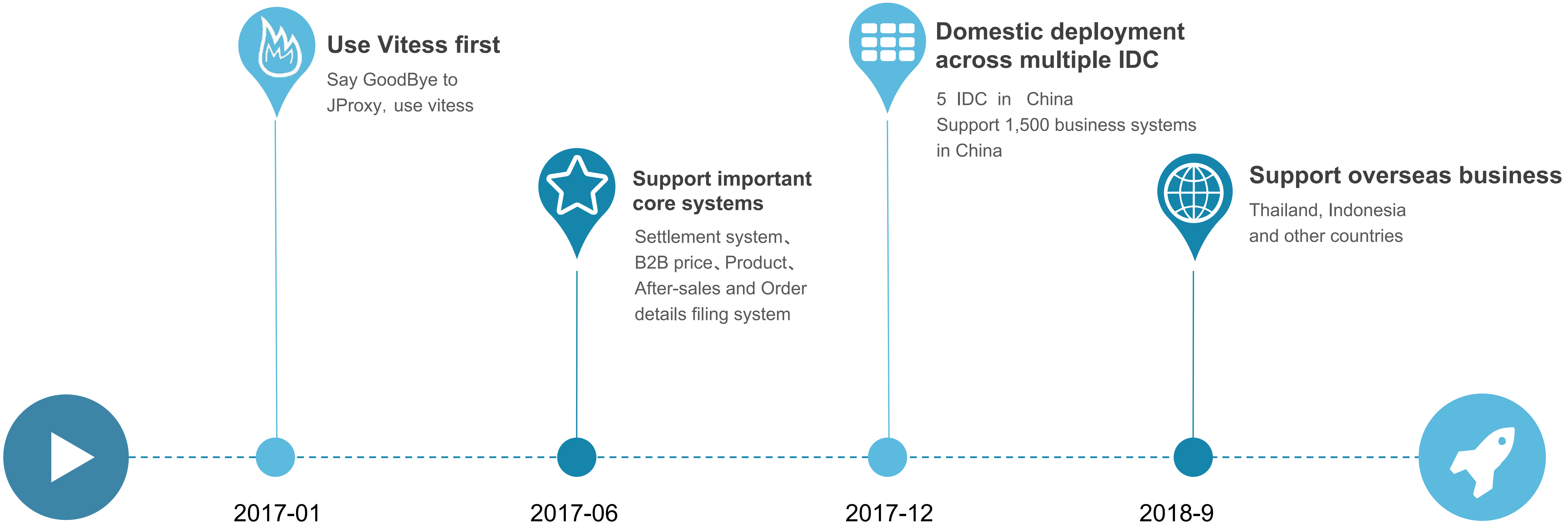
MySQL Protocol Compatibility

- Supports most SQL query statements
- Is compatible with mysql client and mysql JDBC driver

Integration with kubernetes

Vitess can integrated with kubernetes natively and 80% of databases run on docker in JD.COM, these container is scheduled and managed by kubernetes.

RoadMap



Use Vitess first

Say GoodBye to
JProxy，use vitess

2017-01

Support important core systems

Settlement system、
B2B price、Product、
After-sales and Order
details filing system

2017-06

Domestic deployment across multiple IDC

5 IDC in China
Support 1,500 business systems
in China

2017-12

Support overseas business

Thailand, Indonesia
and other countries

2018-9

Deployment

The world’s largest and most complex Vitess deployment

Deployment

KeySpace:1911
DataCenter：8
Shard：4438
Tablet：11416
Tables：552104
Most Shards/KeySpace:72

Data Size

146 TB
252 billion Rows

MySQL

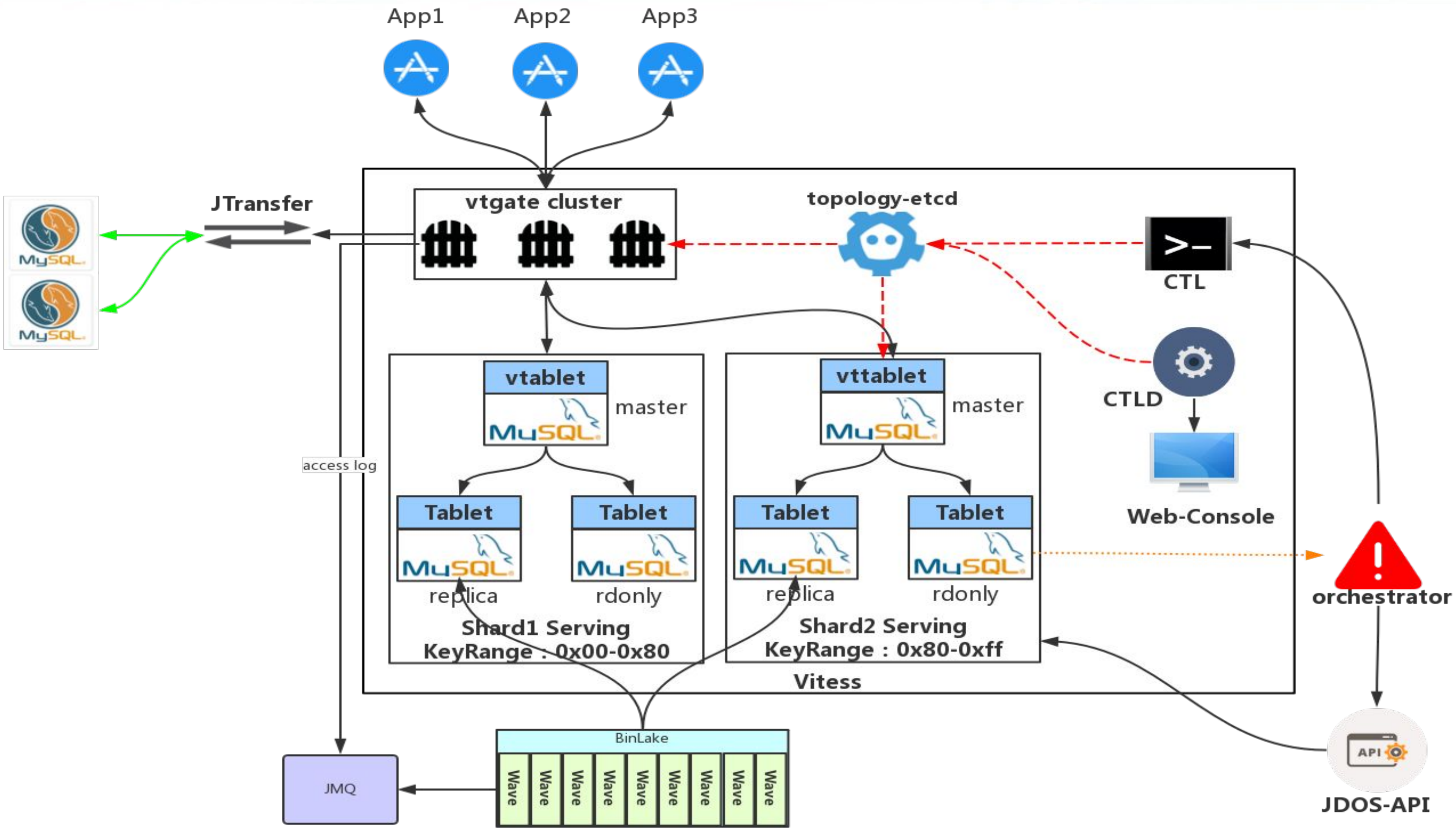
Support Business

project：1731
business：Settlement system、order details system、B2B Price、Cis_pop、Logistics billing system、Coupon and so on, OLTP

Increase

10 KeySpaces/week
10TB/week
20 billion Rows/week

Deployment


App1
App2
App3
JTransfer
vtgate cluster
topology-etcd
CTL
CTLD
Web-Console
vtablet
master
vttablet
master
Tablet
replica
Tablet
rdonly
Tablet
replica
Tablet
rdonly
Shard1 Serving
KeyRange：0x00-0x80
Shard2 Serving
KeyRange：0x80-0xff
Vitess
access log
orchestrator
JDOS-API
BinLake
Wave
Wave
Wave
Wave
Wave
Wave
Wave
Wave
Wave
JMQ

JD’S Work On Vitess

Bug Fix

- Fixed 20 Bugs
- Polling channels closed leads to high CPU utilization #3745
- Vttablet always in restore state after restart #3885
- Cannot parse SQL with some special annotations #3807
- Thread safety issues during resharding #3029
- Vtgate returns non-utf8 encoded string #2583
- Rename table bug #3774
- Refact the way of storing content in vschema

Improve the grammar

- multi-Query #3683
- Begin、commit、rollback support #3671
- Specail sql suport #3801
- Prepare #3864
- Set and auto commit #3896
- Distinct、Load、Union、Exists、ZeroFill、Having

Performance improvement

- The parallel copying
- The performance of VtGate is doubled by controlling GC frequency
- Improve the performance of sorted queries by streaming queries

MySQL Protocal

- mysql-client
- jdbc-driver
- php driver
- node js drvier
- COM_FILED_LIST #3936

JD’S Work On Vitess

Ecological

- JTransfer
- BinLake
- Data access audit
- Manage System

Improve Resource utilization

- All In One Container
- OverUse OF CPU
- 1 master、1 replication 1 readonly

Elastic scaling

Local instant capacity expansion

Split with one action

Anti-compatibility scheduling

Multiple engine

- RocksDB
- TokuDB

Challenges

Splitting

Slow
Manual

Scale Up

Can not scale up immediately

MetaData

The design
of metadata storage result in
can not deploy too large vitess
cluster

orchestrator

The design of orchestrator
result in can not manager too
many instances

Solutions

Splitting

Challenge

- There are many instances with the amount of data that more than 1 TB, lead to the split of these instances process is very slow
- There are too many business system, so it is not practical to split each shard manually

Solution

- Control the amount of data each shard strictly , make it less than 512 GB
- Parallel copy and replication, speed up the split process
- Realize the function of a key split, can automatically or manually triggered

Solutions

Scale Up

Challenge

- Peak twice every year: 618 and 11.11
- JD often make promotion
- We need to be able to improve database service ability rapidly

Solution

- Increase CPU locally without service down
- Monitor the load of physical machines and pods
- Migration with one click

Solutions

Metadata

Challenge

- So many keyspace and vschema storage design result in the vschema info’s size if larger than 1.5 MB which seriously affects the stability of etcd and leads to etcd instance oom frequently

Solution

- Store url in vschema，and get the contents of vschema from the url
- Split the value of entire vschema into the metadata of many individual keyspaces

Solutions

orchestrator

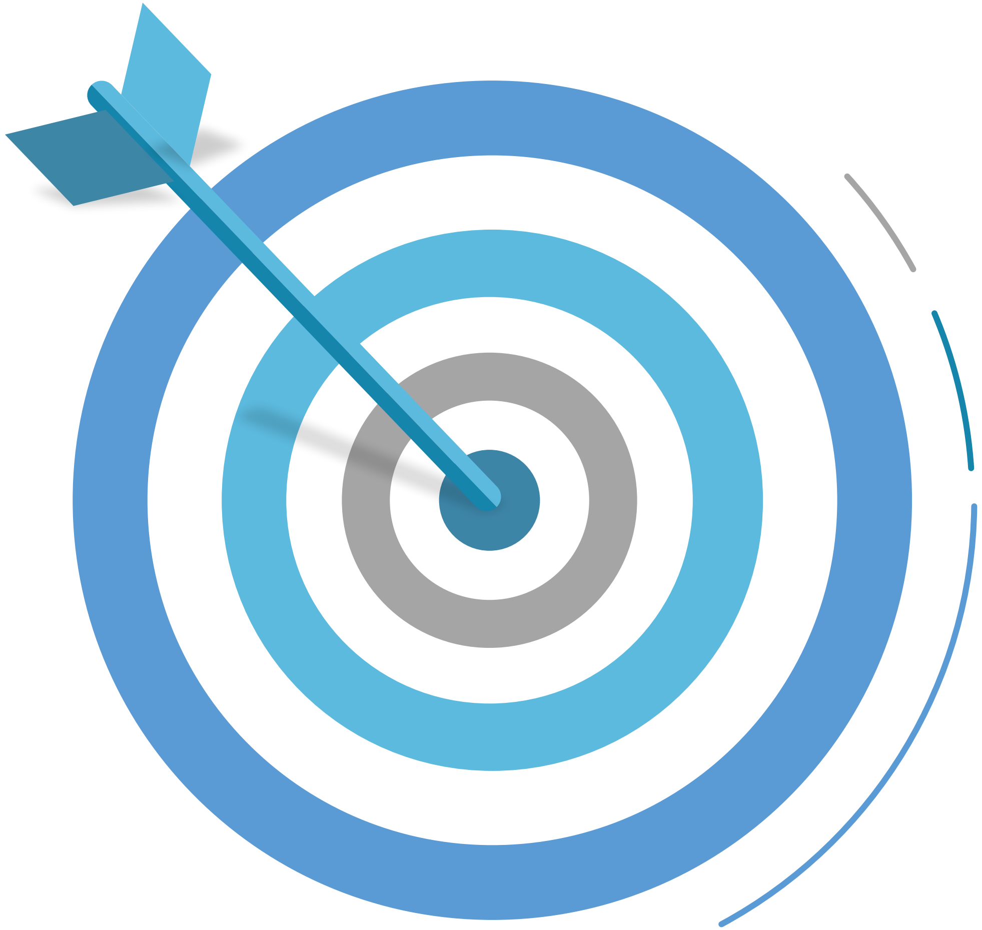
Challenge

- When the cluster monitored by orchestrator has more than 5,000 instances, the orchestrator always changing leader looply and to can not provide services

Solution

- One orchestrator per cell
- Control the number of instances in one cell below 5000

Ongoing Work And Next Step



Resharding Isolation

Each Worker is responsible for splitting up a Shard and achieving the independence of each Worker splitting without mutual influence.

Refact VSchema

Vschena's content is currently stored in one Value
We will split the vschema's content into many individual keyspace content

Auto-Balance

Automatic scaling capacity, splitting and migration of database load are realized based on monitoring data

KubeCon | CloudNativeCon

China 2018