Jurisdiction aware clusters using Vitess

2019

Quick Intro

Jiten Vaidya - CEO and Co-Founder

- Google -> Youtube -> Dropbox -> US Digital Service
- Managed groups that operated YouTube databases at massive scale.
- jiten@planetscale.com
- **@yaempiricist**

PlanetScale

- Founded in February 2018
- Venture backed: a16z, SignalFire
- 30 employees mostly in Mountain View, CA

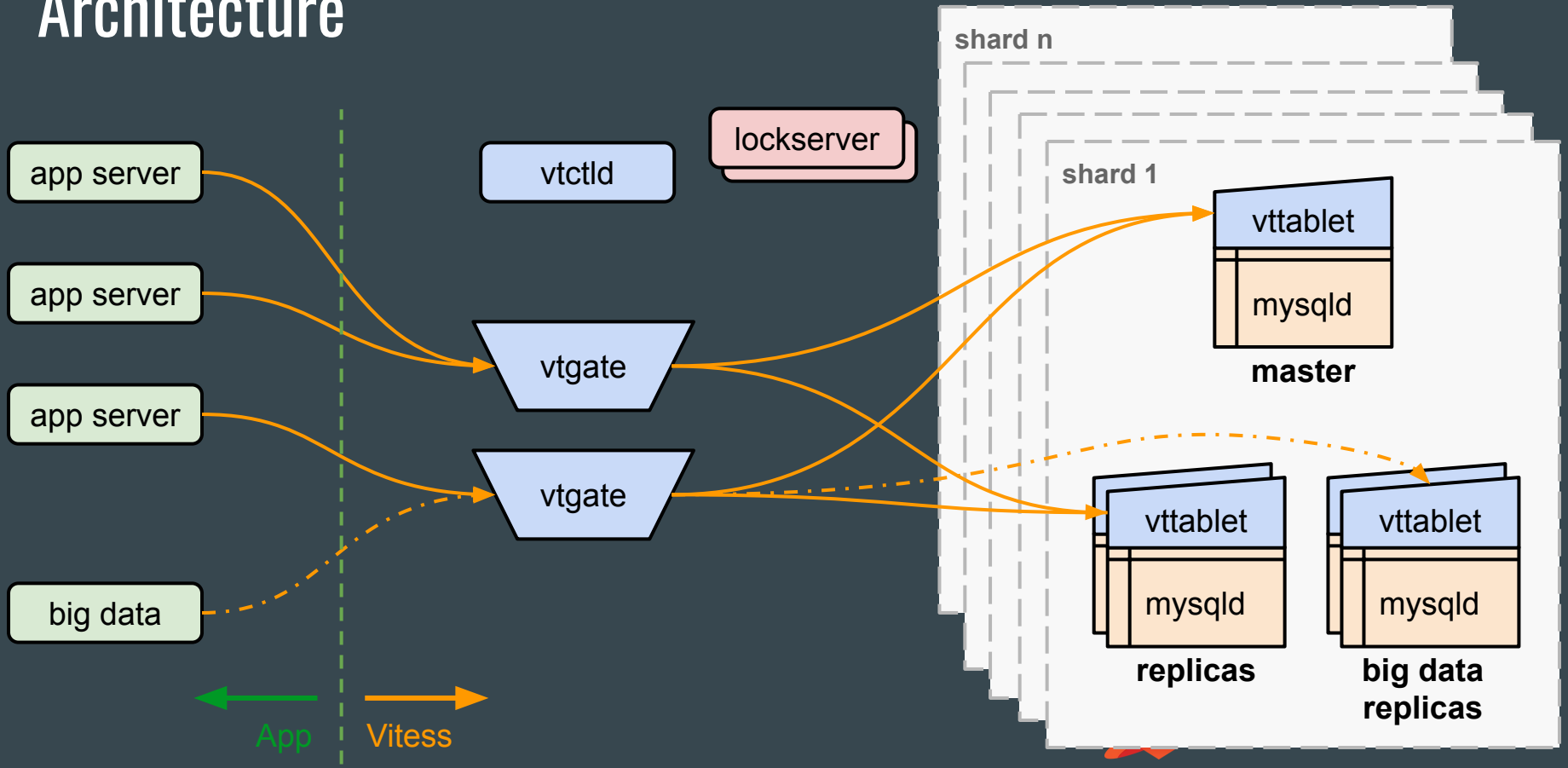
PlanetScale

Agenda

- Vitess Overview and Concepts
- Problem statement
- Cluster Design
- Demo

PlanetScale

Architecture



Concepts: *cell*

- A group of servers and network infrastructure
- A failure domain: isolated from failures in other cells.
- Examples:
  - a full data center
  - a subset of a data center, aka availability zone.
  - a Kubernetes cluster

PlanetScale

Concepts: *keyspace*

- logical database.
- If no sharding, keyspace => single MySQL database
- If using sharding, keyspace => multiple MySQL databases (all with identical schema)
- In either case, a keyspace appears as a single database from the standpoint of the application.

PlanetScale

Concepts: *Keyspace ID*

- keyspace is sharded by keyspace ID ranges.
- Each row is assigned a keyspace ID
- “Street Address” of a row
- Used to decide on which shard a given row lives
- ~~Keyspace ID: equivalent of a NoSQL sharding key~~
- Internal to Vitess, The application does not need to know anything about it.
- Not stored, computed

PlanetScale

Concepts: *Vindex*

- A way to compute Keyspace ID for any row in a table
- Vindex for a Table is defined by
  - Column name
  - Sharding function name
- KeySpaceIDForRow = ShardingFunction(ColumnValueForRow)
- For example, table name: customer, sharding column: id sharding function: hash
- For a row where id is 123, KeySpaceId = hash(123)

PlanetScale

Concepts: *Shard*

- Keyspace ID range (Begin, End)
- If Begin < KeyspaceID <= End, then row with KeyspaceID belongs in that shard.
- One master
- Multiple replicas
- Located in one or more ***cells***

PlanetScale

Concepts: *Sharding functions*
You can also add your own custom sharding function!

binary	Identity
binary_md5	md5 hash
hash	3DES null-key hash
numeric	Identity
numeric_static_map	A JSON file that maps input values to keyspace IDs
unicode_loose_md5	Case-insensitive (UCA level 1) md5 hash
reverse_bits	Bit Reversal

PlanetScale

Jurisdiction aware clusters: the problem statement

- Create a database cluster that stores data for a given country in its jurisdiction
- The client application need not be aware of jurisdictions
- For this demo:
  - **Four jurisdictions/Eight countries**
  - **americas -> USA, Canada**
  - **asia -> India, China**
  - **africa -> Kenya, Nigeria**
  - **europe -> France, Germany**

PlanetScale

Cluster design

Create a Vitess “cell” for each jurisdiction
- uswest -> americas
- asia1 -> asia
- africa1 -> africa
- europe1 -> europe

Create a “keyspace” with four shards, one shard resident in each cell

Use a *custom_geo* V-index so that data is distributed correctly using the “country” field in each row.

PlanetScale

Country to Jurisdiction Map - Used by V-index for mapping

```
{
    "jurisdictions": {
        "europe":[0,64],
        "asia":[64,128],
        "americas":[128,192],
        "africa":[192,255]
    },
    "countries":{
        "europe":["France", "Germany"],
        "asia":["India", "China"],
        "americas":["United States", "Canada"],
        "africa":["Nigeria","Kenya"]
    }
}
```

Sample data

id, name, national_id, country
1,Philip,Roth,123-456-789,United States
2,Gary,Shteyngart,234-567-891,United States
3,Margaret,Atwood,345-678-912,Canada
4,Alice,Munro,456-789-123,Canada
5,Arundhati,Roy,567-891-234,India
6,Shashi,Tharoor,678-912-345,India
.....
13,Chinua,Achebe,405-678-912,Nigeria
14,Buchi,Emecheta,506-789-123,Nigeria
15,Binyavanga,Wainaina,607-891-234,Kenya
16,Grace,Ogot,708-912-345,Kenya

PlanetScale

Demo

Start the cluster
Show cells and shards
Insert data
Show that the data is correctly distributed

Future work

Start from a single shard located in uswest and then demonstrate resharding on the fly while the application continues to write to the DB without downtime.

Seperate out China in it's own jurisdiction without application downtime.

PlanetScale

Questions