Performing online schema changes

Some numbers first

Table records

200M

Records in total

Daily average growth

1.5M

Records per day

I/O operations

6K

Read/Write operations per second during peak hours

The problem(s)

1. A MySQL InnoDB table with close to 200 million records that grows by an average of 1.5 million records per day
2. The need to add a new column to the table without any downtime
3. At this scale, online schema changes fail if I/O operations are not interrupted and it takes a long time for the operations to fail
4. Metadata Locking is real and is an unsolved problem https://dev.mysql.com/doc/refman/5.7/en/metadata-locking.html

The alternatives

1. MySQL Online DDL [4]
2. Facebook OSC [3]
3. pt-online-schema -change from Percona [1]
4. gh-ost from GitHub [2]

gh-ost

Triggerless design

Cut-over

Lag-aware

Replica readings

Uses binary log

Can be paused

On the fly changes

Throttling

Written in Go

Open Source

How it went...

```
# panic-flag-file: /tmp/ghost.panic.flag
# Serving on unix socket: /tmp/gh-ost.robokiller.calls.sock
Copy: 180907290/180907290 100.0%; Applied: 5918548; Backlog: 0/1000; Time: 58h20m32s(total),
55h27m25s(copy); streamer: mysql-bin.021750:101736747; State: migrating; ETA: due
2018-11-18 15:23:22 INFO Setting RENAME timeout as 3 seconds
2018-11-18 15:23:22 INFO Session renaming tables is 105855
2018-11-18 15:23:22 INFO Issuing and expecting this to block: rename /* gh-ost */ table
`robokiller`.`calls` to `robokiller`.`_calls_del`, `robokiller`.`_calls_gho` to
`robokiller`.`calls`
```

Foreign keys killed the scalability star

“Foreign keys complicate the tool’s operation and introduce additional risk. The technique of atomically renaming the original and new tables does not work when foreign keys refer to the table. The tool must update foreign keys to refer to the new table after the schema change is complete.”

-- Percona

Foreign keys killed the scalability star

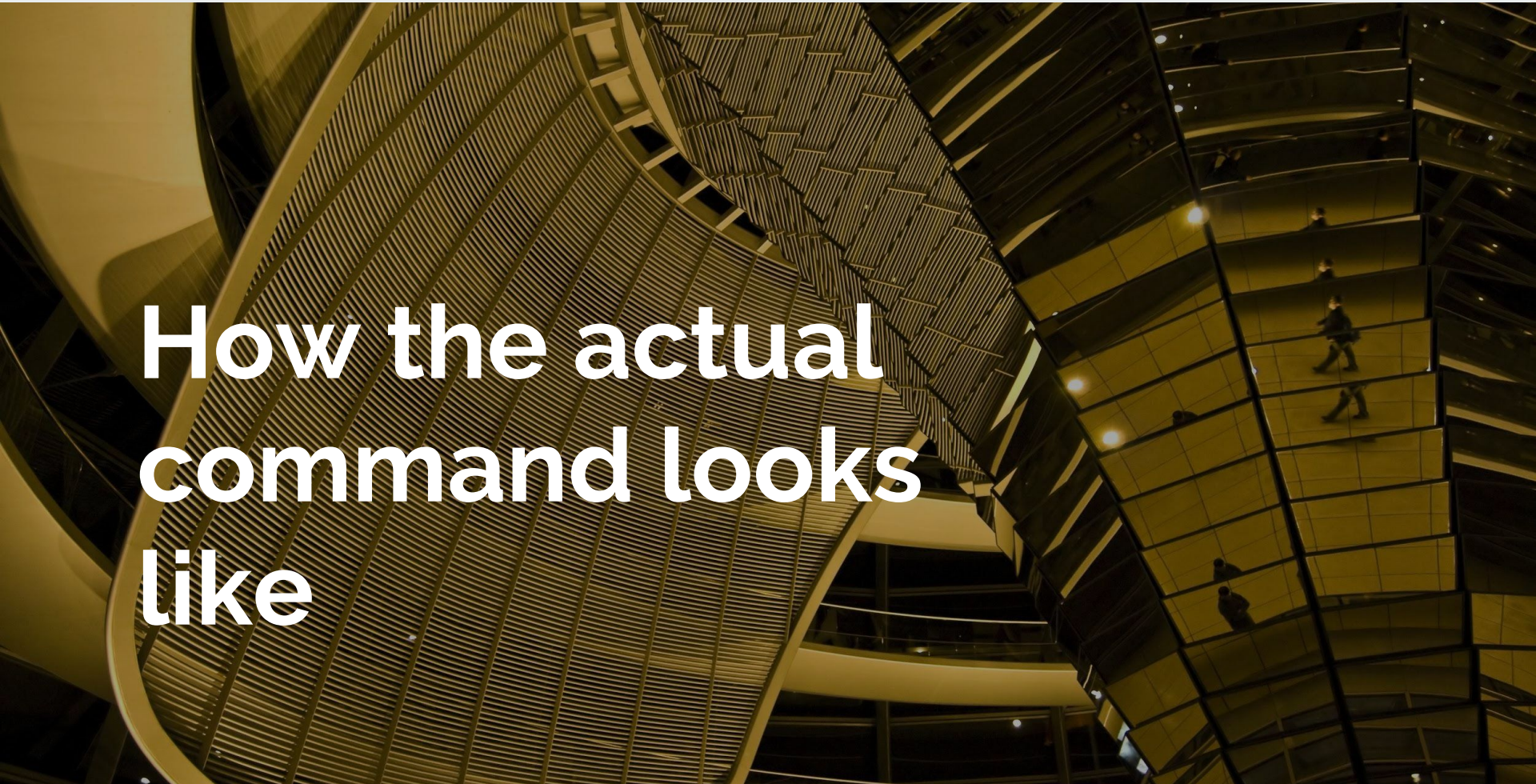
“gh-ost does not support foreign keys on migrated tables (it bails out when it notices a FK on the migrated table)”

-- gh-ost

On JOINs

“Normalized structure and a lot of joins is right way to design your database as textbooks teach you… but when dealing with large data sets it could be recipe for disaster. The problem is not the data size – normalized data normally becomes smaller, but dramatically increased number of index lookups which could be random accesses.”

-- Percona Blog


How the actual command looks like

```
./gh-ost \
--max-load=Threads_running=25 \
--critical-load=Threads_running=1000 \
--chunk-size=100 \
--throttle-control-replicas="35.196.131.244" \
--max-lag-millis=1500 \
--user="mauricio" \
--password="<redacted>" \
--host=104.196.196.65 \
--allow-on-master \
--database="robokiller" \
--table="calls" \
--verbose \
--alter="ADD COLUMN call_type TINYINT UNSIGNED AFTER ip_address__version" \
--switch-to-rbr \
--allow-master-master \
--cut-over=default \
--exact-rowcount \
--concurrent-rowcount \
--default-retries=120 \
--panic-flag-file=/tmp/ghost.panic.flag \
--postpone-cut-over-flag-file=/tmp/ghost.postpone.flag \
--execute
```

Thank you.