Migrating 3B rows to TiDB
for a high-traffic application

The tale of an ambitious migration

FOSDEM

Sorin Dumitrescu
CTO

OMNICONVERT

SaaS Platform: Explore

- A/B Tests
- Pop-ups
- Surveys
- Segmentation
- Goal tracking & Analytics

OMNICONVERT

Platform Explore in numbers

- 18k RPM normal traffic (min: 9k / max 25k)
- 3.5B experimentation records in the database
  - 1.5B visits for experiments | 2.7M new / day
  - 1.3B conversions | 2.3M new / day
  - 0.7B associations for above
  - 0.1B “other”: leads, consent, survey responses

OMNICONVERT

Topics for today

- Context & Approach
- Database Migration to TiDB
  - TiDB Architecture
  - PoC & Results
  - Performance & Stress Test Results
  - Transition plan
  - Lessons Learned
- End-user perception
- Personal conclusions

OMNICONVERT

Context

- Issues with (our setup for) MongoDB
  - Terrible reporting query performance
  - No easy way to scale horizontally (manual process)
  - Very expensive to increase storage space (no sharding)
  - No security patching
- Legacy System
  - MongoDB 3.4
  - PHP 7.0 on VMs
  - Upgrade deadlock

Context

- Issues with (our setup for) MongoDB
  - Terrible reporting query performance
  - No easy way to scale horizontally (manual process)
  - Very expensive to increase storage space (no sharding)
  - No security patching
- Legacy System
  - MongoDB 3.4
  - PHP 7.0 on VMs
  - Upgrade deadlock

Database Migration

OMNICONVERT

Database Migration

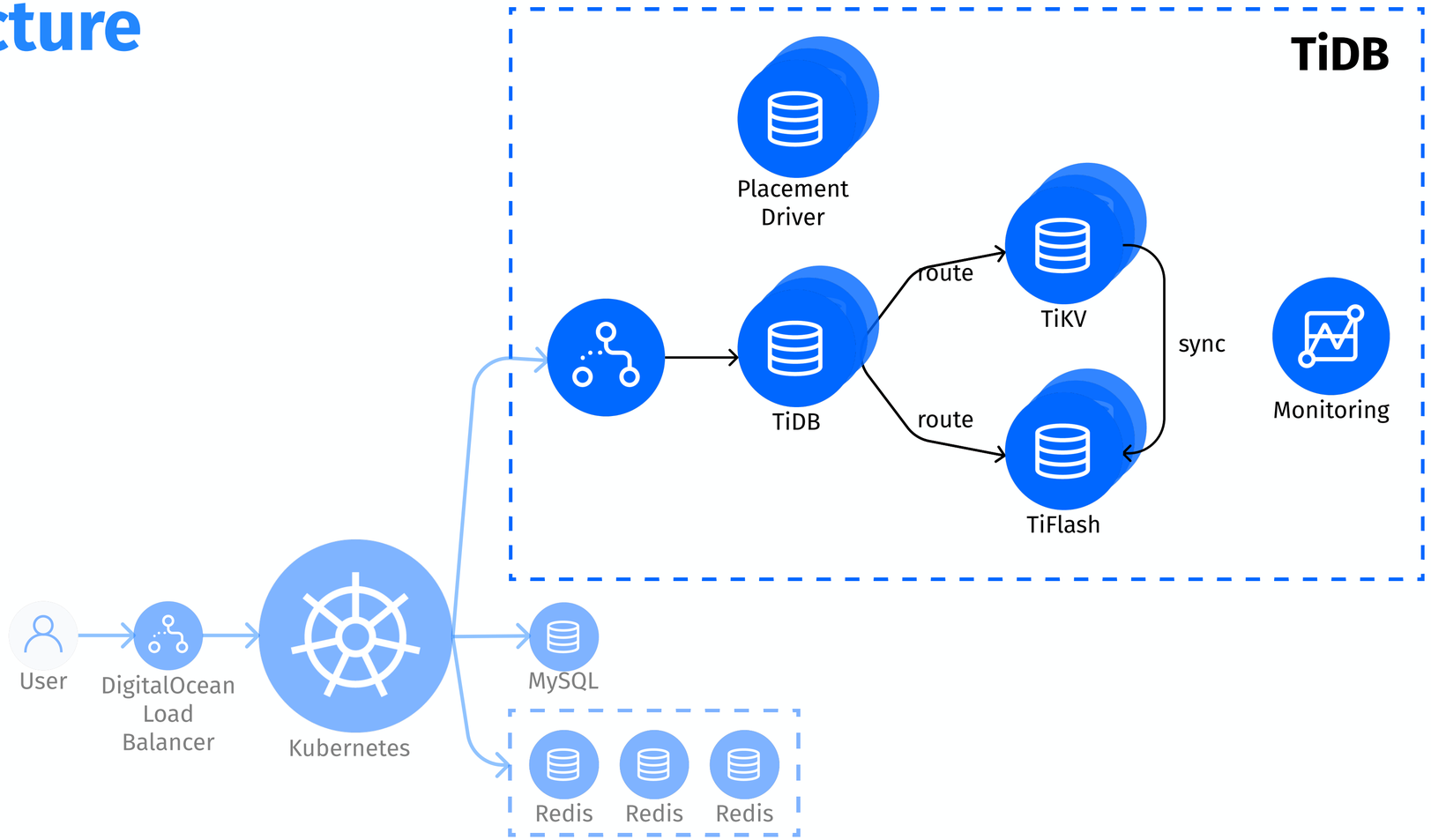
Criteria

- Good aggregation query performance
- Easy to set up and maintain by developers
- High availability
- Horizontal scalability
- Budget < $1500 / month
- Potential candidates
  - ClickHouse
  - Timescale (PostgreSQL)
  - BigQuery / Redshift
  - **TiDB**
  - Snowflake
  - Apache Cassandra
  - Apache HBase

OMNICONVERT

Database Migration

TiDB Architecture



OMNICONVERT

Database Migration

Proof of Concept (PoC)

- Query performance - make or brake
- Stress Test / Load Test

OMNICONVERT

Query Performance

MongoDB 3.4
TiDB 7.5

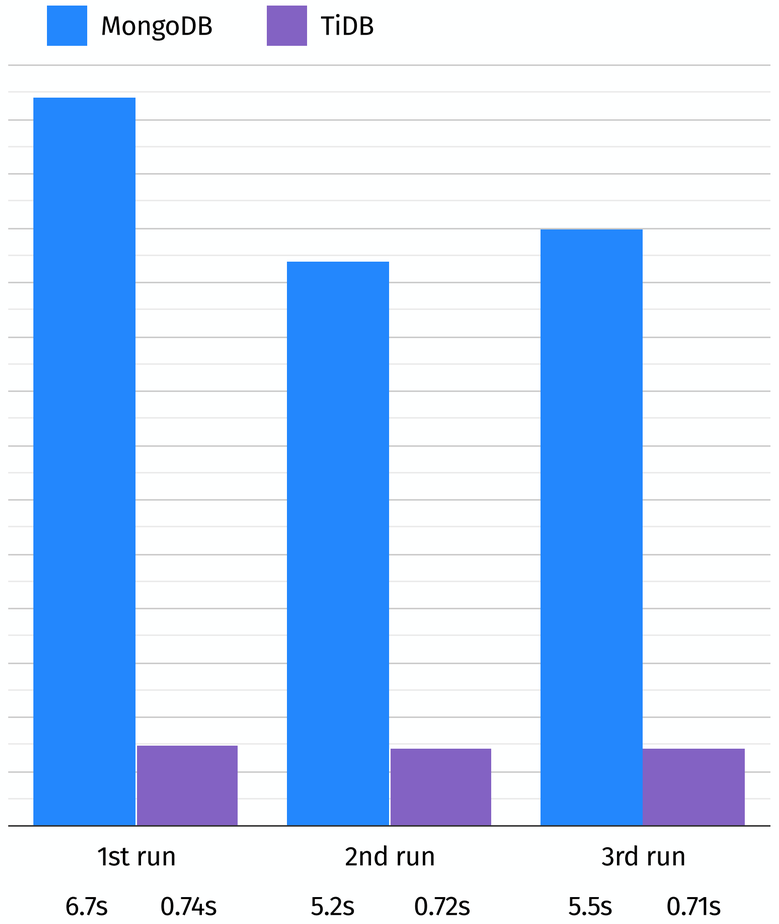
OMNICONVERT

Query Performance

Representative Query Results

```
SELECT COUNT(DISTINCT user) AS total
FROM visit_experiment
WHERE id_experiment = 59910
  AND device_type   = "mobile"
  AND date_created BETWEEN "2023-06-28"
                       AND "2023-12-28"
```

- 1.0+B records in table
- 2.2M total rows for experiment
- 800k matching rows
- 276k distinct rows
- 7.3-9x faster on TiDB



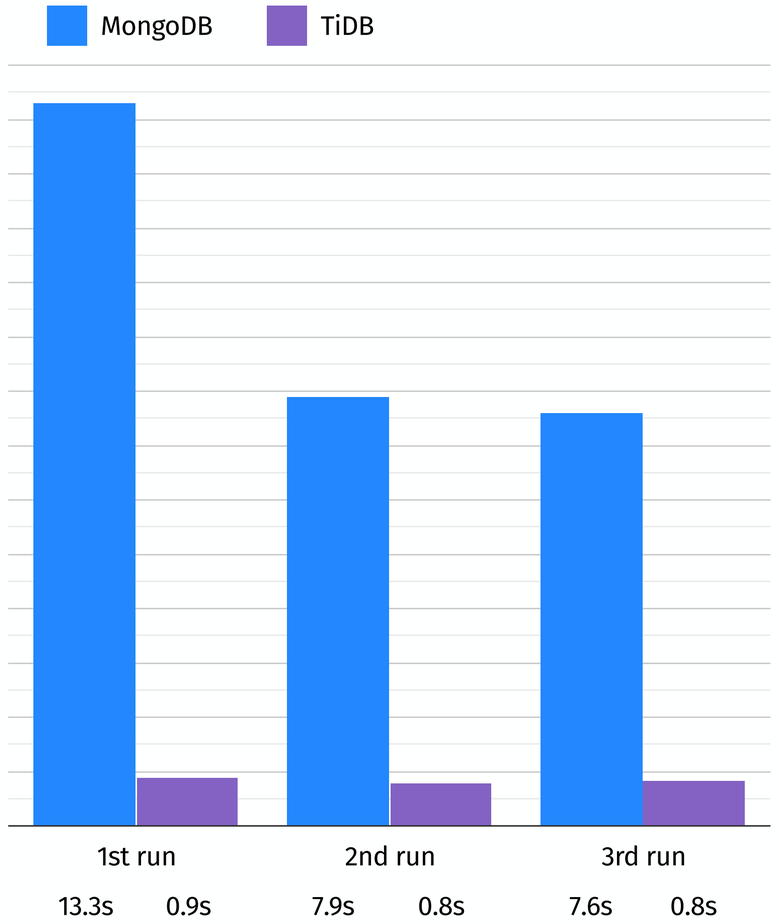
OMNICONVERT

Query Performance

Representative Query Results, by day

```
SELECT COUNT(DISTINCT user) AS total,
       YEAR(date_created)   AS Y,
       MONTH(date_created)  AS M,
       DAY(date_created)    AS D
FROM visit_experiment
WHERE id_experiment = 59910
  AND device_type   = 'mobile'
  AND date_created BETWEEN "2023-06-28"
                       AND "2023-12-28"
GROUP BY YEAR(date_created),
         MONTH(date_created),
         DAY(date_created)
```

9.5-14.7x faster on TiDB

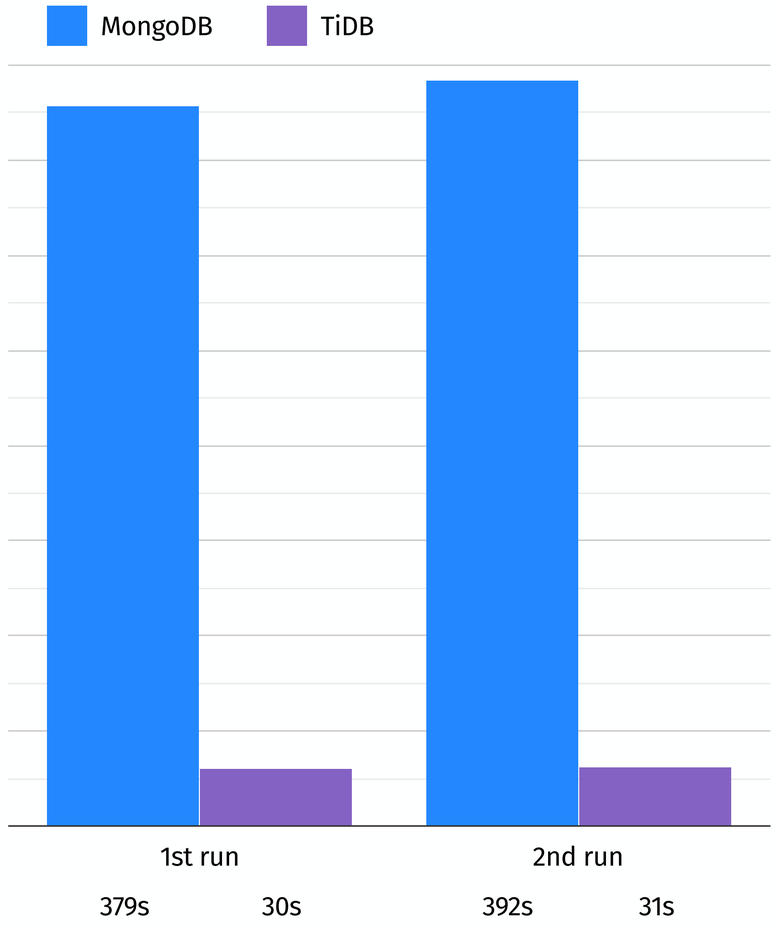


OMNICONVERT

Query Performance

Full Experiment Cache

- Representative experiment, no cache
  - 1.22M views
  - 2 variations
  - 143k conversions
  - 43 goals
- 12.6x faster on TiDB

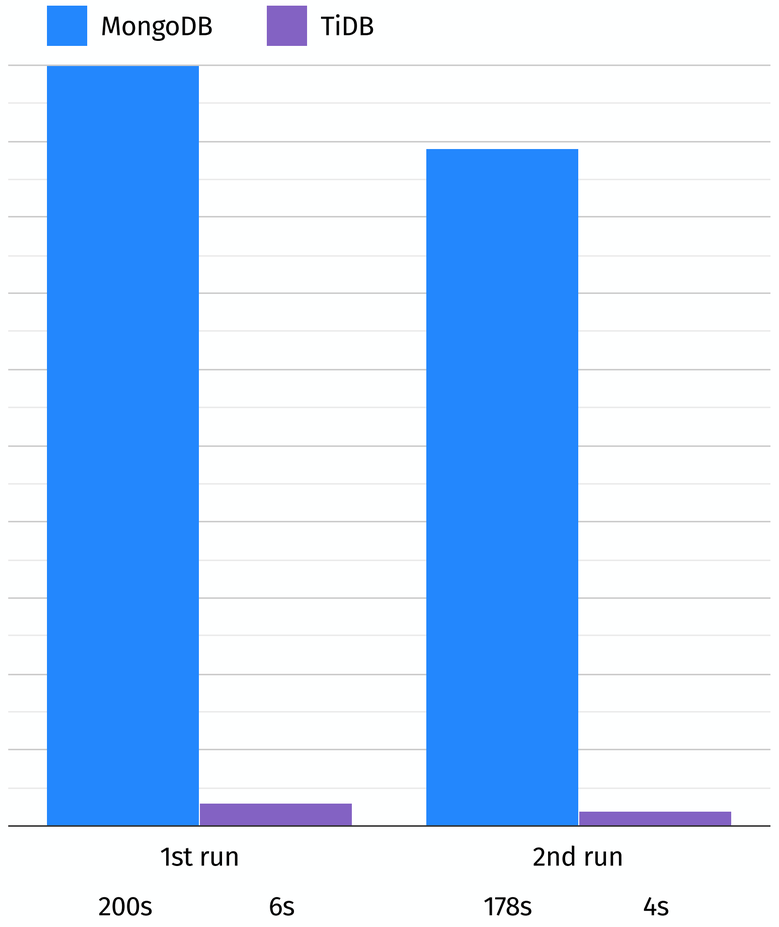


OMNICONVERT

Query Performance

Custom Period Report

- Representative experiment, no cache
  - 1.22M views
  - 2 variations
  - 143k conversions
  - 43 goals
- 33-45x faster on TiDB



OMNICONVERT

Stress Test

OMNICONVERT

Stress Test

Container Results

- Ran stress test with Grafana K6
- First, determined the limit for a container
  - 1 VM with 16 vCPU / 32GB
  - 3k Virtual Users
  - 23.5k RPM
  - 40ms response time
    <20ms in production
  - Utilized 50-60% of the CPU
  - Best: 3 PHP-FPM workers / 2.5vCPU

```
group= iter=3 request_id=94ef17c1-1fca-469c-6b4f-959501c6f1a4 scenario=default source=http-debug vu=2

   data_received..................: 16 GB  19 MB/s
   data_sent......................: 219 MB 271 kB/s
   http_req_blocked...............: avg=17.92µs  min=713ns   med=2.59µs  max=22.87ms  p(90)=3.59µs   p(95)=4.04µs
   http_req_connecting............: avg=3.09µs   min=0s      med=0s      max=21.23ms  p(90)=0s       p(95)=0s
   http_req_duration..............: avg=129.72ms min=12.34ms med=78.64ms max=6.68s    p(90)=272.2ms  p(95)=385.01ms
     { expected_response:true }...: avg=129.72ms min=12.34ms med=78.64ms max=6.68s    p(90)=272.2ms  p(95)=385.01ms
   http_req_failed................: 0.00%  ✓ 0          × 440863
   http_req_receiving.............: avg=13.41ms  min=37.65µs med=80.85µs max=6.63s    p(90)=918.31µs p(95)=1.08ms
   http_req_sending...............: avg=10.68µs  min=3.67µs  med=9.41µs  max=917.95µs p(90)=15.79µs  p(95)=17.99µs
   http_req_tls_handshaking.......: avg=11.86µs  min=0s      med=0s      max=17.75ms  p(90)=0s       p(95)=0s
   http_req_waiting...............: avg=116.3ms  min=12.25ms med=77.69ms max=1.18s    p(90)=259.94ms p(95)=353.6ms
   http_reqs......................: 440863 544.272235/s
   iteration_duration.............: avg=3m27s    min=2m13s   med=3m27s   max=4m55s    p(90)=3m55s    p(95)=4m4s
   iterations.....................: 8970   11.074012/s
   vus............................: 4      min=4        max=3000
   vus_max........................: 3000   min=3000     max=3000


running (13m30.0s), 0000/3000 VUs, 8970 complete and 2558 interrupted iterations
default ✓ [======================================] 0000/3000 VUs  13m0s
root@tidb-poc:~/stress-test/tracking-tests#
```

OMNICONVERT

Stress Test

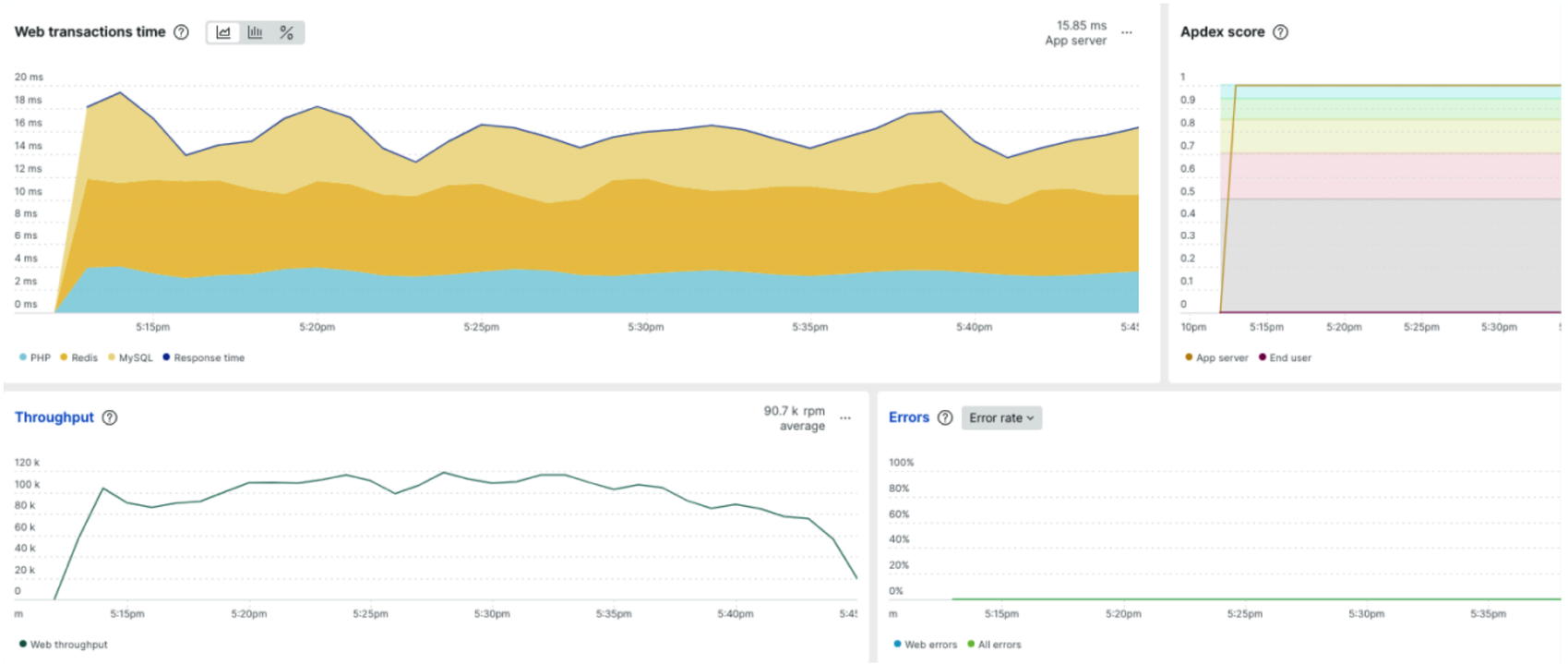
System Results

- Target: 100k RPM for the whole system
- Test launched from 10 VMs
- Tracking App on Kubernetes cluster
  - 10x VMs with 4 vCPU / 8GB
  - Better efficiency when distributed
    4.25x requests with 2.5x vCPUs
  - Utilized 80+% of the CPU on VMs
  - Utilized 30-40% of the CPU on TiFlash

Stress Test

System Results

- Longer period, 30+ min
  - Max 119k RPM
  - Avg 100k RPM
  - Avg response 15ms
    same as production
- Sustainable 100k RPM
  4-5x current traffic



OMNICONVERT

Transition Plan

OMNICONVERT

Transition Plan

Development Plan

- Rewrite ALL Reporting queries
- Rewrite ALL Tracking INSERT and SELECT queries
- Create a mechanism to INSERT in both databases
- Create a flippable switch to SELECT from a given database
  - Multi-layer cache in Redis
    cache missed = SELECT + write in cache

OMNICONVERT

Transition Plan

Release Plan

- **Ph 0** Migrate initial data from MongoDB into TiDB
  Initially migrated the data up until “today”
- **Ph 1** Write into both MongoDB and TiDB, (still) read from MongoDB
  Start writing into TiDB with preset timestamp
- **Ph 2** Migrate rest of data from MongoDB into TiDB
  Migrate gap data (since Ph0 up until Ph1 timestamp)
- **Ph 3** Monitor for differences
  Created scripts to check ALL data (various angles) and use internally
  A lot of internal manual testing
- **Ph 4** Read from TiDB, still write into both MongoDB and TiDB
- **Ph 5** Drop MongoDB and cleanup code

Lessons Learned

OMNICONVERT

Database Migration

Lessons Learned

- Flags to optimize JOINs and DISTINCTs
  - `tidb_opt_agg_push_down`
  - `tidb_opt_distinct_agg_push_down`
- JOINs will not go on forever
  Needed to rewrite a query so it filters results in WHERE instead of JOINing with condition
- Distributed SQL - concurrency is a potential issue
  SELECT + INSERT or INSERT + UPDATE statements can work inconsistently
- Order of inserted IDs is not monotonically increasing
  Querying for last inserted records should not be done with ORDER BY queries
- Over-excited and dropped daily cache completely
  Was re-instated

OMNICONVERT

Database Migration

Lessons Learned

- Optimizer issues: TiKV selected at all times
  - Tried to ANALYZE, no luck
  - No dials/knobs to adjust the Cost Model
- TiKV query time scales with dataset size; TiFlash not so much

rows	TiFlash	TiKV
0k	3.5s	53ms
8k	6.5s	4.7s
24k	9.4s	4.7s
33k	7.6s	10.5s
40k	7.0s	19.9s
88k	9.7s	21.6s
235k	9.5s	21.2s
1.8M	5.1s	> 6m

Personal Conclusions

OMNICONVERT

Personal Conclusions

Facts & Feelings

- Very **bold** project
- Less than 6 months implementation and delivery
- No major issues
- TOTALLY WORTH IT
  - No regrets

OMNICONVERT

Migrating 3B rows to TiDB
for a high-traffic application

Let’s connect!

The tale of an ambitious migration

FOSDEM

Sorin Dumitrescu
CTO

OMNICONVERT