From MySQL to Scylla and Back

Sergei Kuzmichev, Investing.com
July 1, 2024

Budapest, Hungary 2024
PERCONA UNIVERSITY
Open Source Databases Meetup

In partnership with FerretDB

Who am I?

- Database and systems engineer
- Chief Architect at Investing.com
- Co-author, “Learning MySQL, 2nd edition”
- Previously, lead support engineer at Percona, DevOps engineer, Oracle dba…

Who we are?

- Leading global financial news & data platform
- Alexa top 200 site, RIP Alexa
- Millions of page views per day
- Terabytes of financial data
- Hundreds of thousands of QPS

What is this talk about?

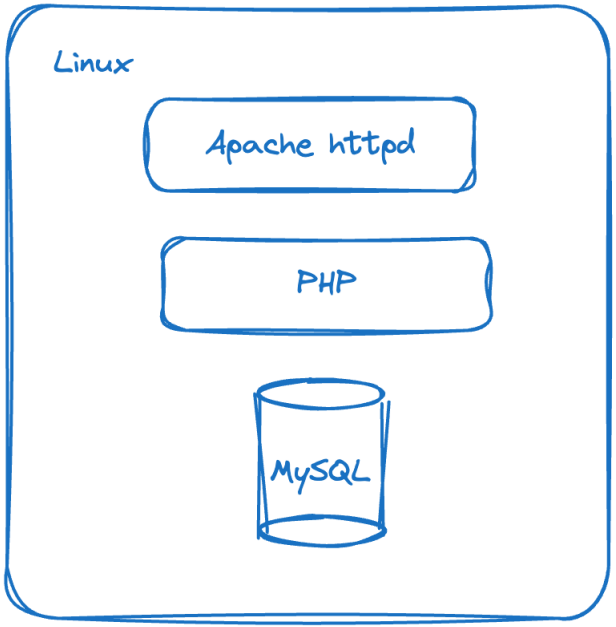
- An ongoing project in one of our subsystems
- Silver bullets
- Choosing storage engines
- And a bit about modern MySQL

What is this talk not about?

- Face-on comparison of MySQL to Scylla
- Bashing databases for no reason
- Moving all the data to a new system that worked better in one case

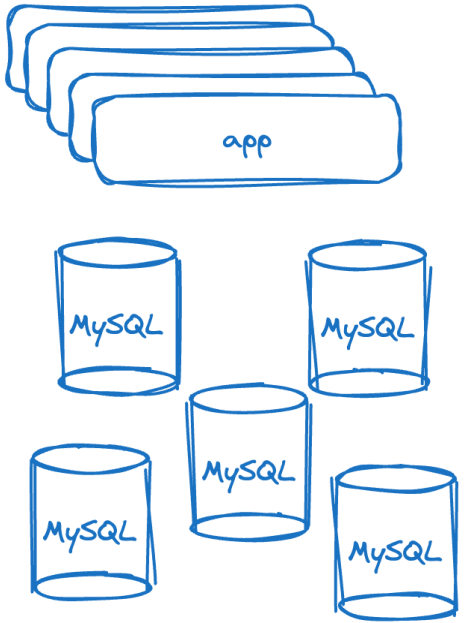
The story

- Initially, MySQL
- And PHP
- Simple times



The story

- Then, lots of PHP
- Then, lots of MySQL
  - With a single primary



Is MySQL Perfect?

- MySQL is not perfect
- Can’t scale writes horizontally without “manual” sharding
- Not simple to make highly available (think 5-10 years ago)
- Might be tricky to optimize without experience

New storage system

- Cassandra (and Scylla) offer many things out of the box
- Scalability including writes scaling
- Reliability
- Simple(r) operations (until it’s not, as usual)
- Good fit? Yes, for some data and access patterns

The story continues

- Since Scylla is good, let’s use it
- Mixing the databases
- Moving the datasets

Unforeseen consequences

- MySQL is old(er)
- MySQL might not be working amazingly well (in its current setup)
- Scylla is new(er)
- Scylla works great for the current data

Unforeseen consequences

- Move all the things to Scylla that are slow in MySQL!
- Some are duplicated
- MySQL is still growing…

Hidden problems

- Scylla is fully adopted
- Works great
- Robust
- Reliable
- Performant
- But… complicated
- And doesn’t fit every need
- Especially when moving from an “old-SQL”

Building charts

- Charts are a basic tool for understanding financial data
- Ubiquitous and very important
- Ideally, updated in real time

Investing.com

Building charts: data model

- Basic data point is a candle
- Each candle “lives” a set time interval
- Essentially a log with a hot head
- We constantly need to search for the last one
- Less frequently for the last with some parameters
- Small data

Building charts: data model

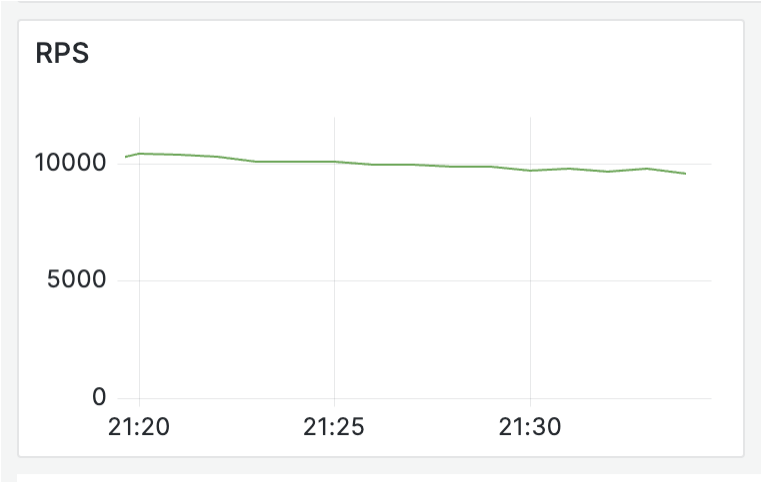
```
CREATE TABLE candles_keyspace.candles_60 (
        instrument int,
        start_timestamp int,
        parameters…
        PRIMARY KEY (instrument, start_timestamp)
) WITH CLUSTERING ORDER BY (start_timestamp DESC)
AND other keyspace options…
```

Getting to the problem

- Last year, we started moving to a cloud feed provider
- Developed a completely new feed infrastructure
- Aimed to overcome inherent limitations of the old system
- Re-used the database to retain compatibility and simplify transition

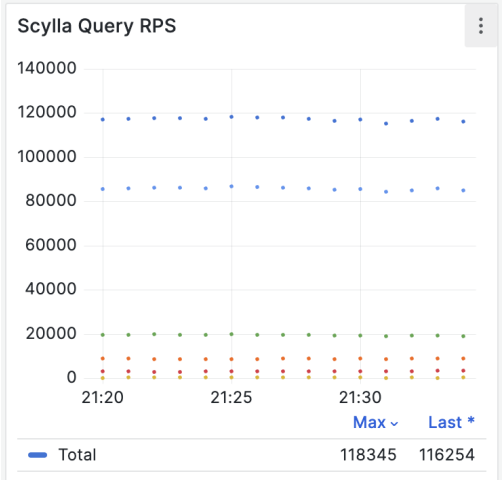
Reaching performance limits

- With the new system comes new possibilities for performance
- And new requirements for performance as well
- Suddenly, DB became a bottleneck
- Overall system RPS observed



Investigating the bottleneck

- Each system RPS translates into multiple DB QPS
- With queries latency observed, theoretical maximum is reached
- Point selects are great
- Our more complicated queries are not

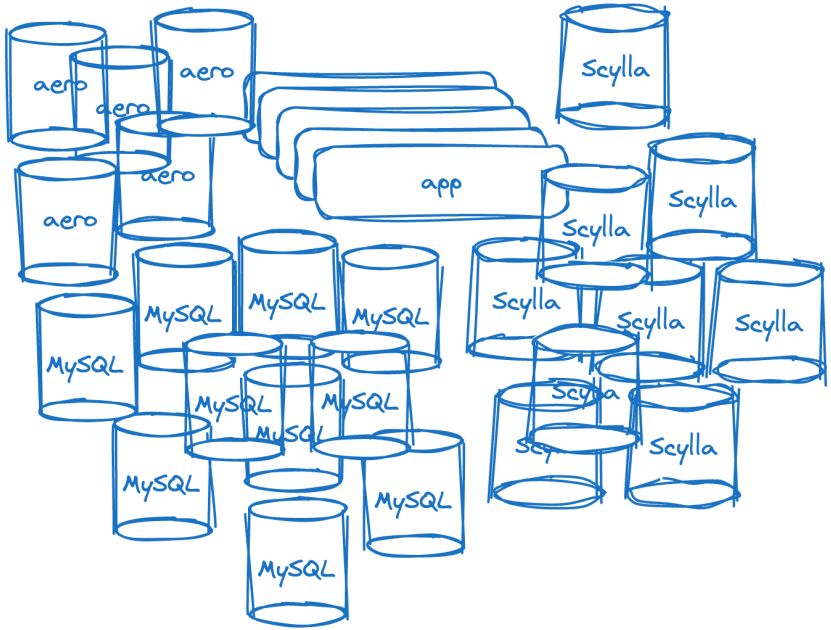


Assigning blame

- Is Scylla at fault here?
  - Not really
- Don’t fit what doesn’t fit
- Be cautious when moving from regular RDBMS
- At least adjust the data model and operations
- Poor planning can have consequences much, much later

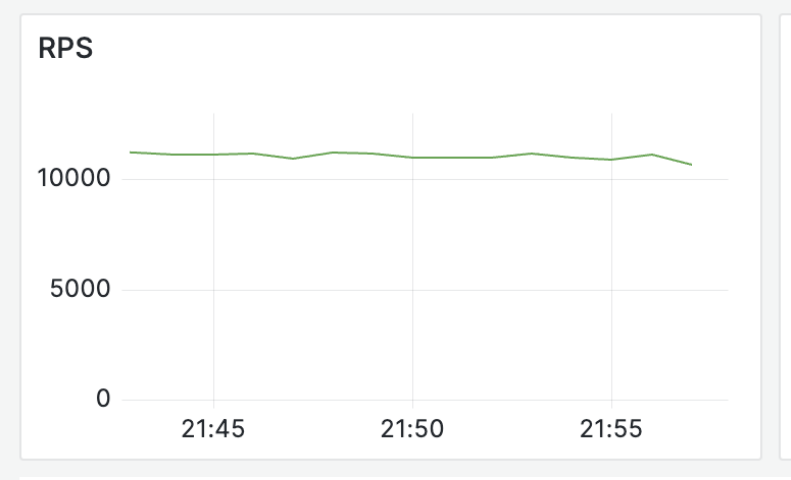
Fixing the performance: evolution

- Since data's in Scylla
- And we have a bottleneck
- Let's materialize some of it
- Adding more complexity



Fixing the performance: evolution

- Could be worse
- Much more complex system
- And even less consistent
- Gets difficult to reason



Fixing the performance: revolution

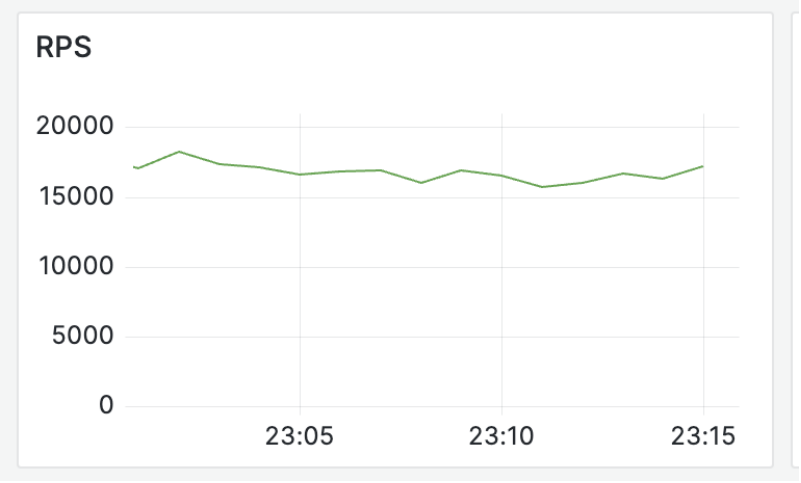
- We’re building a new system
- We can spend time on modeling the data
- Exploring the operations
- Picking the storage engine
- Choosing the features we need
- Can we simplify and improve?

Fixing the performance: considering options

- Current setup
- Time series databases
- Clickhouse
- PostgreSQL
- MySQL
  - InnoDB
  - MyRocks

Fixing the performance: devolution

- Single MySQL server using InnoDB
- 25% bigger machine, but 1 instead of 3
- Not crazily optimized
- Still possible to scale vertically
- Fully consistent
- Bias alert!



Fixing the performance: leaner schema

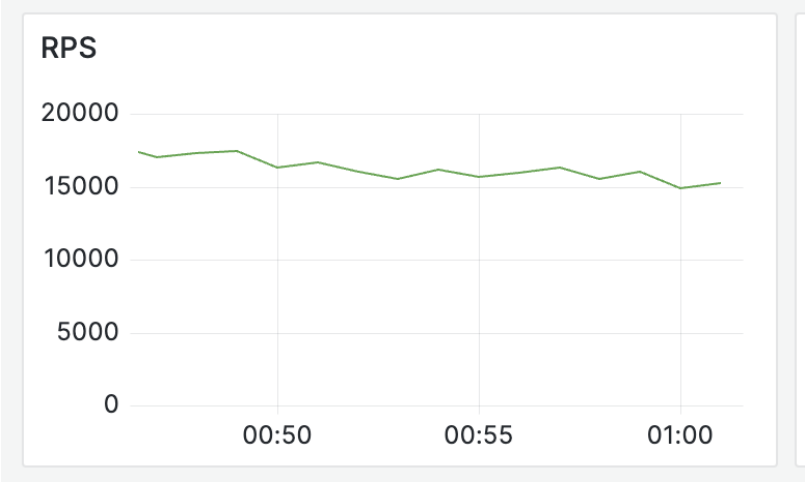
```
CREATE TABLE `candles` (
  `pair_id` int NOT NULL,
  `start_timestamp_utc` datetime NOT NULL,
  `interval_id` varchar(20) NOT NULL,
…
  PRIMARY KEY (`pair_id`,`start_timestamp_utc`,`interval_id`)
) ENGINE=InnoDB
/*!50500 PARTITION BY LIST  COLUMNS(interval_id)
(PARTITION p1minute VALUES IN ('1') ENGINE = InnoDB,
…
```

MySQL: the missing pieces

- It’s nice to have the room to grow
- It’s amazing to have an extremely simple setup
- But single-node MySQL is not Scylla
- No HA, no horizontal scaling even for reads
- Can we make it a closer match?

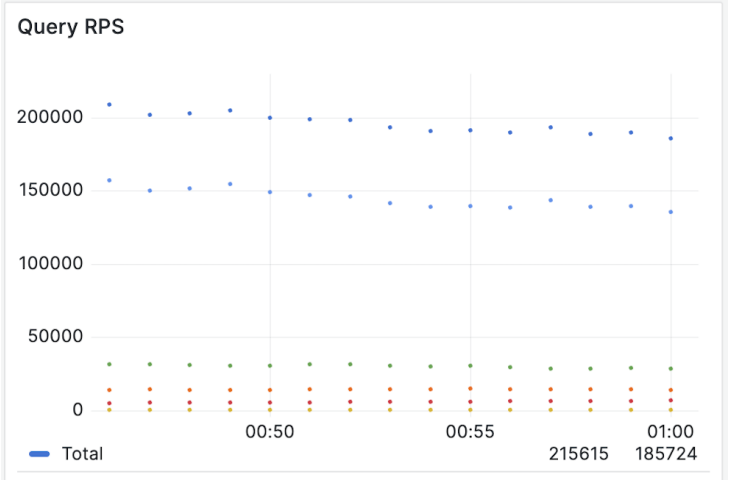
MySQL: the outcome

- InnoDB Cluster
  - Highly Available
  - Simpler maintenance (compared to regular replica)
  - Performant enough
  - Simple* to reason on consistency
- MySQL Router
  - Simple with no r/w split
  - All ops go to single primary
  - Topology-aware
  - Simpler than smart driver of Scylla



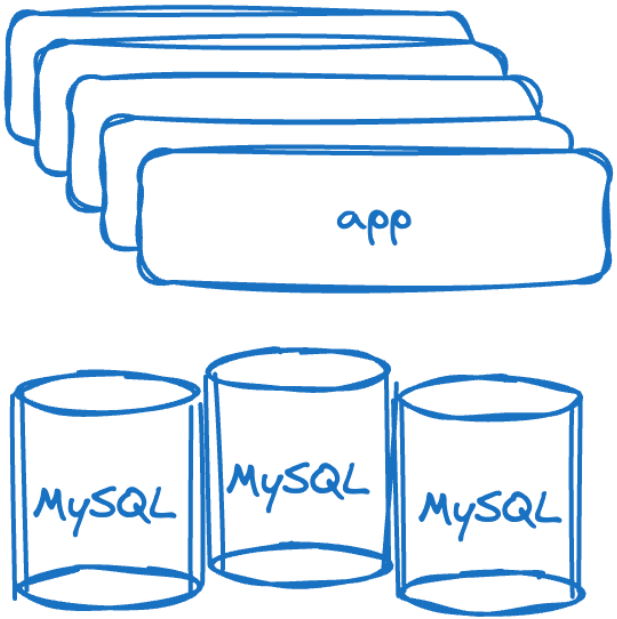
MySQL: the outcome

- We can still trade off some simplicity for performance
- Spread reads to other nodes
  - Forfeits consistency
- Move out the partitions
  - But probably switch from MySQL



MySQL: the outcome

- Isolated storage for chart candles
- Tailor-made, not chosen by default
- Necessary features
- Suitable performance



State of MySQL

- MySQL in 2024 is easier to “get right” than ever before
- InnoDB Cluster provides great quality of life with MySQL Shell
- Default topology and setup seems safe and good enough
- Router is simple (ProxySQL an option for more control)
- Can be extended with ClusterSet
- Good option when massive scale and volume is not expected
- Still not an autopilot…

Summary

- When your old storage is “bad,” and the new storage is “good,” do not (just) move everything to the new storage
  - Approach the move cautiously
- When you have a “new” or “good” storage, keep considering options
- Simpler is sometimes better
  - But only for a particular use-case
- MySQL is still a viable option

In partnership with FerretDB