From Elasticsearch to ClickHouse: 4 years later

CONTENTSQUARE

Ryad Zenine

Senior Data Engineer

CONTENTSQUARE

Contentsquare is the global market leader in experience analytics

We build tools to analyse and improve user experience on digital plate-forms
(websites, mobile apps, etc...)

Some Customers

Adobe Rakuten Walmart Microsoft verizon

Founded in 2012

Head count > 1000

A typical day at the office

1 Billion

pageviews

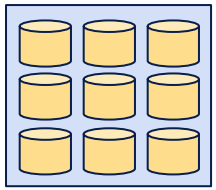
100k

analytics queries

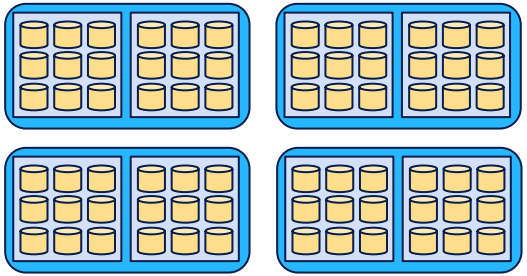
~200 Ms

Average response time

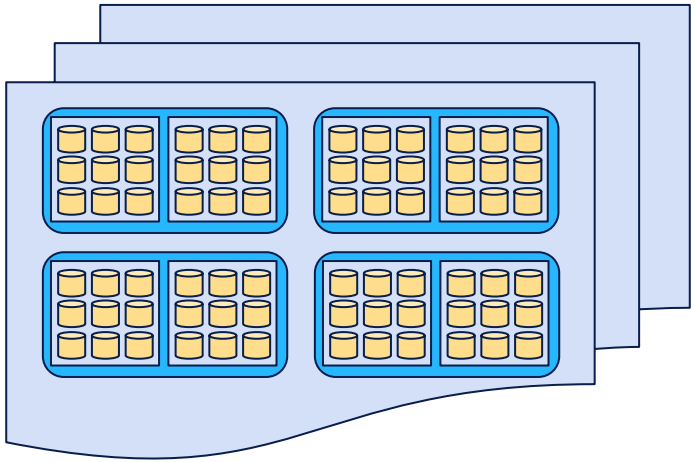
100+ machines running ClickHouse



9 **shards**
per **cluster**

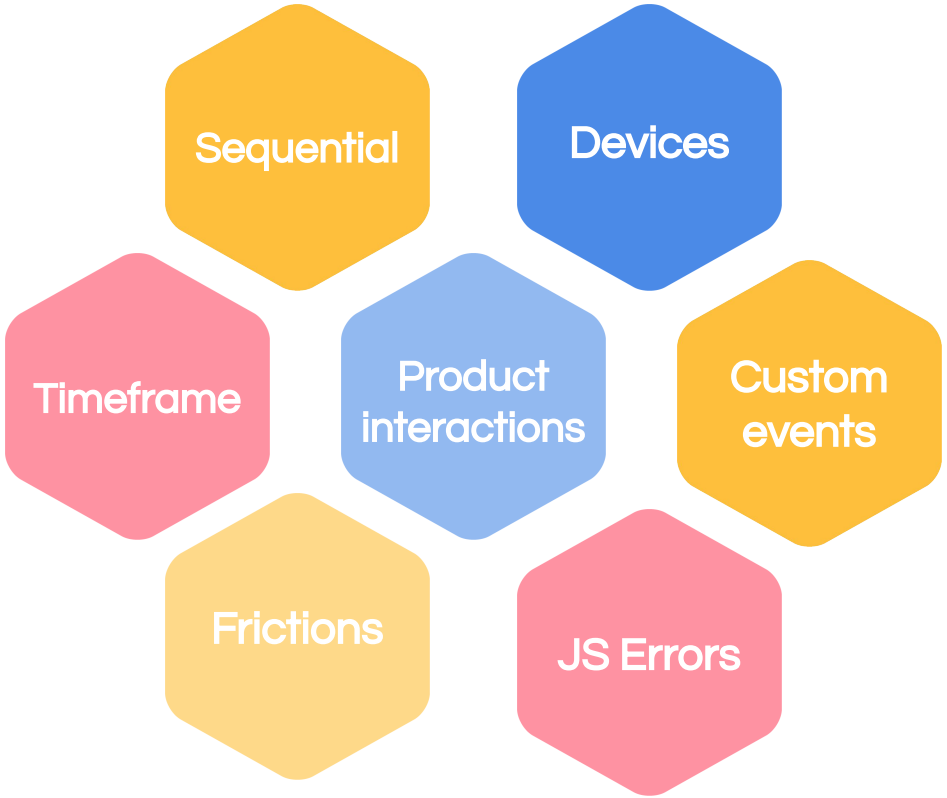


2 **clusters**
per **region**



x2 **replication factor**

Our main challenge



All conditions **compose together** on up to **13 months** of data.

Everything needs to be **computed on the fly**.


Our journey from
Elasticsearch
to
ClickHouse

Our Elasticsearch setup

- **Elastic Search sizing**
  - *14 clusters*
  - *27 m5.4xl per cluster*
  - *3 master node per cluster*

- **Main struggles: Horizontal Scalability & Cost effectiveness**
  - Maintaining multi-tenant clusters was difficult
  - Handling very large accounts was impossible without aggressive sampling

We looked at many alternatives

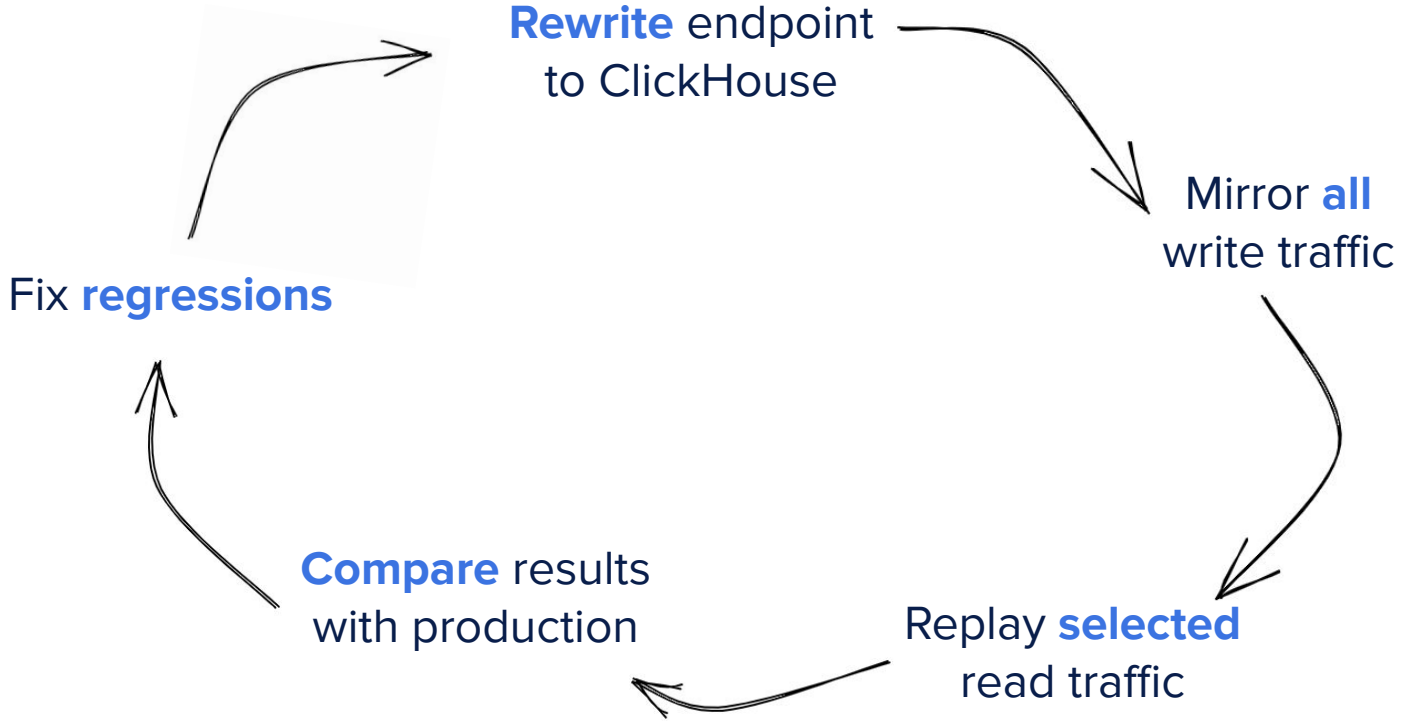
Migration timeline

Phase 1: We tried clickhouse on a new commercial product

+ We learned how to properly use clickhouse
+ We built the automation (terraform, ansible etc...)
+ We built Monitoring & Alerting
+ Got some production experience, although at smaller scale

- Less feedback about performance at scale

Phase 2: Mirror production to new endpoints



Phase 2: Lesson learned

- **Never** break the API contracts
- **Automate** the data ingestion, replay and comparison
- **Split** the team: 50% on legacy, 50% on new endpoints
- Don’t be afraid to **duplicate work**: new features will have to be built twice, it’s ok.

Phase 3: Progressive rollout to all customers

We migrated customers
one by one over 4~6 months.

We had exactly
zero regressions.

ClickHouse technical advantages

11x cheaper

with 6x more data

10x faster

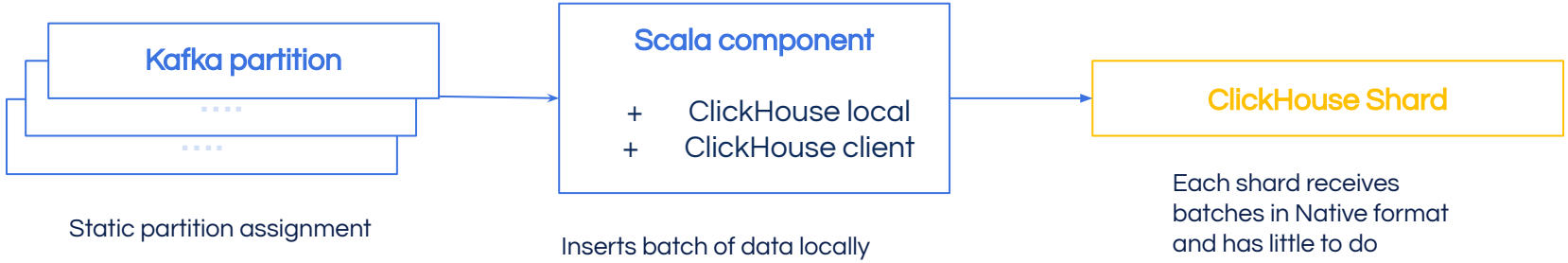
on P99

ClickHouse unlocked new product features

- Allowed us to query on **3 months** instead of only 1
- Population analysis based on **sequential behaviors**
- **Form interaction** analysis
- And many more ...


Adapting
ClickHouse to
our use cases

We optimized our insertion pipeline to have as little overhead as possible



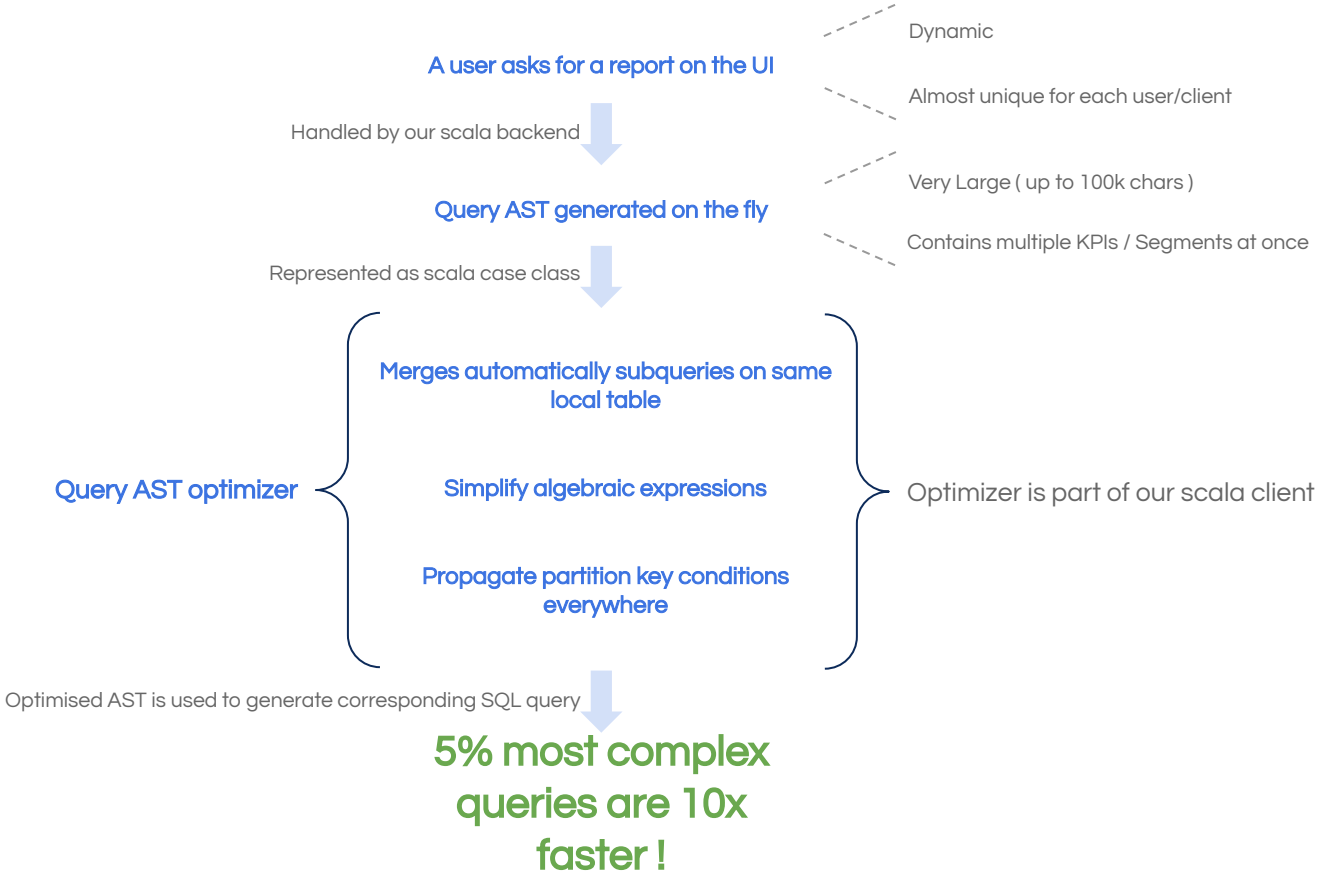
Pros: Smallest insertion overhead possible

- Production cluster receives data in native format
- Batch is already optimized when received by the shard
- Static assignment of kafka partitions per shard

Cons: Each schema update requires more operations

- Tight coupling between scala component and production schema
- Schema change on production becomes a multi-step process.

We generate all our queries on the fly depending on the requested segment



We had to invest a lot in building tooling around ClickHouse

Backup tools

Automatic user management integrated with vault

Automatic schema management

GDPR/CCPA compliance tooling

Scala client: build , optimize and run clickhouse queries in Scala (with the optimizer)

Chproxy improvements

CONTENTSQUARE

Engineering Blog

engineering.contentsquare.com