Large scale data processing pipelines at trivago: a use case

2016-11-15, Sevilla, Spain
Clemens Valiente
Clemens Valiente
Senior Data Engineer
trivago Düsseldorf

Originally a mathematician
Studied at Uni Erlangen
At trivago for 5 years

Email: clemens.valiente@trivago.com
de.linkedin.com/in/clemensvaliente
Data driven PR and External Communication

Price information collected from the various booking websites and shown to our visitors also gives us a thorough overview over trends and development of hotel prices. This knowledge then is used by our Content Marketing & Communication Department (CMC) to write stories and articles.
Data driven PR and External Communication

Price information collected from the various booking websites and shown to our visitors also gives us a thorough overview over trends and development of hotel prices. This knowledge then is used by our Content Marketing & Communication Department (CMC) to write stories and articles.
Data driven PR and External Communication

Price information collected from the various booking websites and shown to our visitors also gives us a thorough overview over trends and development of hotel prices. This knowledge then is used by our Content Marketing & Communication Department (CMC) to write stories and articles.
The past: Data pipeline 2010 – 2015
The past: Data pipeline 2010 – 2015

Java Software Engineering

trivago

Java

Expedia.de
Booking.com
Hotels.com
The past: Data pipeline 2010 – 2015

Java Software Engineering  →  Business Intelligence

trivago

Java  →  mySQL

Expedia.de
Booking.com
Hotels.com
The past: Data pipeline 2010 – 2015

Java Software Engineering

Business Intelligence

trivago

Java

mySQL

Expedia.de
Booking.com
Hotels.com

CMC
The past: Data pipeline 2010 – 2015

Facts & Figures

Price dimensions
- Around one million hotels
- 250 booking websites
- Travellers search for up to 180 days in advance
- Data collected over five years
The past: Data pipeline 2010 – 2015
Facts & Figures

Price dimensions
- Around one million hotels
- 250 booking websites
- Travellers search for up to 180 days in advance
- Data collected over five years

Restrictions
- Only single night stays
- Only prices from European visitors
- Prices cached up to 30 minutes
- One price per hotel, website and arrival date per day
- “Insert ignore”: The first price per key wins
The past: Data pipeline 2010 – 2015

Facts & Figures

Price dimensions
- Around one million hotels
- 250 booking websites
- Travellers search for up to 180 days in advance
- Data collected over five years

Restrictions
- Only single night stays
- Only prices from European visitors
- Prices cached up to 30 minutes
- One price per hotel, website and arrival date per day
- “Insert ignore”: The first price per key wins

Size of data
- We collected a total of 56 billion prices in those five years
- Towards the end of this pipeline in early 2015 on average around 100 million prices per day were written to BI
The past: Data pipeline 2010 – 2015

Java Software Engineering → Business Intelligence

trivago → Expedia.de → mySQL → CMC

Booking.com → Hotels.com
The past: Data pipeline 2010 – 2015

Java Software Engineering

Business Intelligence

MySQL

Expedia.de
Booking.com
Hotels.com

CMC
The past: Data pipeline 2010 – 2015

Java Software Engineering

Business Intelligence

trivago

Java

MySQL

Expedia.de

Booking.com

Hotels.com

CMC
The past: Data pipeline 2010 – 2015

Java Software Engineering

Business Intelligence

trivago

Expedia.de
Booking.com
Hotels.com

CMC
The past: Data pipeline 2010 – 2015

Java Software Engineering

MySQL

Expedia.de™
Booking.com
Hotels.com

Business Intelligence

CMC
Refactoring the pipeline: Requirements

- Scales with an arbitrary amount of data (future proof)
- reliable and resilient
- low performance impact on Java backend
- long term storage of raw input data
- fast processing of filtered and aggregated data
- Open source
- we want to log everything:
  - more prices
    - Length of stay, room type, breakfast info, room category, domain
  - with more information
    - Net & gross price, city tax, resort fee, affiliate fee, VAT
Present data pipeline 2016 – ingestion
Present data pipeline 2016 – ingestion

trivago

Düsseldorf

Java

Kafka

venere.com

Expedia.de

Booking.com

Hotels.com
Present data pipeline 2016 – ingestion

trivago San Francisco
Java \(\rightarrow\) Kafka

trivago Düsseldorf
Java \(\rightarrow\) Kafka

trivago Hong Kong
Java \(\rightarrow\) Kafka

Venere.com
Expedia.de
Booking.com
Hotels.com
Present data pipeline 2016 – processing

Kafka → Camus → HDFS
Present data pipeline 2016 – processing

Kafka → Camus → HDFS → Hive → Hadoop
Present data pipeline 2016 – processing

Kafka → Camus → Impala

Oozie

Hive

HDFS

Hadoop
Present data pipeline 2016 – processing

Kafka → Camus → Hive → HDFS → Impala → R shiny

CMC
Cluster specifications
- 51 machines
- 1.7 PB disc space, 60% used
- 3.6 TB memory in Yarn
- 1440 VCores (24-32 Cores per machine)
Present data pipeline 2016 – facts & figures

Cluster specifications
- 51 machines
- 1.7 PB disc space, 60% used
- 3.6 TB memory in Yarn
- 1440 VCores (24-32 Cores per machine)

Data Size (price log)
- 2.6 trillion messages collected so far
- 7 billion messages/day
- 160 TB of data
### Cluster specifications
- 51 machines
- 1.7 PB disc space, 60% used
- 3.6 TB memory in Yarn
- 1440 VCores (24-32 Cores per machine)

### Data Size (price log)
- 2.6 trillion messages collected so far
- 7 billion messages/day
- 160 TB of data

### Data processing
- Camus: 30 mappers writing data in 10 minute intervals
- First aggregation/filtering stage in Hive runs in 30 minutes with 5 days of CPU time spent
- Impala Queries across >100 GB of result tables usually done within a few seconds
Present data pipeline 2016 – results after one and a half years in production

- Very reliable, barely any downtime or service interruptions of the system
- Java team is very happy – less load on their system
- BI team is very happy – more data, more resources to process it
- CMC team is very happy
  - Faster results
  - Better quality of results due to more data
  - More detailed results
  - => Shorter research phase, more and better stories
  - => Less requests & workload for BI
Uses for price information
- Monitoring price parity in hotel market
- Anomaly and fraud detection
- Price feed for online marketing
- Display of price development and delivering price alerts to website visitors
Present data pipeline 2016 – use cases & status quo

Uses for price information
- Monitoring price parity in hotel market
- Anomaly and fraud detection
- Price feed for online marketing
- Display of price development and delivering price alerts to website visitors

Other data sources and usage
- Clicklog information from our website and mobile app
- Used for marketing performance analysis, product tests, invoice generation etc
Present data pipeline 2016 – use cases & status quo

Uses for price information
- Monitoring price parity in hotel market
- Anomaly and fraud detection
- Price feed for online marketing
- Display of price development and delivering price alerts to website visitors

Other data sources and usage
- Clicklog information from our website and mobile app
- Used for marketing performance analysis, product tests, invoice generation etc

Status quo
- Our entire BI business logic runs on and through the kafka – hadoop pipeline
- Almost all departments rely on data, insights and metrics delivered by hadoop
- Most of the company could not do their job without hadoop data
Future data pipeline 2016/2017

Kafka → Camus → Impala → shiny

CMC

trivago
Future data pipeline 2016/2017

Message format: CSV
Protobuf / Avro

Kafka → Camus → HDFS → HIVE → Impala → R shiny

CMC
Future data pipeline 2016/2017

Message format: CSV, Protobuf / Avro

Kafka → Camus → Impala → shiny

Stream processing
Kafka Streams
Streaming SQL

HDFS

trivago
Future data pipeline 2016/2017

Message format: CSV, Protobuf / Avro

Kafka

Stream processing
Kafka Streams
Streaming SQL

Kafka Connect or Gobblin

Impala

R shiny

CMC

trivago
Future data pipeline 2016/2017

Message format:
- CSV
- Protobuf / Avro

Kafka → Kafka Connect or Gobblin → Impala → R shiny

Stream processing
Kafka Streams
Streaming SQL

CMC
Future data pipeline 2016/2017

Message format:
- CSV
- Protobuf / Avro

- **Kafka**
  - Stream processing
  - Kafka Streams
  - Streaming SQL

- **Kafka Connect** or **Gobblin**

- **Oozie**

- **Impala**
- **Kylin / Hbase**

- **Spark**

- **R**
- shiny

- CMC
Future data pipeline 2016/2017

Message format:
- CSV
- Protobuf / Avro

Kafka → druid → R shiny

Stream processing
Kafka Streams
Streaming SQL

CMC
Future data pipeline 2016/2017

Kafka Streams → local state → R shiny → CMC

Key challenges and learnings

Mastering hadoop
- Finding your log files
- Interpreting error messages correctly
- Understanding settings and how to use them to solve problem
- Store data in wide, denormalised Hive tables in parquet format and nested data types
Key challenges and learnings

Mastering hadoop
- Finding your log files
- Interpreting error messages correctly
- Understanding settings and how to use them to solve problem
- Store data in wide, denormalised Hive tables in parquet format and nested data types

Using hadoop
- Offer easy hadoop access to users (Impala / Hive JDBC with visualisation tools)
- Educate users on how to write good code, strict guidelines and code review
- deployment process: jenkins deploys git repository with oozie definitions and hive scripts to hdfs
Key challenges and learnings

Mastering hadoop
- Finding your log files
- Interpreting error messages correctly
- Understanding settings and how to use them to solve problem
- Store data in wide, denormalised Hive tables in parquet format and nested data types

Using hadoop
- Offer easy hadoop access to users (Impala / Hive JDBC with visualisation tools)
- Educate users on how to write good code, strict guidelines and code review
- deployment process: jenkins deploys git repository with oozie definitions and hive scripts to hdfs

Bad parts
- HUE (the standard GUI)
- Write oozie workflows and coordinators in xml, not through the Hue interface
- Monitoring impala
- Still some hard to find bugs in Hive & Impala
- Memory leaks with Impala & Hue: Failed queries are not always closed properly
Thank you!

Questions and comments?

Clemens Valiente
Senior Data Engineer
trivago Düsseldorf

Originally a mathematician
Studied at Uni Erlangen
At trivago for 5 years

Email: clemens.valiente@trivago.com
LinkedIn: de.linkedin.com/in/clemensvaliente
Resources

- Gobblin: https://github.com/linkedin/gobblin
- Impala connector for dplyr: https://github.com/piersharding/dplyrimpaladb
- Hive on Spark: https://cwiki.apache.org/confluence/display/Hive/Hive%3A+Getting+Started
- Parquet: https://parquet.apache.org/documentation/latest/
- ProtoBuf: https://developers.google.com/protocol-buffers/

Thanks to Jan Filipiak for his brainpower behind most projects, giving me the opportunity to present them.