Aggregating, reporting, and visualizing 10s of billions of records per day from a global footprint

AUDIENCESCIENCE

Who

Frank Conrad

- Chief Architect at AudienceScience
- frank@fc-tb.de
- https://www.linkedin.com/in/frankconrad/

- Software development since 80‘s
- Co-founder of 7d / wunderloop
- all about performance, scale, efficiency

- deep knowledge
  - scale services
  - scale processing hadoop, MR, storm, kafka, cassandra,voldemort,…
  - mesos, marathon
  - Unix system/kernel, production, monitoring

- looking for new challenges

Background

- AudienceScience provides fully integrated, end to end, advertising solutions for the world's largest brand advertisers.
- AudienceScience receives, processes and responds (in real-time) to over 80 billion incoming requests a day, in over 42 countries.
- Our solutions allow advertisers to effectively manage and leverage their consumer data to produce industry leading ROI on their advertising spend.
- Global Distribution of Five Points of Presence to Central DC
- Where we were
  - 20 Billion TPD in 2014

The Challenge

- collection
  - billing relevant
- transfer
- backup / disaster recovery
- processing
- reporting

- traffic increased by 10x in years

The edge (pod)

- Services very distributed

- scribe
- rsync
- kafka
- kafka fallback

experience with kafka

- scales and is performing
- runs rock solid
- think what happens if kafka fails

- hardware
  - cpu, memory
  - disk, raid
  - network
- monitoring
  - kafka-manager

Pod local processing

- full traffic
- Up to 500k msg/s with 2.5kb

disaster recovery

- reprocess / rebuild
- broken DC
- Edge
  - To S3 (secor)
  - asap, as raw make sense
- DC
  - consolidated data
  - periodic backups
- secor scale enhancements for S3

central DC

- first implementation
  - straight forward
- facing problems
  - not enough parallelism
  - didn’t leverage resources evenly

redesign goals

- scale, efficiency, performance
- maximize mapper side work
- optimize / minimize data read, shuffle
- splittable big files (immutable)
- flexible workflows, simplified restart,…
- simpler to maintain

solution we choose

- Basket
  - add / maintain data (in files)
  - data in avro files
  - simple to iterate over data / get file list
- Basket type campaign
  - per campaign
  - user oriented
- Basket type billing
  - per exchange. timeline
  - transaction oriented
- main processing done in java MRV2/yarn

Result

- pros
  - 6x performance (lower runtime) at start
  - leverage cluster more evenly
  - one run could produce up to 30 reports in parallel
  - same cluster could handle later > 10x of traffic
- cons
  - updating baskets needs rewrite of whole part file like parquet
  - need nested data format, not so nice with pig,…
  - non standard way / thinking ;)

What we did

- partitioning, always double, depend on size
- meta data in files
- mutable files, always work with file lists
- merge on reducer, by read old file directly

UI / visualisation

- product/customer want new reports/analytics every week
- engineering and UI can't keep up
- did analysis of the existing reports
- 80% of reports could run on pre aggregated data
  - huge amount of rows in a few tables every day
- POC
  - use redshift as data warehouse
  - use periscope data as query visualization

UI / visualization solution

- data warehouse
  - Snowflake
- visualization
  - Looker

file formats

- json
- protobuf
- avro

Thank you

- Adam Shepard (senior architect)
- Prashant Kumar (development manager) and team lead for all hadoop, snowflake implementation mention above

- As mentioned, Audiencescience has closed down
- So we are looking for new challenges

- Contact me
  frank@fc-tb.de