Movie Flix -

-----------

Use real time recommendations -

If a person start watching new videos, based on threshold filter in kafka stream - generate new recommendations based on similar videos database.

If recommendations does not change, keep showing earlier recommendations.

Keeps some data in global k-table of userIds with videos watched for past 15 minutes.  Generate recommendations corresponding to those videos.

If any video gets added, recommendation changes and those recommendations are displayed to users.

Show position topic -

highly distributed  ----> if volume > 30 partitions. How many brokers - 30 brokers ??? leader election will be less, instead of one heavy broker.. lightweight brokers are preferred...

Show positions data gets backed up in hadoop for backups and generating historical data sets for machine learning, for analytics purpose

Get Taxi -

User position service -  user\_ position ==>   Get user densities for example number of users in grid > threshold > surge price for that grid  - Kafka streams for surge pricing computations - Data goes to surge pricing topic based on region where it is pulled by application

Taxi position service -  taxi\_position  ==>  Get Taxi densities for example number of taxis in a grid  > threshold > surge price for that grid - Kafka streams for surge pricing computations - Data goes to surge pricing topic based on region where it is pulled by application

Other topics can be weather, events (possible events - depending on kind of cab)  - can be included in Kafka streams application to compute surge pricing.

Taxi position data ephemeral - Need not be kept for 7 days.

From Taxi position topic update driver location hash table

social media app -

===========

(Command Query Responsibility Seggregation)

Posting Service ==> posts kafka topic                            ==> Trending posts in past hour

(High retention)

                                                                                         ==> Total likes/Comments Computation

Like/Comment Service ==> comments kafka topic

Posts with counts goes in a separate topic  ==>

data is recorded on kafka in form of events (website events) ==>  events JSON

Posts/Likes/Counts gets branched via kafka streams and gets stored in three different topics

Refresh feeds Service which gives updated real time feeds built in on top of existing feeds architecture using LinkedHashMap and periodic generation.

Trending feeds which gives updated real time trending posts

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Common questions while giving a kafka topic design -

How many producers ( Can it be multiple).

How many partitions - does it need to be highly distributed (Keep partitions high, scale brokers on demand)

Retention period of data for the topic.

Big Data Ingestion -

Useful for offloading data from Kafka to Amazon s3, Elasticsearch. As data cannot be ingested with that much velocity directly in amazon s3 (For example upload latencies) or elasticsearch.

No trending posts in facebook (feed is sufficient)

Financial Application

Logs and Metrics Aggregation

Application pushing high volume Prometheus metrics routed to prometheus datastore/Elasticsearch/Splunk via kafka

Application log forwarders kafka connect into splunk