Ad Click Event Aggregation

xxx

March 2023

Main idea:

- still the database is the most important thing
- do aggregation and store the data
- need to scale more, so more scaling and map-reduce
- has some TCP latency and data missing issue
- this is an extension of Chapter 5

1 Step1

Understand the Problem and Establish Design Scope
functional support:

- aggregate ad_id in M minutes
- return top 100 most clicked ads in every minute
- support filter by country, region
- ad click 1 billion per day, so QPS = 10000

2 Step2

Propose High-level Design

- API1:
  aggregate count
- API2:
  reurn most clicked ad in minute

- data model
  write heavy, not read heavy
  use Casandra or InfluxDB

High level flow:

- log watcher
- data aggregation service (push mode)
  in push mode, we can use kafka as buffer
- database write two kinds of data:
  - raw data as cold data
  - aggregated data: map-reduce
- query service

3 Step3

Design Deep Dive
items to discuss:

- streaming and batching:
  why we need streaming and batching:
  streaming to process current data, batching to re-process old cold data
  two architecture design:
  - lambda design, two pipelines, one for batching one for streaming
  - kappa design, streaming and batching merged in one pipeline
- time and aggregation window
  some msg arrive late, getting wrong order and outside window
- delivery guarantee:
  use kafaka queue to guarantee delivery semantics
- scale the system:
  hotspot issue: in map-reduce, a ad_id is too popular
- data monitor and correctness
- fault tolerance