Metrics Monitoring and Alerting System

xxx

March 2023

The main idea of this chapter:

- how to choose a proper databse
- what is the connection between different components

1 Step1

Understand the Problem and Establish Design Scope

- data size: 100 million per day
- change resolution, down-sample old data
- channel communication: HTTP
- Special: it is about low level metrics, (e.g. CPU, memory), not high level metrics like QPS, coverage etc.

2 Step2

Propose High-Level Design

- data model:
  choose a good database to store everything
  use time series specialized DB, e.g Influx DB
- High-level design:
  - metrics source (the machines being monitored)
  - metrics collector (consider the channels, and data mode)
    - push mode:
      machines push to collectors, need a load balancer in between

    - pull mode:
      collectors pull from machines, need a config to tell collectors which are the machines to collect from
      Drawback: might miss short-lived jobs (it lives between two pull intervals)
  - time series DB (a mature product)
  - query service
  - visualization system
  - alert system

3 Step3

Design Deep Dive:

3.1 1

How to scale between metrics sources and metrics collectors:
use consistent hash ring
can scale up and down collectors

3.2 2

How to scale metrics transmission pipeline between collectors and DB:
use Kafka queue

3.3 3

where aggregation can happen:

- collection agents: (inside metrics source, e.g. 1 min buffer)
- ingestion pipeline
  before writing into DB.
  but this will lose the raw data
- query side:
  controlled by user