

业务监控及告警技术选型

2016-11-28

未来产品一组 FlymeTV

目录

- 1、 概述 3
 - 1.1 应用背景 3
 - 1.2 业务目的 3
- 2、 总体架构 3
 - 2.1 整体架构描述 3
- 3、 技术选型 4
 - 3.1 监控数据采集组件选型 4
 - 3.2 存储组件选型 4
 - 3.3 数据分析与展示组件选型 6
 - 3.4 数据缓冲组件 7

1、概述

1.1 应用背景

魅族度量平台已经做了监控，为何我们仍需要建立自己的监控与告警系统：

- 1、魅族度量平台暂时不支持对应用单个服务接口 qps,rt 等监控指标的直观展示(折线图)，但在实际观测与指标分析中需要知道某一接口在每一时段具体的请求数及时延，以此来做有针对性的优化；
- 2、我们需要清楚的知道各接口返回的业务状态码分布情况，以此来分析当前服务业务逻辑是否运行正常，仅仅只统计 200 与非 200 的状态是远远不够的，而度量平台也不支持此类监控与分析；
- 3、一旦某接口返回大量非正常业务状态码，此时我们需要及时收到系统告警，快速介入并修复系统异常，甚至有些时候系统可能还未出现异常，但有出异常的趋势，此时我们需要有邮件或短信告知，提前预防。

1.2 业务目的

- 1、时刻掌握系统及各服务接口运行情况
- 2、业务异常信息及时推送至相关负责人
- 3、可支持定制化的业务告警规则
- 4、服务扩容及优化的数据依据来源

2、总体架构

2.1 整体架构描述

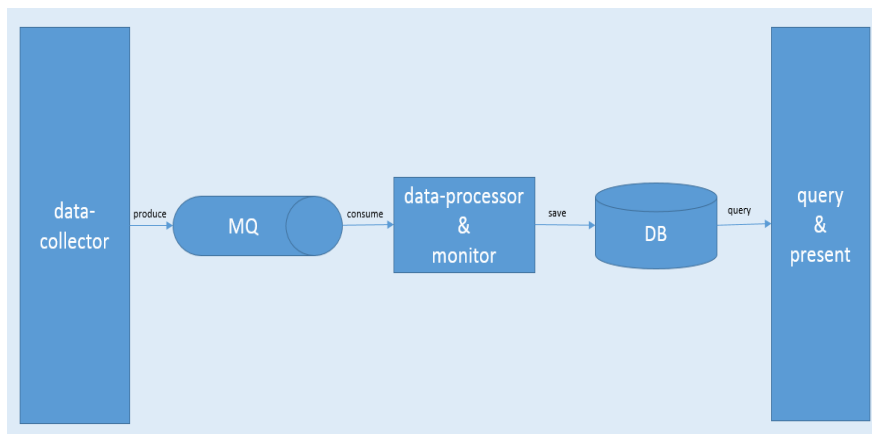


图 2.1.1 监控告警整体架构示意图

监控告警主要涉及到以下几个模块：

- 1、数据采集模块（业务接口数据采集，基础组件数据采集，自定义指标数据采集）

- 2、 数据分析及告警模块(对采集到的数据进行聚合类等操作分析,匹配告警规则)
- 3、 数据存储 (数据入库, 及归档、过期策略等)
- 4、 数据查询与展示模块 (自定义数据展示, 支持个性化图表定义)

3、 技术选型

3.1 监控数据采集组件选型

数据采集模块需对 java 有很好的支持, 且能够搜集以下类型的监控数据:

- 1、 系统中某指标瞬时值
- 2、 某时段内的数据流的平均数值及 pct99, pct95, pct90 数值
- 3、 最好是能够搜集到 jvm 相关信息, 如线程相关指标、内存指标等
- 4、 可支持自定义待搜集数据指标 (如对某一事件的监控)

Java-metrics (<http://metrics.dropwizard.io/3.1.0/getting-started/>)

Metrics 是一个给 Java 提供度量工具的包, 在 JAVA 代码中嵌入 Metrics 代码, 可以方便的对业务代码的各个指标进行监控。

Metrics 提供 5 种基本的度量类型: Gauges, Counters, Histograms, Meters 和 Timers。

Gauges: 是最简单的度量类型, 只有一个简单的返回值, 他用来记录一些对象或者事物的瞬时值。如当前线程数、当前任务队列长度等;

Counter: 是一个简单 64 位的计数器, 他可以增加和减少。比如我们可以定义两个 Counter 类型的计数器, 用来统计所有服务请求数目, 和目前正在处理的请求总数;

Meter: 是一种只能自增的计数器, 通常用来度量一系列事件发生的比率。他提供了平均速率, 以及指数平滑平均速率, 以及采样后的 1 分钟, 5 分钟, 15 分钟速率。比如需要统计请求的速率, 比如统计平均每分钟内有多少次请求进来。只需要定义一个 metric;

Histogram: 是用来度量流数据中 Value 的分布情况, Histogram 可以计算最大/小值、平均值, 方差, 分位数 (如中位数, 或者 95th 分位数), 如 75%, 90%, 98%, 99% 的数据在哪个范围内。比如, 我们想度量, 所有传进来服务的请求参数的长度分布。那么, 可以定义一个 histogram;

Timer: 是 Histogram 跟 Meter 的一个组合, 比如要统计当前请求的速率和处理时间则可以使用 Timer;

3.2 存储组件选型

3.2.1 监控数据特性

业务监控数据有以下特性:

- 1、**数据量急剧膨胀**: 随着业务提升及更多项目的接入, 监控数据将面临高并发写入;
- 2、**实时写入及近实时获取**: 大量的 collector 搜集的数据从四面八方、实时涌来(每分钟成千上万, 甚至每秒钟成千上万条记录), 若想能尽快知道应用是否正常做到近实时的异常告警, 就需要数据能高速入库, 同时还能允许及时查询刚入库的新鲜数据;
- 3、**按时间点高速检索数据**: 监控平台在实际使用过程中往往需要快速定位某一时间点程序运行状况;
- 4、**数据价值随时间推移而降低**: 一个星期之前的数据不如当前时间的数据有价值;
- 5、**成本可控**: 我们不是专业的大数据团队, 机器等资源是项目设计必须考虑的要素, 所以最多使用一台或几台普通虚拟机即可满足部署及运行需求;

3.2.2 关系型数据库存储监控数据的弊端

- 1、按时间断面查询速度慢: 关系型数据库无法按事件时间点快速检索数据, 其只擅长 seek (数据定位);
- 2、入库速度慢;
- 3、特定的函数支持不够友好 (数据查询时分析)

3.2.3 influxDB 时序数据库介绍

时序数据库, 全称为时间序列数据库。时间序列数据库主要用于指处理带时间标签 (按照时间的顺序变化, 即时间序列化) 的数据, 带时间标签的数据也称为时间序列数据。

Influx 官网地址: <https://www.influxdata.com/time-series-platform/influxdb/>

它有三大特性:

1. Time Series (时间序列): 你可以使用与时间有关的相关函数 (如最大, 最小, 求和等)
2. Metrics (度量): 你可以实时对大量数据进行计算
3. Events (事件): 它支持任意的事件数据

特点:

- schemaless(无结构), 可以是任意数量的列
- SQL-like query
- min, max, sum, count, mean, median 一系列函数, 方便统计
- Native HTTP API, 内置 http 支持, 使用 http 读写
- Powerful Query Language 类似 sql
- Built-in Explorer 自带管理工具

InfluxDB 除了拥有上述特性之外, 其在时序数据库中的排名也一直都是靠前, 另外社区较活跃。同时, 更重要的是我们下面将要讲到的 grafana 也原生对 influxDB

支持。

InfluxDB 单机性能优异，对硬件要求较低，结合 TV 项目实际情况，使用一台虚拟机部署 influxDB 足够支撑最近两年内的监控需求。

本地 influxDB 地址: <http://172.16.114.3:8083/>

3.3 数据分析与展示组件选型

Grafana (<http://grafana.org/>)



Grafana 是 Graphite 和 InfluxDB 仪表盘和图形编辑器。Grafana 是开源的，功能齐全的度量仪表盘和图形编辑器，支持 Graphite，InfluxDB 和 OpenTSDB。

Grafana 主要特性:

- 灵活丰富的图形化选项;
- 可以混合多种风格;
- 支持白天和夜间模式;
- 多个数据源; Graphite 和 InfluxDB 查询编辑器等等。

Graphite 指标编辑器:

- Graphite 指标表达解析器
- 功能齐全的查询功能
- 快速添加和编辑函数和参数
- 模板化查询

图形化:

- 快速渲染，甚至是较大的时间跨度
- 点击和拖拽缩放
- 多个 Y 轴
- 条形，折线，点
- 智能 Y 轴格式化
- 系列切换和颜色选择
- Legend values 和格式化选项
- 网格阈值，轴标签
- Annotations
- 仪表盘

之所以选 grafana，主要是其具有非常灵活的扩展性，几乎可以随心所欲

来定义任何想查看的指标，最重要的一点是还不需要写任何代码，完全甚至远远满足 TV 监控项目对数据查询展示的需求，另外其还自带阈值告警功能（但不打算用，告警模块还是自己写比较好，且必须要支持短信告警）。

本地 grafana 地址：

<http://172.16.114.3:3000/dashboard/db/flymetv?panelId=1&fullscreen&edit&from=now-24h&to=now&editorTab=Metrics>

3.4 数据缓冲组件

为了能尽可能的减少监控数据搜集组件对程序性能的影响，在数据搜集和入库之间加入了一层缓冲层，该缓冲层计划使用全局 `metaq`，`collector` 采集到样例数据后异步发送至 MQ，最终由入库程序消费 MQ 消息并写入到时序数据库。