

# 监控方案

由 马文静创建, 最后修改于44分钟以前
 20 views since 26 Apr 2022

- 背景
  - 目标
  - 开源方案调研
    - 1.web-report + zanePerfor
      - 1.方案介绍
      - 2.数据报表
      - 3.总结
      - 4.结合业务改进方案
    - 2.webTrack sdk+阿里云日志平台
      - 1.方案介绍
      - 2.整体流程
      - 3.数据报表
      - 4.总结
    - 3.perfume + Google Analytics
      - 1.方案介绍
      - 3.数据报表
      - 4.总结
    - 4.自建perf.js + pipe
    - 5.业界成熟的商业化平台参考

## 背景

随着业务的发展，后续将不断围绕着用户体验进行一些优化手段的实施，就需要相应的评估工具来对优化前后的效果进行对比，以更好的对真实用户体验进行一定的认知，同时更有针对性的开展一些优化方案。虽然用户体验是用户在使用产品过程中建立起来的一种纯主观感受，但是我们可以通过一些客观的手段把用户体验数字化，使其成为一个衡量标准。

## 目标

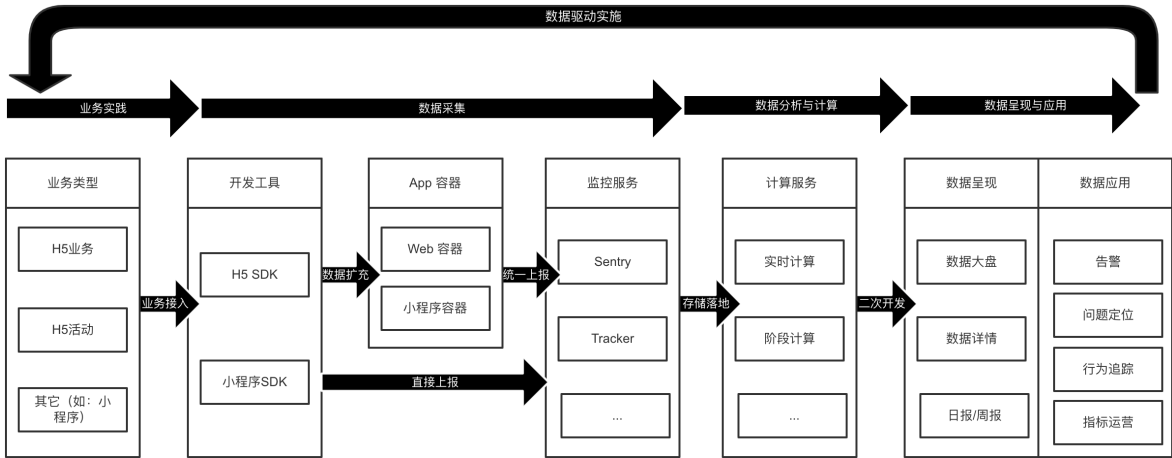
一套标准化监控方案，帮助业务优化用户体验。

【目标拆解】：

- 量化用户感官体验指标
  - 节省业务监控接入成本
  - 定位性能问题，验证优化手段的有效应

【相应策略】：

- 统一指标口径：白屏时间、首屏时间等
  - 构建通用SDK: 业务无感知通用化采集方案，同时集成现有上报SDK
  - 监控服务：基于开源自建 OR 现有成熟的服务分析平台
  - 实现数据应用：报表、大盘、日/周报



## 开源方案调研

# 1. web-report + zanePerfor

## 1.方案介绍

zanePerfor是一个服务于性能监控的业务平台项目

指标采集sdk：基于web-report 进行数据上报

数据服务：语言 — node.js, 技术框架 — egg.js, 数据库 — mongodb, 缓存服务 — redis 可选，支持单机多数据库模式，也支持redis集群 mongodb集群 服务集群

web-report 是一款浏览器端页面性能， ajax, fetch ,错误信息，资源性能上报SDK，资源小巧，性能强悍。可以基于该sdk做二次封装，收集我们关注的核心性能指标

parameter name	describe	explain
url	上报页面地址	
markUv	统计uv标识	
markUser	用户标识	可用来做UV统计， 和用户行为漏斗分析
isFristIn	是否是每次会话的第一次渲染	可以用来做首屏渲染性能统计分类
type	上报类型	1:页面级性能上报 2:页面ajax性能上报 3： 页面内错误信息上报
screenwidth	屏幕宽度	
screenheight	屏幕高度	
preUrl	上一页面	
errorList	错误资源列表信息	
->t	资源时间	
->n	资源类型	resource, js, ajax, fetch,other
->msg	错误信息	
->method	资源请求方式	GET, POST
->data->resourceUrl	请求资源路径	
->data->col	js错误行	
->data->line	js错误列	
->data->status	ajax错误状态	
->data->text	ajax错误信息	
performance	页面资源性能数据(单位均为毫秒)	
->dnst	DNS解析时间	
->tcpt	TCP建立时间	
->wit	白屏时间	
->domt	dom渲染完成时间	
->lodt	页面onload时间	
->radt	页面准备时间	
->rdit	页面重定向时间	

parameter name	describe	explain
->uodt	unload时间	
->reqt	request请求耗时	
->andt	页面解析dom耗时	
resoruceList	页面资源性能数据	
->decodedBodySize	资源返回数据大小	
->duration	资源耗时	
->method	请求方式	GET,POST
->name	请求资源路径	
->nextHopProtocol	http协议版本	
->type	请求资源类型	script, img, fetchrequest, xmlhttprequest, other

2. 数据报表

3. 总结

前端接入web-report-sdk，进行数据上报， 可以自定义上报url，后端部署**node** 服务，MongoDB等数据存储，可以根据业务量级选择最终的搭建方案

优点

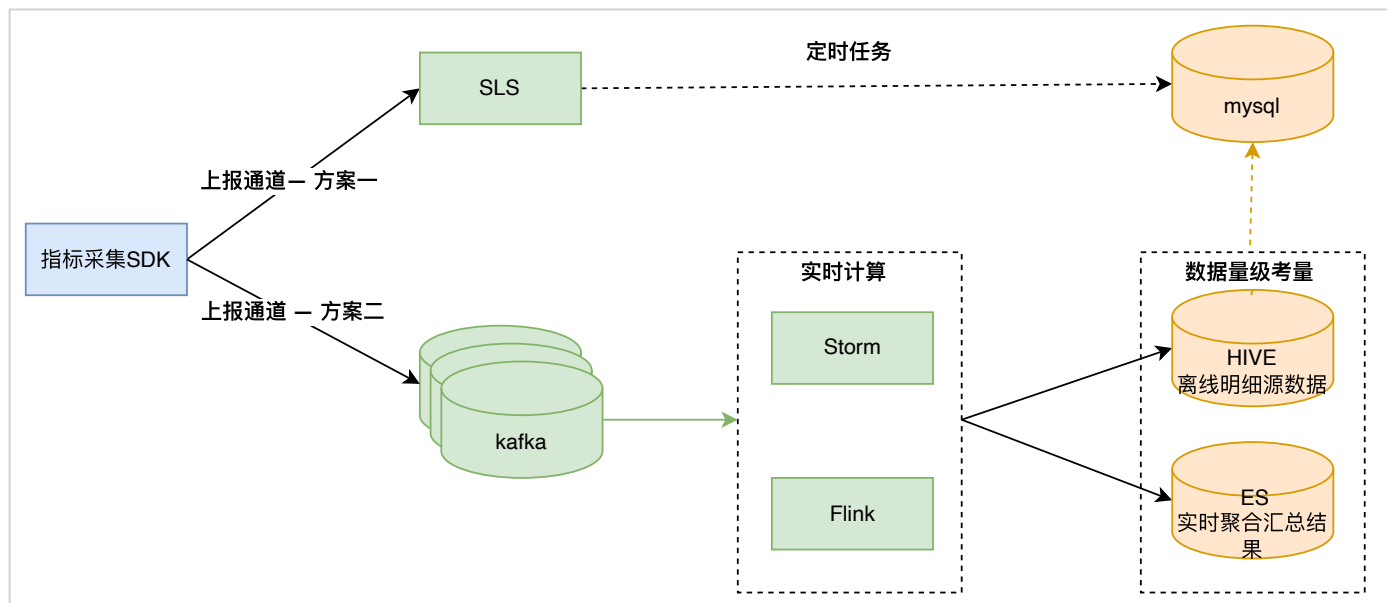
- 项目开源，可以在此基础上二次开发，但需要一定学习成本与后端支持
- star 1.3k

缺点

- 需要自己搭建环境及服务及部署，相比于现有的成熟分析平台，具有一定成本
- zanePerfor 属于个人开发者，目前还不完善，处于开发初期，项目还需要持续优化

#### 4. 结合业务改进方案

根据业务量级考虑高并发场景：



参考文档

github: <https://github.com/wangweianger/zanePerfor>

## 2.webTrack sdk+阿里云日志平台

### 1.方案介绍

webTrack是一个通过get请求上报日志的SDK，上报到阿里云日志服务sls，可上报performance性能数据包括DNS解析时间，TCP建立时间，白屏时间，dom渲染完成时间，页面onload时间，页面准备时间，页面重定向时间，unload时间，request请求耗时，页面解析dom耗时 等指标，除此之外，还可以统计pv、uv、上报自定义关键事件，开启全局错误监听等

### 2.整体流程

开通方式

- 一、阿里云控制台-创建日志服务,开通Web Tracking
- 二、sdk方式(通过Log Service Java SDK开通Web Tracking)

前端上报流程

- 落地页安装webtrack npm包
- 配置config(阿里云sls相关配置) 生成webtrack实例
- 埋点上报

数据分析流程

- 阿里云aliyunLogstore配置索引
- 阿里云日志存储，查询分析验证
- 配置相关仪表盘等监控图标数据
- 通过get请求上报日志的SDK
- 上报到阿里云日志服务sls

### 3.数据报表



4.总结

优点

- 数据平台接入sls，前端只需接入web track sdk，节省时间成本
- 阿里云sls支持数据存储及可视化展示，节省开发可视化后台及存储平台这部分时间成本
- web track 等sdk 可以监控性能，还可以自定义上报事件

缺点

- sls数据仪表盘等需要单独配置，针对具体数据编写对应sql分析语句，有一定不灵活性

参考文档：

接入文档：[https://help.aliyun.com/apsara/enterprise/v\\_3\\_13\\_0\\_20201215/sls/enterprise-ascm-user-guide/web-tracking.html](https://help.aliyun.com/apsara/enterprise/v_3_13_0_20201215/sls/enterprise-ascm-user-guide/web-tracking.html)

3.perfume + Google Analytics

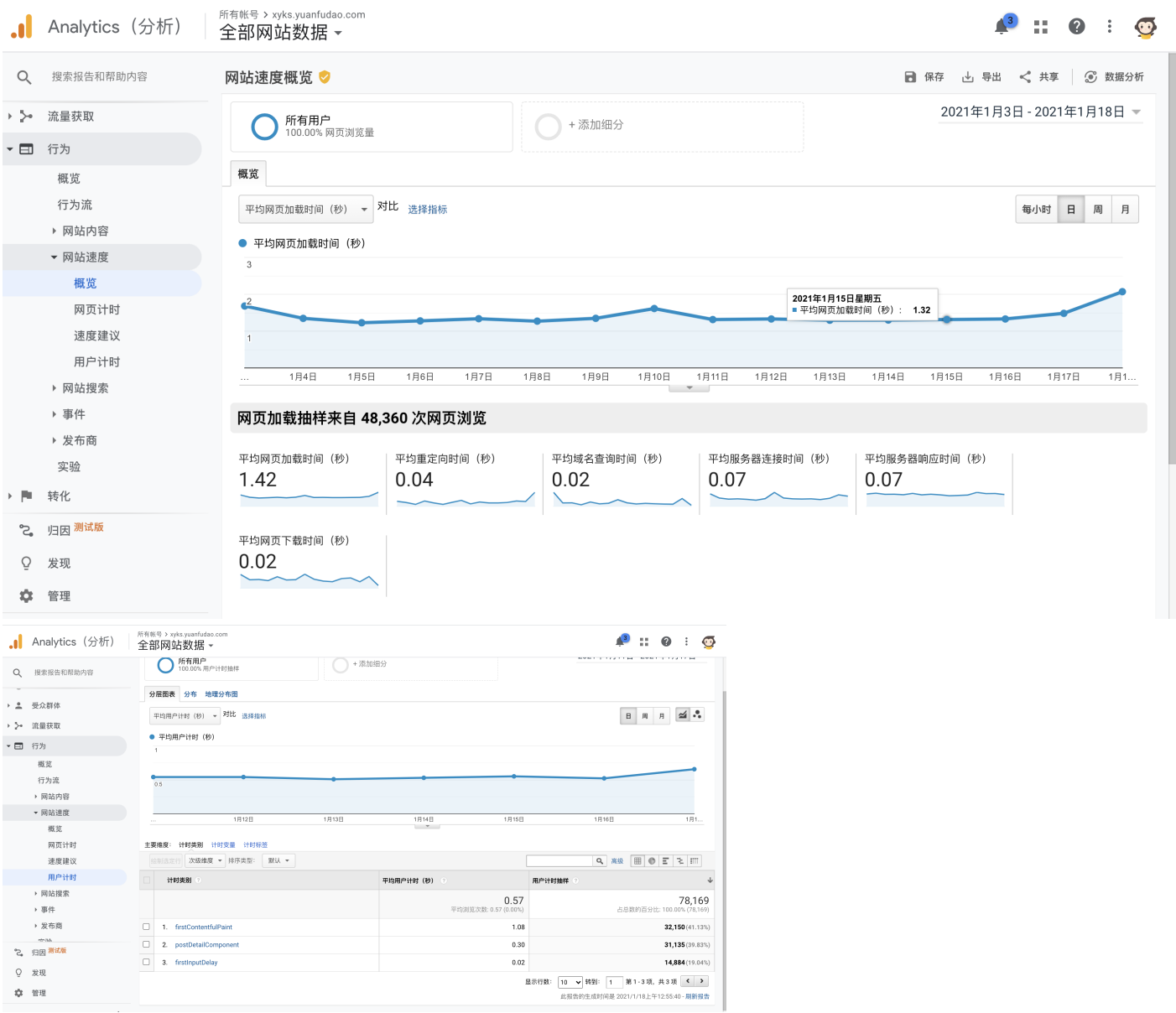
1.方案介绍

Perfume 是一个微小的网络性能监控库，利用Performance得到性能信息,可以监控的性能指标：FP/FCP/FID/可自定义定义打点计算时间间隔，来监控组件生命周期等，支持数据上传至Google Analytics

Web Vitals	Good	Needs Improvement	Poor
Fist Paint (fp)	0-1000	1001-2500	Over 2500
First Contentful Paint (fcp)	0-1000	1001-2500	Over 2500
Largest Contentful Paint (lcp)	0-2500	2501-4000	Over 4000

Web Vitals	Good	Needs Improvement	Poor
First Input Delay (fid)	0-100	101-300	Over 300
Cumulative Layout Shift (cls)	0-0.1	0.11-0.25	Over 0.25
Cumulative Layout Shift Final (clsFinal)	0-2500	2501-4000	Over 4000
Total Blocking Time (tbt)	0-200	201-600	Over 600
Navigation Total Blocking Time (tbt)	0-200	201-600	Over 600

3.数据报表



4.总结

优点

- Google Analytics 可以更全方位的针对网站数据进行统计，比如pv、uv、用户受众群体、活跃等其他指标
- perfume 包体积小，支持上报GA，可以自定义组件渲染时间，更加准确 star 2.6k

缺点

- GA 针对自定义上报的数据 局限性比较大，数据不是特别灵活
- perfume 针对性能的指标相对来说比较少，部分指标需要二次开发

参考文档：

中文文档：[https://github.com/Zizzamia/perfume.js/blob/master/README-zh\\_CN.md](https://github.com/Zizzamia/perfume.js/blob/master/README-zh_CN.md)

github: <https://github.com/Zizzamia/perfume.js>

#### 4.自建perf.js + pipe

<https://pipe.zhenguanyu.com/#/tableshow/162797290958832581?navId=2903&projectName=%E5%B0%8F%E7%8C%BF%E5%8F%A3%E7%AE%97&chartType=7&dateStatus=-1&startDate=2021-09-13&endDate=2021-10-30&dateType=2&selects=0%7C0%7C2%7C0%7C1>

#### 5.业界成熟的商业化平台参考

1.mmtrix

<http://www.mmtrix.com/enterprise>

2.oneapm

<https://www.oneapm.com/bi/feature2.html>

3.岳麓

<https://yueying.effirst.com/pricing>

4.阿里云监控平台ARMS

[https://help.aliyun.com/document\\_detail/58652.html](https://help.aliyun.com/document_detail/58652.html)

无标签