

墨迹天气服务器架构简介

墨迹简介

- 累计设备数：3.3亿
- 日活：3000万
- 天气接口日PV：2.5亿
- 定位接口日PV：1亿
- 支持天气城市：35W
- 服务器组人数：25人

服务器组功能模块

- 天气接口
- 全球定位
- 时景社区
- 墨迹商城
- **App Store**
- 皮肤小铺
- 空气果
- 广告后台
- 推送
- 官网/论坛/指数页面等等
- 问题反馈

业务特点

- 大并发，高流量
- 高性能，高可用性
- 服务化
- 跨平台
- 多机房部署
- 避免单点故障
- 使用成熟技术
- 国际化

整体架构

天气
接口

时景

商城

App
Store

空气果

后台

RPC

Web Service

数据
中心

定位

用户
中心

消息
中心

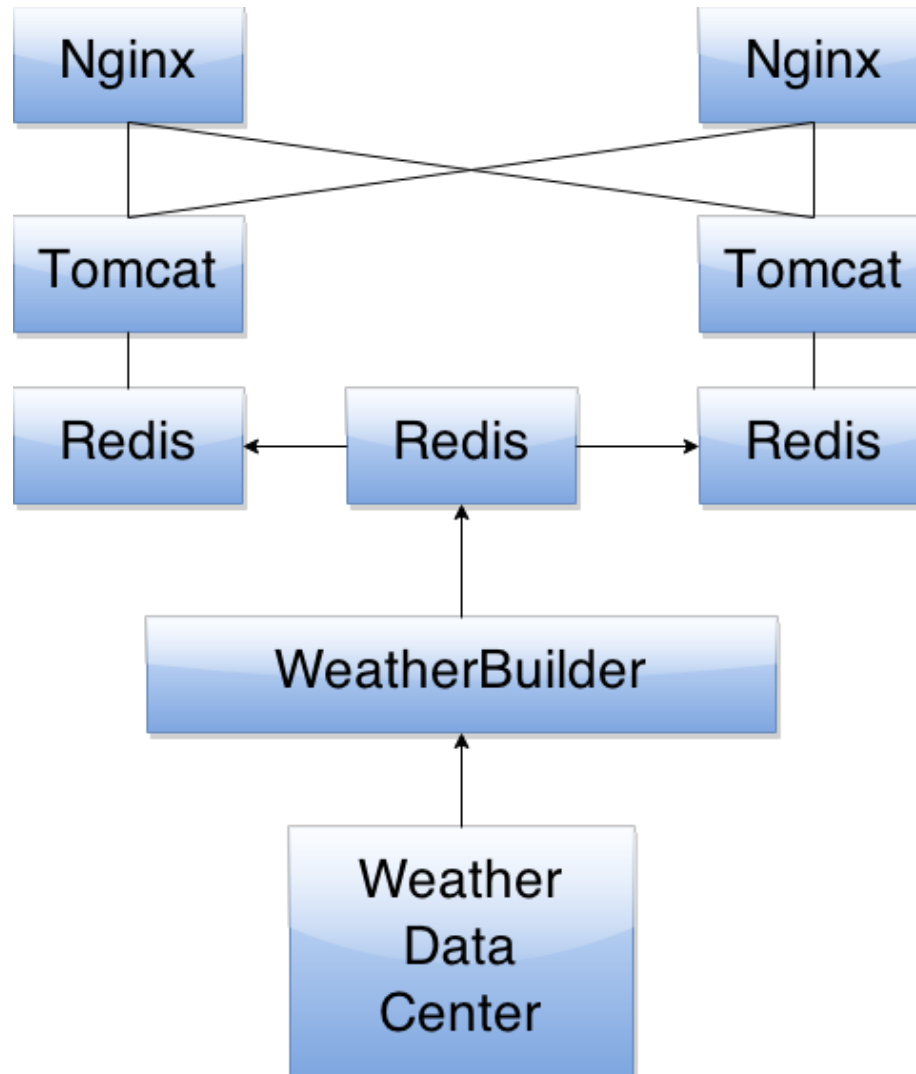
支付

统计

技术简介

- JAVA（业务接口）、Python（天气数据中心，统计及大数据）、PHP（商城，后台，官网论坛等），少量的C和Go
- Nginx作为前端负载均衡，HAProxy作为服务端负载均衡
- 数据库使用Redis+MySQL的两层架构，少量使用mongoDB（统计），PostgreSQL（定位）
- RPC层传输使用OMQ实现（类似zerorpc），使用Protobuf作为传输协议
- 两地三机房+云服务的防灾机制

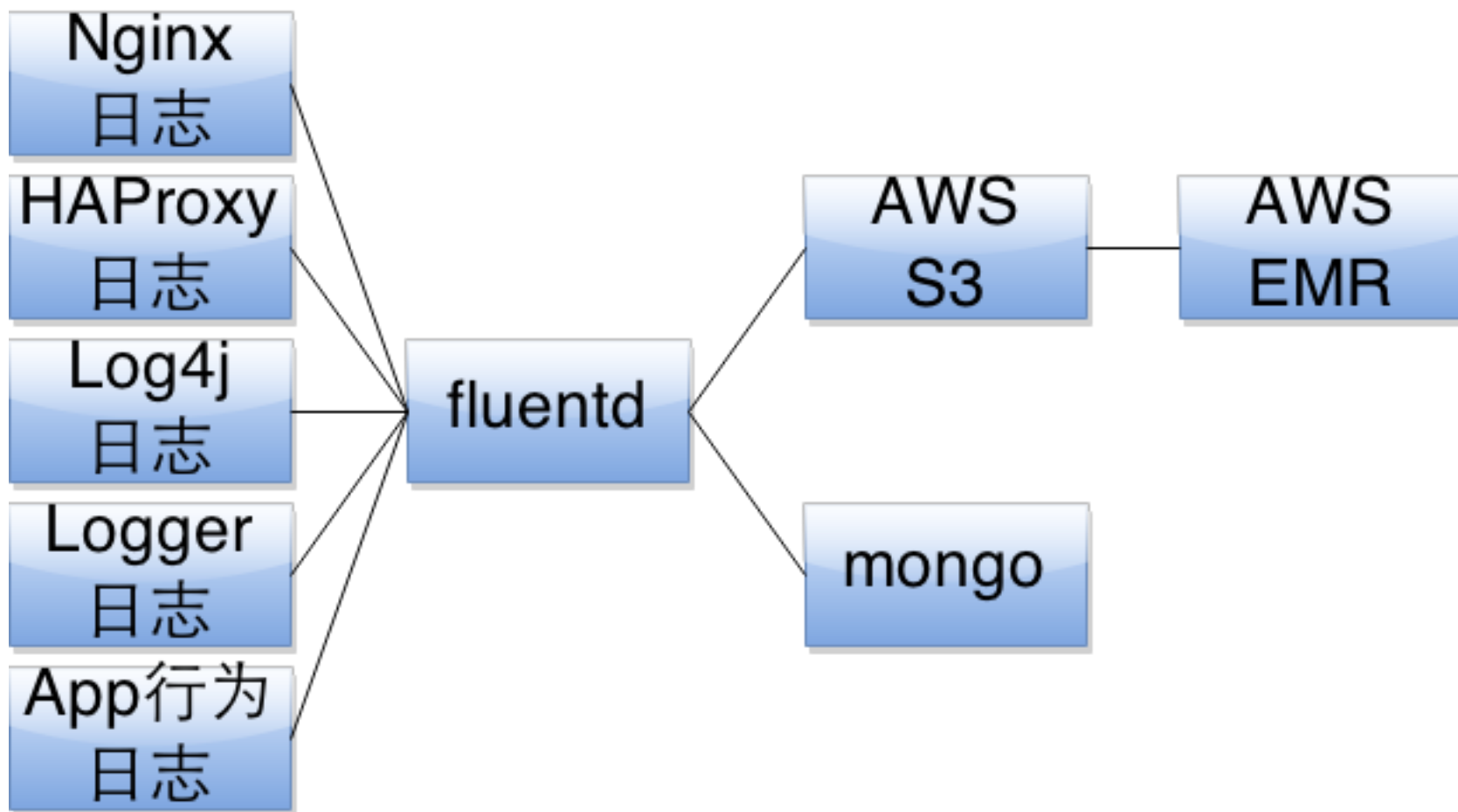
天气接口设计



天气接口设计

- Tomcat层有各项数据的LRU cache
- Nginx根据冷热数据会发往不同的Tomcat集群提高cache hit
- 根据时间戳只返回更新的数据
- 单机4个Tomcat实例，共用1份Redis数据
- WeatherBuilder向天气数据中心注册后，天气变更时会收到天气数据中心的**通知**，更新主Redis，同时还会接收消息中心及广告模块的**通知**。
- Redis是多实例架构，并有一份冗余
- 天气数据中心在多机房有热备
- 最后一道保险：WeatherBuilder除了更新redis外会更新一份静态天气数据文件放到CDN上，保证主要天气数据可以正常更新

日志处理和大数据



日志处理和大数据

- 考虑过使用kafka+storm+HDFS+hadoop的解决方案，主要矛盾是复杂度过高和运维人员不足
- 使用fluentd中间件，两个后端，AWS S3和mongoDB
- mongoDB使用capped collection用来做准实时日志数据结构化，分析，报警等使用
- AWS EMR使用同步到AWS S3上的数据做中长期数据统计及大数据分析

Thanks