



GOPS2017  
Shanghai



# GOPS

# 全球运维大会

2017

上海站

指导单位:  数据中心联盟  
Data Center Alliance

主办单位:  高效运维社区  
GreatOps Community

 开放运维联盟  
OOPSA Open OPS Alliance

大会时间: 2017年11月17日-18日

大会地点: 上海光大会展中心国际大酒店 (上海徐汇区漕宝路67号)





GOPS2017  
Shanghai

# 微信海量数据监控的设计与实践

陈晓鹏      腾讯微信 高级工程师

# 微信后台系统现状



GOPS2017  
Shanghai

- 月活跃用户： > 9亿
- 后台调用数： > 200亿/min
- 后台模块数： > 5k
- 服务器数： > 5w

庞大且复杂的后台系统，单靠人力难以维护。

# 运维监控系统的作用



GOPS2017  
Shanghai

故障报警

故障分析

自动化策略



GOPS2017  
Shanghai

# 目录



1

监控数据收集轻量化

2

微信数据监控的发展过程

3

海量监控分析下的数据存储设计思路

# 常见数据收集流程



GOPS2017  
Shanghai



- 日志量 > 2000亿/min。
- 文本日志处理消耗过大。
- 日志格式太多难以维护。

**如何才能实现分钟级、秒级数据监控？**

# 微信运维监控数据处理



GOPS2017  
Shanghai



# 数据分类



GOPS2017  
Shanghai

- 实时故障监控分析
- 非实时数据统计（业务报表等）
- 单用户异常分析

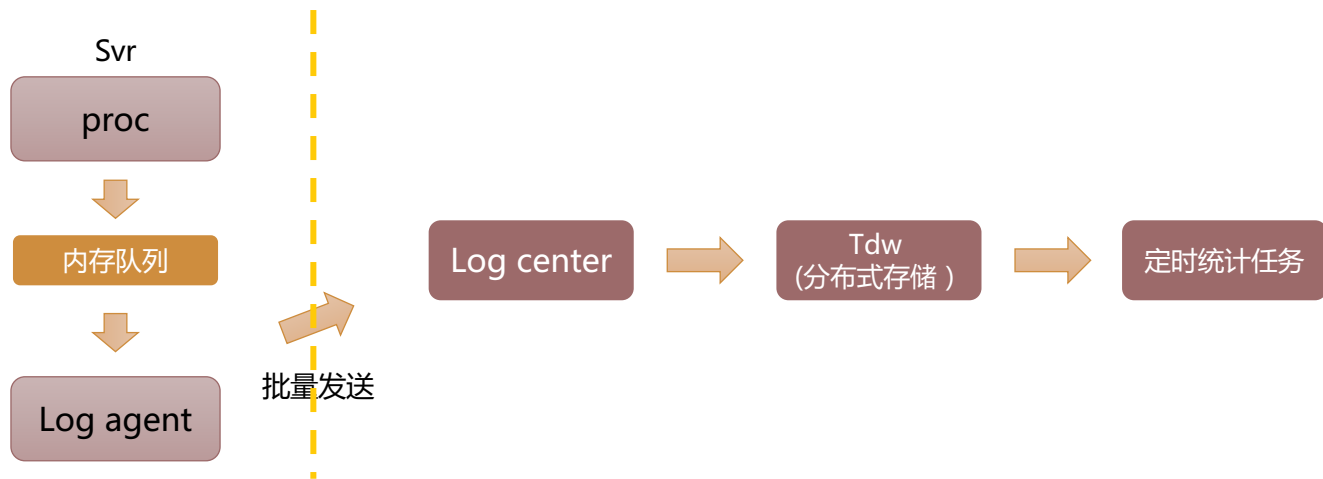


# 非实时数据统计



GOPS2017  
Shanghai

- 数据申请：logid + 自定义数据字段 -> protobuf
- 数据上报：

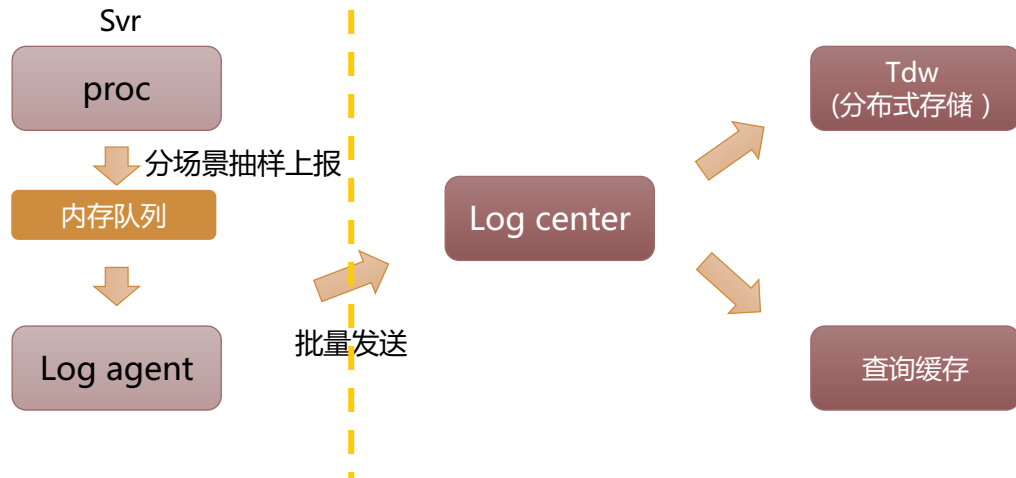




GOPS2017  
Shanghai

# 单用户异常分析

- 固定格式：logid + 固定数据字段 (服务器IP+返回码等) -> protobuf
- 数据上报：



# 实时监控数据



GOPS2017  
Shanghai

数据分类

简化数据

定制快速处理  
策略

# 实时监控数据 —— 数据分类



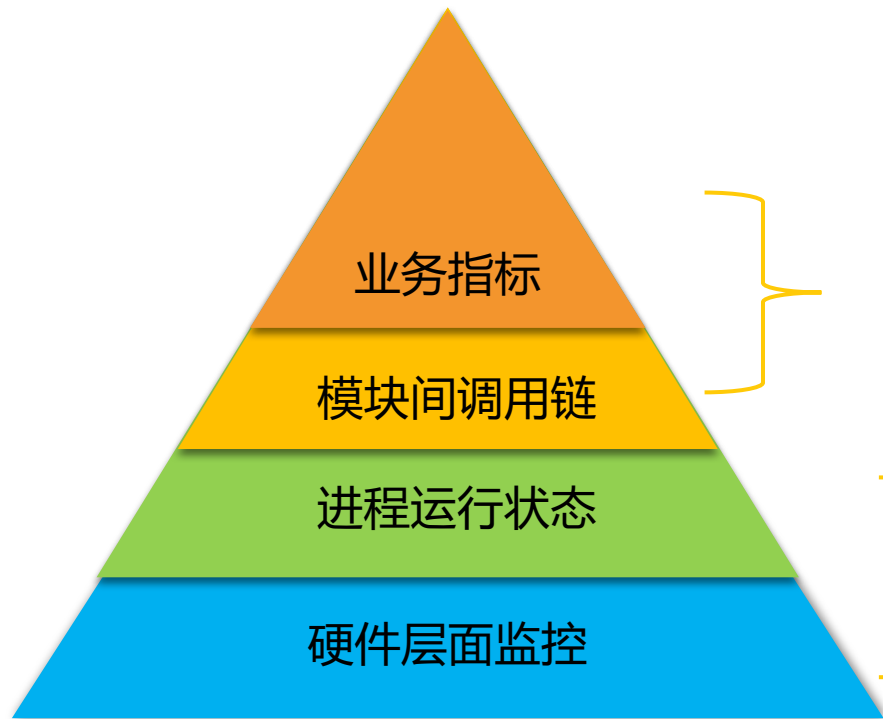
GOPS2017  
Shanghai

- 后台数据监控
- 终端数据监控
- 对外监控服务（商户、小程序）

# 后台数据监控



GOPS2017  
Shanghai



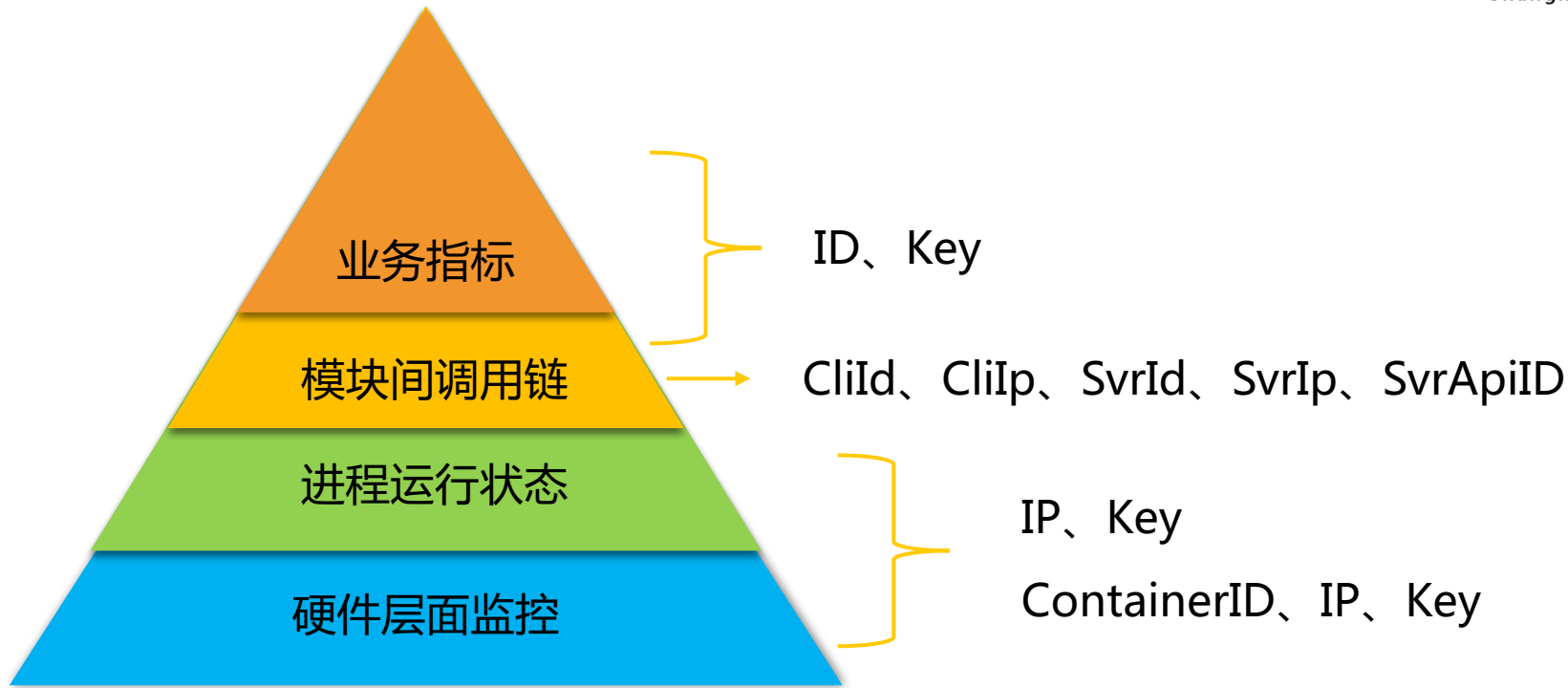
上报量： > 2k亿/min  
汇总结果： > 2亿/min

> 4kw/min

# 简化数据格式



GOPS2017  
Shanghai



# IDKey数据



GOPS2017  
Shanghai

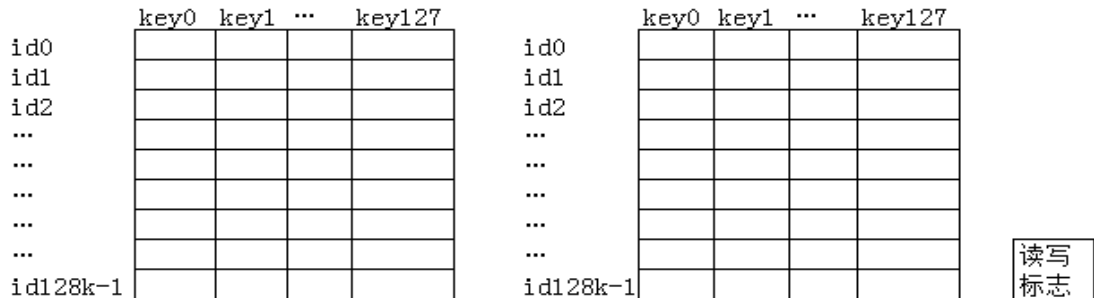
- 关键监控数据，分钟级、秒级监控。
- > 2k亿/min，日志式数据收集？No!
- 快速内存汇总。

# IDKey数据



GOPS2017  
Shanghai

## 共享内存



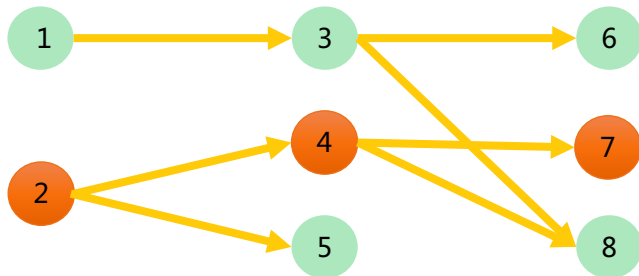
- `OssAttrInc(uint32_t id, uint32_t key, uint32_t val)` //累加
- `OssAttrSet(uint32_t id, uint32_t key, uint32_t val)` //设置新值
- `OssAttrSetMax(uint32_t id, uint32_t key, uint32_t val)` //设置最大值
- `type __sync_fetch_and_add (type *ptr, type value)`
- `bool __sync_bool_compare_and_swap (type *ptr, type oldval, type newval)`
- 单机平均上报数据量：1k，大幅降低秒级汇总难度。



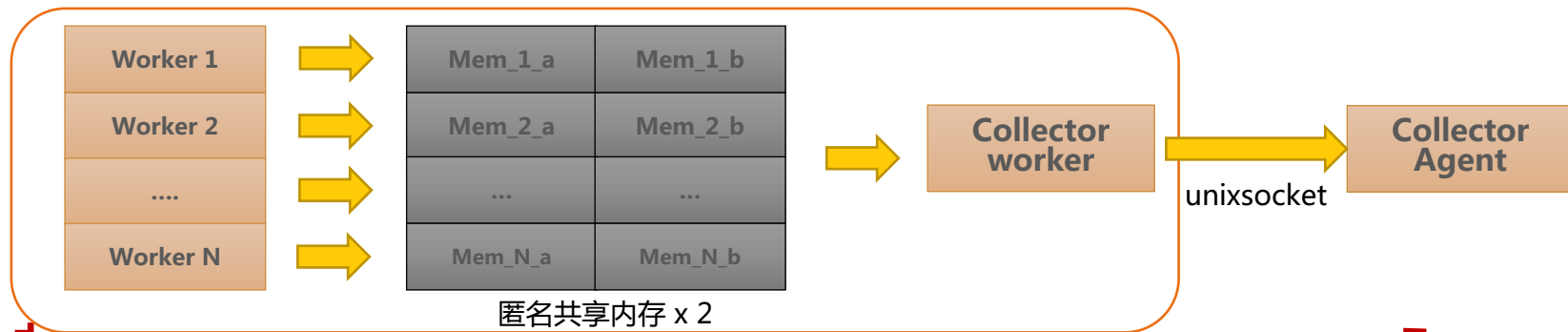


# 调用关系数据

- A模块 a机 b进程 -> B模块 x机 y接口，调用数、失败数、耗时。
- 可以定位故障点（机器、进程、接口）及影响面。



- 上报方式



# 终端数据监控



GOPS2017  
Shanghai

客户端性能、异常

后台性能、异常

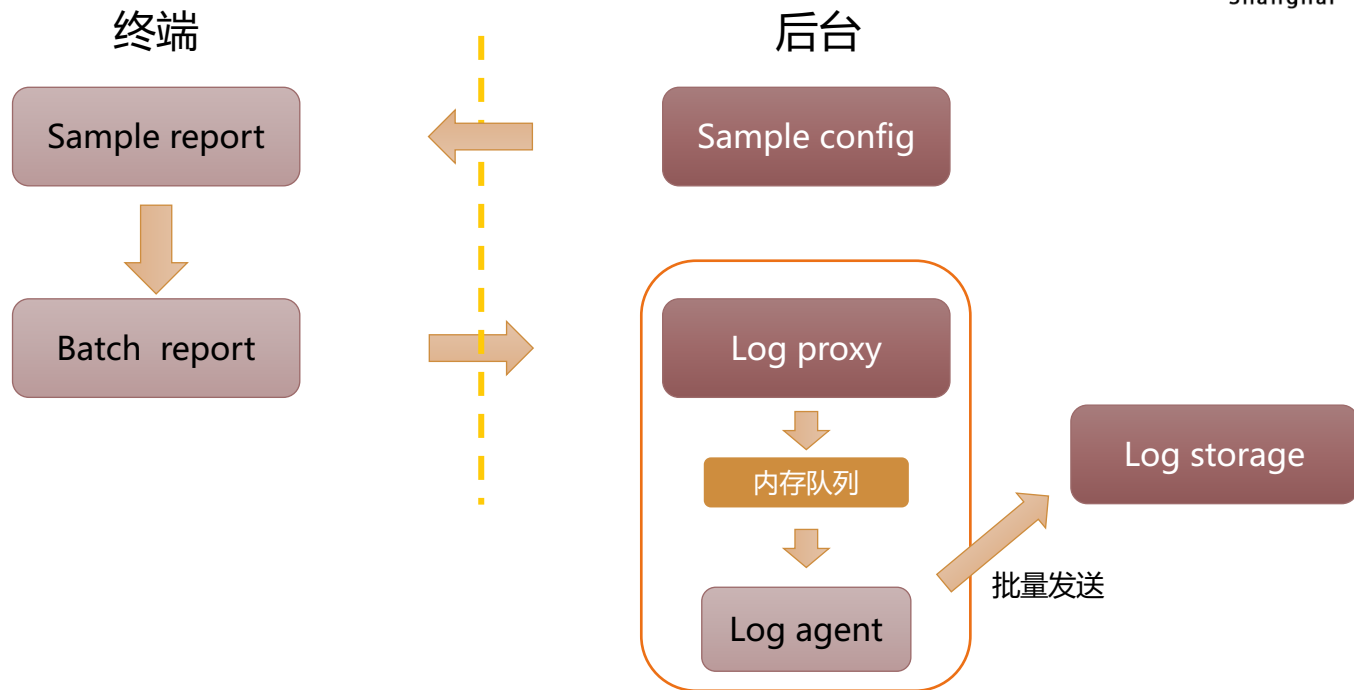
网络异常

AppType、AppVer、ID、Key

# 终端监控数据上报



GOPS2017  
Shanghai



# 对外监控服务



GOPS2017  
Shanghai

- 用户可自行配置数据格式及监控
- 数据格式：

bus_id	int
uid	int64
namespace	string
metric	string
dimension1~n	string (n<=5)
value	double

# 对外监控服务



GOPS2017  
Shanghai

微信支付 | 商户平台

首页 交易中心 商户中心 营销中心 产品中心 数据中心

交易数据  
实时交易数据  
历史交易数据


消息中心  
新增商户订单

### 支付有异常，微信通知你

订购微信支付，当微信支付使用出现异常时，立即在微信支付通知中心收到异常通知。  
[了解异常详情](#)

**微信支付渠道**

微信支付和跨境支付本商户的微信支付通知群，请扫描下方二维码加入微信群，及时接收异常通知。



微信支付  
微信支付通知群  
管理群成员

**告警项**

告警项	说明	订购状态	操作
支付调用异常	当商户调用微信支付出现异常时，进行告警	未订购	<a href="#">订购</a>
支付回调异常	当微信支付回调商户出现异常时，进行告警	未订购	<a href="#">订购</a>

## 运维中心

错误查询 错误告警

性能监控 告警设置


### 性能监控

接口类别 当前总数

网络接口 (成功) 4006614

2小时

调用成功





GOPS2017  
Shanghai

# 目录

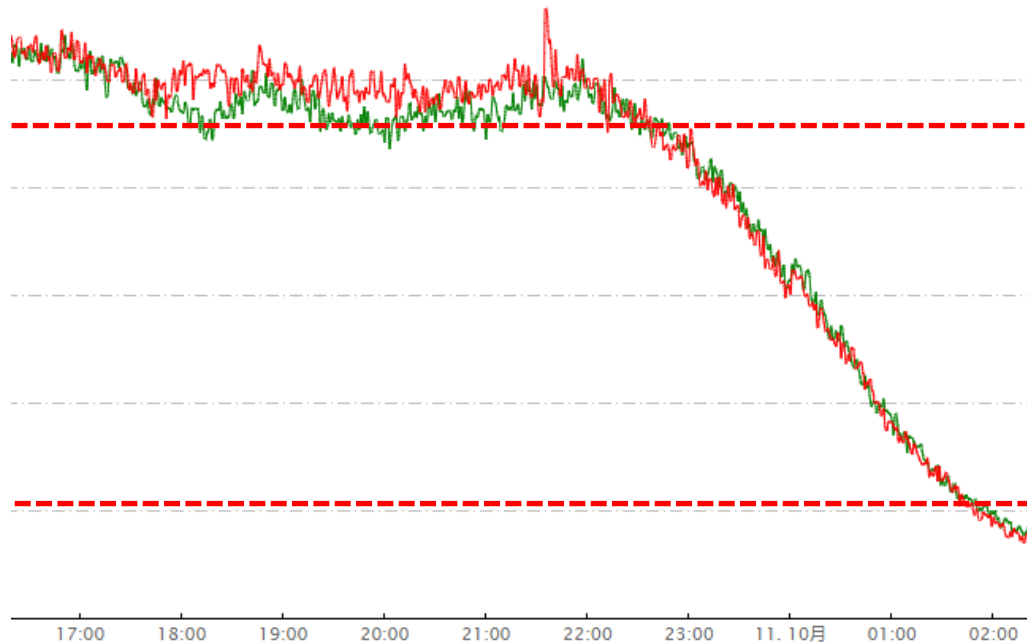
1 监控数据收集轻量化

➔ 2 微信数据监控的发展过程

3 海量监控分析下的数据存储设计思路

# 常见异常检测方法

- 阈值
- 只适用很少量的场景



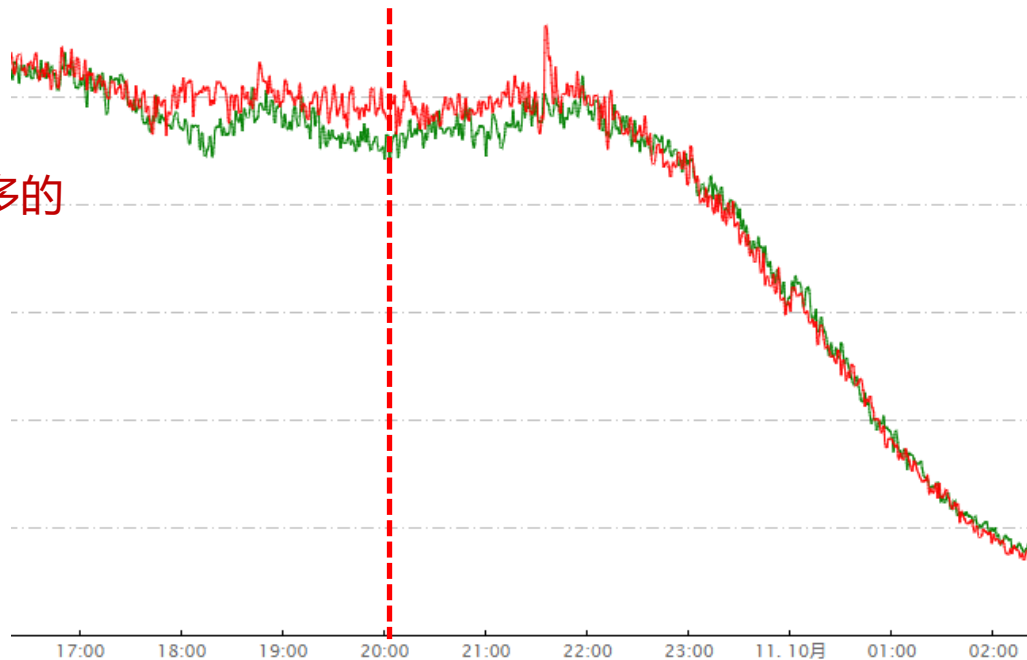
GOPS2017  
Shanghai

# 常见异常检测方法



GOPS2017  
Shanghai

- 同比
- 前后两天同一时间的数量有差异比较多的情况，数量级比较小时尤其明显。
- 只有降低敏感度才能保证准确性。



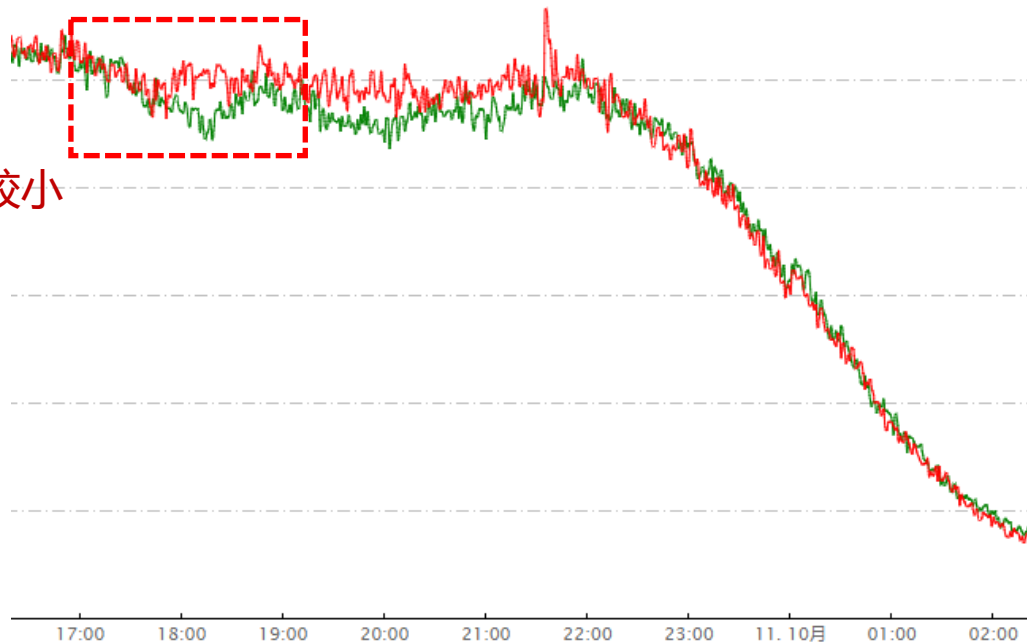


# 常见异常检测方法



GOPS2017  
Shanghai

- 环比
- 相邻的数据并非平稳变化，数量级比较小时尤其明显。
- 同样只有降低敏感度才能保证准确性。



# 算法改进



GOPS2017  
Shanghai

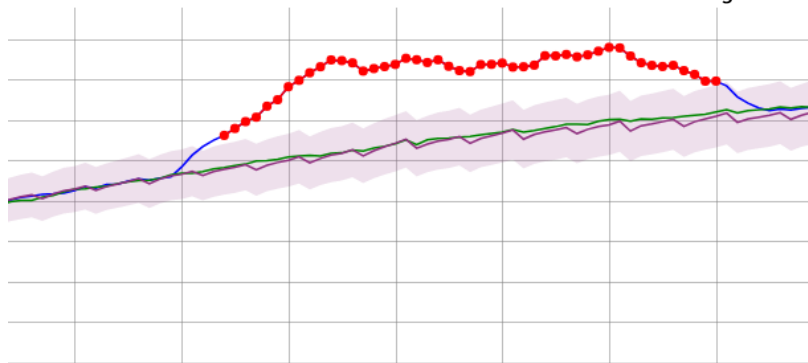
- 均方差

$$s^2 = \frac{1}{n} [(x_1 - x)^2 + (x_2 - x)^2 + \dots + (x_n - x)^2]$$

取过去1个月每天同一时间的数据计算平均值与均方差。

- 用多天数据适应数据的抖动情况。

- 对于每天数量差异大的曲线，敏感度很低，容易漏报。



# 算法改进

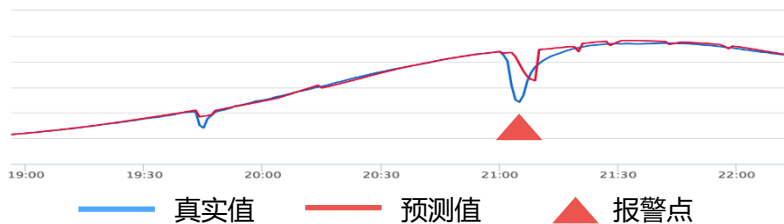


GOPS2017  
Shanghai

- 多项式拟合预测

$$y(x, w) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M = \sum_{i=0}^M w_i x^i$$

应用于周期稳定曲线，通过历史数据预测数据趋势。



- 历史数据平稳的曲线，出现很长时间的缓慢变化也不会判断为异常。

# 监控配置问题



GOPS2017  
Shanghai

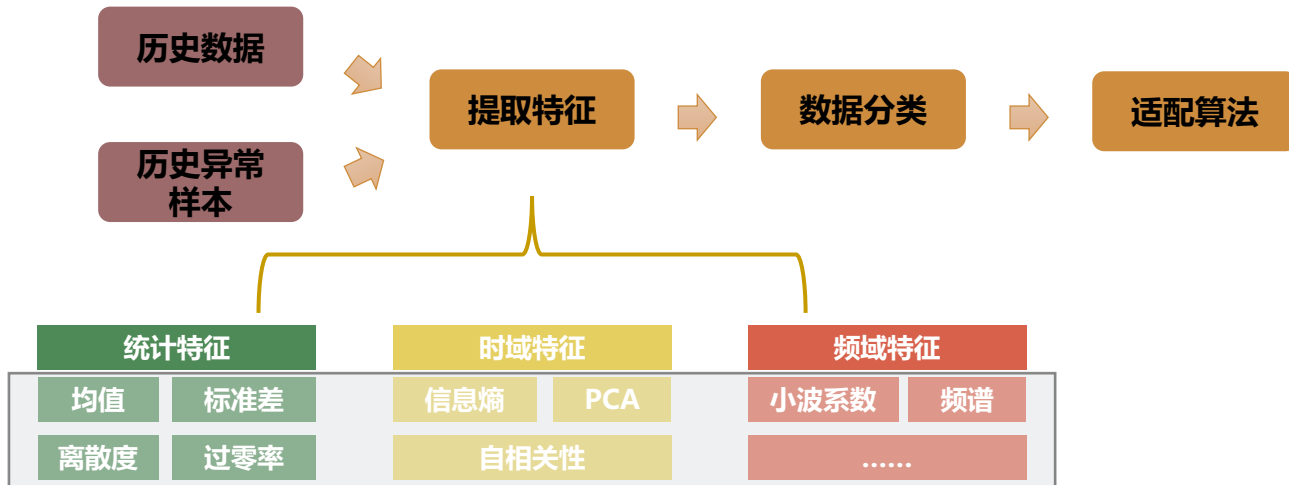
- 超过30w的监控项要人手配置。
- 观察曲线选择不同算法。
- 选择不同的敏感度。
- 隔一段时间需要进行调整。

**人工处理难以持续！**

# 监控数据自动分类



GOPS2017  
Shanghai





GOPS2017  
Shanghai

# 目录

1 监控数据收集轻量化

2 微信数据监控的发展过程

➔ 3 海量监控分析下的数据存储设计思路



GOPS2017  
Shanghai

mmclimonitorlogicsvr (图: 调用 失败 耗时)  
mmclimonitorlogicsvr (图: 调用 失败 耗时)  
mmvardnode(mmvardnode1000) (图: 调用 失败 耗时)  
mmyardnode(mmyardnode1000) (图: 调用 失败 耗时)

被调 概况 主调

接口  
接口被调: 主机

服务  
服务器名

接口被调: 调用源 接口 返回码 主调: 目标主机

基础数据: CPU 内存 IO 磁盘 流量 网卡 进程 cgroup yard gpu IDKEY 变更 shell操作 shell操作(新)

Name	Cpu	Use	Sys	Virt (KB)	Rss (KB)	Shared (KB)	Read (KB)	Write (KB)	句柄	Restart	Core
mmsnsmentionidx	75%	54%	20%	71,922,760	7,184,478	4,655,522	0	1	3,006	0	0
mmmsgbroker	56%	44%	12%	170,216,301	41,142,438	28,475,135	2	0	6,324	0	0
mmauthcheck	51%	45%	6%	6,284,866	1,083,529	804,925	0	80	729	0	0
connagent	40%	17%	23%	1,986,443	630,341	531,372	0	0	5,104	0	0
mmochwsteprank	32%	22%	9%	107,716,080	23,060,783	18,176,464	9	0	2,934	0	0
mmemotionstorelogicsvr	32%	27%	5%	80,025,616	9,761,859	7,089,579	0	25	783	0	0
mmcontactasynmqworker	30%	19%	11%	64,421,501							0
mmmemcachesvr	15%	5%	9%	10,618,876							0
ossattragent	6%	2%	2%	16,055,628							0
mmoctvappsvr	5%	2%	2%	70,705,691							0
mmibagent	4%	4%	0%	1,171,795							0
mmcontactasynmq_svr	3%	1%	1%	2,768,595							0
mmochwsteprankmq	3%	2%	1%	2,850,998							0
mmauthcheckmqworker	2%	1%	0%	21,003,207							0
mmdataagent	2%	2%	0%	3,946,587	863,288	832,447	0	0	105	0	0
zkagent	2%	2%	0%	4,062,299	843,971	798,178	0	7	264	0	0
mmsecureagent	0%	0%	0%	3,654,564	380,586	355,910	0	0	21	0	0
SendDayRankResultMessage	0%	0%	0%	4,344,145	650,381	556,400	51	0	48	0	0

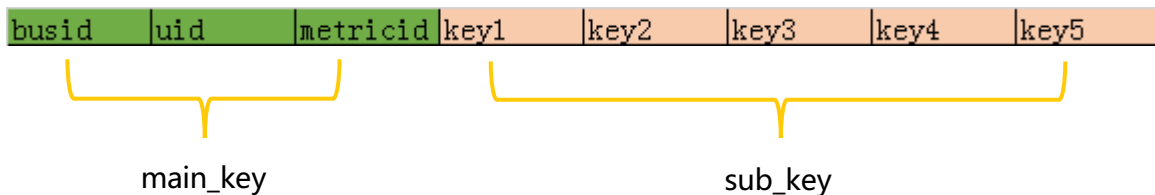




GOPS2017  
Shanghai

# 时序数据库设计要求

- 单机入库量  $> 500w/min$
- 数据监控读取量  $> 50w * 22天 / min$
- 故障定位读取任意时间  $50w * 2天 / s$
- 支持多维度key :



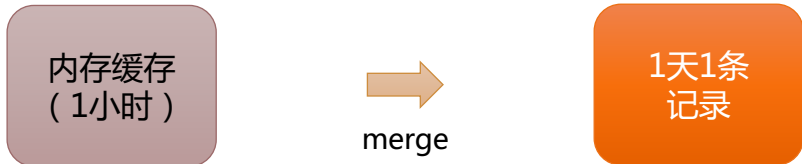


# 监控数据入库



GOPS2017  
Shanghai

- 一分钟一条记录，数据量过大。
- 先缓存一定时间的数据，再合并成一天一条记录。

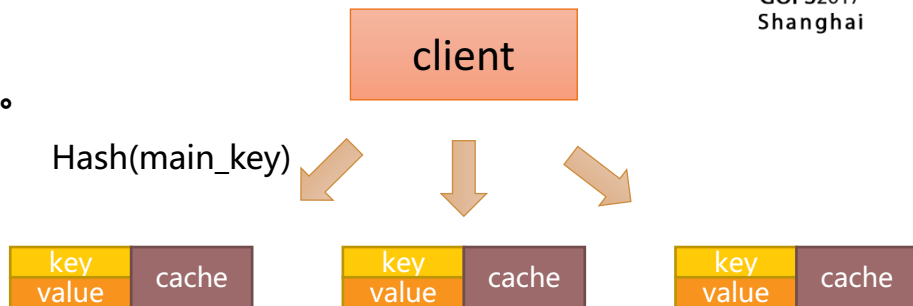




GOPS2017  
Shanghai

# 监控数据存储V1

- 自行实现Key-Value存储，Key常驻内存。
- Hash(main\_key)加速批量查询。



- Key使用二分查找，支持前置匹配查询：
- 单机读性能 > 100w/s。

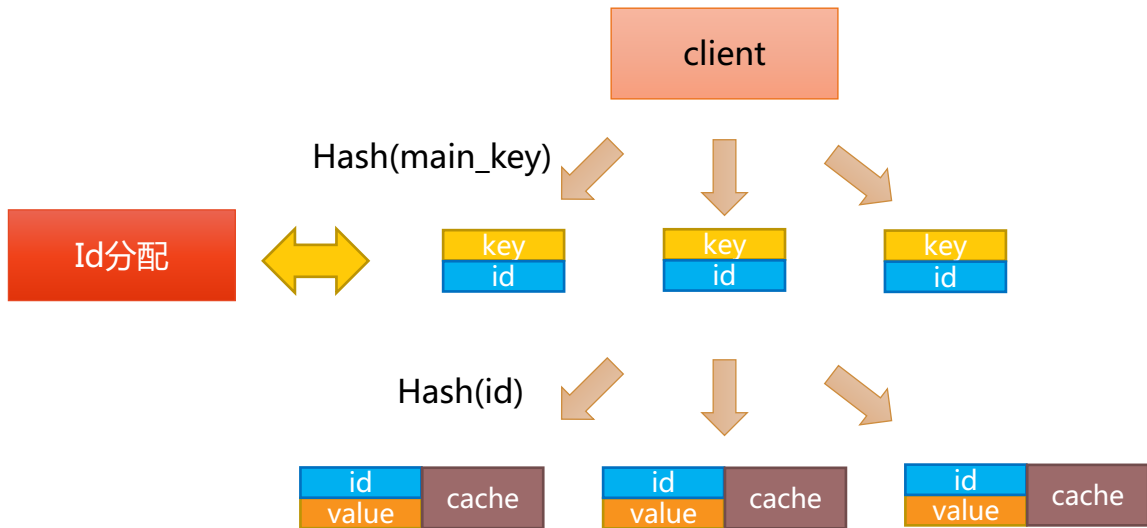
11	12	13	21	22	23	31	32	33
1*	1*	1*	2*	2*	2*	3*	3*	3*

- Hash(main\_key)数据极不均衡。
- 一天一条记录，key占内存稍多。

# 监控数据存储V2



GOPS2017  
Shanghai

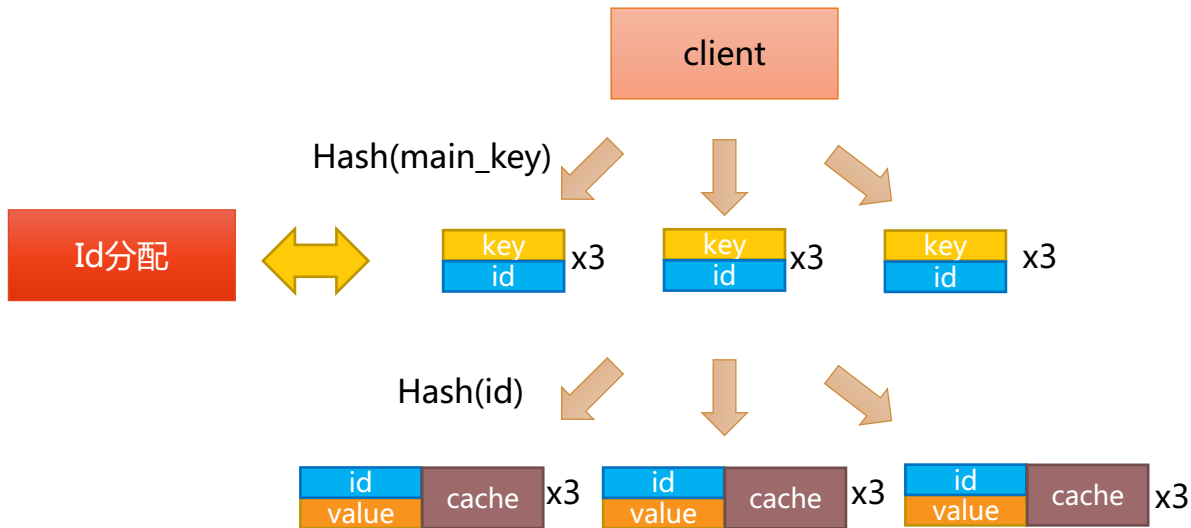


- Key-value拆分成key-id-value ( key-id、 id-value )。
- 通过id分配服务控制value数据均衡。
- key-id 7天重新分配一次，常驻内存。

# 监控数据存储V2 —— 数据容灾



GOPS2017  
Shanghai



- 使用**微信开源的phxpaxos**框架进行容灾。
- Phxpaxos框架的多master特性，让并发读性能更高。



GOPS2017  
Shanghai



# Thanks

高效运维社区  
开放运维联盟

荣誉出品