



GOPS2017
Shanghai



GOPS

全球运维大会

2017

上海站

指导单位:



主办单位:



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

百亿访问量的监控平台如何炼成

——罗马 (Roma)

WiFi万能钥匙 李春旭 高级架构师



自我介绍：李春旭

- ✓ 十年互联网研发经验，喜欢折腾技术，曾供职于快钱、阿里巴巴、平安健康等公司，专注于以下领域：分布式监控平台、调用链跟踪平台、统一日志平台、应用性能管理、稳定性保障体系建设、私有云及混合云等
- ✓ 2016年加入WiFi万能钥匙，目前主要负责公司的全链路监控体系、稳定性保障体系相关项目建设



Interested in : Monitoring、Logging、Tracing、APM、DevOps、AIOps、Distributed System.....

1

Overview



背景介绍



架构设计



最佳实践

产品介绍——WiFi万能钥匙·最安全的免费上网服务

- ✓ 帮助更多无法上网的人接入网络
- ✓ 用户连接体验->基于大数据场景化
- ✓ 安全上网
 - 安全云感知：提前预知风险
 - 安全隧道加密：实时保障安全
 - 千万级安全险：快速安全理赔
 -



产品数据——WiFi万能钥匙·数据规模



用户规模仅次于微信和QQ，国内最大的移动互联网工具软件

以上公开数据截止到2016年6月份

用户体验——可用性·可靠性·故障率·量化

Availability	Downtime per year	Downtime per quarter	Downtime per month	Downtime per week	Downtime per day	Downtime per hour
90%	36.52 days	9.13 days	3.04 days	16.80 hours	2.40 hours	6.00 minutes
95%	18.26 days	4.57 days	1.52 days	8.40 hours	1.20 hours	3.00 minutes
99%	3.65 days	21.91 hours	7.30 hours	1.68 hours	14.40 minutes	36.00 seconds
99.5%	1.83 days	10.96 hours	3.65 hours	50.40 minutes	7.20 minutes	18.00 seconds
99.9%	8.77 hours	2.19 hours	43.83 minutes	10.08 minutes	1.44 minutes	3.60 seconds
99.95%	4.38 hours	1.10 hours	21.91 minutes	5.04 minutes	43.20 seconds	1.80 seconds
99.99%	52.59 minutes	13.15 minutes	4.38 minutes	1.01 minutes	8.64 seconds	0.36 seconds
99.999%	5.26 minutes	1.31 minutes	26.30 seconds	6.05 seconds	0.86 seconds	0.04 seconds

数据来源：<https://dastergon.github.io/availability-calculator>

品牌

股价

监控现状——WiFi万能钥匙•监控现状与问题

功能特性	Zabbix	开源框架	自研框架
健康检测	端口检测	不支持	进程检测
应用监控	不支持	支持	支持（URL、Spring、Exception、SQL等多维指标）
系统监控	支持	不支持	支持
业务监控	支持（统计维度单一）	支持（统计维度单一）	支持（单维、多维，多种聚合计算方式）
中间件监控	不支持	不支持	支持
调用链跟踪	不支持	支持（改造成本较高、非本地化）	支持（适配公司内部的中件体系）
日志搜索	不支持	不支持	支持
多渠道	支持	支持	支持（掌信、短信、邮件，容错、重试等）
多实例	不支持	不支持	支持（应用端口切分、便于定位问题）
多机房	支持	不支持	支持（单元化部署、数据集中化、带宽占用少等）
多语言	支持（HTTP）	不支持	支持（HTTP、LOG）
易用性	差（用户体验较差、不友好）	中（数据丢失、侵入性强）	良（低侵入性、有服务树、易用性强）
.....			

监控目标——监控平台解决的问题

外



应用外：

计算机硬件、网络设备、操作系统、计算机软件

内



应用内：

应用性能、应用异常、业务打点、日志输出等

间



应用间：

链路分析、依赖关系、性能分析、容量规划等

业务数据

前端/应用/中间件

计算机软件

操作系统

硬件/网络设备

➤➤➤ 一体化、自动化、智能化、全链路

参考案例——国内外主流的技术方案

Dapper, Zipkin, Pinpoint, Hydra, Mtrace, NewRelic , OneAPM

Cacti, Nagios, Ganglia, Zabbix

CAL, CAT, Open-Falcon, Mercury

Roma

Comsat, Hubbler, Dragoon, Xflush, Alimonitor, EagleEye

Observability : Monitoring、Logging、Tracing

- ✓ 背景介绍
- ✓ 架构设计
- ✓ 链路跟踪

2

Architecture



架构原理·Data Flow



架构原则

性能影响

1

低侵入性

2

无内部依赖

3

单元化部署

4

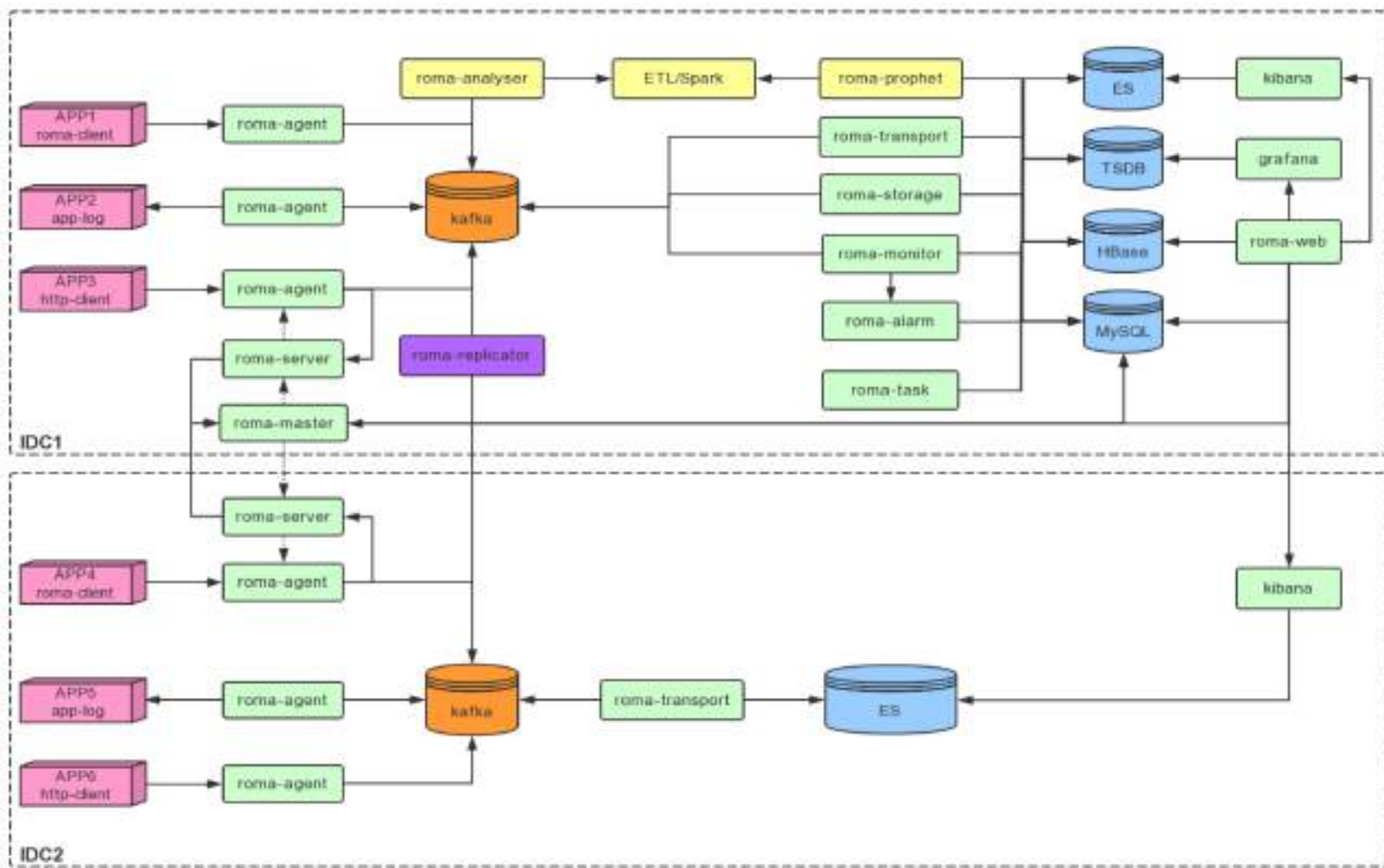
数据集中化

5

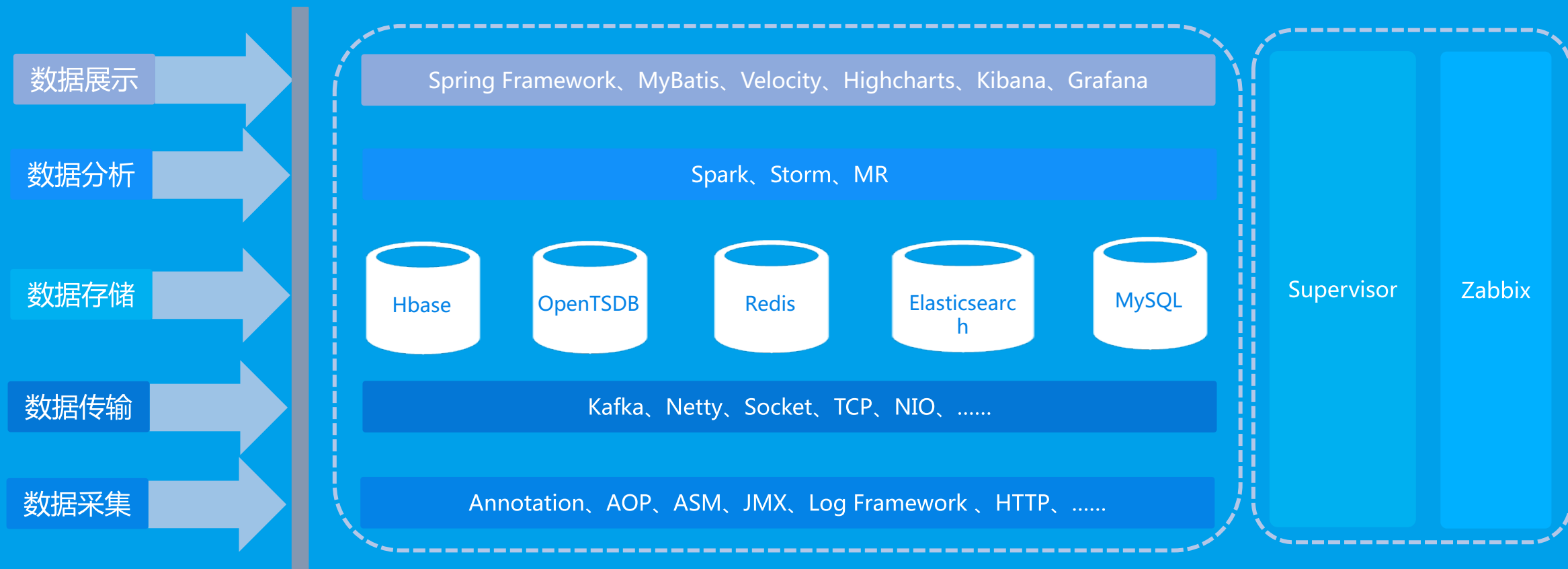
业务架构



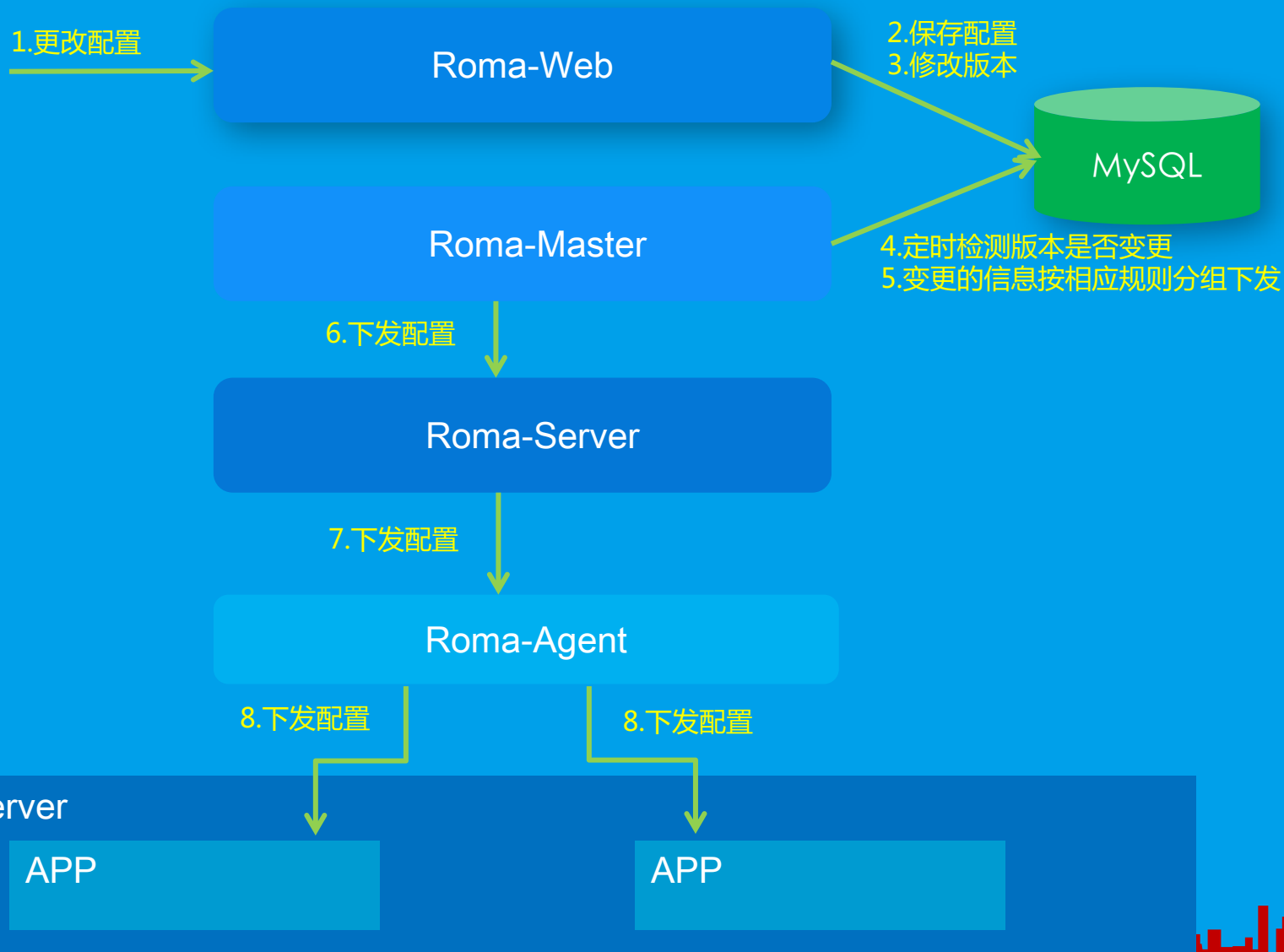
应用架构



技术架构



配置下发

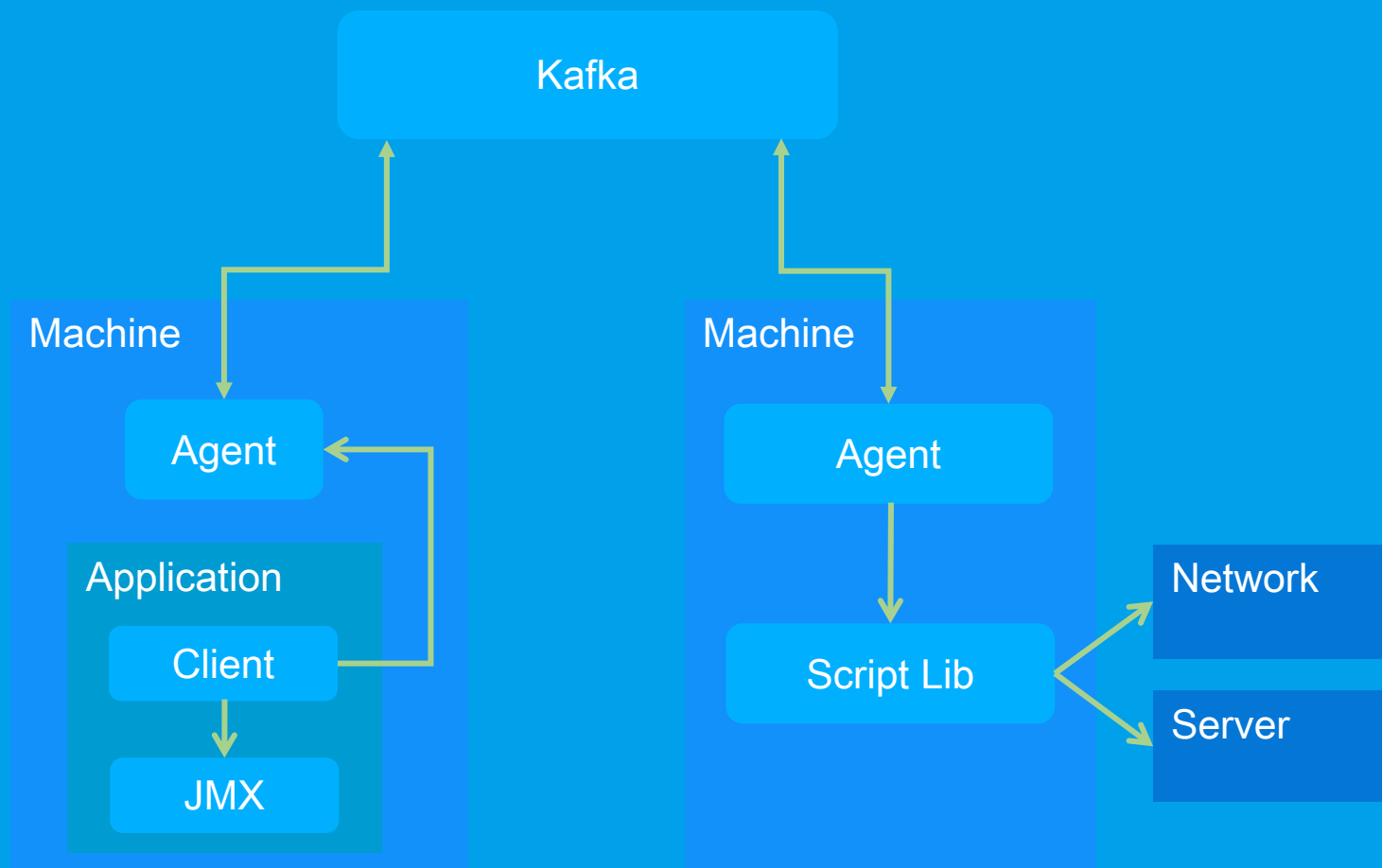


数据采集·对比分析

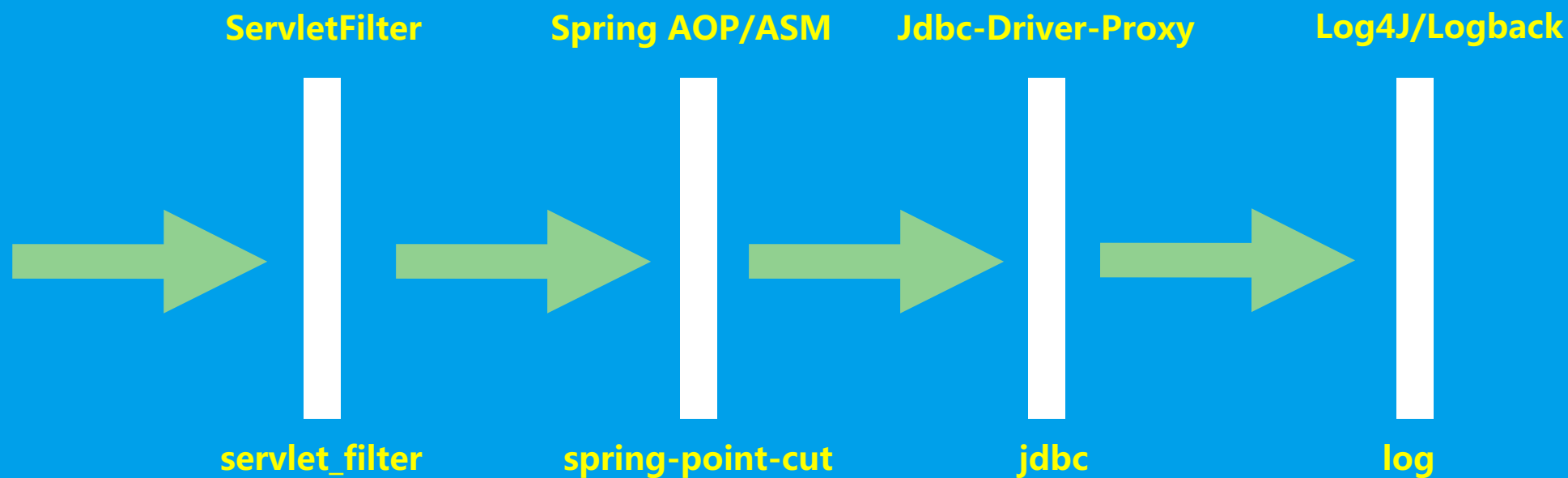
采集方式	易用性	复杂度	侵入性	可靠性	性能影响	采集策略	指标粒度	部署策略
主动采集	中	中	高	高	低	全量 / 采样	多	分布式
被动采集	低	低	中	低	低	全量	少	集中式
日志输出	高	中	低	中	中	全量 / 采样	中	分布式
探针技术	高	高	低	低	高	采样	中	集中式
旁路监听	高	高	低	低	中	采样	少	集中式
.....								

关于各种数据采集方式的对比分析说明

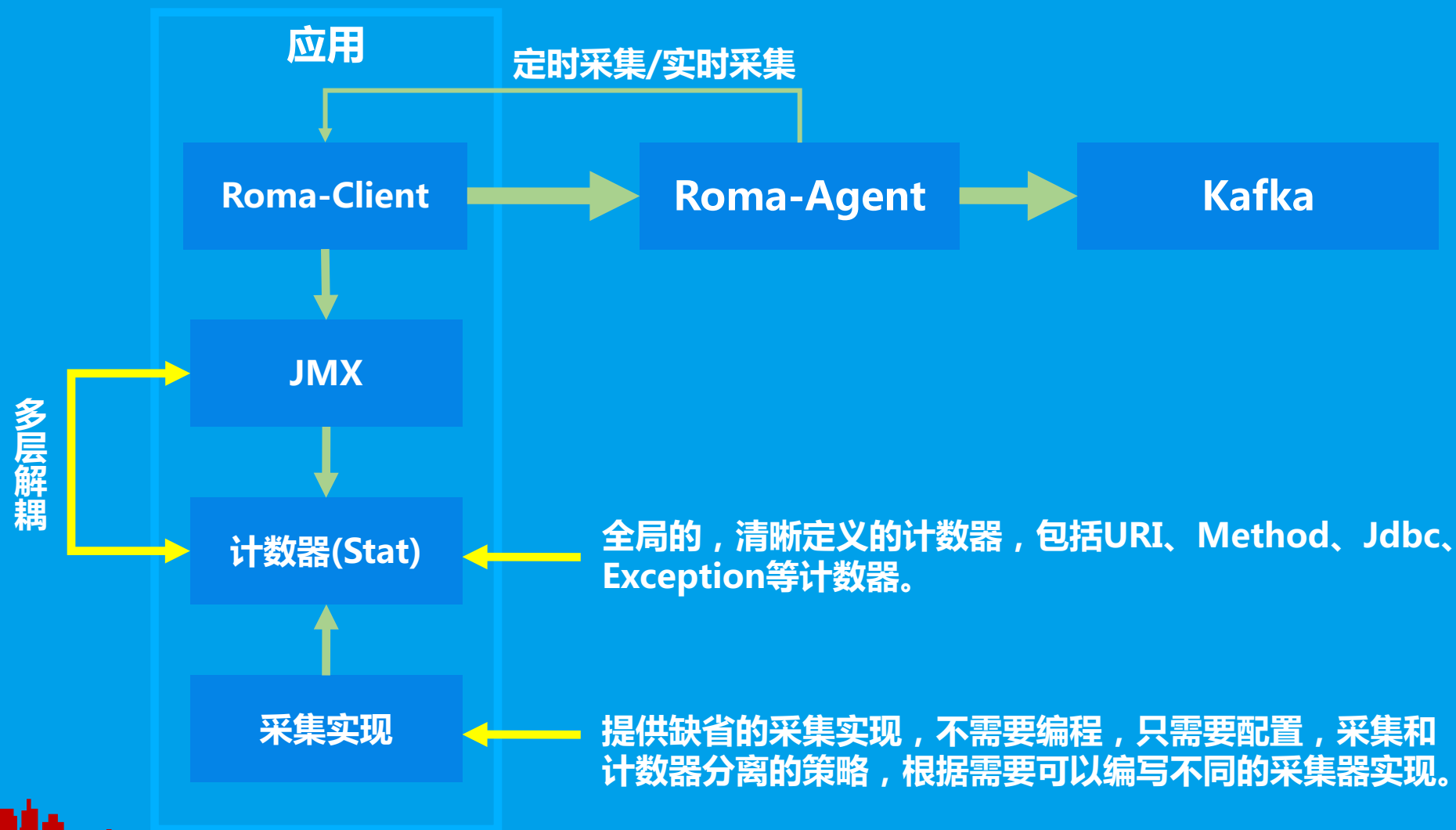
数据采集·罗马实现



应用监控·Pipeline



数据采集·应用监控



数据传输·网络协议

- ✓ TCP+JSON+NIO
- ✓ Kafka
- ✓ Compression : Gzip
- ✓ TLV : Type Length Value
- ✓ Type :
 - 0 : Raw Binary
 - 1 : UTF-8 JSON TEXT
 - 2 : XML

✓ 消息格式 :

```
[  
  #header  
  {  
    "S": number,#序号  
    "T": string#消息类型  
  },  
  #body  
  {  
  }  
]
```

✓ 消息内容 :

```
[  
  {  
    "S": 1,  
    "T": "GetAttribute"  
  },  
  {  
    "OBJ_NAME":  
    "com.wifi.roma:type=JdbcStats",  
    "ATTRS": [SqlList],  
    "OPTS": [reset=true list_unpack_field=SqlList]  
  }  
]
```

数据同步

✓ 需求背景

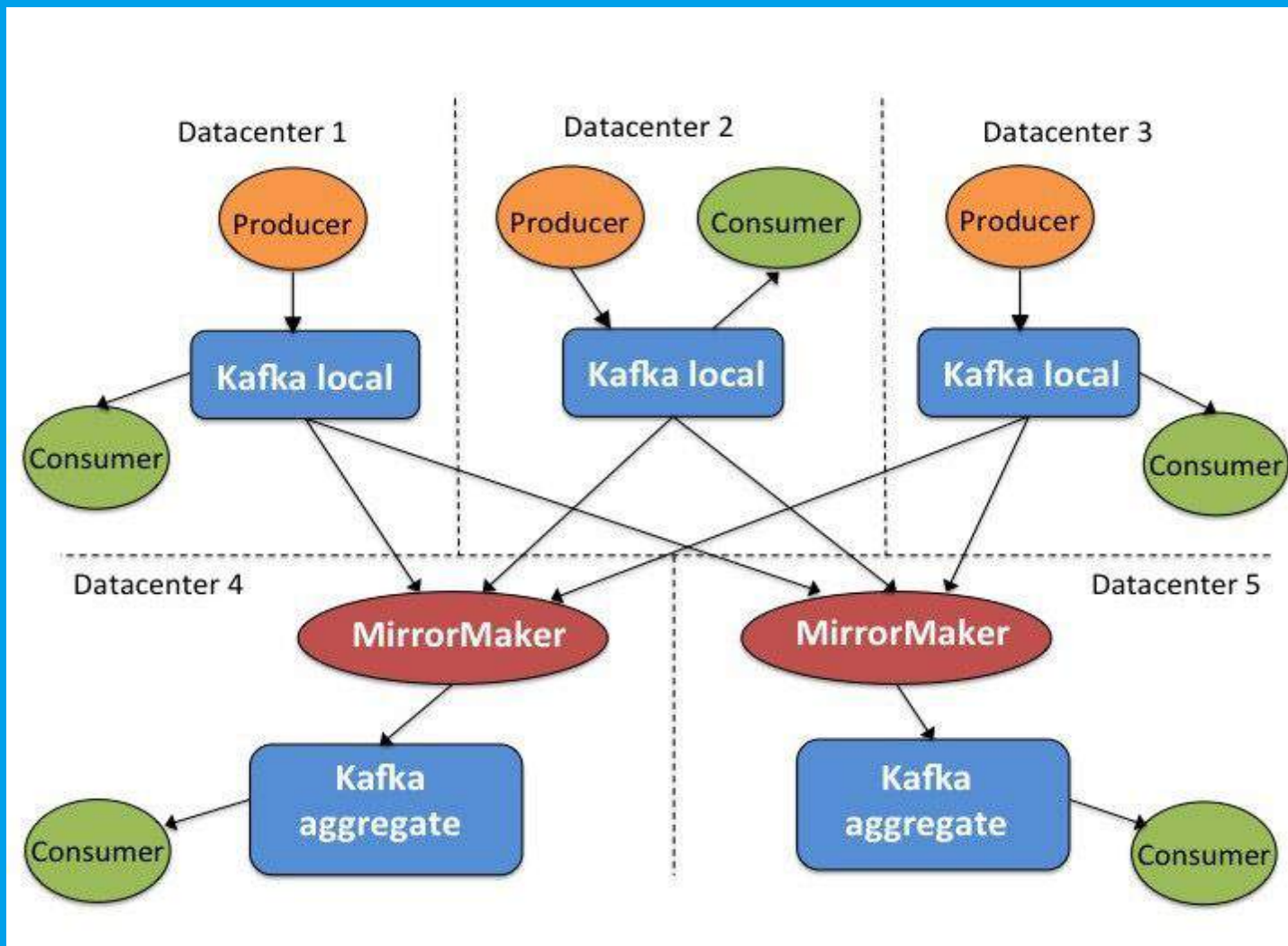
- 应用部署在多个IDC机房
- 单元化部署、数据集中化
- 不做机房间容灾、多活等

✓ 常见方案

- Direct Cluster
- active/passive、active/active
- MirrorMaker、uReplicator

✓ 对比分析

- 网络延迟、生产者及消费者耗时
- 集群之间Offsets管理（一致性）
- Topic、Partitions、Mirrormaker



数据分析

- ✓ Spark
- ✓ HBase
- ✓ Elasticsearch
- ✓ 数据异常
 - 采集异常
 - 网络不通
 - 节点宕机
 - 硬件故障
 -

过期数据

- 1) 专门异常处理通道、不再触发报警；
- 2) 数据聚合处理异常、制定处理规则；
- 3) 数据归档处理异常、归档数据修复；
- 4) 指标数据的准确性以及完整性缺失；

数据追踪

- 1) 用户能够方便的了解自己配置的监控项的运行状况；
- 2) 如果运行异常，用户可以在页面上查看错误类型和返回的错误信息；
- 3) 异常日志会被保留，用户可以通过页面查询异常日志；

数据聚合

- 1) 忽略未上传的数据，将已有的数据聚合；
- 2) 舍弃：丢弃数据，在该时间点不做聚合；
- 3) 重试：不是所有监控项都可以进行重试；

应对策略

- 1) 监控机和被监控机可以缓存部分数据；
- 2) 延时上报的数据走专门的数据修复流程，不影响主流程的数据，不报警；
- 3) 对于无状态的监控项，设置Retry机制；
- 4) 舍弃、重试、延迟等待；

数据存储

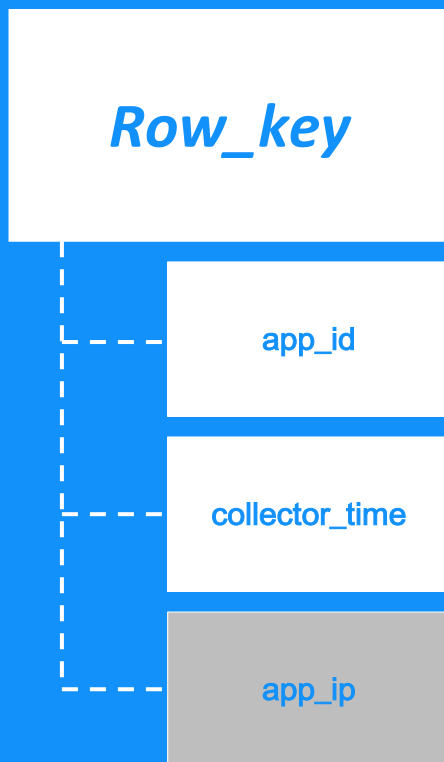
✓ Elasticsearch

- 应用日志：7天
- 链路日志：7天
- 业务日志：30天

✓ OpenTSDB

✓ HBase

- 最新数据：当前数据
- 明细数据：一个月
- 分钟数据：三个月
- 小时归档：六个月
- 按天归档：一年



1、md5(sql_name)

2、md5(web_uri)

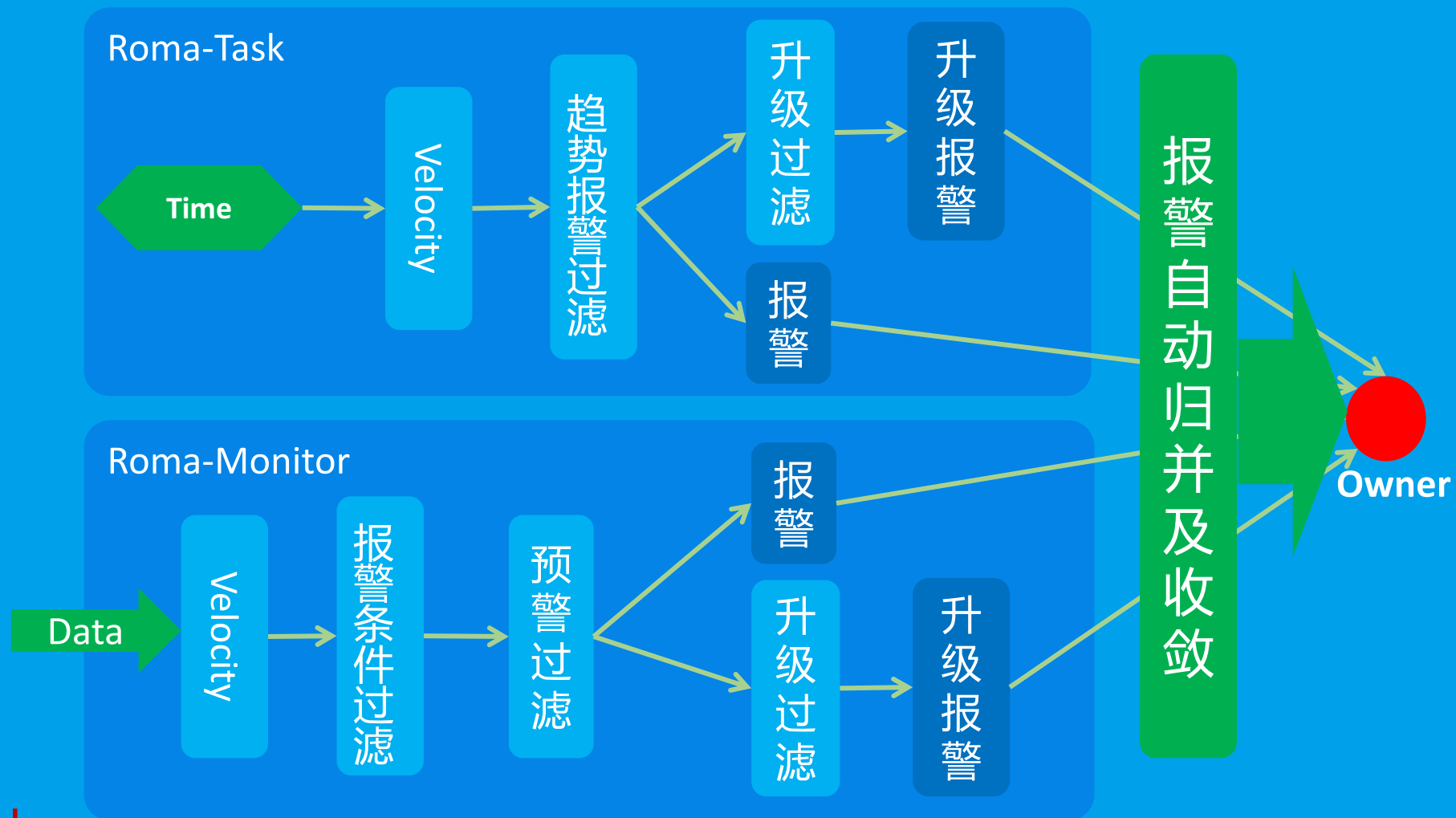
3、md5(class_name + method)

4、md5(interface_name+method)

5、.....

聚合计算：min、max、sum、count

报警处理



性能测试·网关压测

对比分析	TPS	性能影响	是否满足期望
无Roma、无XXX	14920		
无Roma、有XXX	13540	9.24%	否
有Roma、无XXX	14207	4.77%	是
有Roma、有XXX	10309	30%	否

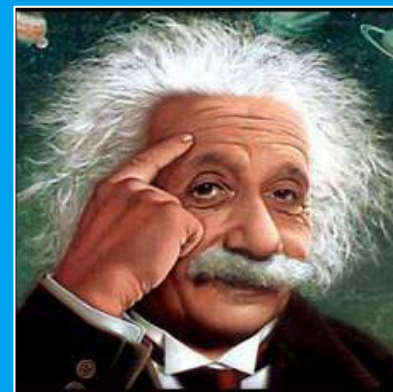
Roma对应用的性能影响：1%~5%

使用Roma、下线XXX：TPS上升5%、CPU节省5%

架构挑战

问题思考

- ✓ 如何保证分布在多个IDC机房的服务器都能正常分配注册？
- ✓ 分布式网络环境中，Roma-Agent如何与Roma-Server重连？
- ✓ 控制台Roma-Web配置修改后，如何快速准确地下发至Roma-Agent？
- ✓ 如何检测配置下发内容的正确性？如何应对配置下发过程中的网络故障？
- ✓ 海量的监控指标数据如何保证高性能存储及数据查询高效性？
- ✓ 如何保证分布式环境下海量日志数据的采集、存储、检索等？
- ✓ 如何提高报警的准确性、及时性？如何减少漏报、误报等相关问题？
- ✓ 如何实现多机房成千上万台机器上Roma-Agent的自动化安装部署？
- ✓



3

Best Practices



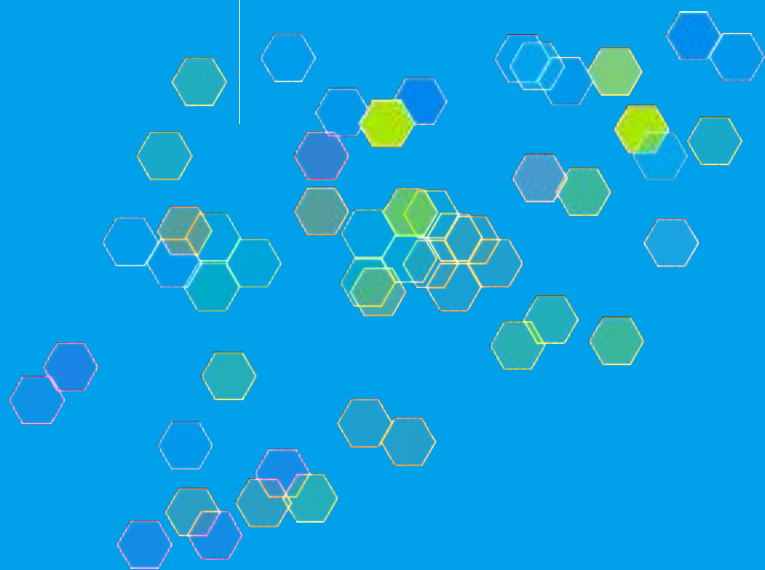
背景介绍



架构设计



最佳实践



调用链跟踪

日趋复杂的分布式服务体系

- ✓ 分布式远程服务框架：Motan
- ✓ 分布式消息服务框架：RocketMQ
- ✓ 分布式任务调度框架：Blanc
- ✓ 分布式网关服务框架：Alps
- ✓ 分布式搜索引擎框架：Elasticsearch
- ✓ 分布式配置中心框架：taihe
- ✓ 分布式分库分表框架：Cobar
- ✓ 分布式缓存服务框架：Redis
- ✓

埋点字段	字段描述
eventTime	请求开始时间
traceId	全局上下文ID
spanId	用于标示调用层次关系
type	中间件调用类型标示
hostIp	请求主机IP
hostName	请求主机名称
serviceName	调用的服务名
methodName	调用的方法名
remoteIp	对端IP地址
spanTime	处理时间
requestSize	客户端请求的大小
responseSize	服务端相应的大小
resultCode	调用结果
errorMsg	调用异常信息
extInfo	用于附加的扩展字段

中间件埋点注意事项

- ✓ 各中间件类型进行编码
- ✓ 中间件日志格式差异性
- ✓ 埋点日志的采样标识符
- ✓ 业务埋点日志透传规则：
 - 链路签名
 - 用户标识
 - 入口标识
 - 机房标识
 - 单元化路由
 -

调用链跟踪·埋点实现

✓ TraceId：调用上下文

- 关联一次请求相关的日志，全局唯一，在各个系统间传递。



- 端口号：单机多进程的应用使用
- 标志位：用于调试和标记

✓ SpanId

- 标识日志埋点顺序和嵌套关系，在各个系统间传递。
- 调用关系：同步 / 异步 / 一对多调用
- 表示方式：顺序编号、多级编号

✓ 异步调用：线程上下文切换

- 异步 IO：Send 和 Recv 不在同一线程
- 异步 servlet：业务逻辑在不同线程中切换执行
- 业务逻辑提交到其它线程处理时会丢失上下文

✓ 异步调用：解决方案

- 切换前保存、切换后还原上下文
- InheritableThreadLocal及封装库：不可变对象
- ThreadPoolExecutor及封装库：
beforeExecute、afterExecute、terminated
- Bytecode、ASM、JVMTI (Instrumentation)

调用链跟踪·数据输出·数据存储

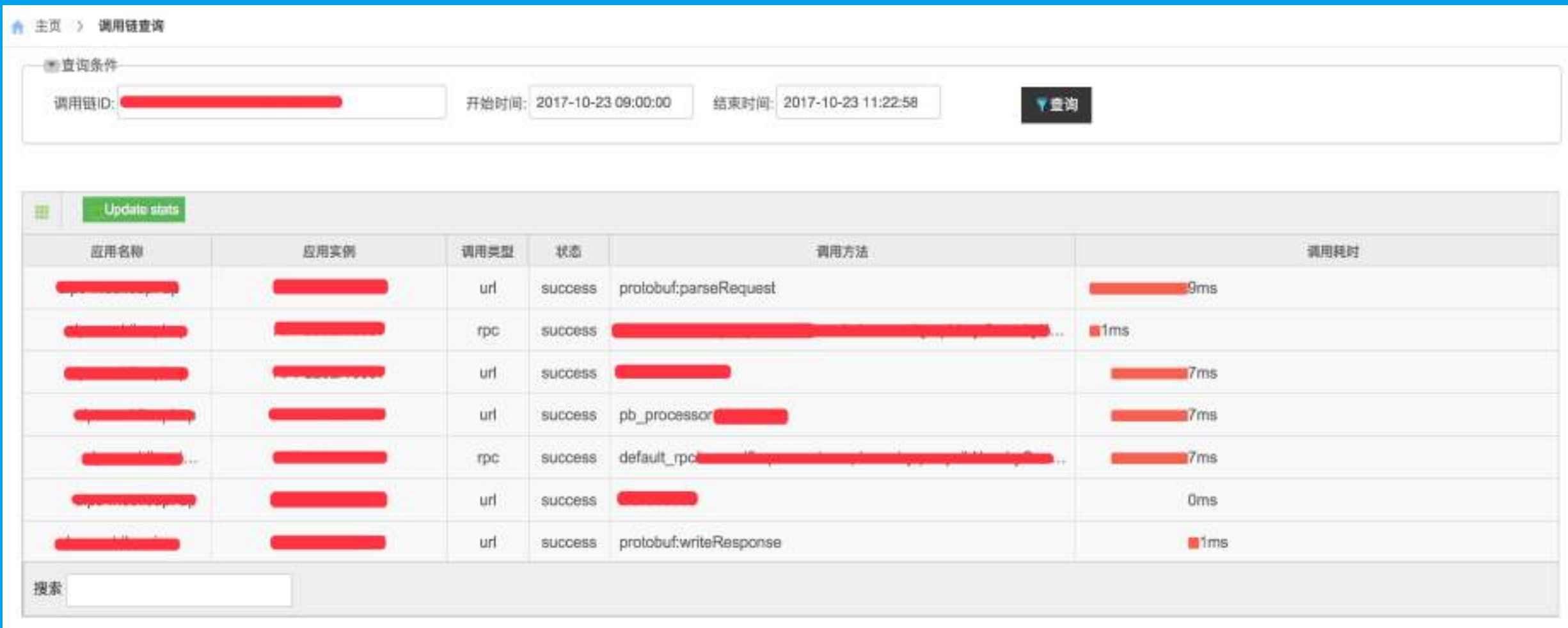
写日志面临的挑战

- ✓ 尽可能减少对业务线程的影响，降低系统消耗
- ✓ 每个网络请求至少1行日志，QPS 越高日志产生越快
- ✓ 解决方案：不进行日志输出、直接采样上报
 - 开关控制以及采样输出（入口采样策略）
 - 链路入口采样控制（调用次数及百分比采样）
 - 调用次数策略、百分比策略、时间间隔策略
 - URI数据采集与链路数据采样相融合（动态采样）
 - 链路数据直接上报（发送至roma-agent）

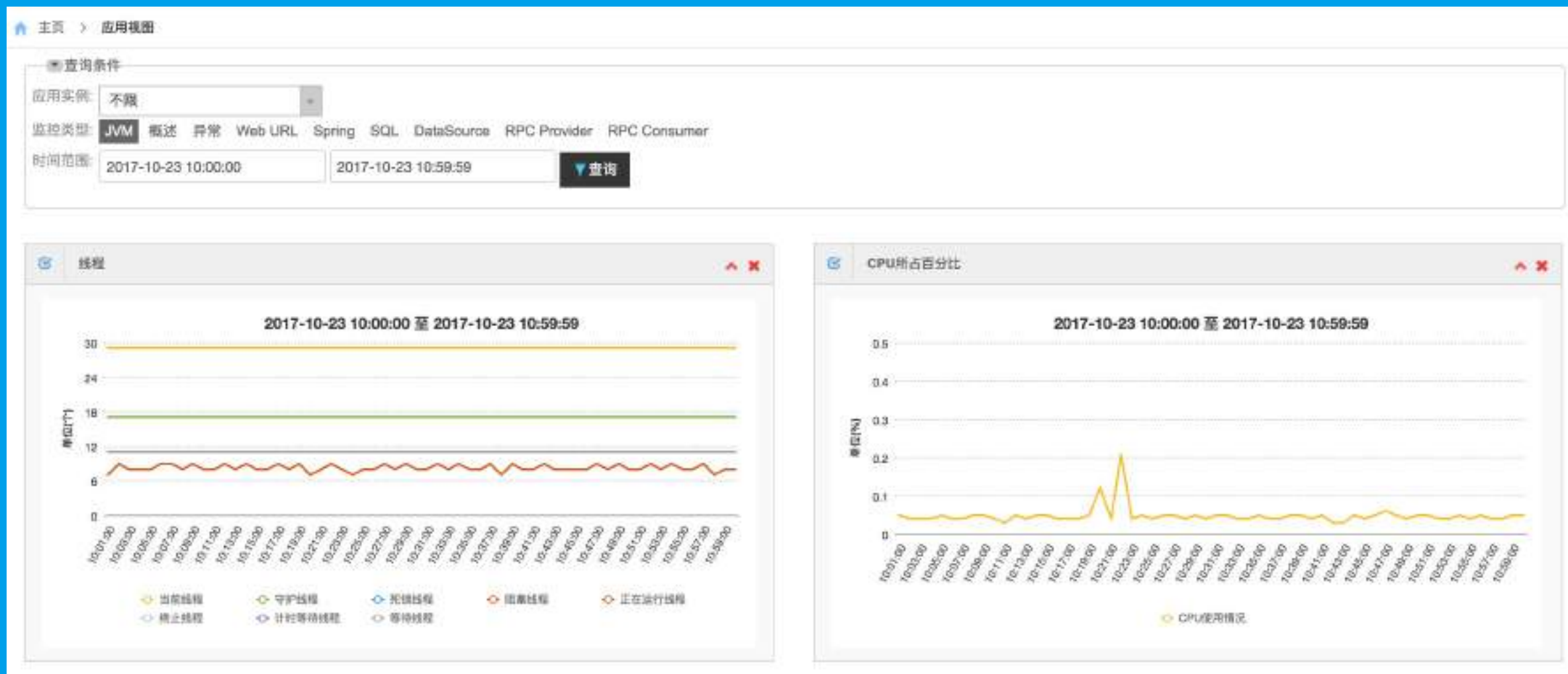
数据存储面临的挑战

- ✓ 调用链的指标数据分散在调用经过的各个服务器上
- ✓ 离线分析需要将同一条调用链的数据汇总在一起
- ✓ 按 TraceId 汇总链路指标数据
 - 基于Elasticsearch存储：用 TraceId 关联查询
- ✓ 调用链指标数据分析
 - 数据关联分析：应用监控、链路跟踪相结合
 - 数据可视化：基于ES查询链路结果数据展示

功能演示·调用链跟踪



功能演示·应用监控·JVM



功能演示·应用监控·异常

Update stats

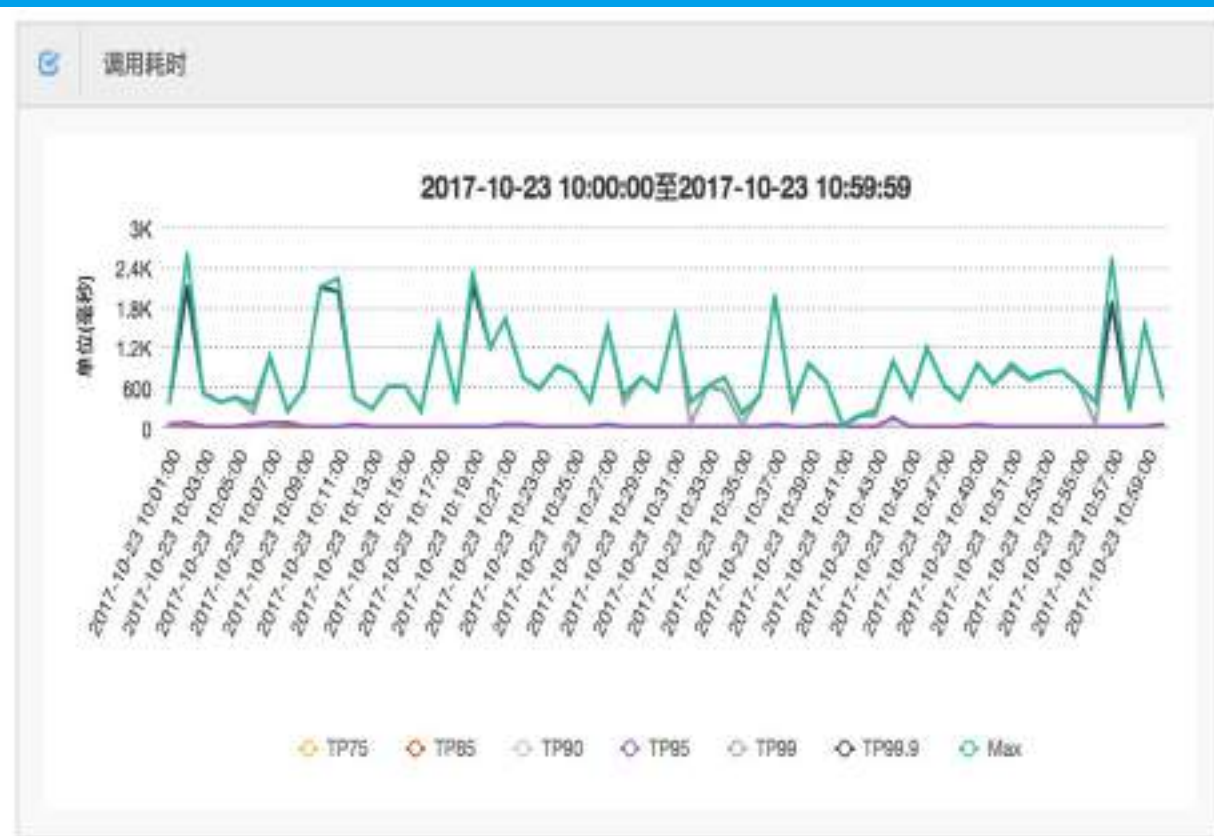
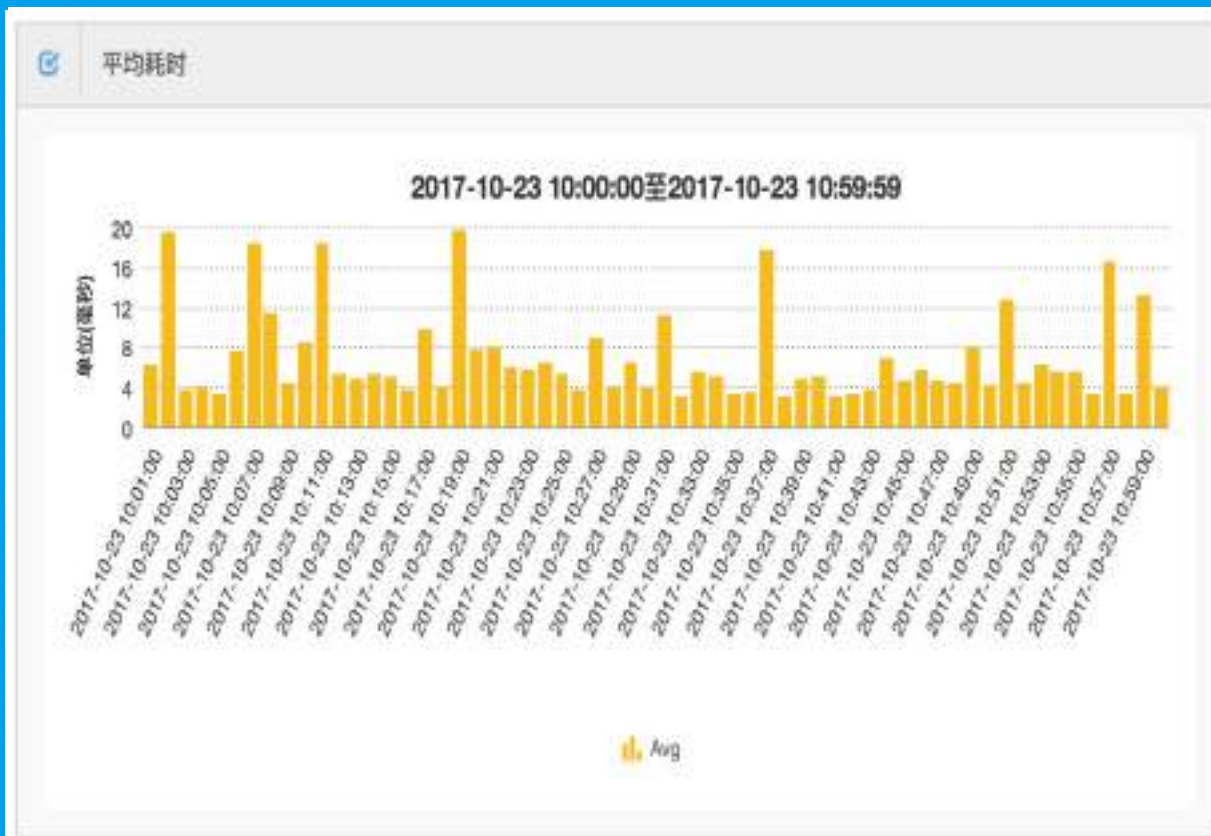
每页显示10条记录

异常类型	异常类	行号	异常时间	异常次数
java.lang.NumberFormatException	ExceptionScheduledTask.executeExceptionTask	33	2017-10-23 10:59:18	252
java.lang.NullPointerException	ExceptionScheduledTask.executeExceptionTask	42	2017-10-23 10:59:18	242

搜索

异常详情	
线程名称	pool-2-thread-9
MDC	null
异常信息	0 executeExceptionTask,listAdd:null
堆栈信息	<pre>java.lang.NullPointerException at [redacted].ExceptionScheduledTask.executeExceptionTask(ExceptionScheduledTask.j ava:40) at sun.reflect.GeneratedMethodAccessor129.invoke(Unknown Source) at sun.reflect.DelegatingMethodAccessorImpl.invoke(Dele gatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:498) at org.springframework.scheduling.support. ScheduledMethodRunnable.run(ScheduledMethodRunnable.java:65) at org.springframework.scheduling.support.DelegatingErrorHandlingRun nable.run(DelegatingErrorHandlingRunnable.java:54) at java.util.concurrent.Executors\$RunnableAdapter.call(Executors.java:511) at</pre>

功能演示·应用监控·Web URL



功能演示·日志搜索

日志文件读取配置 - (应用名称: [REDACTED])

文件配置组

文件配置

类型

[REDACTED]

文件

[REDACTED]

添加文件

是否从头开始读取

否

是否为多行日志

是

多行分隔符匹配正则表达式

^201,.*

添加文件配置

更新

返回



感谢聆听

Thank you for your listening

End



GOPS2017
Shanghai



Thanks

高效运维社区
开放运维联盟

荣誉出品



GOPS2017
Shanghai



想第一时间看到
高效运维社区公众号
的好文章吗？

请打开高效运维社区公众号，点击右上角小人，如右侧所示设置就好

