WOTD

51CTO

全球 软件开发技术峰会

World Of Tech 2017

2017年12月1日-2日 • 深圳中洲万豪酒店

DEVELOPMENT

WOTD

51CTO

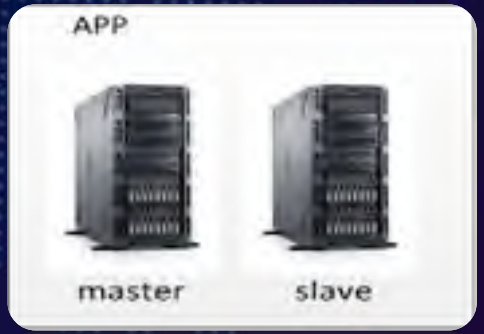
日志平台案例

杨津萍

目录

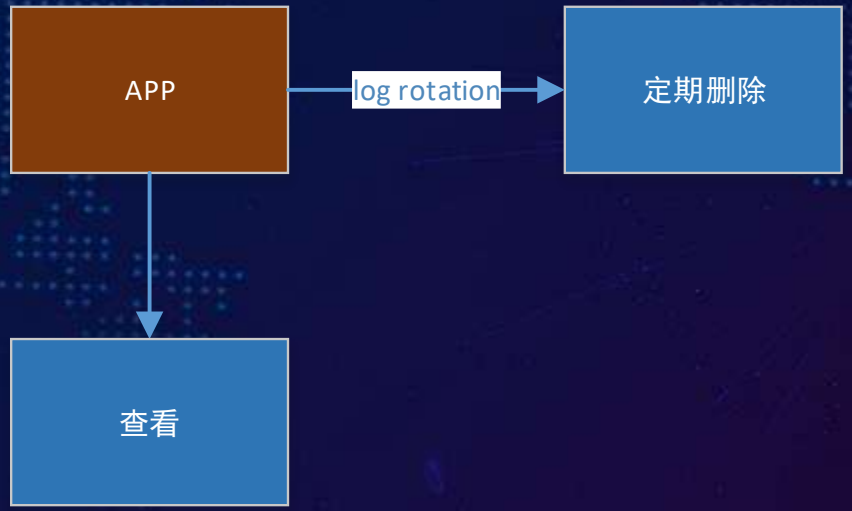
应用场景

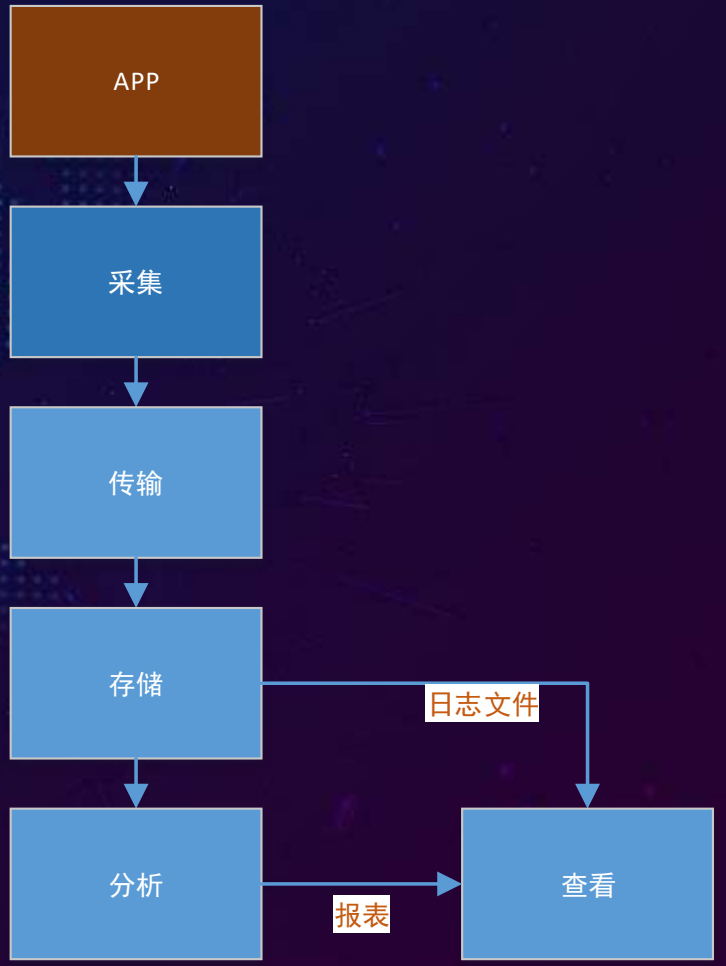
01 简单应用



02 复杂应用

日志业务流程





日志的数据场景

场景	实时	准实时	离线
Error/Exception	一级应用	其他	
分析	一级应用		其他应用
追溯			所有应用

日志-基准

优化思路

基本优化

- 内存
  - 如内存分配，垃圾回收，缓存
- 网络
  - 传输协议（如压缩，序列化），策略
- CPU
  - 如多线程，提高利用率和负载
- 磁盘
  - 如 减少寻道次数，采用ssd等
- 系统/容器
  - 修改句柄，关闭无用服务

平台扩展

- 做加减法
  - 在系统加分布式缓存等
- 纵向扩展
  - 单机扩展磁盘，cpu等
- 横向扩展
  - 分布式，集群

数据分治

- 数据分类
  - 数据走不同的数据链路
- 数据分级
  - 某类数据，如info类直接丢掉
- 数据热点
  - 打散热点

系统降级

- 业务分级
  - 黄金链路

目录

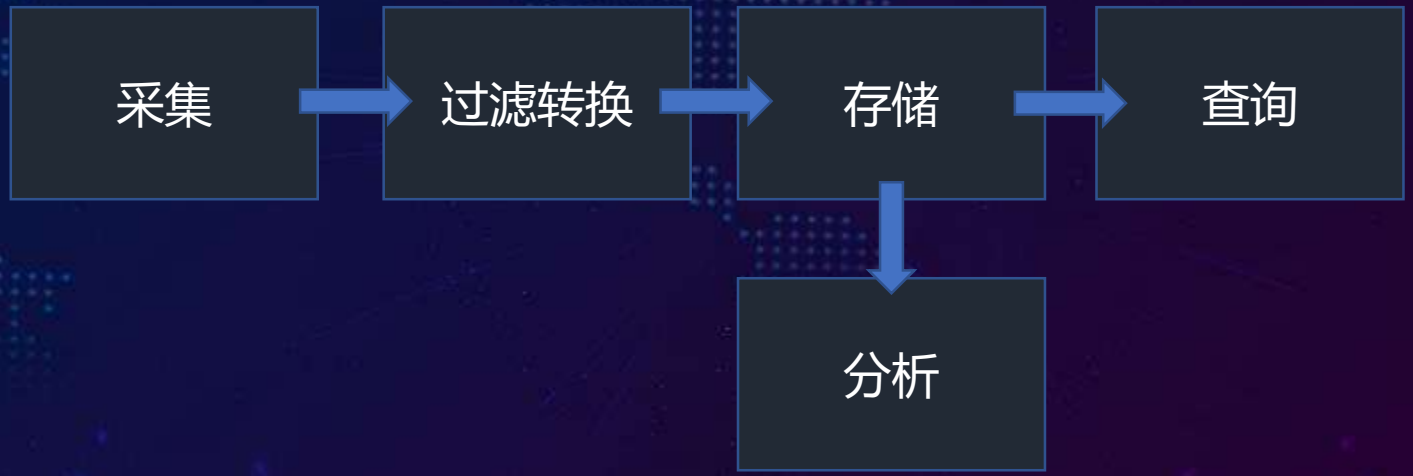
案例背景

01 日志量大 – 百亿/天

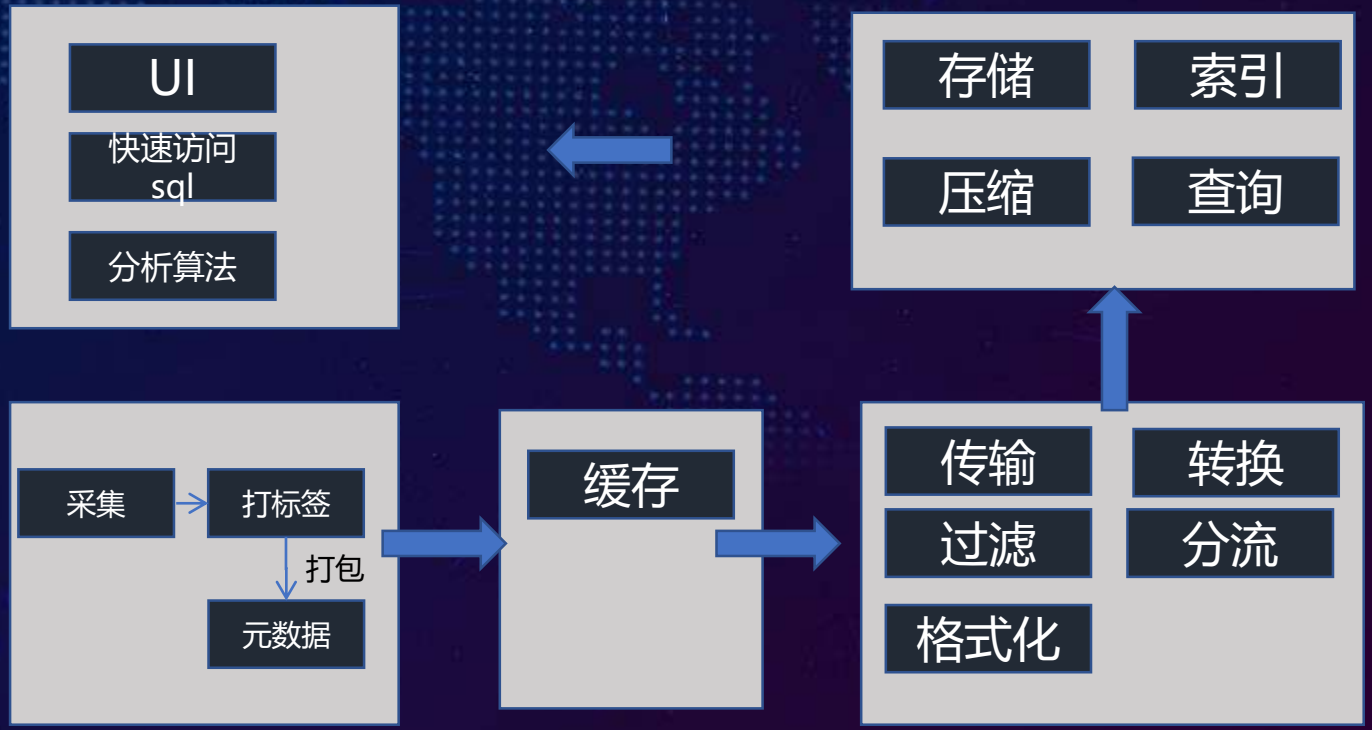
02 生产环境隔离 – 不能直接查看数据

03 代理资源限制 – 日志采集占用资源不能超过一个核

一级业务架构

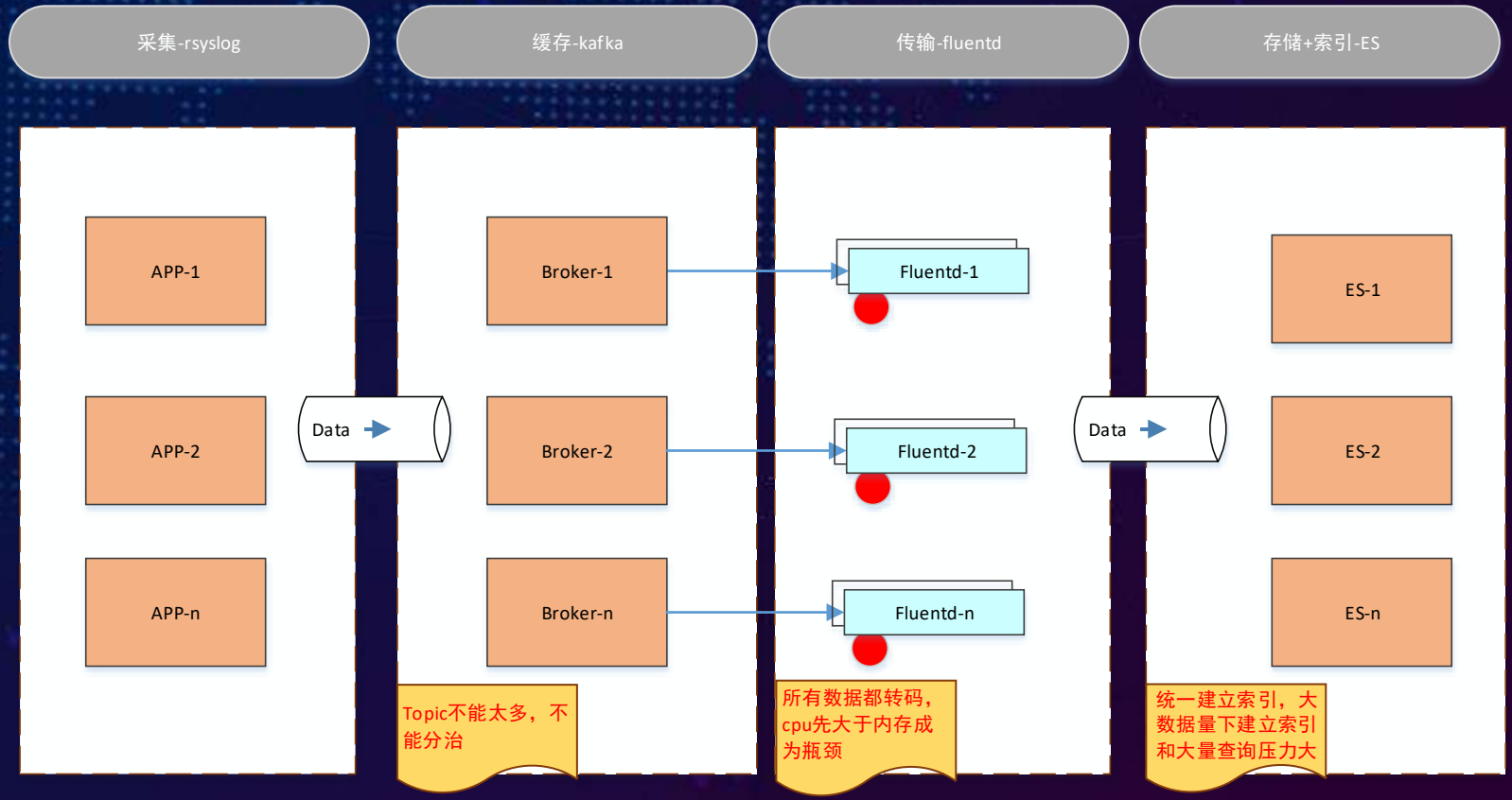


二级业务架构



一级技术架构

二级技术架构


采集-rsyslog
缓存-kafka
传输-fluentd
存储+索引-ES
APP-1
APP-2
APP-n
Data
Broker-1
Broker-2
Broker-n
Fluentd-1
Fluentd-2
Fluentd-n
Data
ES-1
ES-2
ES-n
Topic不能太多，不能分治
所有数据都转码，cpu先大于内存成为瓶颈
统一建立索引，大数据量下建立索引和大量查询压力大

Rsyslog

1. 资源占用少

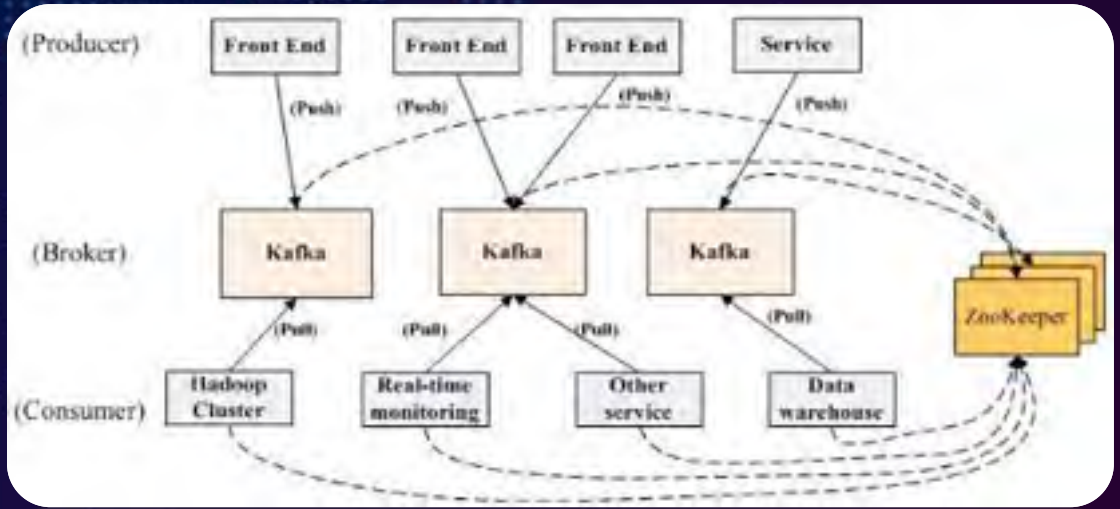
2.可以添加规则，如时间

候选	Logstash forwarder	Fluentd	Heka	Rsyslog
写入 2048 event/s 时cpu%	35 (max 366)	7.9 (max 8.9)	10.9 (max 12.9)	2.7 (max 3.3)
写入 2048 event/s 时mem%	1.6	1.4	0.3	0.0
写入 2048 event/s 时发送性能 (event/s)	1869	1942	1942	1950
写入 16384 event/s 时cpu%	210 (max 364)	50 (max 59)	70 (max 81.3)	18.0 (max 19.3)
写入 16384 event/s 时mem%	2.2	1.2	0.3	0.0
写入 16384 event/s 时发送性能 (event/s)	9524	13333	13333	14012
写入 32768 event/s 时cpu%	250 (max 355)	100 (max 102)	130 (max 166.2)	32.2 (max 32.9)
写入 32768 event/s 时mem%	2.4	1.8	0.3	0.0
写入 32768 event/s 时发送性能	9524	25000	25000	27128
备注	接收端为 Logstash null 输出	接收端为TCP端口	接收端为TCP端口	接收端为TCP并写入ramfs
实现语言	Go	C + CRuby	Go + Lua	C
接收端失效后行为	事件堆积，内存持续增长	缓存到内存（或磁盘），队列满后暂停发送，记录位置（文件输入）	缓存到磁盘，到达上限后暂停发送，内存不变	缓存到内存，队列满后再缓存到磁盘，再满丢弃输入（优先级可配），暂定发送，内存不变，CPU 略增
日志轮转后行为	同时打开新、旧文件，fd 数目持续增长（Issue #308)	旧文件保持打开若干秒（可配置）然后关闭	丢弃旧文件，打开新文件	同时打开新、旧文件，fd 数目持续增长（Issue #488 已修复)

kafka

1.顺序存储，快速存取，有topic

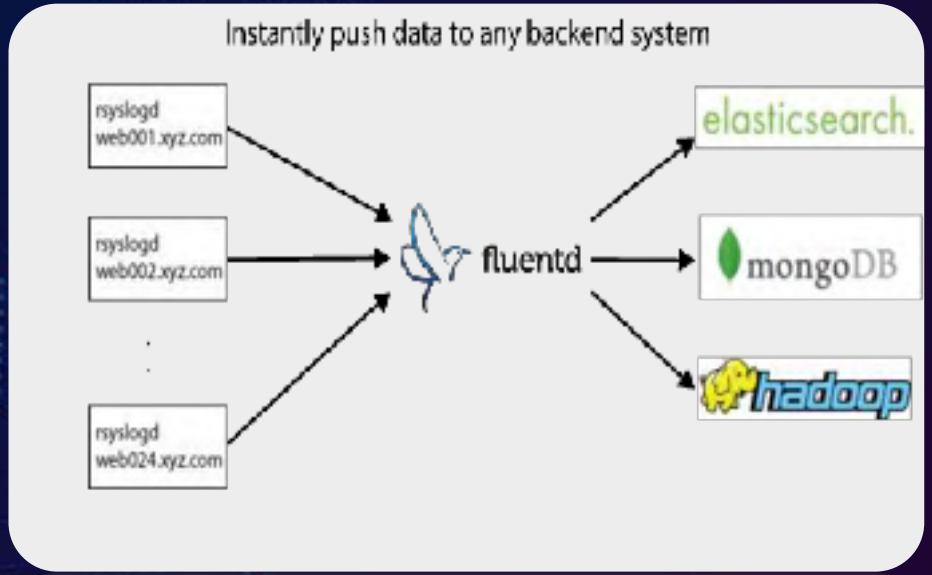
2.增加一次缓存，防止日志丢失

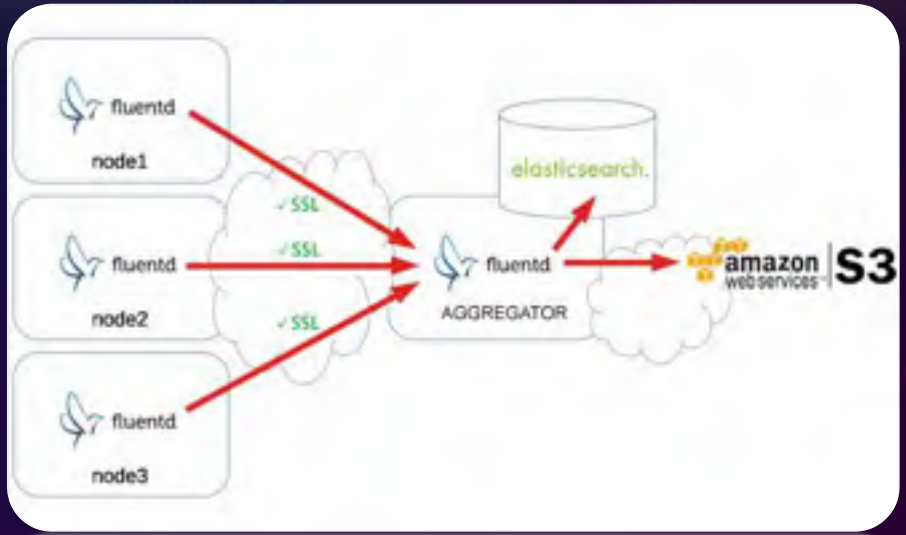


fluentd

1.负责传输过滤

2.灵活搭配，可以与rsyslog直接整合





Elasticsearch2.2+kibana

1.快速索引非结构化数据

2.平行扩展

问题定位与解决

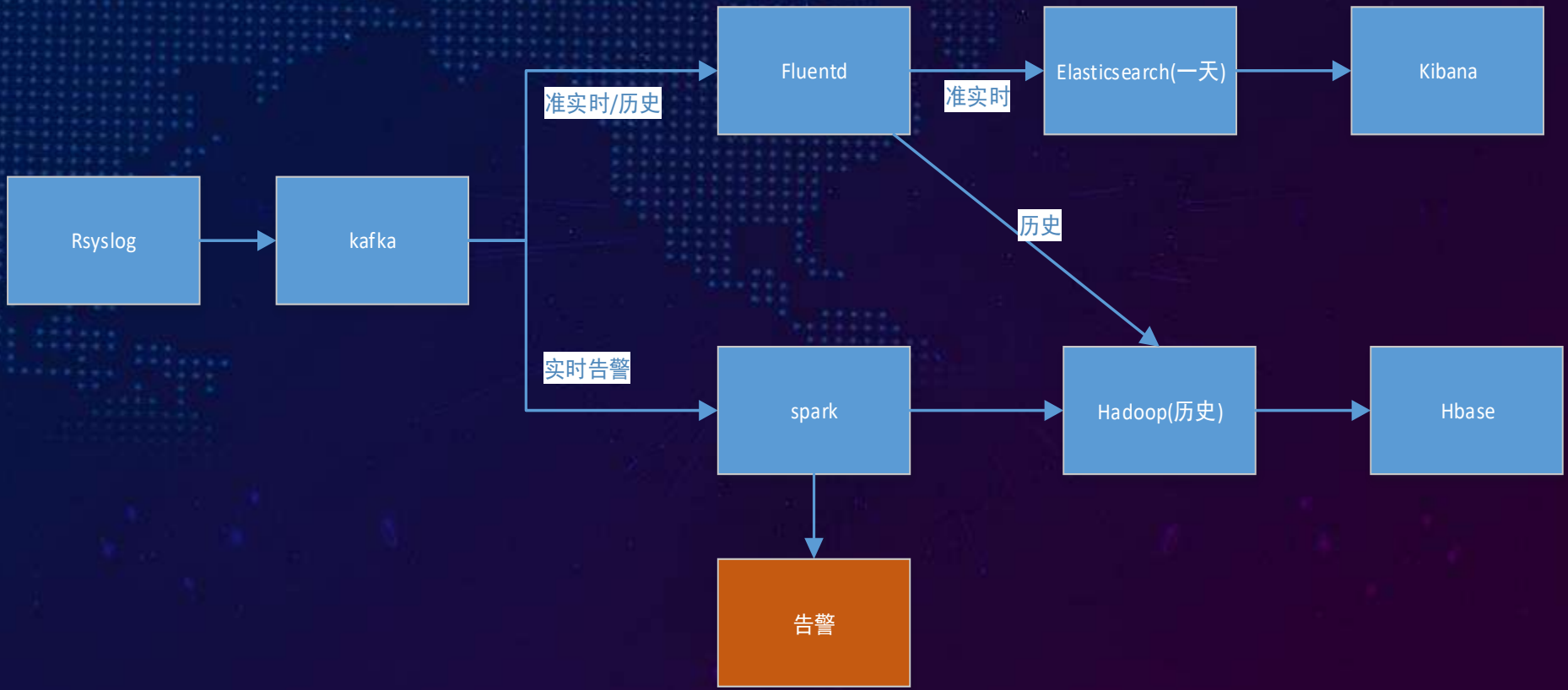
01 全部写入Kafka，所有数据都要走规则链

02 Fluentd的host轮询机制 造成高水位频发

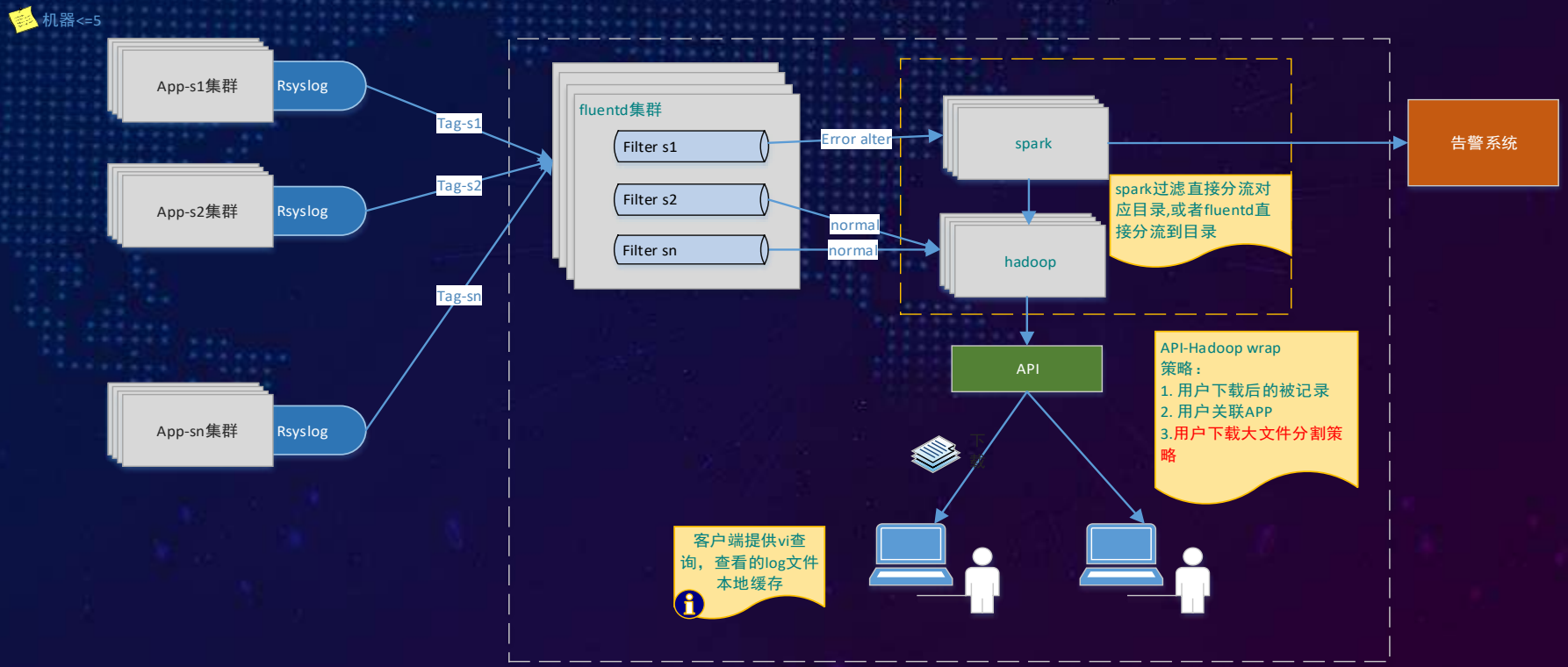
03 ES存储用到raid0, 存储时间过长，成本过高，找替代方案

04 实时数据达到写入写出平衡

改造一 降存储


Rsyslog
kafka
准实时/历史
Fluentd
准实时
Elasticsearch(一天)
Kibana
历史
实时告警
spark
Hadoop(历史)
Hbase
告警

数据分治


机器<=5
App-s1集群
Rsyslog
App-s2集群
Rsyslog
App-sn集群
Rsyslog
Tag-s1
Tag-s2
Tag-sn
fluentd集群
Filter s1
Filter s2
Filter sn
Error alter
normal
normal
spark
hadoop
告警 系统
spark过滤直接分流对应目录,或者fluentd直接分流到目录
API
API-Hadoop wrap
策略：
1. 用户下载后的被记录
2. 用户关联APP
3.用户下载大文件分割策略
客户端提供vi查询，查看的log文件本地缓存

效果

01 机器资源有效的利用

02 传输单核每秒3000条上升到1.5万条

03 很少触发ES保护机制

04 历史数据有效存储与追溯

目录

日志优化总结

01 日志的特点是低频次查询-- 把历史数据放入廉价存储

02 时间越长，意义越少--有效留存有意义数据

03 顺序写盘替代内存，量大实时考虑SSD

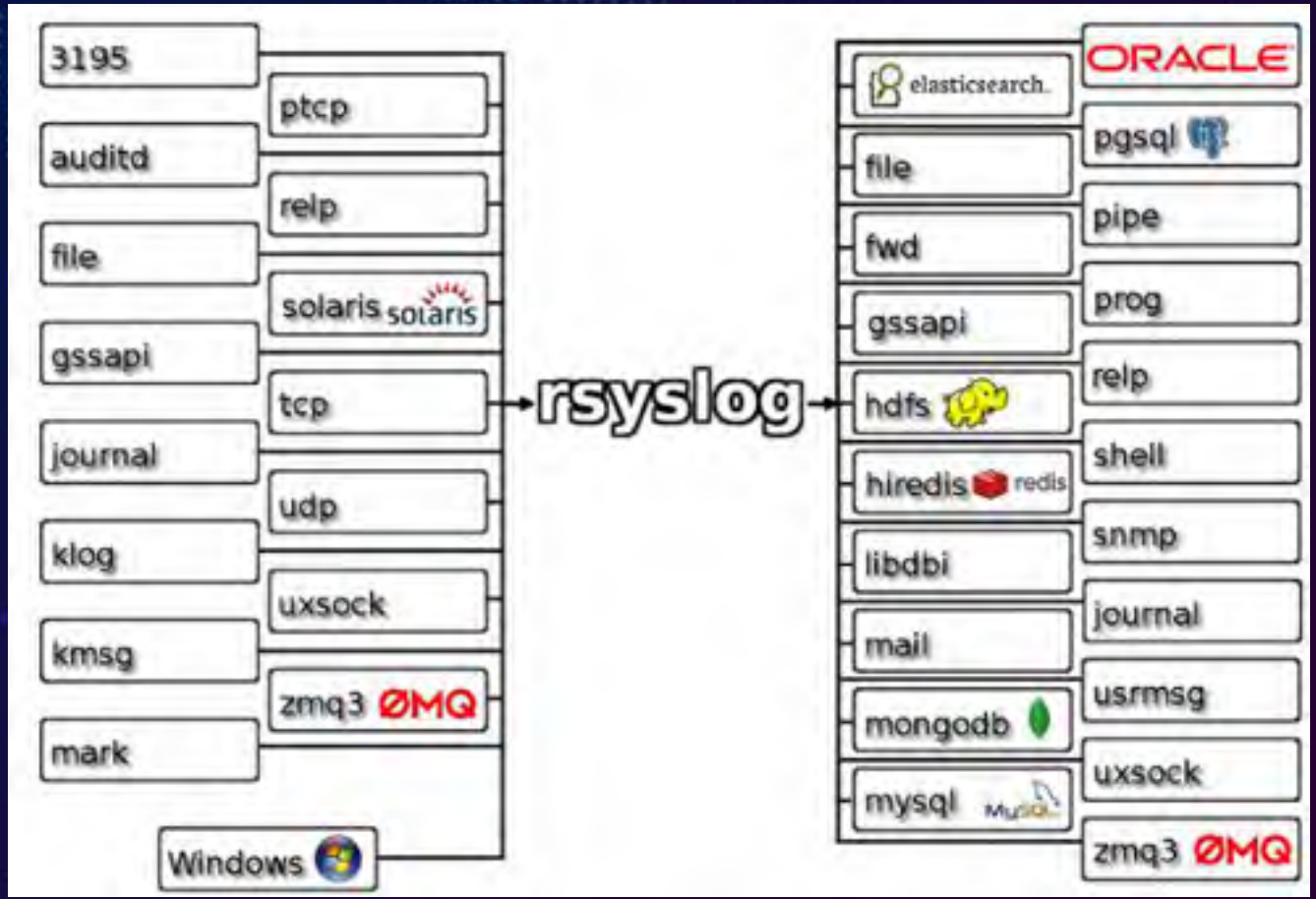
04 提前定制规范，能够有效解决后期的分析等工作

日志格式

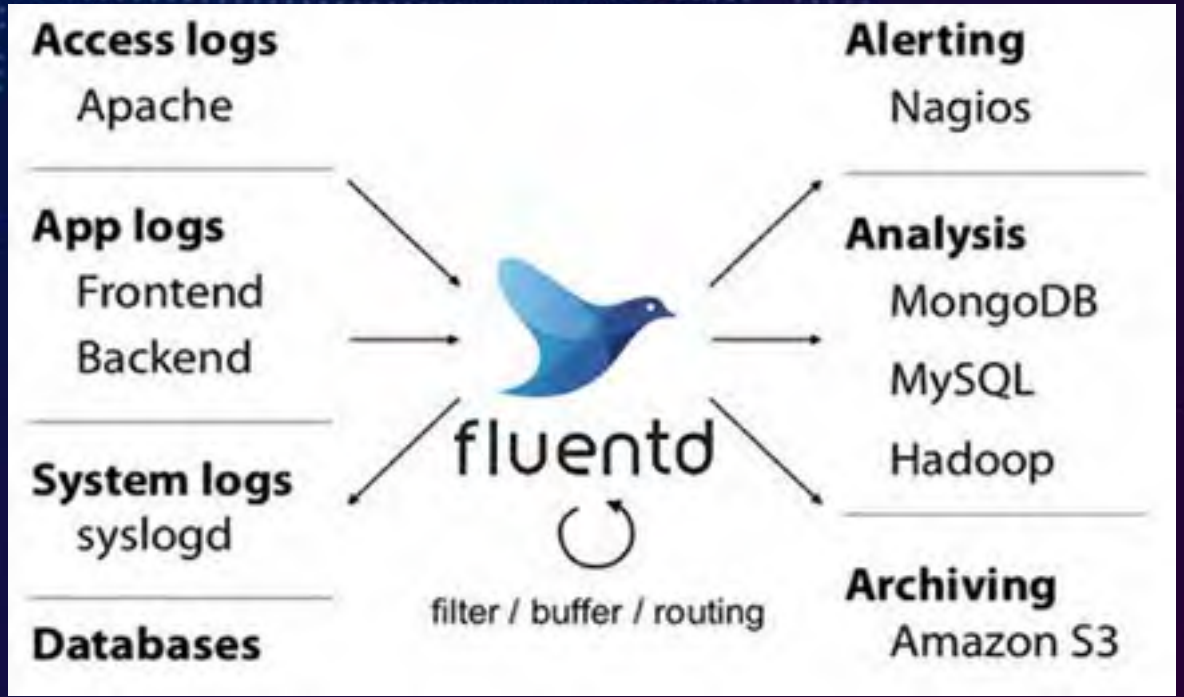
类型	
uuid	
timestamp	
type	Type of message i.e. “WebLog”.
logger	Data source i.e. “Apache”, “TCPInput”, “/var/log/test.log”
severity	Syslog severity level.
payload	Textual data i.e. log line, filename.
env_version	
pid	
hostname	
fields	Array of Field structures.

目录

日志方案


3195
auditd
file
gssapi
journal
klog
kmsg
mark
Windows
ptcp
relp
solaris
tcp
udp
uxsock
zmq3
rsyslog
elasticsearch
file
fwd
gssapi
hdfs
hiredis
libdbi
mail
mongodb
mysql
ORACLE
pgsql
pipe
prog
relp
shell
snmp
journal
usrmsg
uxsock
zmq3

日志方案


Access logs
Apache
App logs
Frontend
Backend
System logs
syslogd
Databases
fluentd
filter / buffer / routing
Alerting
Nagios
Analysis
MongoDB
MySQL
Hadoop
Archiving
Amazon S3

Thank you!