Open-Falcon

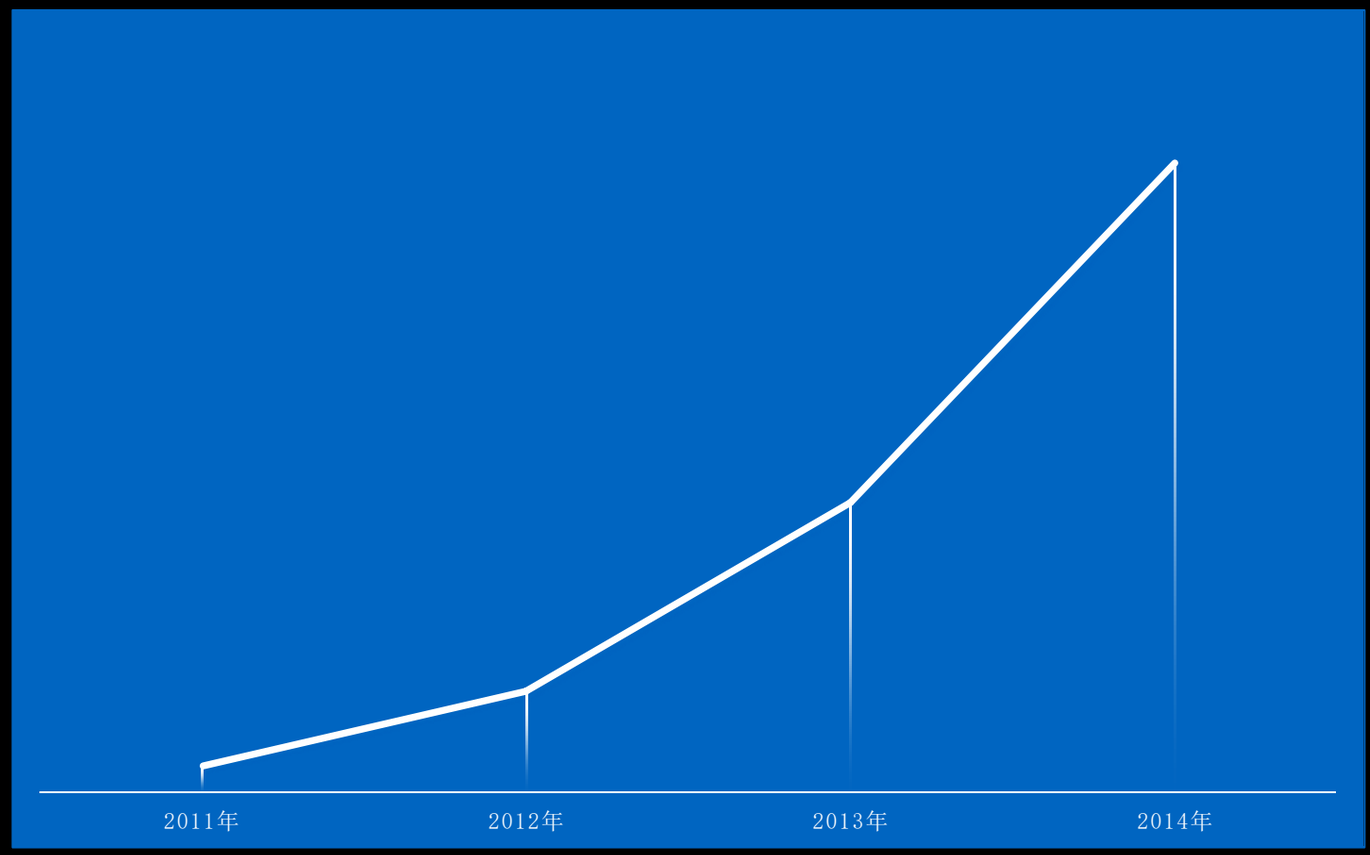
监控现状

- 小公司/创业团队，<1000台服务器规模

- BAT级别，>10W台服务器规模
  - Noah、XFlush、alimonitor、TMP、TNM2

早期的小米

- 规模
  - 2个同城机房
  - 多个自建CDN节点
- Zabbix
  - 对其进行二次开发、封装、调优
  - 基本满足监控需求



Zabbix遇到的问题

- 学习成本高，用户使用效率低下
- 系统自身不具备水平扩展能力，性能下降，无法支撑业务发展
- 告警策略维护、变更代价很大，运维工程师耗费大量时间
- 不利于自动化管理、无法融入内部运维自动化体系
- ……

寻找适合中等规模公司的监控……

寻求新思路：Open-Falcon

业务模型与设计原则

- 业务模型
  - 被动拉取->主动上报
  - 主机人工配置->特征自动匹配
- 设计原则
  - 人性化：易部署、易运维、易使用
  - 高可用：区域自治、支持多机房、无核心单点
  - 可扩展：水平扩展、插件扩展、功能扩展

监控规划

Dash board

主动上报

程序数据 主机数据

外部接入

域名 访问质量 网络监控 …

数据处理

阈值判别 复杂计算 智能分析

报警与联动

报警策略 联动处理 报警丰富 报警跟踪 问题管理

API

特点

- 数据采集：无须预定义、Agent自发现、支持Plugin、主动上报。
- 容量水平扩展：每秒20多万次数据收集。
- 可扩展性：告警、存储、绘图，可持续水平扩展。
- 告警策略：策略模板、模板继承和覆盖、多种告警方式、回调动作。
- 告警设置：最大告警次数、连续性&非连续报警、告警级别、恢复通知、维护周期，告警合并。
- 历史数据效查询：秒级返回上百个指标一年的历史数据。
- Dashboard人性化：多维度的数据展，户定义Dashboard等功能。
- 架构设计高可用：系统无核心单点。

数据采集

400+

自带监控项

- CPU
- 内存
- 负载

20+

社区贡献监控插件

- 磁盘空间
- 磁盘I/O
- 网络相关
- 端口存活
- ss命令采集数据

500+

社区贡献监控项

- 进程存活
- 进程资源消耗
- 时钟偏移

数据采集

- 服务自定义采集项
  - 基
    相
  - 自
  - N
- Pus
  - S
  - A

```python
#!-*- coding:utf8 -*-

import requests
import time
import json

ts = int(time.time())
payload = [
    {
        "endpoint": "test-endpoint",
        "metric": "test-metric",
        "timestamp": ts,
        "step": 60,
        "value": 1,
        "counterType": "GAUGE",
        "tags": "location=beijing,service=falcon",
    },
]
r = requests.post("http://127.0.0.1:1988/v1/push", data=json.dumps(payload))
print r.text
```

主动采集和push

统

r

数据采集

- **metric**
  代表这个采集项具体度量的是什么
- **endpoint**
  代表metric的主体(属主)是什么
- **tags**
  一组逗号分割的键值对，对metric进一步描述和细化
- **timestamp**
  UNIX时间戳，描述产生该数据的时间点
- **counterType**
  COUNTER或者GAUGE ， 描述该采集项的类型
- **step**
  描述该采集项的汇报周期，单位为秒

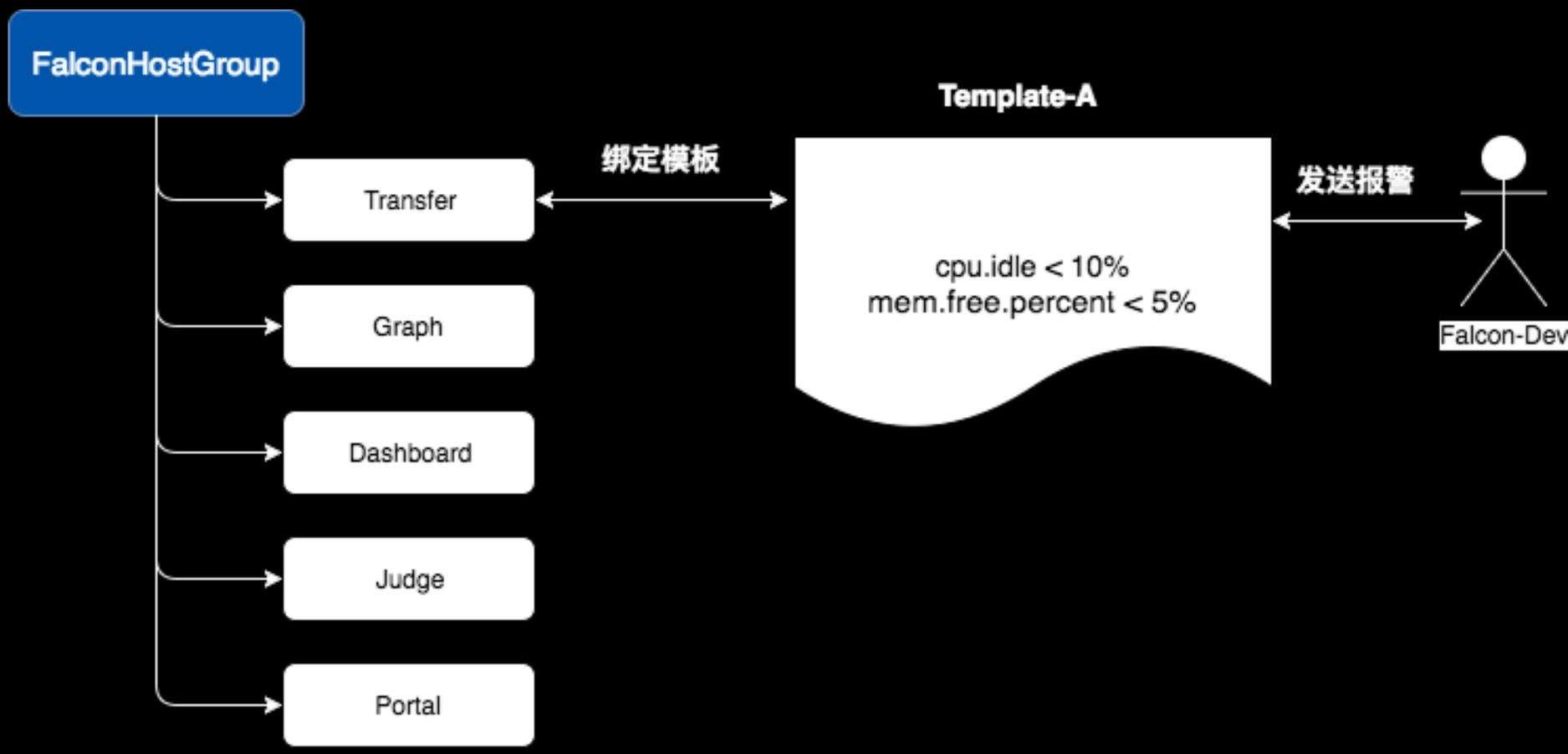
```
{
    metric: df.bytes.free.percent,
    endpoint: hostA,
    tags: mount=/home,
    value: 5,
    timestamp: UNIX时间戳,
    counterType: GAUGE,
    step: 60
}
```

```
{
    metric: df.bytes.free.percent,
    endpoint: hostA,
    tags: mount=/root,
    value: 15,
    timestamp: UNIX时间戳,
    counterType: GAUGE,
    step: 60
}
```

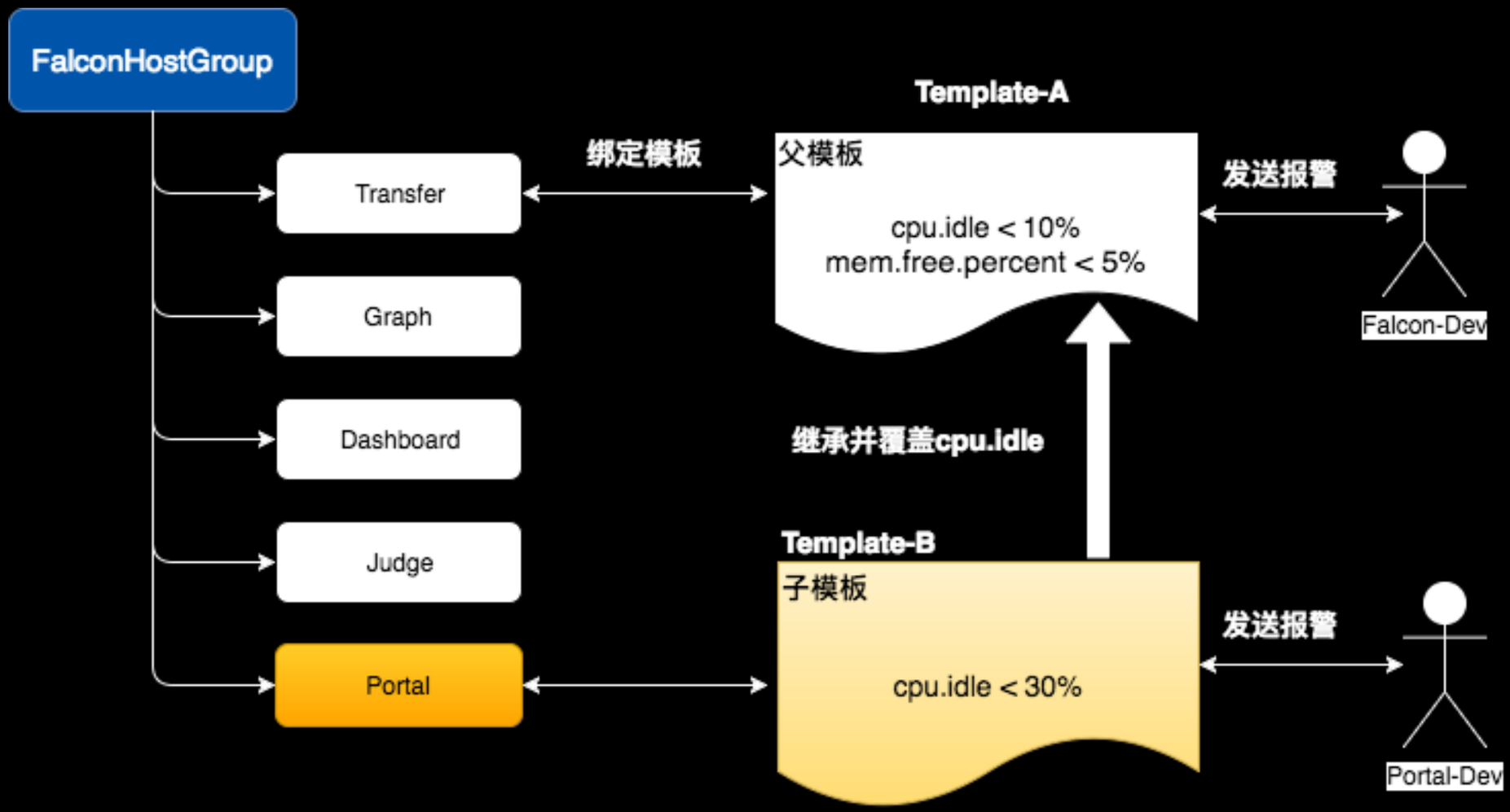
监控告警

- **VS Zabbix**
  - 模板继承与覆盖
  - 服务维度管理模板
  - **Tag化描述策略**
  - 实例上下线自动变更监控

模板继承与覆盖


FalconHostGroup
Transfer
Graph
Dashboard
Judge
Portal
绑定模板
Template-A
cpu.idle < 10%
mem.free.percent < 5%
发送报警
Falcon-Dev

模板继承与覆盖


FalconHostGroup
Transfer
Graph
Dashboard
Judge
Portal
绑定模板
Template-A
父模板
cpu.idle < 10%
mem.free.percent < 5%
发送报警
Falcon-Dev
继承并覆盖cpu.idle
Template-B
子模板
cpu.idle < 30%
发送报警
Portal-Dev

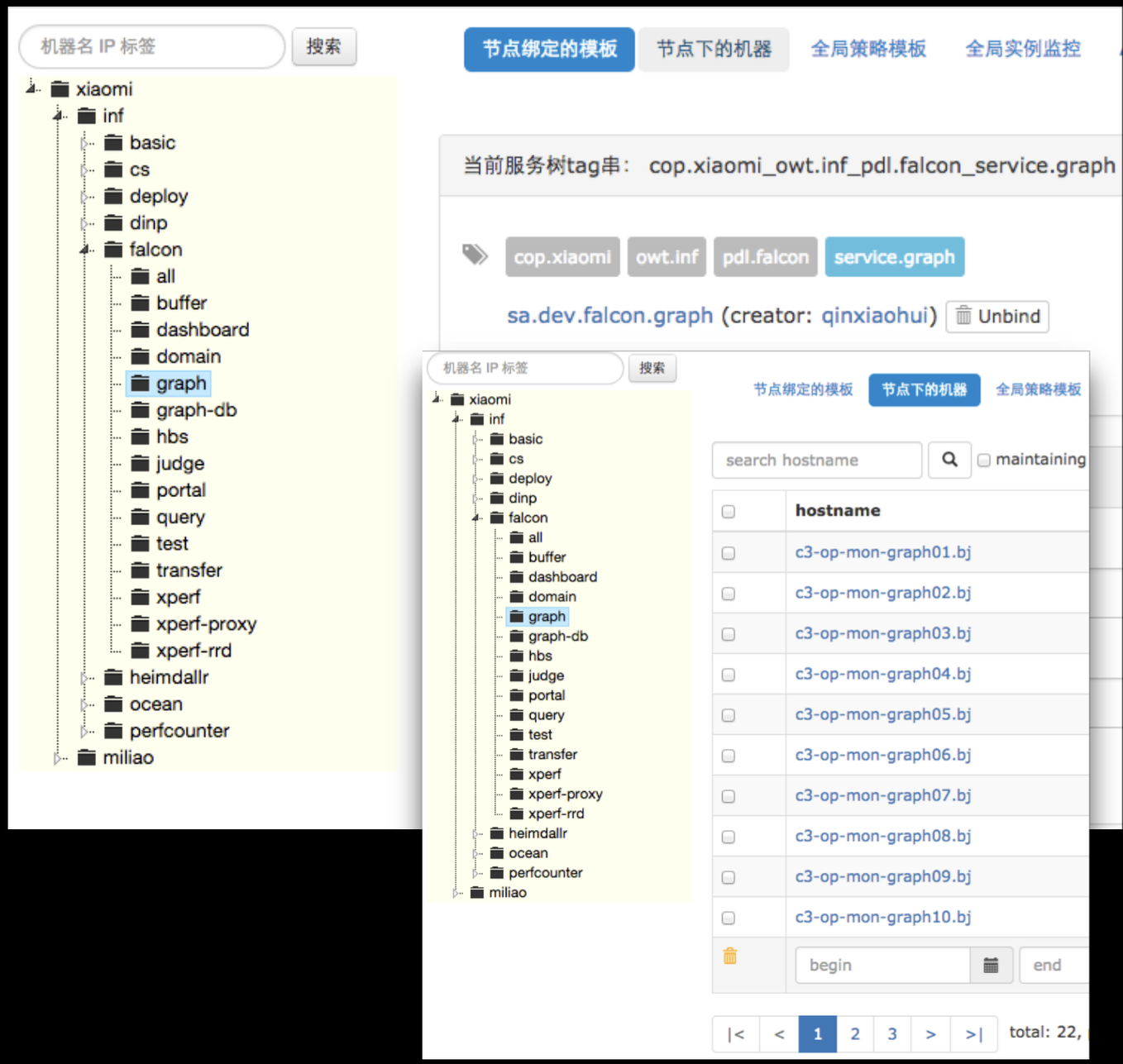
服务维度管理模板

- 服务树节点 和 **HostGroup** 动态对应
- **HostGroup**绑定策略模板

企业私有业务树

数据转换

HostGroup



Tag化描述告警策略

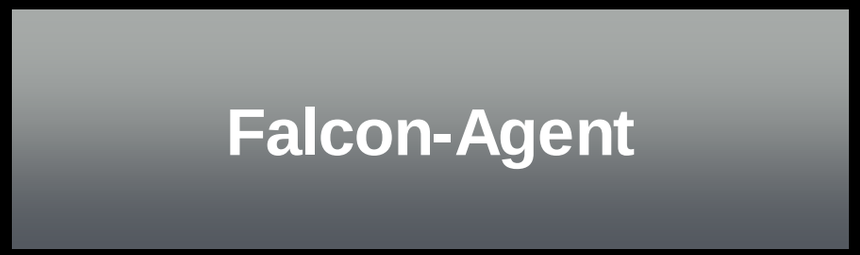
hostA主机有多个磁盘，当所有分区可用空间<5%触发告警
metric=df.bytes.free.percent && endpoint=hostA < 5%

hostA的root分区可用空间<10%，触发告警
metric=df.bytes.free.percent && endpoint=hostA && mount=/root < 10%

Falcon有100台服务器，cpu.idle<20%，触发告警
metric=cpu.idle && service=falcon < 20%

监控自动变更

- 自动添加实例监控
- 业务维护时间



模板中的策略列表

Id	策略基本信息(metric/tags [note])	触发条件	最大报警次数	报警级别	生效时间段
224	common.servicerunning/cop=xiaomi,owt=miliao,pdl=account	all(#3)==0	10	2	
166	cpu.idle	all(#5)<5	3	2	00:00-05:00

监控报警

级别	描述	影响	影响范围	处理要求	报警方式
P0	业务核心功能异常	业务核心功能 部分核心功能不可用	所有用户 部分用户	立即通报 立即处理	短信、邮件、消息工具
P1	业务非核心功能出现问题或业务响应、数据实效性下降	非业务核心功能 部分核心功能不可用	所有用户 部分用户	立即通报 立即处理	短信、邮件、消息工具
P2	内部问题，对业务功能无影响。如服务器宕机、服务实例crash等	无	无	无需通报 及时处理	短信、邮件
P3	内部预警类问题，对业务功能无影响。如磁盘空间、CPU ID等	无	无	无需通报 及时处理	短信、邮件

监控报警

- 持发短信、发邮件
- 支持http的回调接 (动态参数)

模板报警配置，对模板中的所有策略生效

def alarm(): #配置了UIC组才会发报警

报警接收组（在UIC中管理报警组，快捷入口）：

× falcon

def callback(): #高级用法，配置了callback地址才会触发回调

callback地址（只支持http get方式回调）：

回调之前发提醒短信　回调之前发提醒邮件　回调之后发结果短信　回调之后发结果邮件

Save

报警聚合

- 原则
  - 严重问题报警不能延迟
  - 非严重问题延迟不能超过2分钟
  - 异常&恢复均需要合并
- 维度
  - 级别、状态、接收人、metric
- 时间窗口：60s

数据

- 报警数据
  - 最近采集点内存存储，用于报警判别
- 绘图数据
  - **falcon-graph && rrdtool**，数据采样，老化
  - 默认存储**5**年
- 详细数据
  - **opentsdb && hbase** ， 用于数据挖掘分析

Dashboard

搜索Endpoints

Endpoint

c3-op-mon-graph0

可以用空格分割多个搜索关键字

标签(eg: job=appstore-web)

全局搜索 | Limit 50

快速过滤

刷新counter列表

- c3-op-mon-graph01.bj
- c3-op-mon-graph02.bj
- c3-op-mon-graph03.bj
- c3-op-mon-graph04.bj
- c3-op-mon-graph05.bj

搜索Counters

Counter

net if eth0 bytes

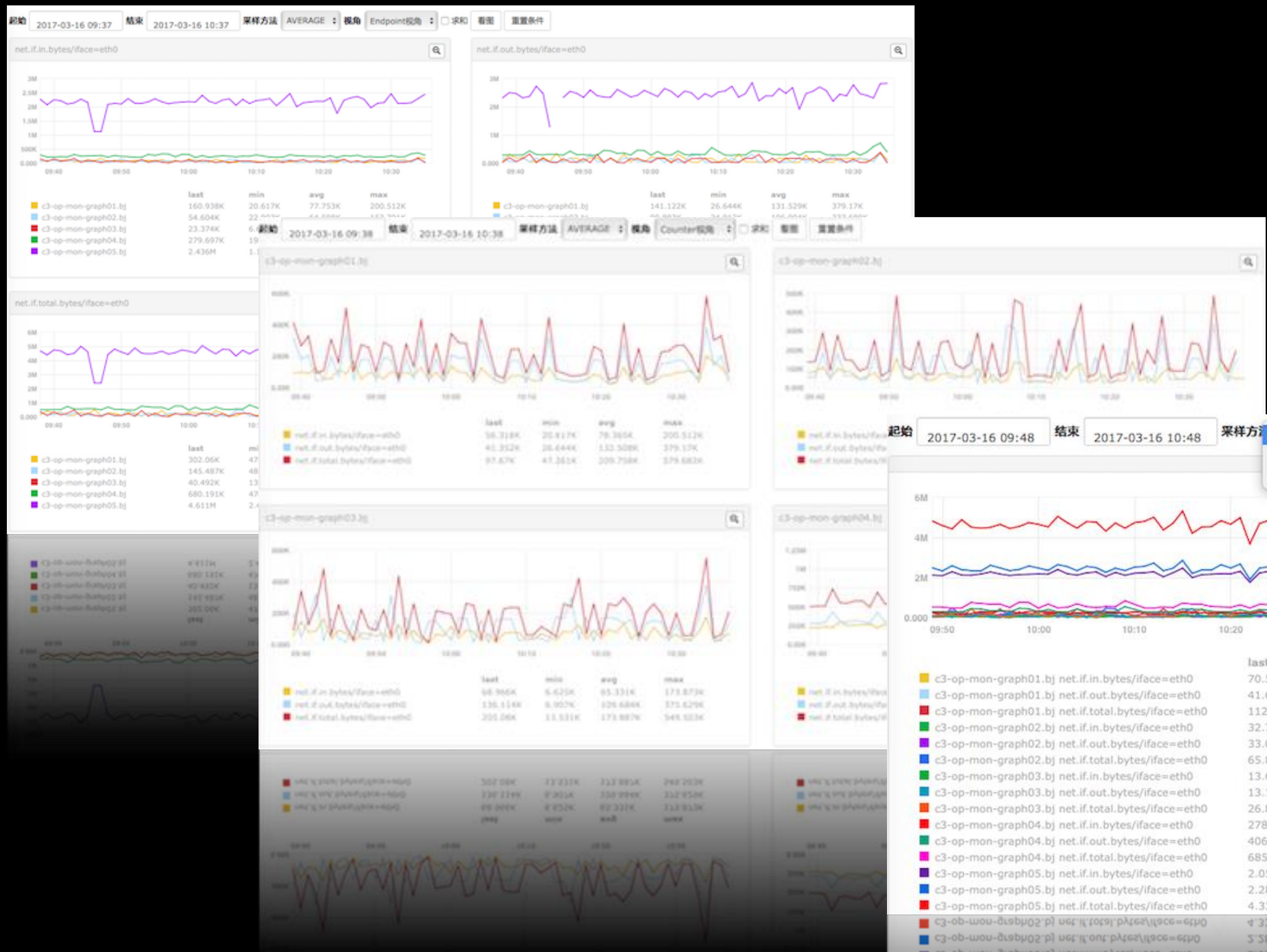
可以用空格分割多个搜索关键字

搜索 | Limit 50 | 快速过滤 | 其他操作 | 看图

Endpoint视角
Counter视角
组合视角

Counters		
net.if.in.bytes/iface=eth0		
net.if.out.bytes/iface=eth0	计数器	60s
net.if.total.bytes/iface=eth0	计数器	60s

Dashboard


起始 2017-03-16 09:37
结束 2017-03-16 10:37
采样方法 AVERAGE
视角 Endpoint视角
求和
看图
重置条件
net.if.in.bytes/iface=eth0
net.if.out.bytes/iface=eth0
net.if.total.bytes/iface=eth0
last min avg max
c3-op-mon-graph01.bj 160.938K 20.617K 77.753K 200.512K
c3-op-mon-graph02.bj 54.604K
c3-op-mon-graph03.bj 23.374K
c3-op-mon-graph04.bj 279.697K
c3-op-mon-graph05.bj 2.436M
起始 2017-03-16 09:38
结束 2017-03-16 10:38
采样方法 AVERAGE
视角 Counter视角
c3-op-mon-graph01.bj
c3-op-mon-graph02.bj
c3-op-mon-graph03.bj
c3-op-mon-graph04.bj
起始 2017-03-16 09:48
结束 2017-03-16 10:48
AVERAGE
MAX
MIN
视角 组合视角
last min avg max
c3-op-mon-graph01.bj net.if.in.bytes/iface=eth0 70.572K 20.617K 79.425K 208.739K
c3-op-mon-graph01.bj net.if.out.bytes/iface=eth0 41.633K 26.644K 130.015K 379.17K
c3-op-mon-graph01.bj net.if.total.bytes/iface=eth0 112.205K 47.261K 208.481K 579.682K
c3-op-mon-graph02.bj net.if.in.bytes/iface=eth0 32.785K 22.903K 73.459K 591.543K
c3-op-mon-graph02.bj net.if.out.bytes/iface=eth0 33.047K 24.912K 119.671K 656.203K
c3-op-mon-graph02.bj net.if.total.bytes/iface=eth0 65.832K 48.622K 193.157K 1.248M
c3-op-mon-graph03.bj net.if.in.bytes/iface=eth0 13.663K 6.625K 59.379K 173.873K
c3-op-mon-graph03.bj net.if.out.bytes/iface=eth0 13.155K 6.907K 101.95K 375.629K
c3-op-mon-graph03.bj net.if.total.bytes/iface=eth0 26.818K 13.531K 160.302K 549.503K
c3-op-mon-graph04.bj net.if.in.bytes/iface=eth0 278.691K 198.045K 249.072K 364.972K
c3-op-mon-graph04.bj net.if.out.bytes/iface=eth0 406.354K 270.133K 354.795K 710.044K
c3-op-mon-graph04.bj net.if.total.bytes/iface=eth0 685.046K 476.341K 603.403K 1.075M
c3-op-mon-graph05.bj net.if.in.bytes/iface=eth0 2.058M 1.763M 2.188M 2.6M
c3-op-mon-graph05.bj net.if.out.bytes/iface=eth0 2.281M 1.909M 2.473M 2.887M
c3-op-mon-graph05.bj net.if.total.bytes/iface=eth0 4.339M 3.671M 4.659M 5.488M

集群监控

合法 计算集群qps平均值
分子：$(qps/module=judge,project=falcon)
分母：$#

合法 计算集群qps总值
分子：$(qps/module=judge,project=falcon)
分母：1

合法 计算集群disk.io.util大于等于40%的机器个数
分子：$(disk.io.util)>=40
分母：1

集群监控

合法 计算整个集群diso.io.util大于40%的比率
分子：$(disk.io.util)>40
分母：$#

合法 计算集群中cpu.idle + cpu.busy为100的机器个数
分子：($(cpu.idle) + $(cpu.busy)) = 100
分母：1

合法 分母与分子配置无差别，都可使用Counter，仅举个例子
分子：$(cpu.busy)
分母：$(cpu.busy)

集群监控

- 集群监控

tag串：

cluster.production-lg_cop.xiaomi_job.frontend_owt.miliao_pdl.im_service.frontend_servicegroup.common

分子：

$(fe-current_connections/cluster=production-lg,cop=xiaomi,job=frontend,owt=miliao,pdl=im,service=frontend,servicegroup=common)

分母：

1

计算得出的监控值要重新push回Falcon，需填写以下信息：

endpoint:

im-fe-check

metric:

fe-current_connections.total.lg

tags:

cop=xiaomi,owt=miliao,pdl=im,servicegroup=common,service=frontend,job=frontend,cluster=production-lg

汇报周期（秒为单位）：

30

容器监控

- **Docker监控**
  - 宿主机上运行**cadvisor**采集基础监控，方便更新
  - 每容器一**IP**，**IP**作为**endpoint**
  - 容器内通过**falcon-agent**上报业务监控，对业务透明
  - 容器退出钩子清理报警事件
  - 监控服务器端老化过期数据（**7**天）
  - 集群监控**+**报警钩子支持服务自动缩扩容

未来展望

- 网络监控
- 报警管理
- 报警信息丰富
- 故障自动判别
- 系统保护机制