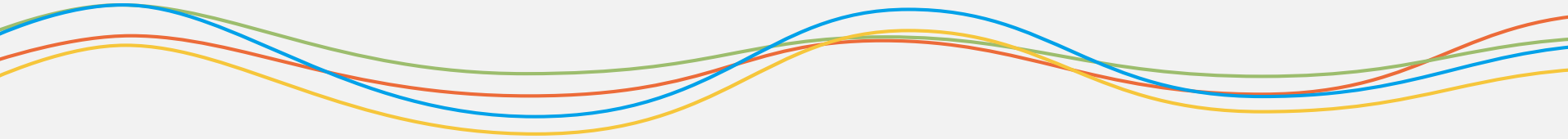
APMCon

去哪儿监控系统实践

张悦shadow@qunar



自我介绍

APMCon

目录

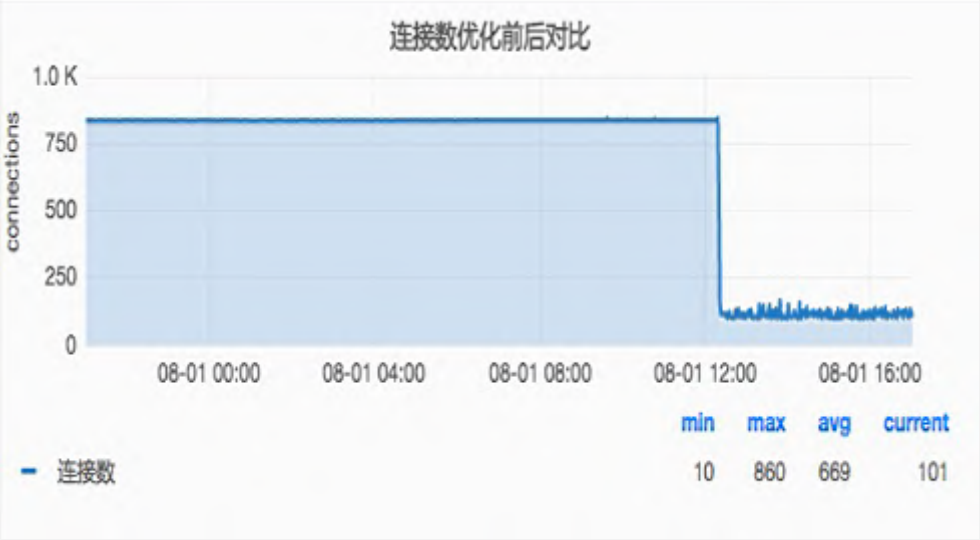
APMCon

- 背景
- Watcher设计&架构
- Watcher实践
- 总结

背景

为什么需要监控

APMCon



监控目的

APMCon

- 实时报警
- 提前预警
- 追查问题
- 容量规划

那些年我们用过的监控工具

APMCon

使用cacti存在的问题:

- 单点
- 监控可视化弱
- 没有API
- 每个业务线一套

需求&目标

APMCon

- 高可用、易扩容
- 高精度数据存储
- `好`看

选型

对比

APMCon

优势
- 基于列存储，可存放千亿级数据
- 支持Tags

劣势
- 集群启动成本较高，不适合每个机房一套
- 需要HBase运线维
- 函数有限

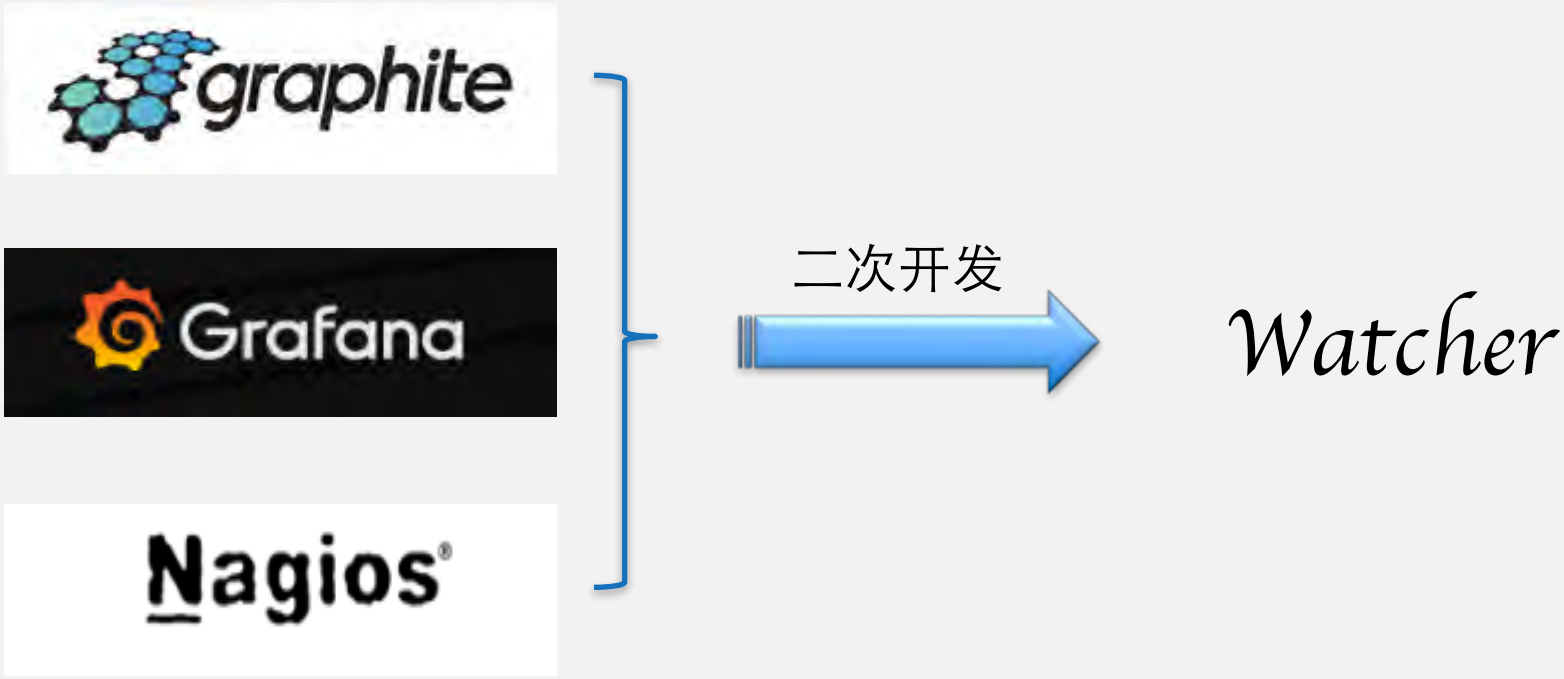
优势
- 易部署，无外部依赖
- 类似sql的查询语言

劣势
- 没有稳定版本

优势
- 每一层都可以横向扩展
- 基于文件存储，易运维
- 支持函数

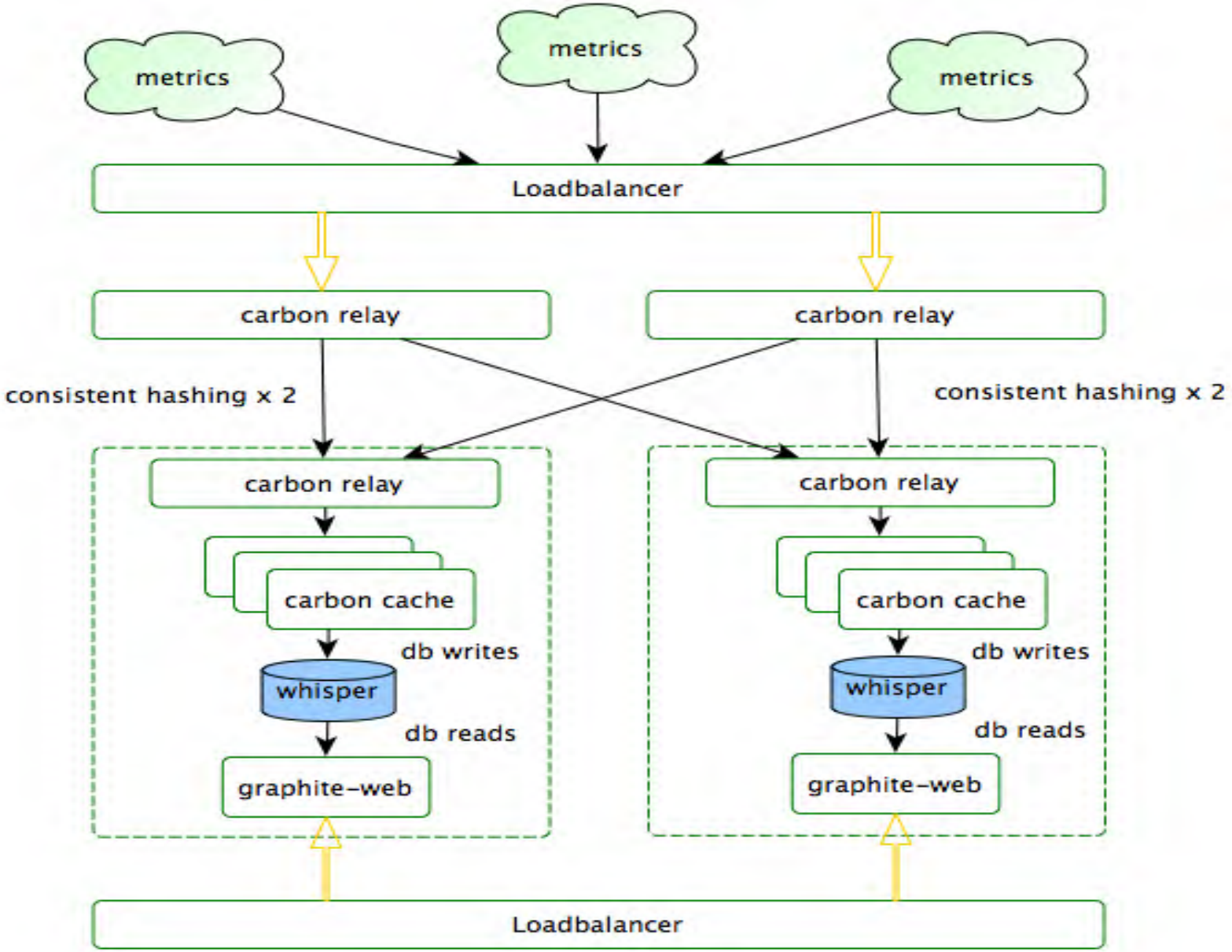
劣势
- 指标多会导致存储空间剧增(跟rrd一样)
- 需要SSD

Watcher设计&架构



架构演进

Watcher v0.1

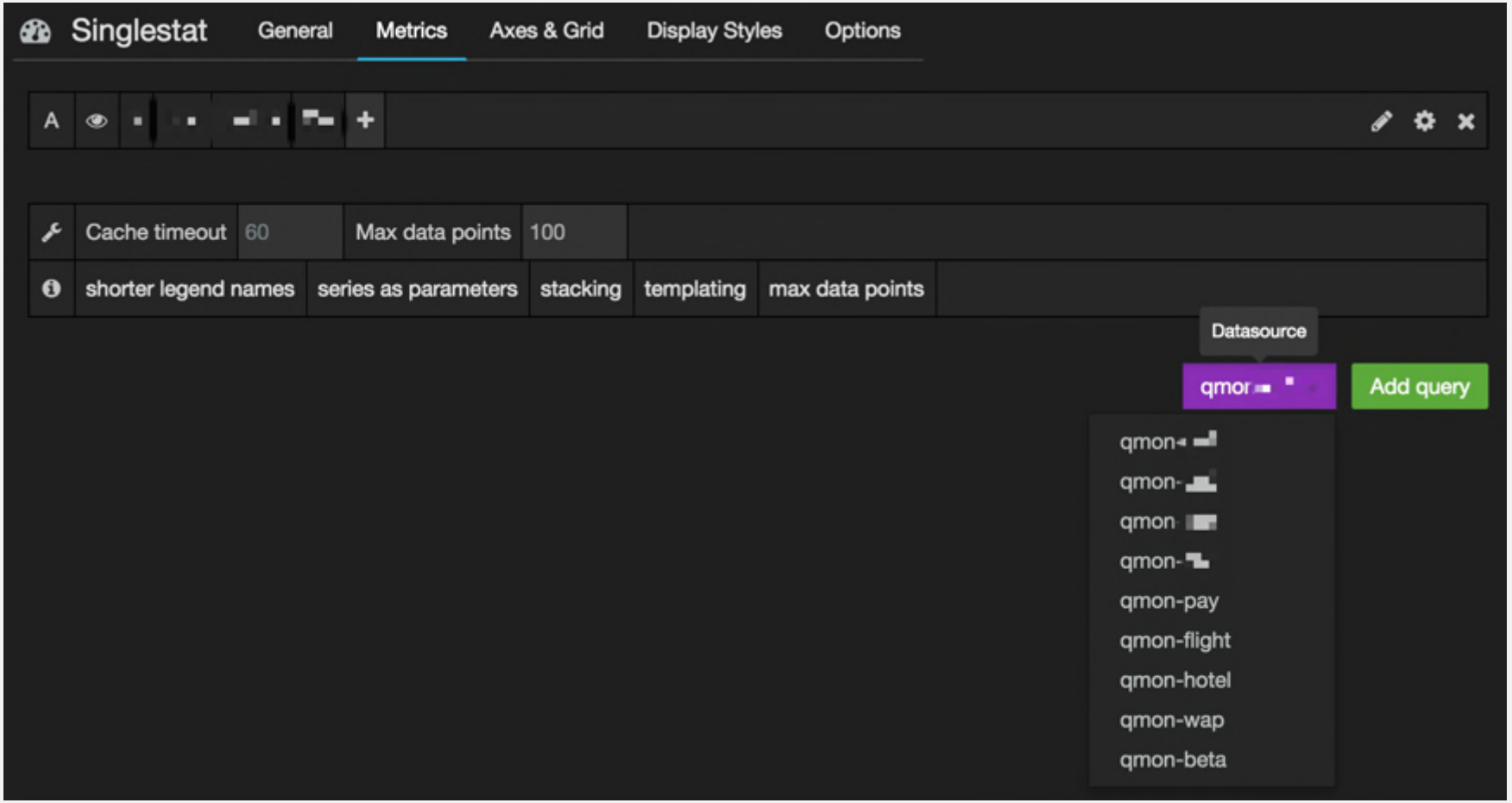


2013 TCP
2014 PICKLE
2015 UDP

遇到的问题

APMCon

图渲染慢！



为什么慢

APMCon

graphite-web

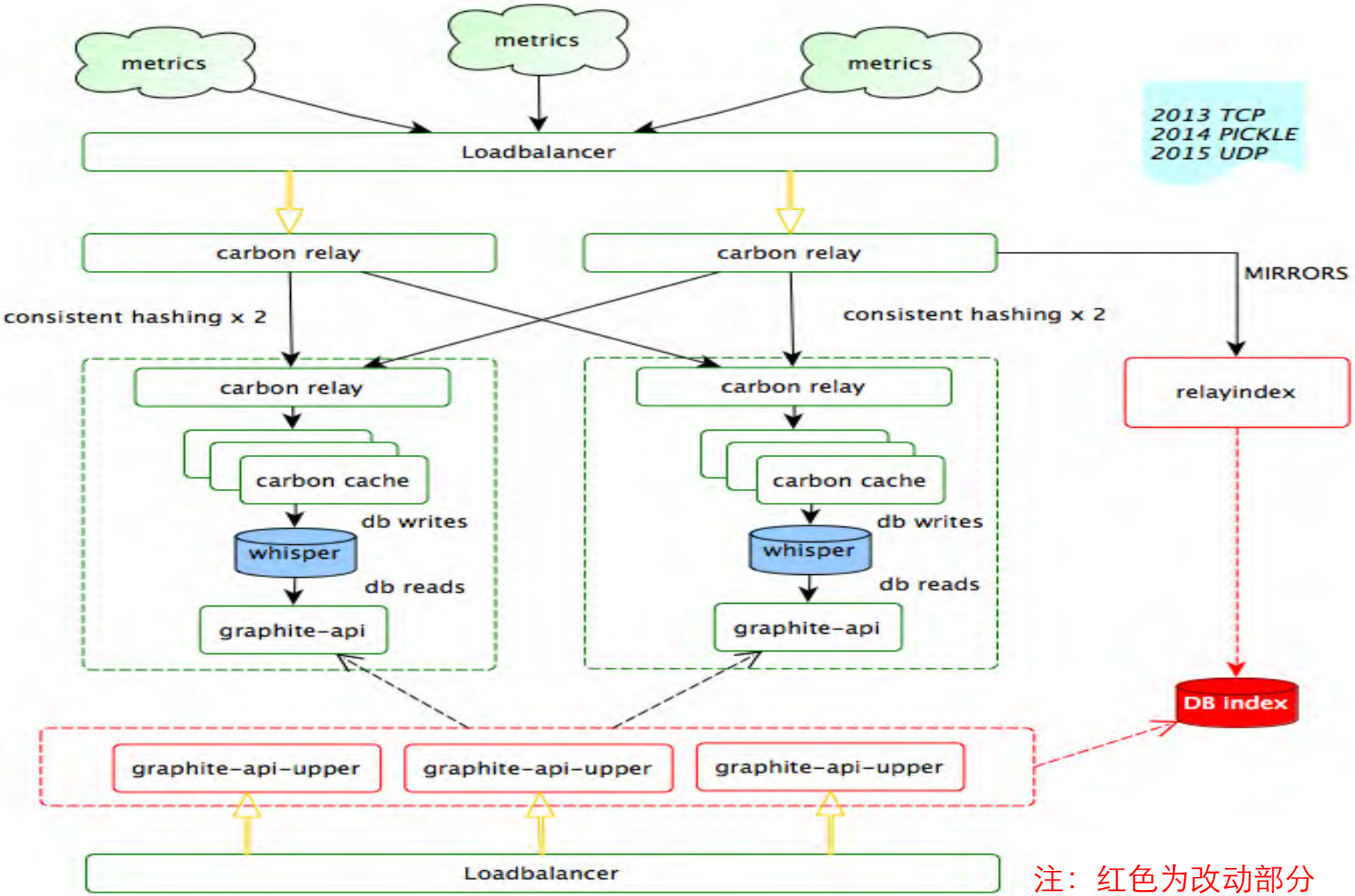
```
############################
# Cluster Configuration #
############################
CLUSTER_SERVERS = ["10.0.2.2:80", "10.0.2.3:80"]

## Remote rendering settings
# List of IP (and optionally port) of the webapp on each remote server that
# will be used for rendering. Note that each rendering host should have local
# access to metric data or should have CLUSTER_SERVERS configured
RENDERING_HOSTS = ["10.0.2.2:80", "10.0.2.3:80"]
```

```
23  class Store:
24    def __init__(self, finders=None, hosts=None):
25      if finders is None:
26        finders = [get_finder(finder_path)
27                   for finder_path in settings.STORAGE_FINDERS]
28      self.finders = finders
29
30      if hosts is None:
31        hosts = settings.CLUSTER_SERVERS
32      remote_hosts = [host for host in hosts if not settings.REMOTE_EXCLUDE_LOCAL or not is_local_interface(host)]
33      self.remote_stores = [ RemoteStore(host) for host in remote_hosts ]
34
35
36    def find(self, pattern, startTime=None, endTime=None, local=False):
37      query = FindQuery(pattern, startTime, endTime)
38
39      # Start remote searches
40      if not local:
41        remote_requests = [ r.find(query) for r in self.remote_stores if r.available ]
42
43      matching_nodes = set()
44
45      # Search locally
46      for finder in self.finders:
47        for node in finder.find_nodes(query):
48          #log.info("find() :: local :: %s" % node)
49          matching_nodes.add(node)
50
51      # Gather remote search results
52      if not local:
53        for request in remote_requests:
54          for node in request.get_results():
55            #log.info("find() :: remote :: %s from %s" % (node,request.store.host))
56            matching_nodes.add(node)
```

Watcher v0.2

APMCon

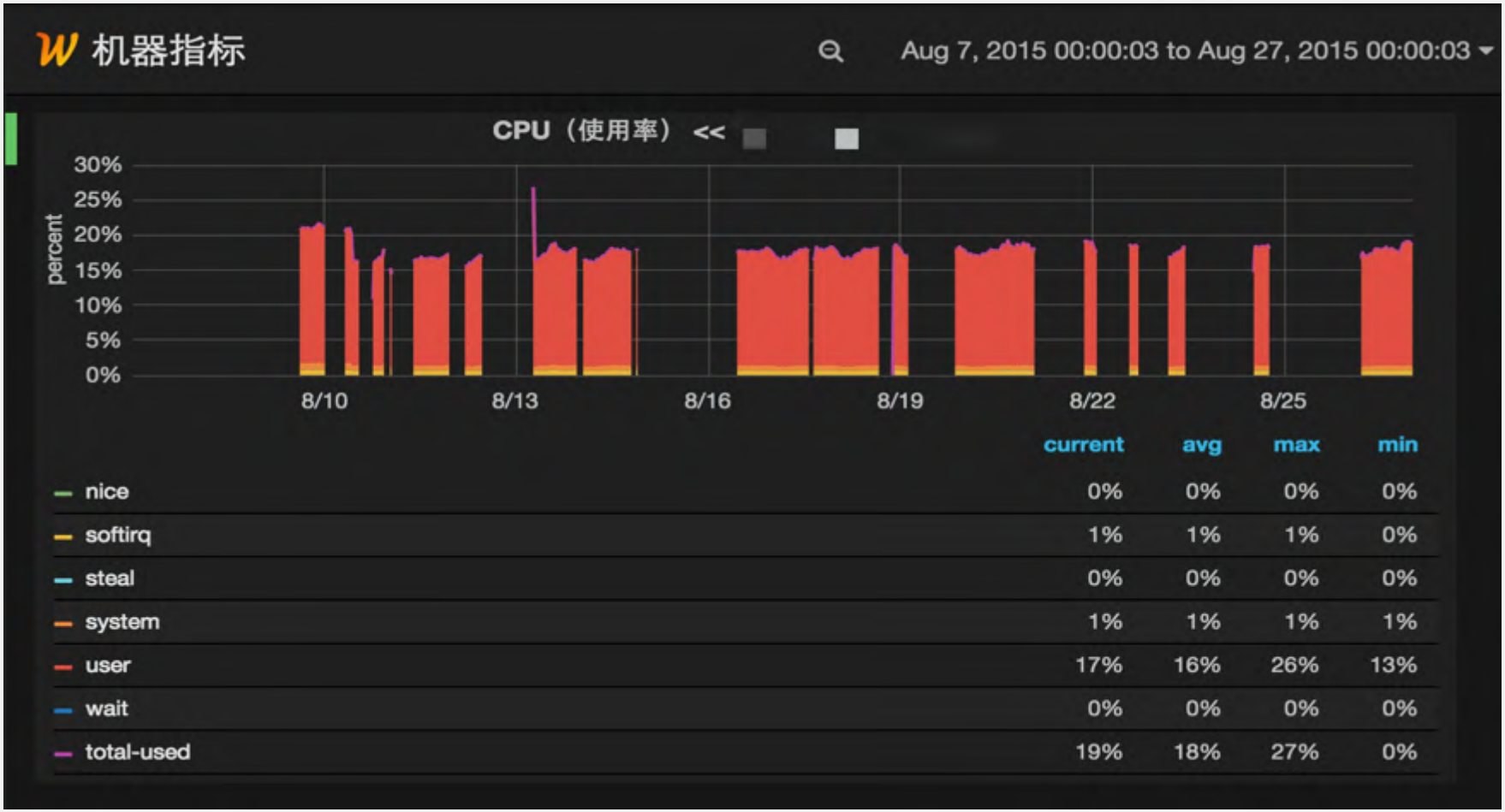


注：红色为改动部分

遇到的问题

APMCon

断点！ 丢数！



原因

APMCon

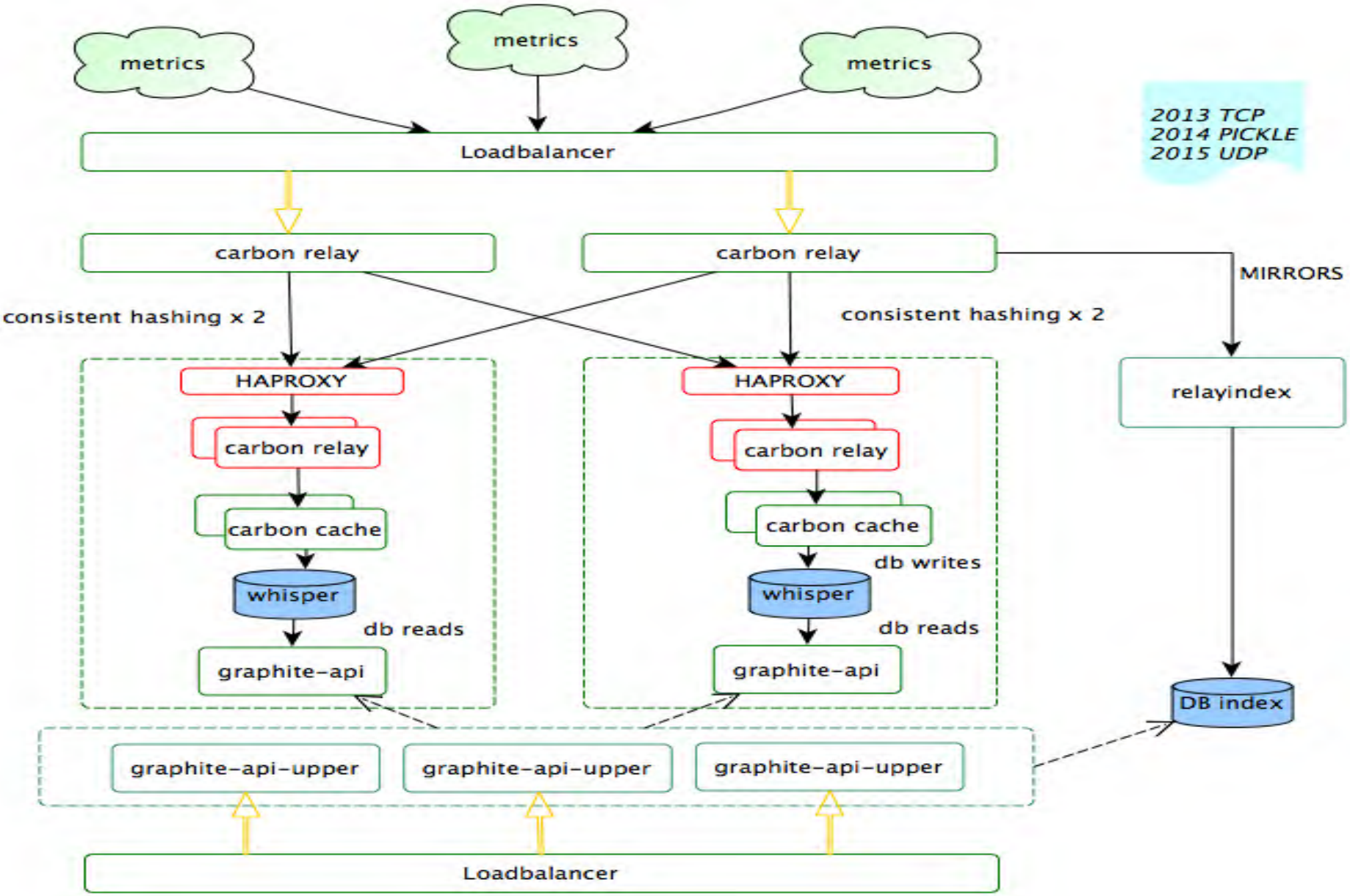
```
17:51:21 :: CarbonClientFactory(127.0.0.1:2124:b) send queue is full (100000 datapoints)
17:51:26 :: CarbonClientFactory(127.0.0.1:2424:e) send queue is full (100000 datapoints)
17:51:44 :: CarbonClientFactory(127.0.0.1:2024:a) send queue is full (100000 datapoints)
17:51:45 :: CarbonClientFactory(127.0.0.1:2224:c) send queue is full (100000 datapoints)
17:52:46 :: CarbonClientFactory(127.0.0.1:2324:d) send queue is full (100000 datapoints)
```

根本原因

- Carbon-relay CPU使用率100%

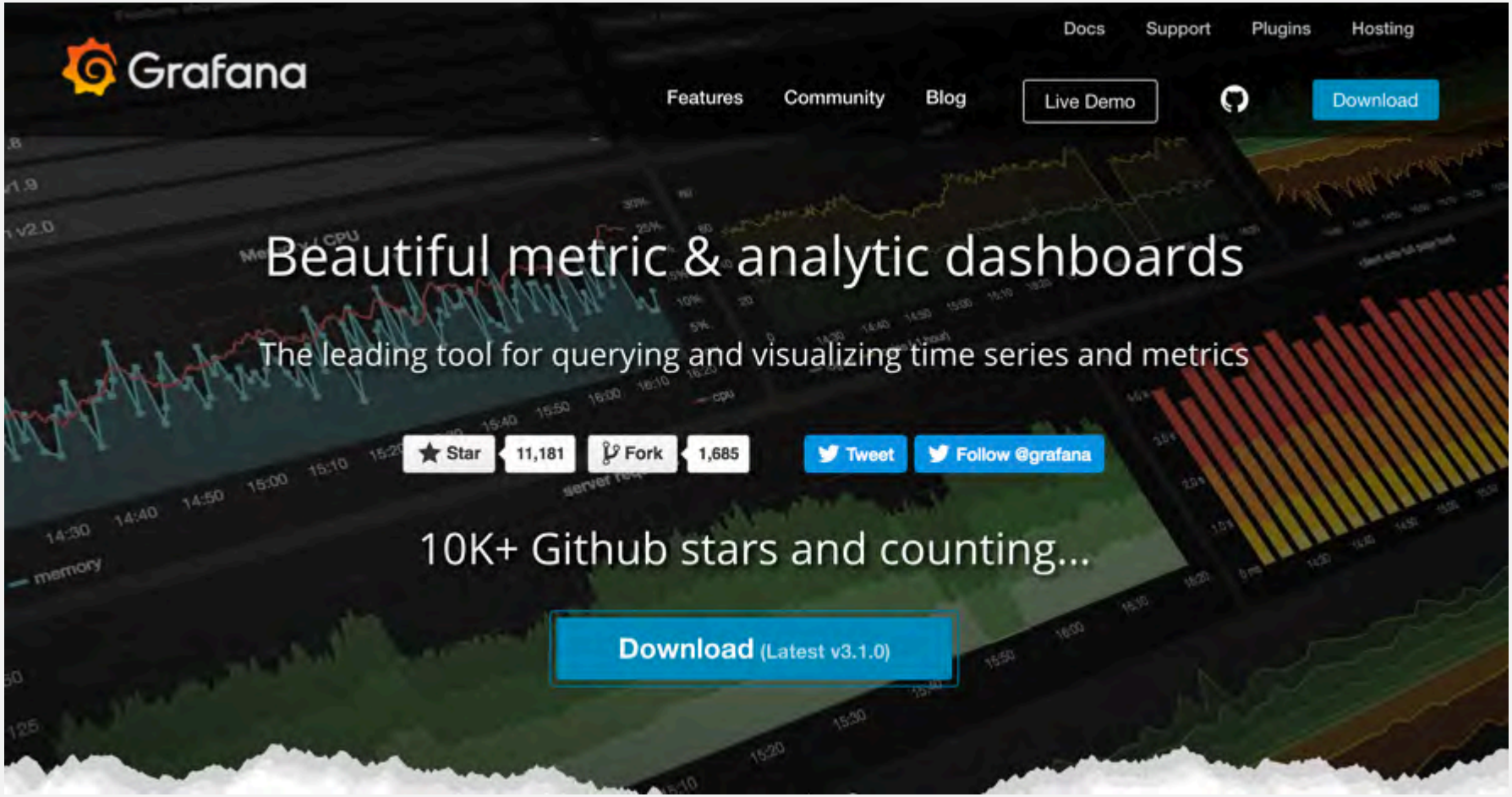
Watcher v0.3

APMCon



监控可视化

基于Grafana二次开发



Grafana特点

APMCon

- 丰富的图形和样式
- 很好的适配 Graphite Functions
- Templating、Annotations…

新增功能

APMCon

- 树形组织结构
- 用户&公共空间
- 版本记录&回滚
- 复制粘贴
- 多维度展示
- 分页
- …

机器维度

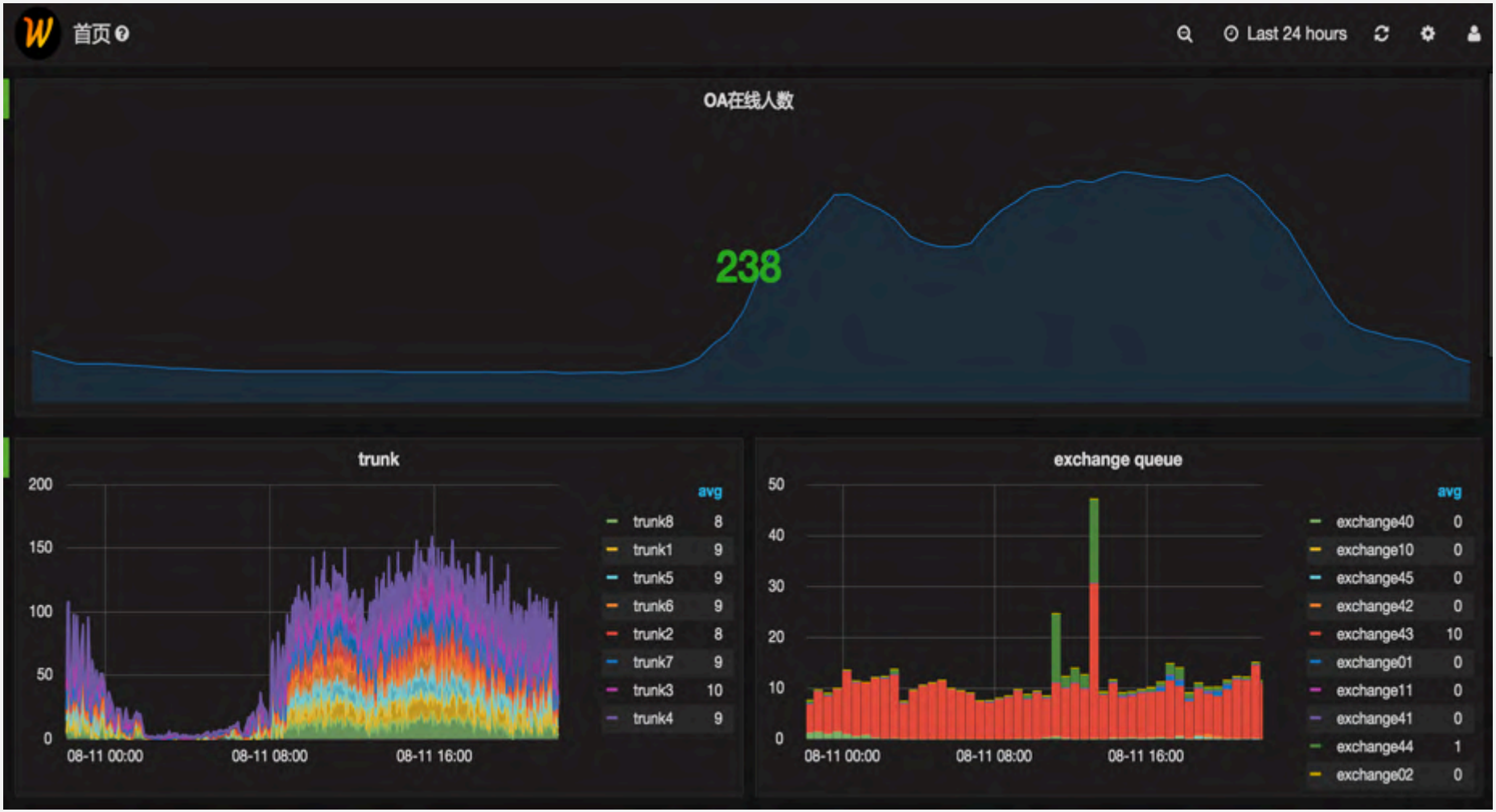

机器指标
Last 1 hour
用户 公共 指标 机器 报警
Linux Windows 网络 宿主 数据库
应用树:
机器名:
模板
CPU
CPU TCP连接 load
swap 内存 温度 用户
电流 磁盘I/O 磁盘空间
网络流量 进程 Nginx
Ngx_status Squid
使用说明:
1. 机器、指标和模板都支持正则表达式
(以|分隔可实现多选)
CPU (使用率) <<
50%
40%
30%
20%
10%
0%
percent
21:30
22:00
min max avg current
nice 0% 0% 0% %
softirq 0% 1% 0% %
steal 0% 0% 0% %
system 6% 9% 8% %
user 29% 36% 31% %
wait 0% 1% 0% %
total-used 36% 45% 40% %
CPU (使用率) <<
60%
50%
40%
30%
20%
10%
0%
percent
21:30
22:00
min max avg current
nice 0% 0% 0% %
softirq 0% 1% 0% %
steal 0% 0% 0% %
system 7% 10% 9% %
user 29% 40% 32% %
wait 0% 1% 0% %
total-used 37% 49% 42% %

应用维度

APMCon


应用名称
Last 1 hour
用户
公共
指标
机器
报警
添加图
监控名称
0.045
0.040
0.035
0.030
0.025
0.020
21:00
21:30
Min: 0.0333 Max: 0.0333 Avg: 0.0333 Current: Total: 1.9667 TotalTim
e(s) * Avg: 118.0000
12.5
10.0
7.5
5.0
2.5
0
21:00
21:30
Min: 0 Ma
x: 10.28 Avg: 5.11 Current: Total: 301.28 TotalTime(s) * Avg: 18.077000 K
2.00
1.75
1.50
1.25
1.00
0.75
21:00
21:30
Min: 0.9
00 Max: 1.800 Avg: 1.348 Current: Total: 79.550 TotalTime(s) * Avg: 4.7730
0000 K
260
240
220
200
180
21:00
21:30
Min:
186 Max: 250 Avg: 216 Current: Total: 12.7484 K TotalTime(s) * Avg: 764.9
030 K

自定义维度


首页
Last 24 hours
OA在线人数
238
trunk
200
150
100
50
0
08-11 00:00
08-11 08:00
08-11 16:00
avg
trunk8 8
trunk1 9
trunk5 9
trunk6 9
trunk2 8
trunk7 9
trunk3 10
trunk4 9
exchange queue
50
40
30
20
10
0
08-11 00:00
08-11 08:00
08-11 16:00
avg
exchange40 0
exchange10 0
exchange45 0
exchange42 0
exchange43 10
exchange01 0
exchange11 0
exchange41 0
exchange44 1
exchange02 0

完善监控

APMCon

系统级别监控

- cpu
- load
- 内存
- swap
- 网卡流量
- 磁盘空间
- 磁盘IO
- SSD
- …

中间件监控

- Mysql
  - 慢查询
  - QPS
  - 连接数
  - 表锁
  - 行锁
- JVM
  - 线程数
  - 堆内存
  - 垃圾收集时间
- Memcached/Redis
  - 命中率
  - 连接数

业务监控

- 接口响应时间
- 请求数
- 端口监控
- 进程监控
- 日志监控
- 订单量、用户数…
- …


指标接收/发送量
400 K
300 K
200 K
100 K
0
07-30 00:00
08-02 00:00
08-03 00:00
08-04 00:00
2016-07-30 13:30:00
指标发送量: 348.5796 K
指标接收量: 347.2023333333333 K
指标发送量
指标接收量
SWAP
2.8 GiB
1.9 GiB
1000 MiB
0 B
07-31
08-02
08-04
used
Memory
9 GiB
7 GiB
6 GiB
4 GiB
2 GiB
0 B
07-31
08-02
08-04
used

报警

灵活的报警配置

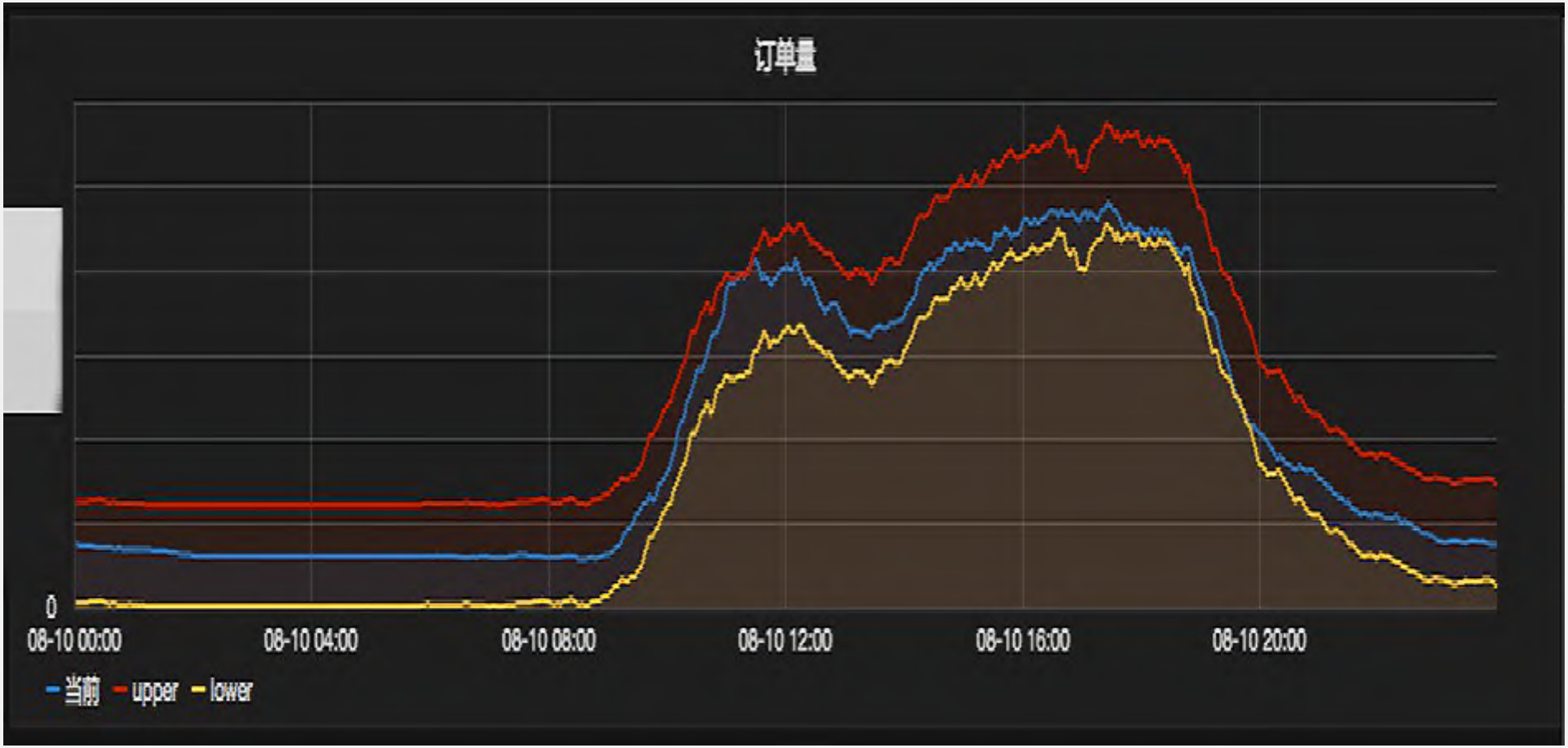
APMCon

- 不同时间段不同报警规则
- 报警联系人支持排班轮询
- 多种通知方式
- 按 OnCall 顺序升级电话通知
- 报警回调
- 临时规则

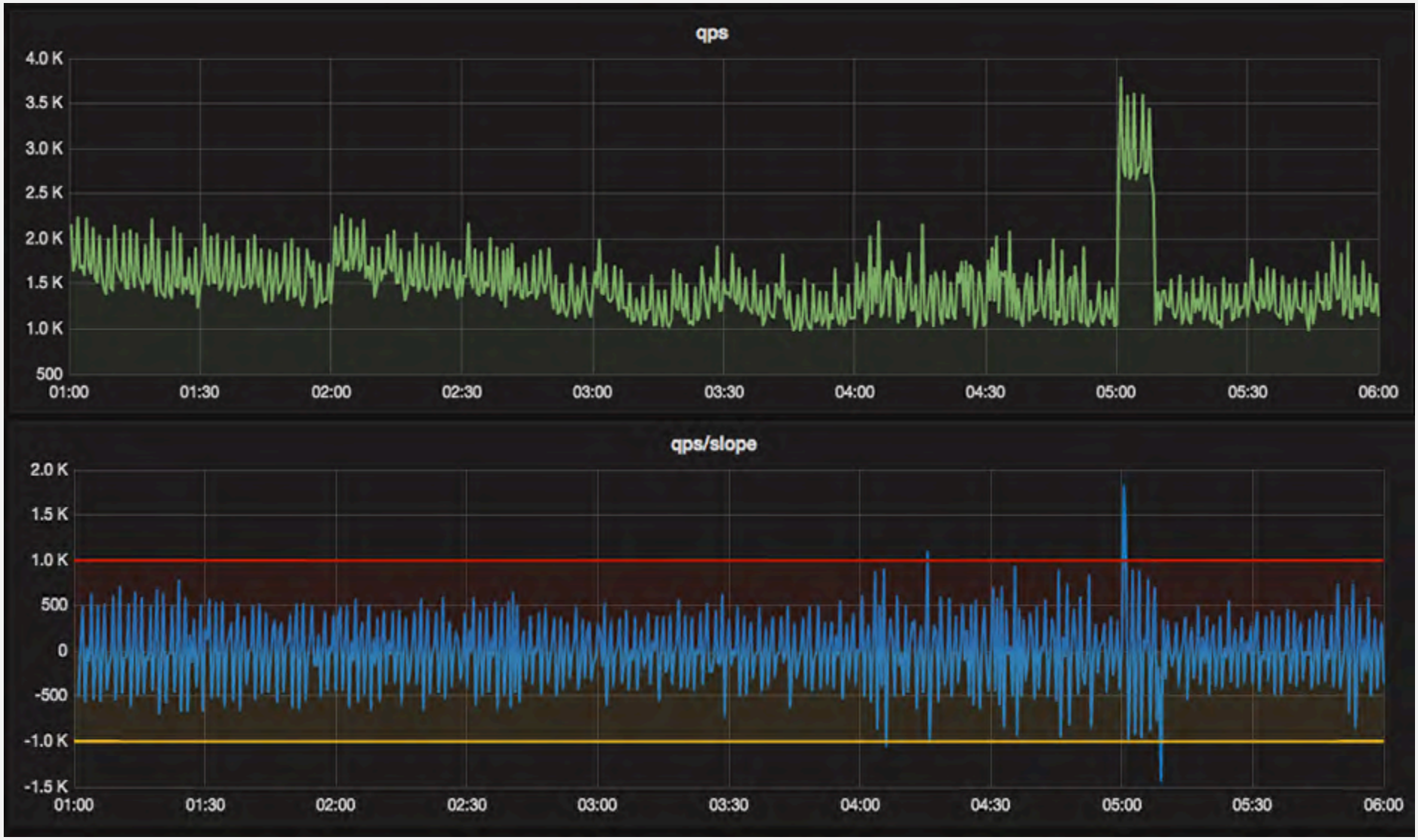
常用报警策略

APMCon

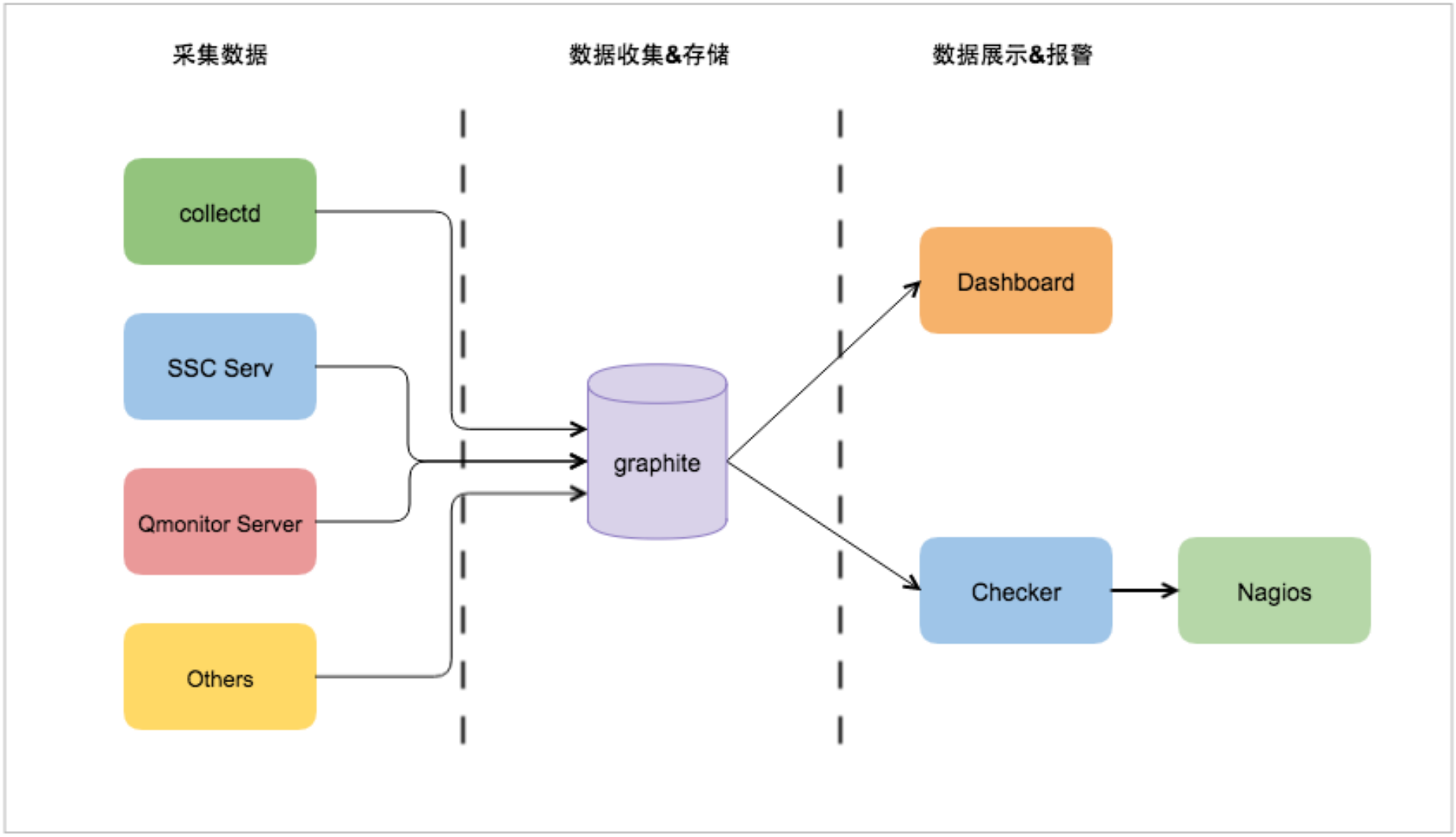
通过环比上周/昨天预测阈值



骤增/骤减 – 设置斜率报警


qps
4.0 K
3.5 K
3.0 K
2.5 K
2.0 K
1.5 K
1.0 K
500
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
qps/slope
2.0 K
1.5 K
1.0 K
500
0
-500
-1.0 K
-1.5 K
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00

数据走向


采集数据
数据收集&存储
数据展示&报警
collectd
SSC Serv
Qmonitor Server
Others
graphite
Dashboard
Checker
Nagios

总结

现状

APMCon

- 基础监控指标500+
- 600w+系统级别监控指标 (24T)
- 200w+业务监控指标 (24T)
- 每分钟1600w指标上报


监控系统
用户
告警
提升可用性
减少故障损失
可视化
分析
决策
运维
容量
水平扩展
在线扩容
稳定性
整体无单点
成熟社区，经过考验的方案
效率
丰富的运维工具
减少人工干预

THANKS!