



大米運維課堂

最前哨開源監控 Prometheus 專題講座

第三讲 Prometheus 监控入门简介

第三讲内容

- prometheus是什么?
- prometheus能为我们带来些什么
- prometheus对于运维的要求
- prometheus多图效果展示

1) Prometheus是什么

我们还是来先看看 官方对它的原始定义

Prometheus is an open-source systems monitoring and alerting toolkit originally built at SoundCloud. Since its inception in 2012, many companies and organizations have adopted Prometheus, and the project has a very active developer and user community. It is now a standalone open source project and maintained independently of any company. To emphasize this, and to clarify the project's governance structure, Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.

大米来帮翻译一下哈=》

Prometheus 是一个开源系统监控和报警的工具集合，由SoundCloud创建 (<http://soundcloud.com/>)，自从2012诞生之后，至今已经有许多公司和组织开始使用它了，这个开源项目拥有大量的积极参与开发和建设的研发人员以及社区用户。目前已经是一个独立运行的开源的由各公司自行维护的监控项目。为了让项目更充实更清晰 2016年 Prometheus 加入了 Cloud Native Computing Foundation (CNCF)，并且成为继Kubernetes(结合 容器/docker)-之后 第二个加入该组织的成员

这个就是来自于官方的介绍

其中更多突出的 还是这个项目基于开源的 和 各种社区组织维护 多重联合开发 的这样一个特性

也就决定了 这个项目 必然是越来越好

当前最新的版本Prometheus 2.0 也是近期刚推出不久，经过本人最近几个月的对最新版本的实践 各种之前遇到的问题 差不多都被修复了 而且性能更加稳定

(大米老师当前所在的企业 也是以Prometheus 作为核心监控，新版本运行 几个月来 没有出现过任何稳定和性能上的问题)

2) Prometheus能给我们带来什么

之前在第二讲最后的部分 我们给大家推出了一个 最终的未来理想化监控

那么prometheus 可以针对未来监控 对于 准确性 和 精确性的 要求 极大的贡献自己的一份力量

为什么这么说呢？ 我们不能凭空判断，咱们用真实数据来说话

认识promethues监控的优质特性

- 基于time series 时间序列模型 (数字 数学)

时间序列 (time series X,Y) 是一系列有序的数据。通常是等时间间隔的采样数据。

- 基于K/V 的数据模型

Key/value 这个 键值的概念 咱们很熟悉

```
{ disk_size : 80 }
```

最大的好处 就是数据格式 简单 速度快 易维护开发

- 采样数据的查询 完全基于数学运算 而不是其他的表达式 并提供专有的查询输入 console

这个特点很独特，所有的查询都基于数学运算公式 例如 (增量 (A) + 增量 (B)) / 总增量 (C) > 固定百分比 =>

- 采用 HTTP pull / push 两种对应的数据采集传输方式

所有的数据采集 都基本采用 HTTP ， 而且分为 pull / push 推和拉两种方式去写采集程序 方便

- 开源，且大量的社区成品插件

这个非常厉害，<https://prometheus.io> 很多prometheus 社区开发的差距已经异常强大和完善 如果 公司对监控要求的不是特别高的话，默认的几个成品插件 装上就可以用到底了 监控成型速度太快了

- push的方法 非常非常的灵活

pull push的方法 我们往后会具体介绍，这里提一句

push的这种采集方法 灵活程度超过你的想象 几乎任何形式的数据 都可以实现

- 本身自带图形调试(sql)

prometheus(查询语句)本身的就再带了 现成的图形成型界面

虽然最终肯定不能跟grafana的效果相比，但是 这种自带图形成图 可以大大帮助运维做 调试

- 最精细的数据采样

大多数市面上的开源监控 采样也就能精确到 半分钟 一分钟的程度

商品化监控产品 就更别提了 （为了缩小数据存储的成本） 有的甚至 5分钟 就是采样最小间隔

prometheus 理论上 可以达到 每1秒采集！！而且可以自行定制频率 （不过 强大的同时 其实 大米 不太建议 细到这个程度 因为数据量太大了 如果1s采样一次）

- 最新的高大上监控 很霸气 面试很有料

呵呵 这个就比较个人主义了， 运维面试的时候 监控方面 如果你懂得这么个高大上的工具，并且对各种监控算法 系统底层相关都很熟悉（prometheus逼着你必须熟悉底层） 的话，优势很大哦

比较显著的特点就介绍如上了

另外：咱们也不能盲目崇拜 也需要客观的对待prometheus

prometheus 还是又一些不足 有待于改进

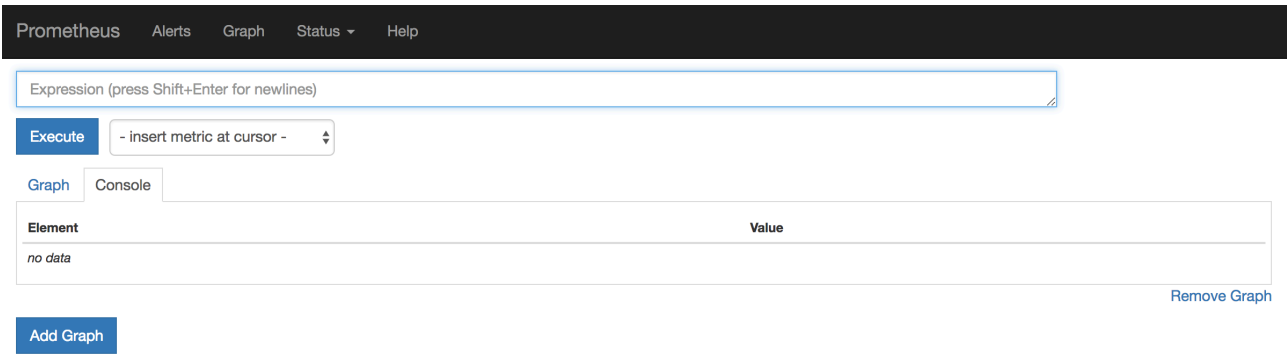
- 不支持集群化 （这个是当前最迫切的需求）
- 被监控集群过大后 本身性能有一定瓶颈（如果有集群 就可以解决这个问题）
- 偶尔发生数据丢失（这个问题 在2.0之前 会偶尔发生几次， 2.0之后貌似已经彻底解决 至少大米这几个月 没有看到丢失）
- 中文支持不好 中文资料也很少（这个问题 也是老生常谈了 往往新的 很牛的国外工具 都不太支持中文）

3) prometheus 对于运维的要求

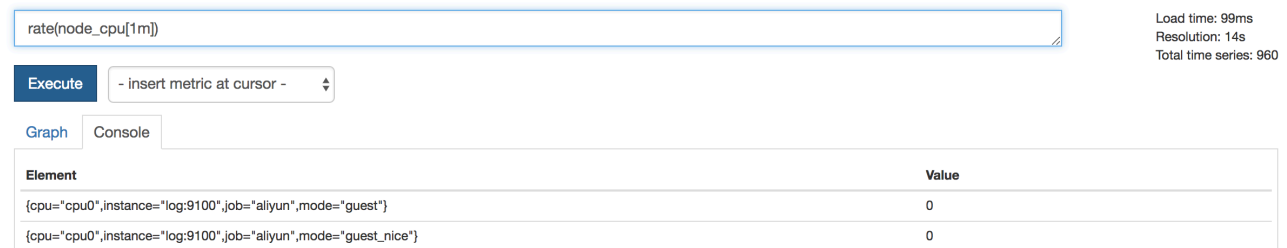
- 要求对操作系统有很深入扎实的知识 不能只是浮在表面 /proc/ nagios zabbix
- 对数学思维有一定的要求 因为它基本的内核就是数学公式组成 T-S (四则运算，算法 - > 微积分， 代数 数论)
- 对监控的经验有很高的要求 很多时候 监控项需要很细的定制

4) prometheus 各种图形展示

- prometheus主界面



- prometheus 数学查询命令行展示



$$(1 - ((\text{sum}(\text{increase}(\text{node_cpu}\{\text{mode}=\text{"idle"}\}[1\text{m}])) \text{ by } (\text{instance}))) / (\text{sum}(\text{increase}(\text{node_cpu}[1\text{m}])) \text{ by } (\text{instance}))) * 100$$
 (较复杂数学公式查询展示)

(1-((sum(increase(node_cpu(mode="idle")[1m])) by (instance)) / (sum(increase(node_cpu[1m])) by (instance)))) * 100

Execute

- insert metric at cursor -

Graph

Console

Element	Value
{instance="log:9100"}	6.317316814551887
{instance="redis2:9100"}	7.631211613599243

Load time: 39ms
Resolution: 14s
Total time series: 7

- prometheus 配置展示

PrometheusAlertsGraphStatus▼Help

Configuration

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  evaluation_interval: 15s
alerting:
  alertmanagers:
    - static_configs:
      - targets: []
      scheme: http
      timeout: 10s
scrape_configs:
  - job_name: prometheus
    scrape_interval: 15s
    scrape_timeout: 10s
    metrics_path: /metrics
    scheme: http
    static_configs:
      - targets:
        - prometheus.server:9090
        - prometheus.server:9100
  - job_name: pushgateway
    scrape_interval: 15s
    scrape_timeout: 10s
    metrics_path: /metrics
    scheme: http
    static_configs:
      - targets:
        - localhost:9091
        - localhost:9092
```

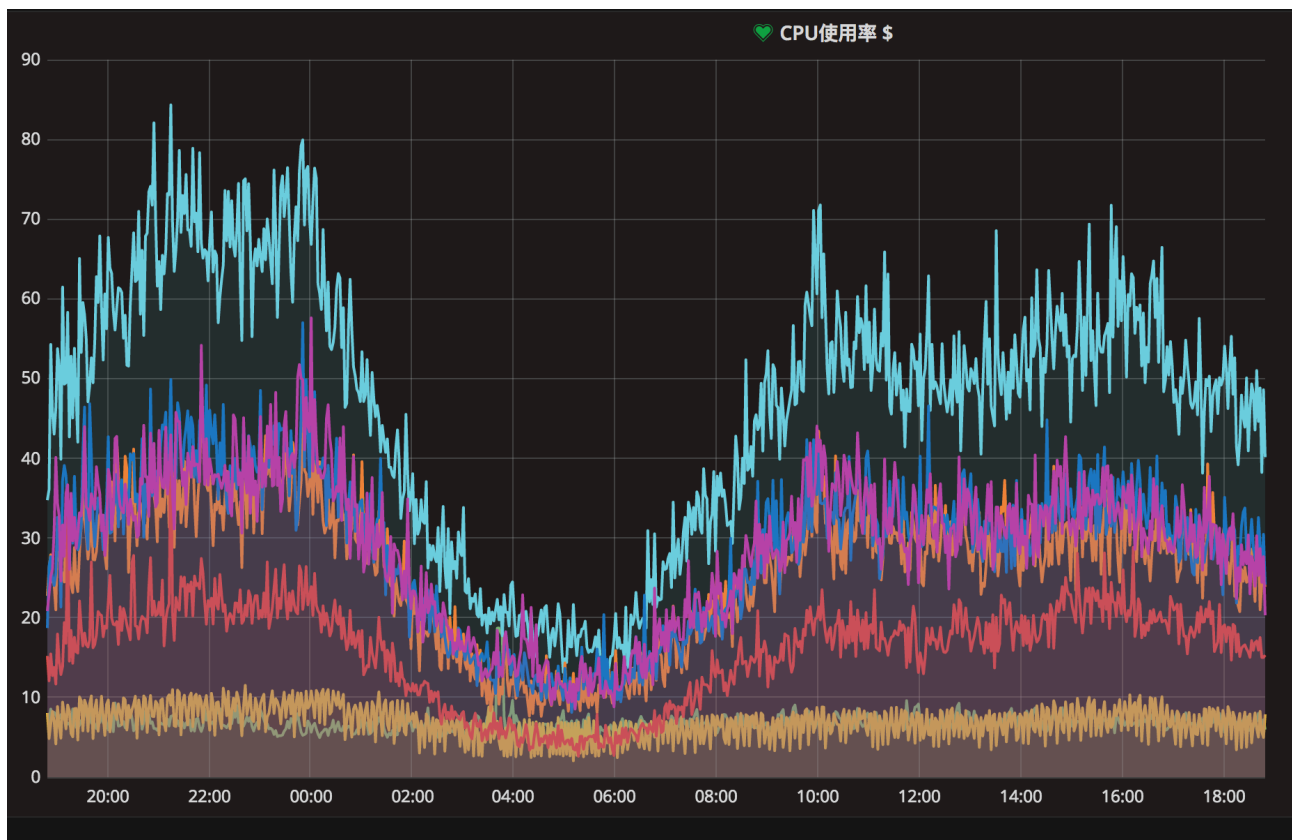
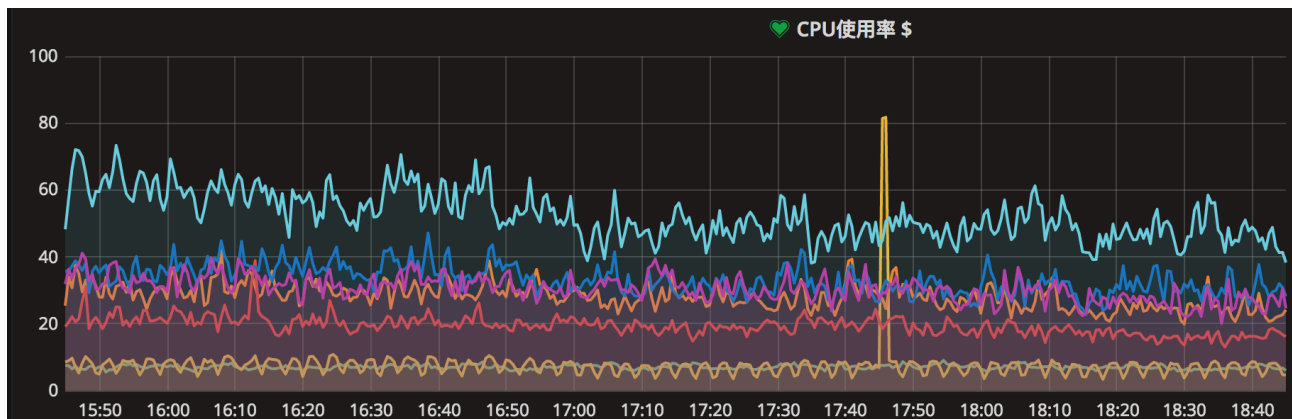
- prometheus targets展示（被监控节点）

Targets

aliyun (7/7 up)

Endpoint	State	Labels	Last Scrape	Error
http://log:9100/metrics	UP	instance="log:9100"	1.507s ago	
http://redis2:9100/metrics	UP	instance="redis2:9100"	8.976s ago	

- prometheus + grafana 监控CPU展示



- prometheus 文件描述符监控 展示

