

# 分布式系统调研 3

## Lua

### 简介

lua 是一款轻量级的应用，可以运行在服务端，一种轻量级的小巧的脚本语言，一般用于做游戏加速引擎、异步消息处理，方便做相关的扩展调用，提供灵活的调用。Lua 有着相对简单的 C API 而很容易嵌入应用中。很多应用程序使用 Lua 作为自己的嵌入式脚本语言，以此来实现可配置性、可扩展性。其内部内置了很多组件，方便做服务端的扩展，如：redis、db、traffic limit 限流等，可以运行在 nginx 上，做相关的扩展，在此基础上做反向 dialing、流量控制，优化了底层的实现流程。在用法上和 C 类似，简单的流程控制、函数、元素、表达式、元操作、协同程序、文件流等操作。

### Lua 特性

**轻量级:** 它用标准 C 语言编写并以源代码形式开放，编译后仅仅一百余 K，可以很方便的嵌入别的程序里。

**可扩展:** Lua 提供了非常易于使用的扩展接口和机制：由宿主语言(通常是 C 或 C++)提供这些功能，Lua 可以使用它们，就像是本来就内置的功能一样。

**其它特性:**

支持面向过程(procedure-oriented)编程和函数式编程(functional programming)；

自动内存管理；只提供了一种通用类型的表 (table)，用它可以实现数组，哈希表，集合，对象；

语言内置模式匹配；闭包(closure)；函数也可以看做一个值；提供多线程（协同进程，并非操作系统所支持的线程）支持；

通过闭包和 table 可以很方便地支持面向对象编程所需要的一些关键机制，比如数据抽象，虚函数，继承和重载等。

### Lua 教程

<https://www.runoob.com/lua/lua-tutorial.html>

## OpenResty

OpenResty（也称为 ngx\_openresty）是一个全功能的 Web 应用服务器。它打包了标准的 Nginx 核心，很多的常用的第三方模块，以及它们的大多数依赖项。

通过揉和众多设计良好的 Nginx 模块，OpenResty 有效地把 Nginx 服务器转变为一个强大的 Web 应用服务器，基于它开发人员可以使用 Lua 编程语言对 Nginx 核心以及现有的各种 Nginx C 模块进行脚本编程，构建出可以处理一万以上并发请求的极端高性能的 Web 应用。

OpenResty 致力于将你的服务器端应用完全运行于 Nginx 服务器中,充分利用 Nginx 的事件模型来进行非阻塞 I/O 通信。不仅仅是和 HTTP 客户端间的网络通信是非阻塞的,与 MySQL、PostgreSQL、Memcached 以及 Redis 等众多远方后端之间的网络通信也是非阻塞的。

因为 OpenResty 软件包的维护者也是其中打包的许多 Nginx 模块的作者,所以 OpenResty 可以确保所包含的所有组件可以可靠地协同工作。

官方网站: <https://openresty.org/cn/>

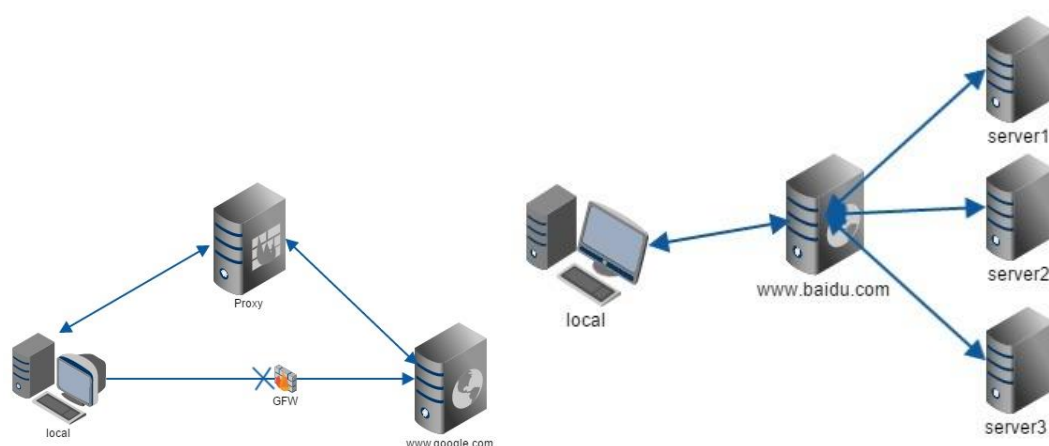
Nginx: 是一个免费、开源、高性能的 HTTP 服务器和反向代理,以及 IMAP/POP3 代理服务器。

*附: Lua + Openresty + Nginx 教程*

<https://moonbingbing.gitbooks.io/openresty-best-practices/content/base/intro.html>

*附: 代理服务器与反向代理服务器*

<https://www.zhihu.com/question/24723688>



## Prometheus

Prometheus 是一个开源的监控软件。非常适合用来监控 docker 容器。

### 简介

Prometheus 是一个开源系统监控和警报工具包,最初在 SoundCloud 构建。自 2012 年成立以来,许多公司和组织都采用了 Prometheus,该项目拥有非常活跃的开发者和用户社区。它现在是一个独立的开源项目,独立于任何公司维护。Prometheus 将其指标收集并存储为时间序列数据,即指标信息与记录时的时间戳以及称为标签的可选键值对一起存储。

### 特点

具有由度量名称和键/值对标识的时间序列数据的多维数据模型

PromQL, 一种 利用这种维度的灵活查询语言

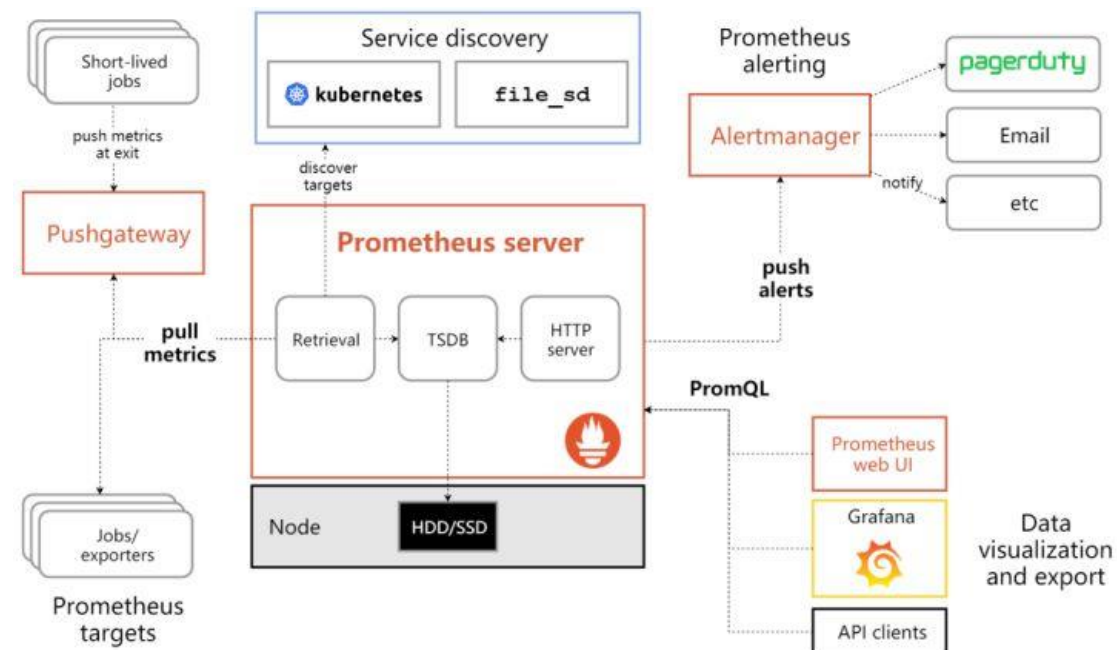
不依赖分布式存储; 单个服务器节点是自治的

时间序列收集通过 HTTP 上的拉模型进行

通过中间网关支持推送时间序列

通过服务发现或静态配置发现目标  
多种图形模式和仪表板支持

## 架构图



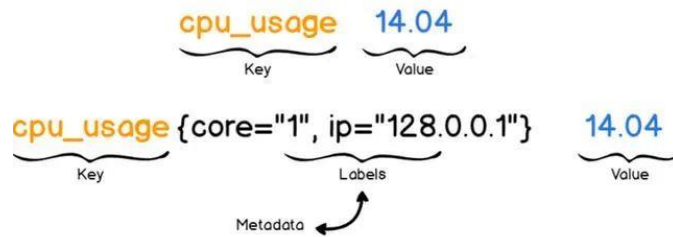
- Prometheus Server: 用数据的采集和存储, PromQL 查询, 报警配置。
- Push gateway: 用于批量, 短期的监控数据的汇报总节点。
- Exporters: 各种汇报数据的 exporter, 例如汇报机器数据的 node\_exporter, 汇报 MongoDB 信息的 MongoDB\_exporter 等等。
- Alertmanager: 用于高级通知管理。

注: 架构图里能看到两个 Pull metrics, 一个是采集器直接被 Server 拉取数据 (pull); 另一个是采集器主动 Push 数据到 Push Gateway, Server 再对 Push Gateway 主动拉取数据 (pull)。

## 工作流程 (采集监控数据)

1. Prometheus server 定期从静态配置的主机或服务发现的 targets 拉取数据 (zookeeper, consul, DNS SRV Lookup 等方式)
2. 当新拉取的数据大于配置内存缓存区的时候, Prometheus 会将数据持久化到磁盘, 也可以远程持久化到云端。
3. Prometheus 通过 PromQL、API、Console 和其他可视化组件如 Grafana、Promdash 展示数据。
4. Prometheus 可以配置 rules, 然后定时查询数据, 当条件触发的时候, 会将告警推送到配置的 Alertmanager。
5. Alertmanager 收到告警的时候, 会根据配置, 聚合, 去重, 降噪, 最后发出警告。

## 数据结构: Key\_Value



## 2 Data Model Filtering



Key 是指标名字，Value 是该指标的值，此外 Metadata(元信息)也非常重要，也可称之为 labels(标签信息)。这些标签信息指定了当前这个值属于哪个云区域下的哪台机器，如果没有 labels，数据有可能会被丢失。

## 监控指标的 4 种基本类型

### 1.Counter(计数器)

计数器最简单的指标类型。比如你想统计某个网站的 HTTP 错误总数，这时候就用计数器。计数器的值只能增加或重置为 0，因此特别适合计算某个时段上某个时间的发生次数，即指标随时间演变发生的变化。

### 2.Gauges

Gauges 可以用于处理随时间增加或减少的指标，比如内存变化、温度变化。这可能是最常见的指标类型，不过它也有一定缺点：如果系统每 5 秒发送一次指标，普罗服务每 15 秒抓取一次数据，那么这期间可能会丢失一些指标，如果你基于这些数据做汇总分析计算，则结果的准确性会有所下滑。

### 3.Histogram(直方图)

直方图是一种更复杂的度量标准类型。它为我们的指标提供了额外信息，例如观察值的总和及其数量，常用于跟踪事件发生的规模。比如，为了监控性能指标，我们希望在有 20% 的服务器请求响应时间超过 300 毫秒时发送告警。对于涉及比例的指标就可以考虑使用直方图。

### 4.Summary(摘要)

摘要更高级一些，是对直方图的扩展。除了提供观察的总和和计数之外，它们还提供滑动窗口上的分位数度量。分位数是将概率密度划分为相等概率范围的方法。

注：Summary 和 Histogram 对比

- 直方图随时间汇总值，给出总和和计数函数，使得易于查看给定指标的变化趋势。
- 而摘要则给出了滑动窗口上的分位数（即随时间不断变化）。

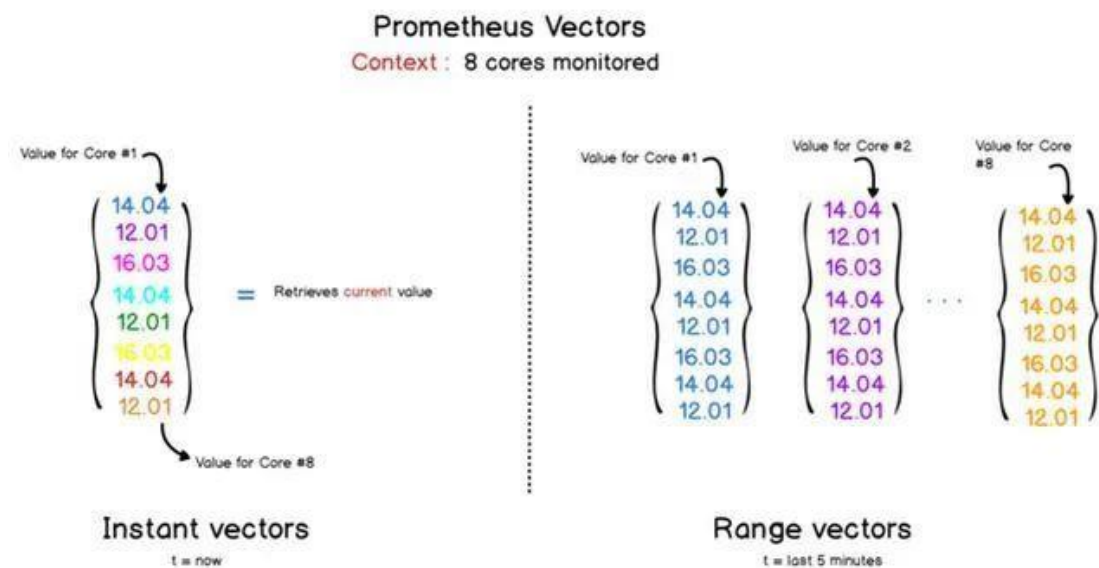
## 数据提取

普罗米修斯内置了自己的 SQL 查询语言用于查询和检索数据，这个内置的语言就是 PromQL。普罗米修斯的数据是用键值对表示的。PromQL 也用相同的语法查询和返回结果集。

PromQL 会处理两种向量：

即时向量：表示当前时间，某个指标的数据向量。

时间范围向量：表示过去某时间范围内，某个指标的数据向量。



针对 8 核 CPU 的使用率

## 可视化数据

Grafana 是一个大型可视化系统，功能强大，可以创建自己的自定义面板，支持多种数据来源，当然也支持普罗米修斯。

通过配置数据源，Grafana 会使用相应的 SQL 拉取并绘制图表，能直接看到普罗米修斯的各个指标数据图表：



监控操作教程: [https://blog.51cto.com/u\\_12082223/3241875](https://blog.51cto.com/u_12082223/3241875)

Grafana: <https://grafana.com/docs/>

Prometheus Server: <https://github.com/prometheus/prometheus#install>

Prometheus Push Gateway: <https://github.com/prometheus/pushgateway>

Prometheus book: <https://yunlzheng.gitbook.io/prometheus-book/introduction>

附: PromQL

<https://yunlzheng.gitbook.io/prometheus-book/parti-prometheus-ji-chu/promql/prometheus-query-language>

PromQL 是 Prometheus 内置的数据查询语言，其提供对时间序列数据丰富的查询，聚合以及逻辑运算能力的支持。并且被广泛应用在 Prometheus 的日常应用当中，包括对数据查询、可视化、告警处理当中。

**PromQL 会处理两种向量：**

即时向量：表示当前时间，某个指标的数据向量。

时间范围向量：表示过去某时间范围内，某个指标的数据向量。

## InfluxDB

时序数据库 InfluxDB 版是一款专门处理高写入和查询负载的时序数据库，用于存储大规模的时序数据并进行实时分析，包括来自 DevOps 监控、应用指标和 IoT 传感器上的数据。

### 特点

- 专为时间序列数据量身打造的高性能数据存储。TSM 引擎提供数据高速读写和压缩等功能。
- 简单高效的 HTTP API 写入和查询接口。
- 针对时序数据，量身打造类似 SQL 的查询语言，轻松查询聚合数据。
- 允许对 tag 建索引，实现快速有效的查询。
- 数据保留策略（Retention policies）能够有效地使旧数据自动失效。

influxDB中的名词	传统数据库中的概念
database	数据库
measurement	数据库中的表
points	表里面的一行数据

Point由时间戳（time）、数据（field）、标签（tags）组成。

Point相当于传统数据库里的一行数据，如下表所示：

Point属性	传统数据库中的概念
time	每个数据记录时间，是数据库中的主索引（会自动生成）
fields	各种记录值（没有索引的属性）也就是记录的值：温度，湿度
tags	各种有索引的属性：地区，海拔

InfluxDB 中文文档：<https://jasper-zhang1.gitbooks.io/influxdb/content/>