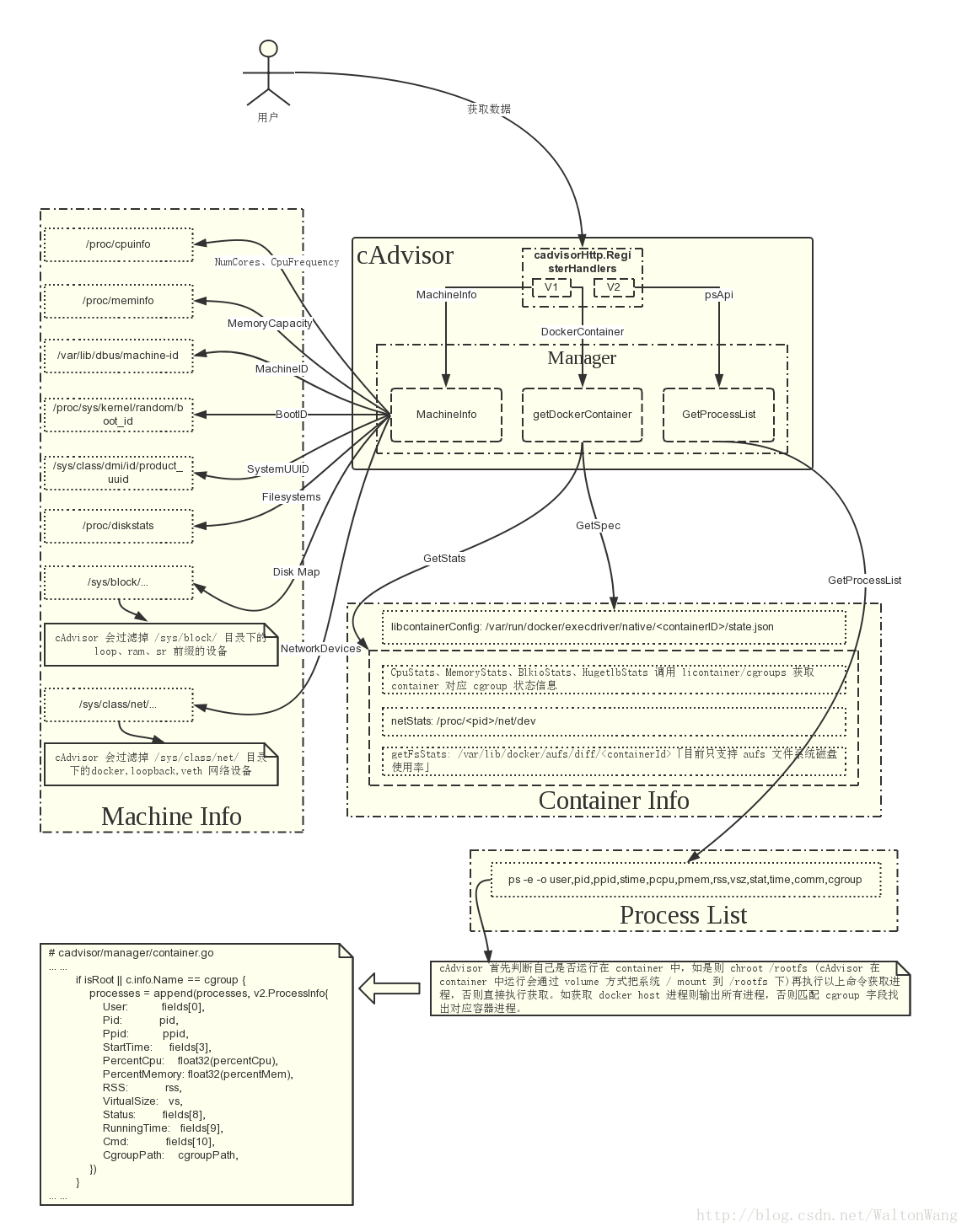
容器监控

1. **cAdvisor**

**简介**：cAdvisor 是谷歌公司用来分析运行中的Docker容器的资源占用以及性能特性的工具,它会收集、聚集、处理并导出运行中容器的信息。这些信息能够包含容器级别的资源隔离参数、资源的历史使用状况、反映资源使用和网络统计数据完整历史状况。

**原理**：



CAdvisor运行时挂载了宿主机根目录，docker根目录等多个目录，由此可以从中读取容器的运行时信息。docker基础技术有Linux namespace，Control Group(CGroup)，AUFS等，其中CGroup用于系统资源限制和优先级控制的。

宿主机的/sys/fs/cgroup/目录下面存储的就是CGroup的内容了，CGroup包括多个子系统，如对块设备的blkio，cpu，内存，网络IO等限制。Docker在CGroup里面的各个子系统中创建了docker目录，而CAdvisor运行时挂载了宿主机根目录和 /sys目录，从而CAdvisor可以读取到容器的资源使用记录。比如下面可以看到容器b1f257当前时刻的CPU的使用统计。

而容器网络流量CAdvisor是从/proc/PID/net/dev中读取的，如上面的容器b1f257进程在宿主机的PID为6748，可以看到容器所有网卡的接收和发送流量以及错误数等。CAdvisor定期读取对应目录下面的数据并定期发送到指定的存储引擎存储，而本地会默认存储最近2分钟的数据并提供UI界面查看。

**CGroup** ：CGroup 是 Control Groups 的缩写，是 Linux 内核提供的一种可以限制、记录、隔离进程组 (process groups) 所使用的物理资源 (如 cpu memory i/o 等等) 的机制。2007 年进入 Linux 2.6.24 内核，CGroups 不是全新创造的，它将进程管理从 cpuset 中剥离出来，作者是 Google 的 Paul Menage。CGroups 也是 LXC 为实现虚拟化所使用的资源管理手段。

**特点**：有界面 运行方便 能监控单个容器的CPU 、内存、网络情况

1. 易于部署程度：※※※※※

2. 信息详细程度：※※

3. 集成度：※

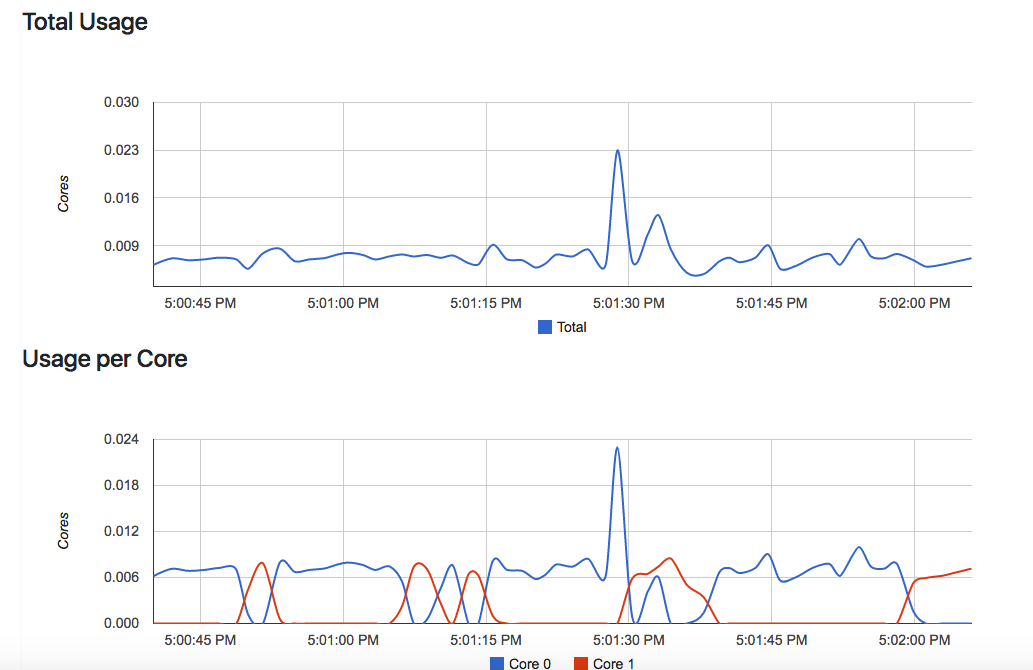
4. 生成警报的能力：无

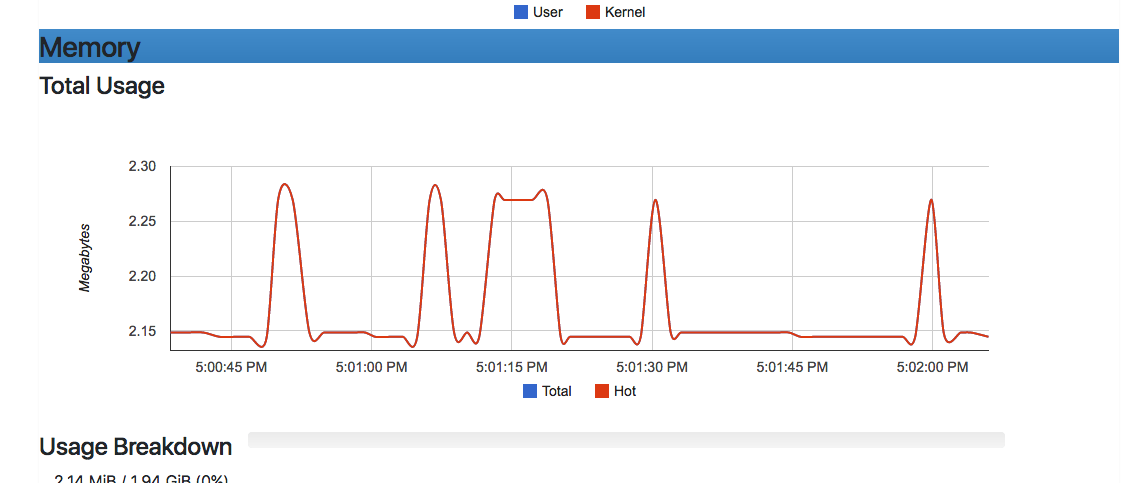
5. 监测非Docker的资源的能力：无

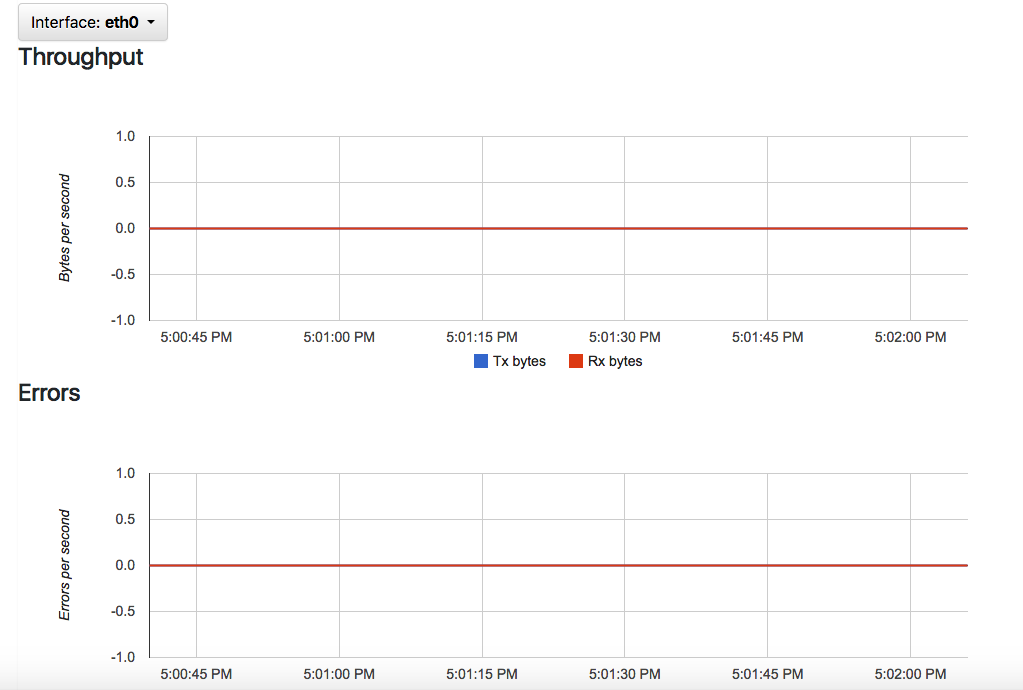
6. 成本：免费

**缺点**：不能监控远程容器，只能监控当前cAdvisor服务器情况，如果多个服务器，则需要每个服务器都得运行cAdvisor

界面：







集成前端：<https://www.jianshu.com/p/9e47ffaf5e31>

Debian默认没有开启 CGroup Memory的支持，CAdvisor默认情况下无法统计到容器内存数据，需要修改GRUB启动参数，修改文件/etc/default/grub，加入下面这行：



然后更新grub2重启即可。



1. **DataLog**

**工作原理**：Datadog的工作方式是在每一台需要监控的服务器上运行它的Agent。Agent不但会收集这台服务器的各类基础性能数据，如CPU使有率，剩余内存空间，剩余磁盘空间，网络流量等，也可以收集用户自定义的性能数据，灵活性很好。Datadog 代理可以直接使用 cgroup 的指标报告每十五秒收集一次 CPU 、内存、网络和 IO 数据。

Datadog另一个好用的地方在于它与众多的云服务和开源项目有整合，我们在实际用的整合有



Datadog也可以对各种监控的性能指标设定阈值，当指标超出阈值范围时，发出警报。

功能：可以方便地自定义指标。这些自定义的指标可以是Web server对每个请求的响应时间，或是数据库中每张表的读写请求数，也可以是Job queue里pending的job数量。

Datadog的自定义指标有两种类型Count和Histogram。Count比较直观，就是用于记数，用来统计某个事件在一个时间区间内发生了多次次，例如我们可以定义一个page\_view来记录应用中每个页面的访问次数。但在性能监控中，Histogram更为有用，它的每个数据点是一个时间戳加上一个浮点值，所以它不但可以统计某个事件在一个时间区间内发生的次数，也可以统计这些事件所对应的浮点值的平均值，中位数，最大值，最小值等等。如果这听起来还是比较抽象，一个实际的例子就是记录服务器对每个请求的响应时间。用Histogram来记录，我们就可以方便地得到某个时间段的平均响应时间和最长响应时间，而这两个数据对于性能监控至关重要。

对发送到Datadog的每个数据点，我们都可以添加多个tag，这对于之后数据的查询与分类非常有帮助，比如我们可以根据数据来自哪个microservice，给它们打上不同的tag。在实际使用中，我们会以更细的粒度在数据点上添加tag。

可以监控报警

1. **ServerStatus探针**



需要安装服务器端和客户端

<https://blog.bbskali.cn/1825.html>

https://www.cmsky.com/serverstatus/

<https://github.com/cppla/ServerStatus>



特点：只能监控物理机，不能监控docker容器情况，免费，部署方便

探针是用来探测空间、服务器运行状况和PHP信息用的，探针可以实时查看服务器硬盘资源、内存占用、网卡流量、系统负载、服务器时间等信息。

ServerStatus中文版是一个酷炫高逼格的云探针、云监控、服务器云监控、多服务器探针~，该云监控（云探针）是ServerStatus（ https://github.com/BotoX/ServerStatus ）项目的中文（优化）版。

它是一种Web脚本程序，实质上是通过网页编程语言（ASP、PHP、ASP.NET等）实现探测服务器敏感信息的脚本文件，通常用于探测网站目录、服务器操作系统、CPU数量、Session超时时间、服务器Session变量、 服务器Application变量、组件支持情况、运算速度、磁盘读写速度以及网络连接速度等。

缺点：探针最耗资源的一项就是浮点运算能力检测了，这需要占用大量的CPU去进行运算，点一次，可能占用的资源不太多，但是，如果你连续点击多次的话……我的CPU是I3 2120，CPU占用已经到达了100%，而且这时网站也无法正常访问了。直到浮点运算结果出来后，才能正常加载。

资料来源：<https://www.mywpku.com/tanzhen.html>

1. **Scout**

Scout是一个新的服务器监端控应用程序，允许自动监控关键指标，拥有80多个插件用于监控您的应用程序，实时（每秒）流式仪表板并提供警报功能。

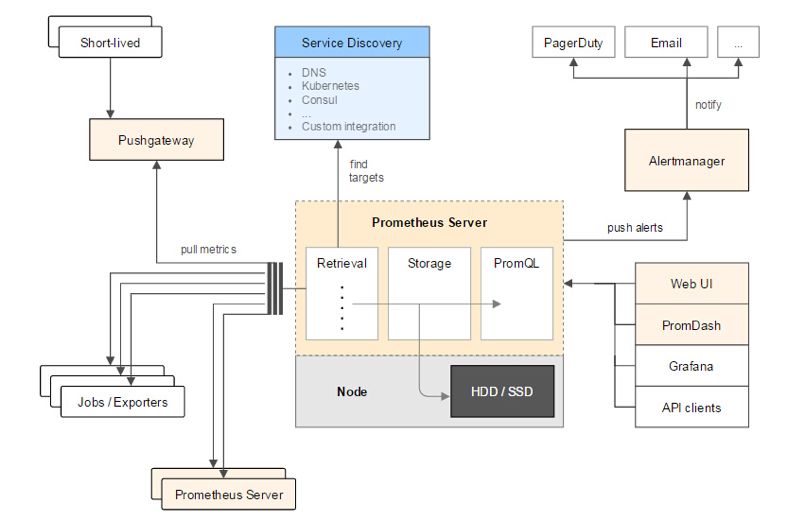
其中“docker-scout”是一个插件，是基于Scout应用程序的容器，你可以使用它来监视您docker环境中的主机和容器。Scout为您提供了在主机上运行的所有内容的详细信息，例如CPU使用情况和内存使用情况，这些并不仅限于在docker内运行的特定进程。

在配置了“docker-scout”插件后，可以从容器中导入数据，如CPU使用率、内存使用率网络吞吐量以及每个主机的容器数量。数据以统计形式以及图形形式提供。

缺点:与cAdvisor不同，它不提供每个主机上运行的各个容器的详细信息。如果您在同一台服务器上托管不同的容器，这是一个严重的限制。另外，这些附加功能的价格，每个监控主机十美元。

1. **Prometheus**

架构图



安装运行

docker run -d -p 9090:9090 --name prometheus prom/prometheus

查看状态信息

localhost:9090/status

查看metrics信息

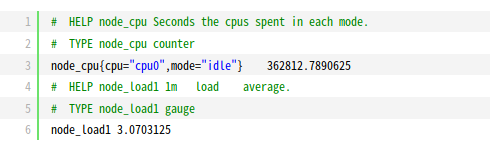
<http://localhost:9090/metrics>

**常用概念**

prometheus采集到的监控数据均以metric（指标）形式保存在时序数据库中（TSDB）

每一条时间序列由 metric 和 labels 组成，每条时间序列按照时间的先后顺序存储它的样本值。

默认情况下各监控client向外暴露一个HTTP服务，prometheus会通过pull方式获取client的数据，数据格式如下：



以#开头的表示注释信息，解释了每一个指标的监控目的和类型

node\_cpu表示监控指标的名称

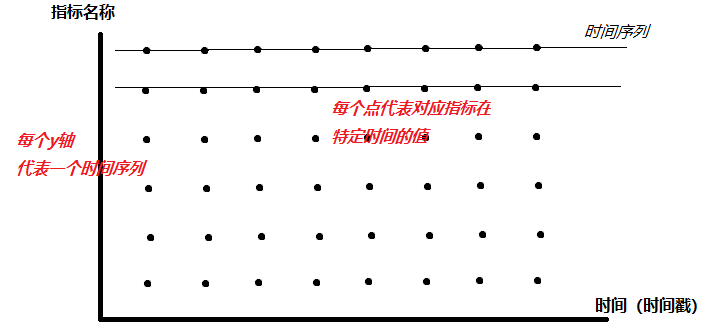
{}内的内容是标签，以键值对的方式记录

数字是这个指标监控的数据

下图横坐标代表的是时间（时间戳的方式记录在TSDB中），纵坐标代表了各种不同的指标名称，坐标系中的黑点代表了各个指标在不同时间下的值。

每一个横线 就是时间序列

每个黑点就是样本（prometheus将样本以时间序列的方式保存在内存中，然后定时保存到硬盘上）





搭建

https://blog.51cto.com/10158955/2167075

1. **aa**