

How to use collectd

1.简介

collectd是一个守护(daemon)进程，用来定期收集**系统和应用程序的性能指标**，同时提供了以不同的方式来**存储**这些指标值的机制。

collectd从各种来源收集指标，例如 操作系统，应用程序，日志文件和外部设备，并存储此信息或通过网络使其可用。 这些统计数据可用于**监控系统**、**查找性能瓶颈**（即性能分析）并**预测未来的系统负载**（即容量规划）等。

• 优点:

由C语言编写(性能很好 可移植性高)、可运行在嵌入式系统上、包含100多种插件(并提供强大的网络特性)

• 缺点:

本身不能生成图形、监控功能只能进行简单的门阀检测()

2.使用

• 下载

```
wget https://storage.googleapis.com/collectd-tarballs/collectd-5.7.2.tar.bz2
```

```
tar xf collectd-5.7.2.tar.bz2
```

• 编译

```
cd collectd-5.7.2
```

```
./configure #编译
```

```
--prefix=/usr --sysconfdir=/etc --localstatedir=/var --libdir=/usr/lib --  
mandir=/usr/share/man --enable-all-plugins(可在编译的同时加一些选项)
```

• 安装

```
make all install #安装 (或者直接执行命令安装 sudo apt-get install collectd)
```

3.目录结构

• 配置文件目录

```
配置文件目录： /etc/collectd/collectd.conf
```

• 启动文件目录

```
启动文件目录： /etc/init.d/collectd
```

• 日志文件目录

日志文件目录：`/var/log/syslog`

• 数据存储目录

数据存储目录：`/var/lib/collectd/rrd/`

4.配置

```
sudo vim /opt/collectd/etc/collectd.conf
```

监控系统所用的插件有:cpu,memory,processes,load,interface,disk,swap等

一般只需要修改**network**插件这一项

因为网络插件可以发送到collectd的**远程实例**中(数据库, csv文件或缓存等介质中),或者接受从远程服务端发来的数据

```
LoadPlugin network          #去掉#就表示载入该插件
<Plugin network>
#      # client setup:
#      Server "10.24.106.1" "25826" #该地址和端口是接收数据的服务器的地址和端口,例如:
#      安装数据库(influxdb等)的服务器的地址和端口
#      <Server "239.192.74.66" "25826"> #若需传输加密数据,就配置这一小块部分
#          SecurityLevel Encrypt
#          Username "user"
#          Password "secret"
#          Interface "eth0"
#          ResolveInterval 14400
#      </Server>
#      TimeToLive 128
#
#      # server setup:
#      Listen "ff18::efc0:4a42" "25826" #如配置这一部分,表示接收从其他collectd实例中
#      发来的数据
#      <Listen "239.192.74.66" "25826">
#          SecurityLevel Sign
#          AuthFile "/etc/collectd/passwd"
#          Interface "eth0"
#      </Listen>
#      MaxPacketSize 1452
#
#      # proxy setup (client and server as above):
#      Forward true
#
#      # statistics about the network plugin itself
#      ReportStats false
#
#      # "garbage collection"
#      CacheFlush 1800
</Plugin>
```

ps:若要监控cluster(集群),则**每台服务器**都需要**下载安装**collectd并进行相应的配置。

5.启动

执行命令 `sudo /etc/init.d/collectd start` 或者 `systemctl start collectd`

并设置开机启动 `systemctl enable collectd`

6.查看数据

若开启了rrdtool插件就可在 `/var/lib/collectd/rrd/` 目录下看到相应的统计数据。

通过下面命令可以查看具体数据：

```
rrdtool fetch *.rrd AVERAGE
```

*.rrd表示任何以.rrd结尾的文件，rrdtool命令更详细的用法可以自行百度。

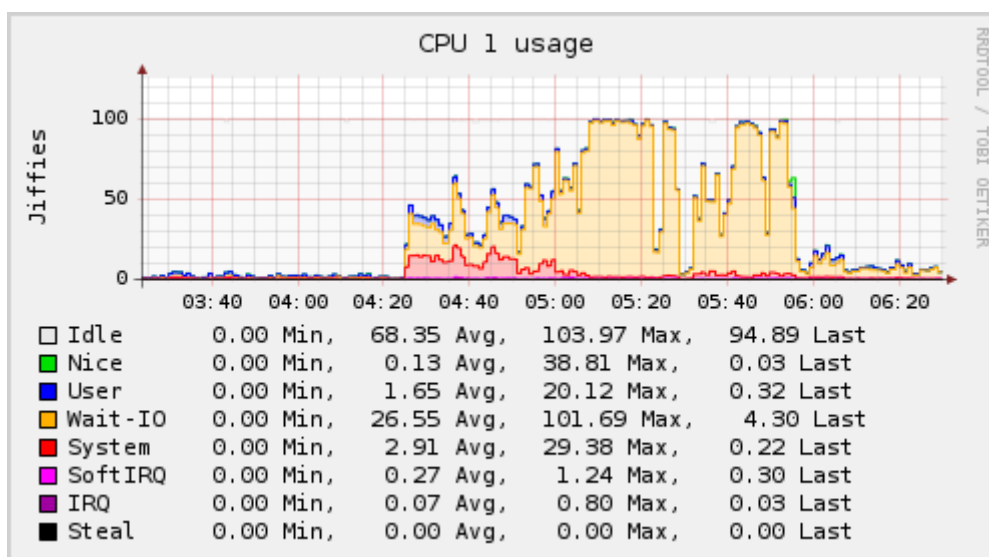
能够看到第一列为timestamp(时间戳)

可以通过命令 `date -d @timestamp` 把timestamp（以秒为单位）转换为和 `date` 命令显示的相同的时间格式,命令 `date +%s` 用时间戳的形式表示时间。

7.主要插件介绍

- Plugin: CPU

可视化显示



cpu

指标释义

jiffies: 是一个单位，jiffies是内核中的一个全局变量，用来记录自系统启动以来产生的节拍数，在linux中，一个节拍大致可理解为操作系统进程调度的最小时间单位，不同linux内核可能值有不同，通常在1ms到10ms之间

user: 从系统启动开始累计到当前时刻，处于用户态的运行时间，不包含 nice值为负的进程。

nice: 从系统启动开始累计到当前时刻，**nice**值为负的进程所占用的CPU时间。

idle: 从系统启动开始累计到当前时刻，除I/O等待时间以外的其它等待时间。

wait-io: 从系统启动开始累计到当前时刻，I/O操作等待时间。

system: 从系统启动开始累计到当前时刻，处于内核态的运行时间。

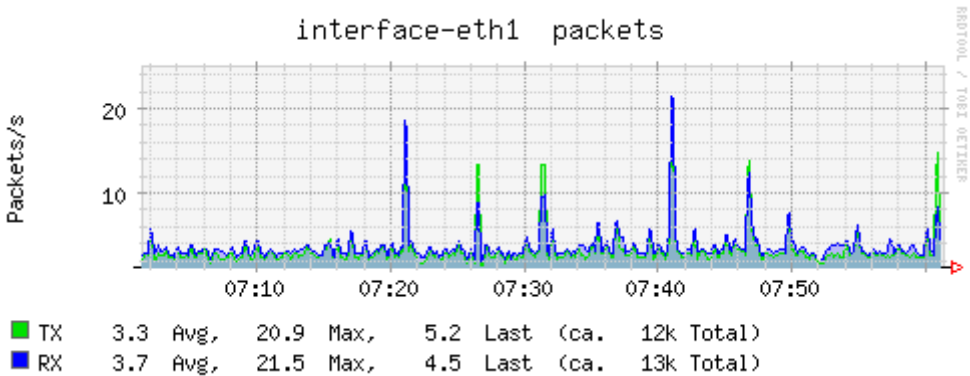
softIRQ: 从系统启动开始累计到当前时刻，软中断时间。

IRQ:从系统启动开始累计到当前时刻，硬中断时间。

steal:运行在虚拟环境中其他操作系统所花费的时间。

- Plugin: interface

Interface插件收集关于流量（每秒八位字节），每秒的数据包和接口错误（一秒钟内）的信息。



interface

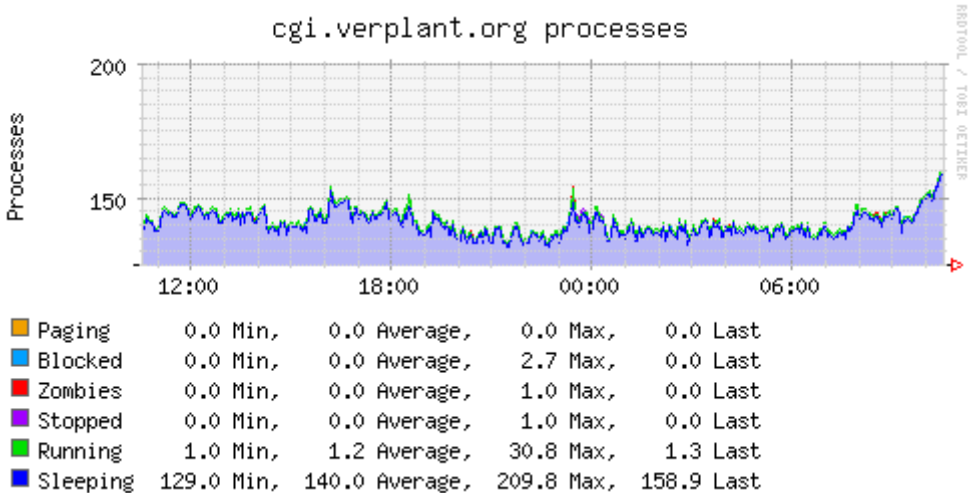
rxpck/s：每秒钟接收的数据包

txpck/s：每秒钟发送的数据包

rxbyt/s：每秒钟接收的字节数

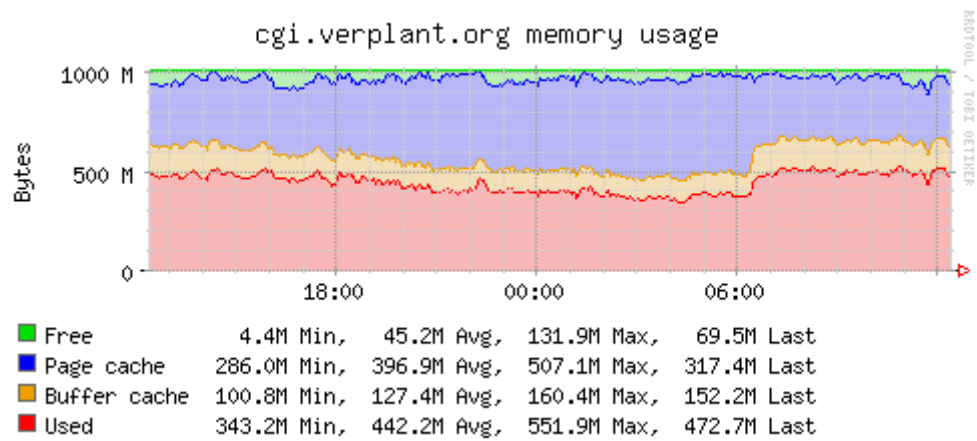
txbyt/s：每秒钟发送的字节数

- Plugin: processes



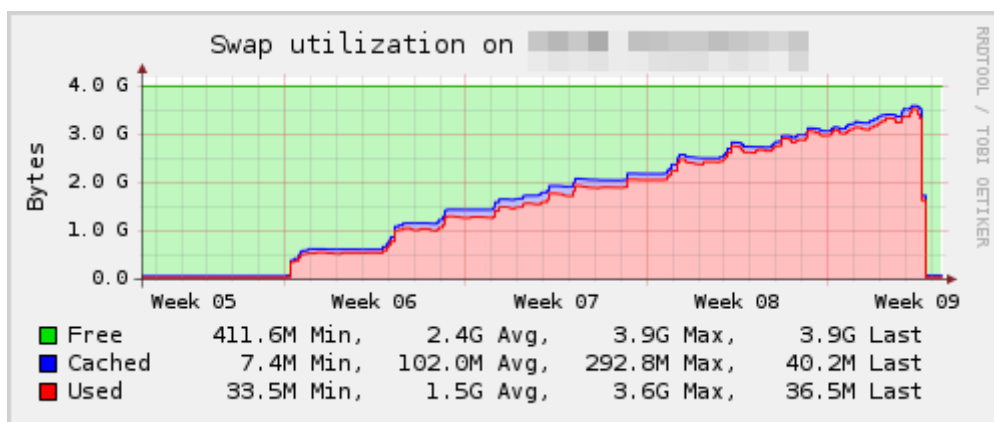
processes

- Plugin: memory 收集系统的物理内存利用率



processes

- Plugin: swap 收集swap空间的使用情况



swap

- Plugin: df 统计文件系统的使用信息
- Plugin: irq 收集操作系统处理中断的数量
- Plugin: disk 收集磁盘的性能统计信息
- Plugin: load 收集系统负载信息，定义为队列中可运行任务的数量

在Linux中，系统相关数据可以通过命令 `cd /proc/` 在该目录下查看。

8.自定义插件

python插件将python解释器嵌入到collectd中并且提供了collectd插件系统接口。