在自已编写监控插件之前我们首先需要对nagios监控原理有一定的了解

Nagios的功能是监控服务和主机，但是他自身并不包括这部分功能，所有的监控、检测功能都是通过各种插件来完成的。

启动Nagios后，它会周期性的自动调用插件去检测服务器状态，同时Nagios会维持一个队列。

所有插件返回来的状态信息都进入队列，Nagios每次都从队首开始读取信息，并进行处理后，把状态结果通过web显示出来。

Nagios提供了许多插件，利用这些插件可以方便的监控很多服务状态。安装完成后，在nagios主目录下的/libexec里放有nagios自带的可以使用的所有插件

如，check\_disk是检查磁盘空间的插件，check\_load是检查CPU负载的，等等。基本上每一个插件可以通过运行./check\_xxx –h 来查看其使用方法和功能。

**Nagios可以识别4种状态返回信息：**

**0(OK)表示状态正常/绿色、**

**1(WARNING)表示出现警告/黄色**

**2(CRITICAL)表示出现非常严重的错误/红色**

**3(UNKNOWN)表示未知错误/深黄色。**

Nagios根据插件返回来的值，来判断监控对象的状态，并通过web显示出来，以供管理员及时发现故障。四种监控状态如下图所示：



当我们知道了nagios是通过命令返回值来判断状态，在shell中也即是通过命令的退出状态码来判断，这样我们可以自已编写一个监控脚本

如编写**监控postgresql数据库的ESTABLISHED连接数**，可以如下的命令及语法进行编写。

[复制代码](javascript:void(0);)

#!/bin/bash

#

if [ "$1" = "-w" ] && [ "$2" -gt "0" ] && [ "$3" = "-c" ] && [ "$4" -gt "0" ]; then

pgsql=`netstat -lnaput| grep ":5432" | grep ESTABLISHED | wc -l`

if [ "$pgsql" -ge "$4" ]; then

echo "Pgsql\_concurrent: CRITICAL Total: $pgsql - concurrent\_count |USED=$pgsql;200;500;;"

$(exit 2)

elif [ "$pgsql" -ge "$2" ]; then

echo "Pgsql\_concurrent: WARNING Total: $pgsql - concurrent\_count|USED=$pgsql;200;500;;"

$(exit 1)

else

echo "Pgsql\_concurrent: OK Total: $pgsql - concurrent\_count|USED=$pgsql;200;500;;"

$(exit 0)

fi

else

echo "check\_PgsqlCount"

echo ""

echo "Usage:"

echo "check\_PgsqlCount -w <warn count> -c <crit count>"

echo ""

echo "Copyright (C) 2014 http://www.cnblogs.com/5201351"

exit

fi

[复制代码](javascript:void(0);)

以上监控脚本为笔者工作所编写，在此作为一种编写nagios监控插件的思路，其中在if语句中的3条echo语句中、我们可以发现输出的内容是由**|**分隔

其中只写入"|"之前的内容也行，"|"之后的的内容为可选见容，**nagios会将"|"之后的内容作为性能数据输出**。如可输出给pnp4nagios，性能数据格式如下：

'label'=value[UOM];[warn];[crit];[min];[max]

需要注意的是：

**1、性能数据的多个选项值之间用分号;分割**

2、如果label中包含空格、等号、或者单引号，则label需要用单引号来括起来

**3、warn/crit/min/max可以为null值,如果有值单位必须统一**

4、如果UOM单位是%，则min和max不需要再指定

5、UOM单位可以是如下： 默认空，表示数量（用于用户数、处理器数等）