大米運維課堂

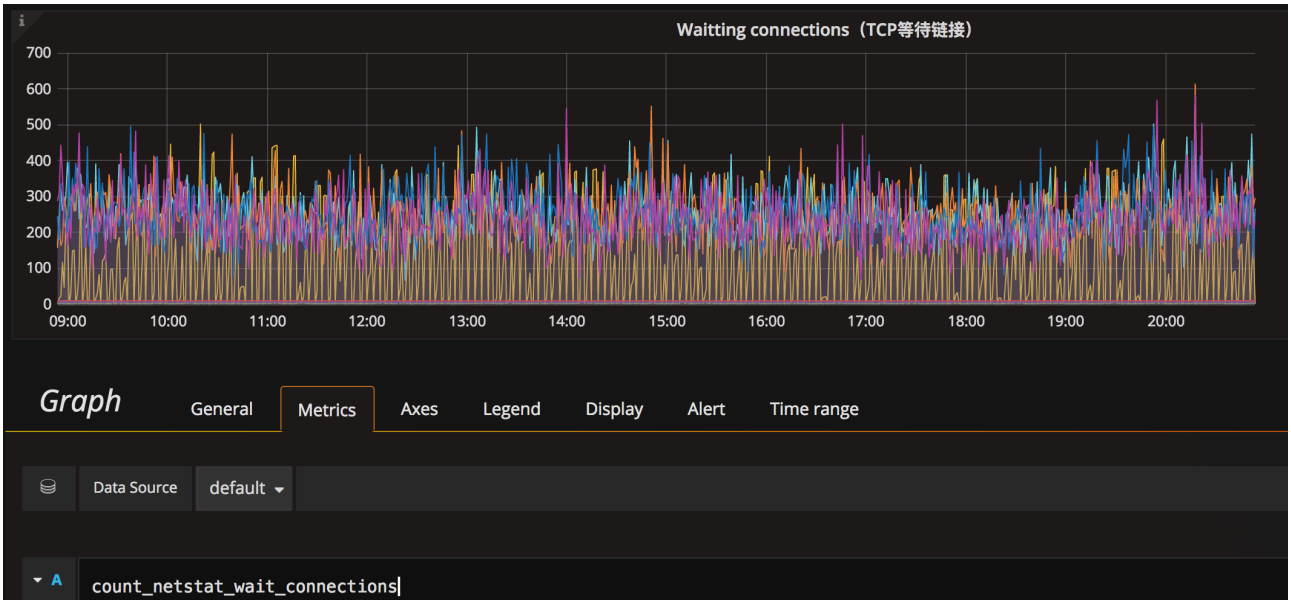
最前沿開源監控 Prometheus專題講座

第十四讲：Prometheus 企业级实际使用(二)

第十四讲内容：

- Prometheus+grafana TCP等待链接监控 企业实际使用

TCP等待链接 其实之前应给大家 用这个举例子好几次了



使用公式
count_netstat_wait_connections 一个key足够了 gauge
数据来源： pushgateway + 脚本

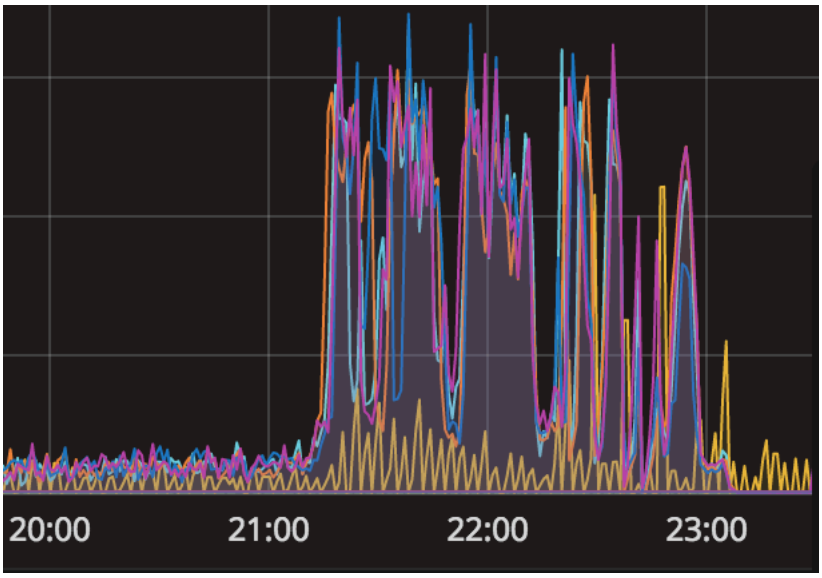
其实 node_exporter 也有对应的 tcp wait
不过 里面提供的 各种 tcp 状态的数据种类 实在太多 太细了（我有点懒得去 一个一个TCP状态 加起来出监控， 不过感兴趣的同学 可以自己尝试）

索性 我这里就用脚本自己写一个了

处于各种 wait状态的 TCP链接 是作为运维 平日排查(网络负载,服务器的负载，DB)的一个重要指标
netstat

Close_wait , time_wait 等等。。(TCP链接的方向) TCP协议

一般当 wait类型的TCP 过大时
一定说明 系统网络负载（流量负载） 出现问题了
内核优化
比如这种状况



导致这种 状况的原因 很多 并非只会因为 网络不给力

还跟 访问请求量 攻击流量 数据库 CPU 等等 都有可能引起

- Prometheus+grafana 文件描述符监控 企业实际使用

Linux系统 我们之前在 系列课程上篇时 提到过 本身就是一个基于文件表达的操作系统

任何资源的使用 都可以映射成一个文件

文件 和 文件句柄（Linux中 叫做文件描述符更准确 不过习惯叫句柄 不好改口了） 虽然并无直接联系 但是 有间接的连带关系

如下是一段 网上对文件描述符的解释 我们来看下

文件描述符是linux/unix操作系统中特有的概念。其相当于windows系统中的句柄。习惯性的，我们也把linux文件描述符称之句柄。
Linux系统中，　每当进程打开一个文件时，系统就为其分配一个唯一的整型文件描述符，用来标识这个文件。标准C中每个进程默认打开的有三个文件，标准输入，标准输出，标准错误，分别用一个FILE结构的指针来表示，即stdin，stout，sterr，这三个结构分别对应着三个文件描述符0,1,2。
文件描述符是一个简单的整数，用以标明每一个被进程所打开的文件和socket。第一个打开的文件是0，第二个是1，PID->文件的句柄　依此类推。
linux　操作系统通常对每个进程l能打开的文件数量有一个限制。
linux系统默认的最大文件描述符限制是1024

- Prometheus+grafana 网络丢包率监控 企业实际使用

网络监控

我们来看一下　一个企业中 实际使用的 对服务器内网流量 ping延迟和丢包率

pushgateway + shell

ping prometheus server

ping + ip

lostpk=`timeout 5 **ping -q -A -s 500  -W 1000 -c 100** prometheus | grep transmitted | awk '{print $6}'`

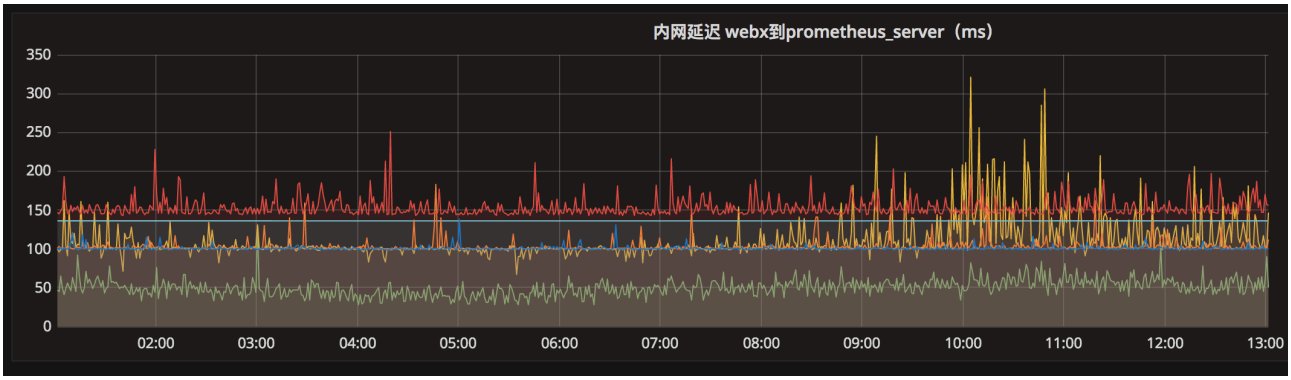
rrt=`timeout 5 ping -q -A -s 500  -W 1000 -c 100 prometheus | grep transmitted | awk '{print $10}'`

-s 一个ping包的大小
-W 延迟timeout
-c 发送多少个数据包

T-S
value_lostpk=`echo $lostpk | sed "s/%//g"`
value_rrt=`echo $rrt | sed "s/ms//g" `

echo "lostpk_"$instance_name"_to_prometheus : $value_lostpk"
echo "lostpk_"$instance_name"_to_prometheus $value_lostpk" | curl --data-binary @- http://prometheus.monitor.com:9092/metrics/job/pushgateway1/instance/localhost:9092

echo "rrt_"$instance_name"_to_prometheus : $value_rrt"
echo "rrt_"$instance_name"_to_prometheus $value_rrt" | curl --data-binary @- http://prometheus.monitor.com:9092/metrics/job/pushgateway1/instance/localhost:9092



如上就是 使用scripts 获取网络延迟和丢包率的 实例

另外 针对企业中 对网络的监控 我们还可以 通过安装 smokeping 等更专业的网络监控工具来采集数据

smokeping本身就提供很专业的 图形展示 命令行输出 -> 脚本 -》 pushgatway

当然 我们为了统一化的考虑 也可以和promethues连用