



互联网监控的艺术

演讲人：田逸(sery@163.com)



监控演变历程

- 用户、老板电话通知
- Ping主机
- 放在线收音机
- 时不时登陆系统查看状态
- 使用监控平台



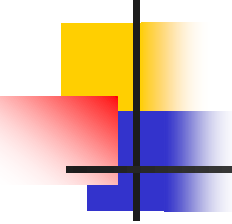
必要性

- 被动方式延误时机，并且给技术人员产生巨大的压力
- **Ping**主机对于服务器数量多的场合会怎么样？
- 服务器在远端，无声卡等，放不了在线收音机音频
- 数以百计甚至上千计的服务器，挨个登录恐怕是个困难
- 部署监控平台，随时了解大规模网络运行的状态，并且在出现意外时及时报警



高可用性的保证措施

- 高可用架构：应用集群、负载均衡、分布式文件系统、数据库集群等
- 设计良好的监控系统



选择何种监控方式

- 自己写程序或者脚本小工具
- 商业解决方案
- 开源的方案

自己写程序或者脚本

- 优点

- 1、现金成本低
- 2、操控性强

- 不足

- 1、集中管理是个问题
- 2、无可视性效果
- 3、调度需要好的技巧
- 4、运营复杂，如监控一个集群，需要对集群维护。因程序被安装在每个主机，因此可能要重复n次操作

商业解决方案

- 优势

- 1、大包大揽
- 2、用户不用承担运营中的责任。假如出了问题，可把责任推给厂商。
- 3、很受欢迎的展现方式，如报表、美观的用户界面

- 不足

- 1、成本高(一般以万计)
- 2、取舍不易：需要的功能可能不足，不需要的功能一大堆



商业解决方案（续）

- 3、巨大的资源占用：就**client**代理而言，包大小基本以百兆字节计
- 4、扩展性差。要新增模块，需要昂贵的支出
- 5、安全问题。来自代理**agent**的权限。



开源解决方案

- 优势

- 1、开源免费
- 2、定制能力强
- 3、完全可控
- 4、集中化管理
- 5、可视性好

- 不足

- 1、需要很强的技术实力
- 2、一切靠自己



推荐的方案

- 开源的方案
- Nagios、zenoss、Ganglia 、zabbix 等
- 本案以nagios为例



谁在使用nagios

- Facebook
- 搜狗www.sogou.com
- 网易www.163.com
- 空中网www.kong.net
- 新浪www.sina.com
- 阿里巴巴
- www.wikipedia.org
- 互动百科www.hudong.com
- Sohu
- 其他更多.....



什么是nagios

- Nagios是一个框架，核心部分是调度器和状态通知模块。
- 调度器调度插件或者任意定制的可执行程序，根据执行结果返回状态值，并根据需要进行相信的报警/通知



Nagios监控机制

- 检查登录用户数的场景
- 条件：
 - 1、当登录数小于5时，属于正常
 - 2、当用户数大于5小于8时，应当警告
 - 3、大于8时为异常
- 脚本：logins.sh



检查用户登录脚本（续一）

- `#!/bin/sh`
- `logins=`who |wc -l``
- `if [$logins -le $1]`
- `then`
- `echo "OK!-login count is $logins"`
- `exit 0`
- `fi`

- `if [$logins -gt $1 -a $logins -le $2]`
- `then`
- `echo "Warnning!-login count is $logins"`
- `exit 1`
- `fi`

- `if [$logins -gt $2]`
- `then`
- `echo "Critical!-login count is $logins"`
- `exit 2`
- `fi`

检查用户登录脚本（续二）

- 手动执行脚本，形如 `./usercon.sh 5 8`
- 输出：
 - 1、有2个用户登录时:OK!-login count is 2

```
[root@MONITOR ~]# ./usercon.sh 5 8  
OK!-login count is 2
```

- 2、有6个用户登录时: Warning!-login count is 6

```
[root@MONITOR ~]# ./usercon.sh 5 8  
Warning!-login count is 6
```

- 3、有9个用户登录时: Critical!-login count is 9

```
[root@MONITOR ~]# ./usercon.sh 5 8  
Critical!-login count is 9
```



调度脚本

- 本地调度：用于检查远程服务端口等
- 远程调度（**Nrpe**）：检查主机资源、检查内部网络服务
- 设定调度时间间隔、重试次数等。注意与**crond**不同。



通知机制

- 退出代码为0，代表正常
- 退出代码为1，代表警告warning
- 退出代码为2，代表极度异常Critical
- 退出代码为3、代表未知（不常用）



监控的表现形式

- **Web方式**：不同的用户查看各自负责的应用运行状态
- **邮件**：故障发生或恢复时发送邮件告警
- **手机短消息**：最及时的通知方式



Nagios的组成

- 3个大的部分：
 - 1、守护进程（core）
 - 2、插件
 - 3、web接口



定制安装nagios-core

- Useradd nagios -s /sbin/nologin
- Tar zxvf nagios-3.x.tar.gz
- Cd nagios-3.x
- ./configure --prefix=/usr/local/nagios
- Make all
- Make install make install-config install-commandmode
- 为保持通用性 舍弃了一些安装选项



安装插件

- Tar zxvf nagios-plugin-x.tar.gz
- Cd nagios-plugin-x
- ./configure --prefix=/usr/local/nagios
- Make all
- Make install



安装nrpe

- Useradd nagios -s /sbin/nologin
- Tar zxvf nrpe-x.tar.gz
- Cd nrpe.x
- ./configure --prefix=/usr/local/nrpe
- Make;make install
- 从nagios-core复制一些插件



配置nagios

- 对象归类。使用多个配置文件，易于维护
- 使用模版，减少重复书写文字
- 用户分级，责任分清（`cgi.cfg`）



监控配置的两难问题

- 对象少了，担心有遗漏；对象多了，既担心冗余，又顾虑“狼来了”



关注点： 紧要程度

- 影响财务 ★ ★ ★ ★ ★
- 组织上的影响。如伤害客户关系 ★ ★ ★
- 对个人的影响。如可能导致某些人被解雇 ★ ★



综合监控

- 主机资源
- 网络服务
- 逻辑



（一）主机资源

- 负载、磁盘使用量、tcp连接数、交换空间利用率
- 监控主机资源可起到预警的作用

(二) 监控网络服务

- 检查端口存活情况
- 有时服务存活，但并不意味服务可用。
例如/**var**分区满了，**web**服务依然存活，
但用户很可能不能正常访问



逻辑监控

- 模拟某个应用的实际行为，通过返还状态确定该应用是否正常。
- 例子1：对mysql主从复制的监控
- 例子2：监控一个web集群

一些效果不佳的监控场景

- 监控测试对象
- 跨机房监控
- 企图用一个nagios core监控数以万计的对象
- 把报警信息发给无关人员
- 监控linux的内存使用等等



大规模网络监控的一些技巧

- 主机资源自动感知
- 配置自动化
- 使用配置模版



（一）主机资源自动感知

- 建立主机列表：临时列表、已存在主机列表、新主机列表等。
- 用nmap指定地址范围扫描，并对扫描内容进行处理
- 对比列表，自动整理扫描到的新主机资源



（二）自动配置nagios

- 以（一）的列表做输入，并遍历这个列表
- 用程序语言自动生成配置文件并自动进行语法检查



(三) 使用模版

- 目的是减少配置文件文本的长度
- 定义一个服务模版：
- `define service {`
- `name commonserv`
- `check_period 24x7`
- `max_check_attempts 4`
- `normal_check_interval 3`
- `retry_check_interval 2`
- `notification_interval 10`
- `notification_period 24x7`
- `notification_options w,u,c,r`
- `register 0`
- `}`

(三) 使用模版续一

- 引用模版:
- `define service {`
- `host_name nagios-server`
- `service_description check-host-alive`
- `contact_groups sagroup`
- `check_command check-host-alive`
- `use commonserv`
- `}`

（三）使用模版续二

- 假如监控3000个对象，使用模版，则节省了 $3000 \times 7 = 21000$ 行
- 使用模版的前提是所有的引用具有一致性。如果一些服务对象有其他要求（如不同的联系组），则可以定义多个模版，来获得这种灵活性



结束语

- 监控的理想状态：沉默是金！！！但事实上，偶尔报几次警，心里还是要踏实一些！！！！



完毕

- 谢谢！
- 田逸
- Email: sery@163.com
- Qq:447877614
- <msn:sery@sohu.com>
- 即将面市作品 《互联网运营智慧》
- 2010/8/5