zabbix的原理

https://blog.csdn.net/scott\_bing/article/details/78613194

Agentd安装在被监控的主机上，Agent负责定期收集客户端本地各项数据，并发送至Zabbix Server端，Zabbix

Server收到数据，将数据存储到数据库中，用户基于Zabbix

WEB可以看到数据在前端展现图像。当Zabbix监控某个具体的项目，改项目会设置一个触发器阈值，当被监控的指标超过该触发器设定的阈

值，会进行一些必要的动作，动作包括：发送信息（邮件、微信、短信）、发送命令（SHELL 命令、Reboot、Restart、Install等）。

mha集群现在写是瓶颈 咋处理

MySQL的MHA+keepalived+负载均衡+读写分离架构

https://blog.csdn.net/Stubborn\_Cow/article/details/48155081

增加主主机；优化表结构；或者读写分离

docker中同主机 不同主机间如何通信

https://www.cnblogs.com/liyuanhong/articles/5851251.html

docker启动时，会在宿主主机上创建一个名为docker0的虚拟网络接口，默认选择172.17.42.1/16，一个16位的子网掩码给容器提供了6553

4个IP地址。docker0只是一个在绑定到这上面的其他网卡间自动转发数据包的虚拟以太网桥，它可以使容器和主机相互通信，容器与容器

间通信。

zabix工作流程

https://blog.csdn.net/u012062455/article/details/53216898

abbix监控过程是这样的：安装在主机上的zabbix\_agentd负责监控主机（具体的监控任务是由agentd端的Item来完成的），并收集数据，

然后将数据发送到zabbix

server端。如果是分布式系统，需要监控的机器较多，为了减轻server端的压力，可能中间还会再搭建一个proxy端，用来暂时接收监控数

据，然后将数据转发到server端。Server端将数据存储到数据库中，zabbix web再将数据在前端以图表或者文字的形式展现出来。

要相对主机的某一项性能进行监控，需要在在web界面执行如下操作

1.在你想要监控的主机下，创建监控项Item，监控指定的数据

2.创建触发器Trigger，将创建好的监控项添加进触发器，设置触发条件

3.创建动作Action，将触发器和用户关联起来，当触发器被触发时，向关联的用户发送告警信息

完成以上操作后，就可以对主机某项性能进行监控，当获取到的数据达到触发条件时，触发器就会被触发，然后通过报警媒介向关联的用

户发送告警信息。

web优化

https://blog.csdn.net/qiansg123/article/details/80131716

数据库读写分离

使用中间件

关系性数据库与非关系性数据库区别

nosql和关系型数据库比较？

优点：

1）成本：nosql数据库简单易部署，基本都是开源软件，不需要像使用oracle那样花费大量成本购买使用，相比关系型数据库价格便宜。

2）查询速度：nosql数据库将数据存储于缓存之中，关系型数据库将数据存储在硬盘中，自然查询速度远不及nosql数据库。

3）存储数据的格式：nosql的存储格式是key,value形式、文档形式、图片形式等等，所以可以存储基础类型以及对象或者是集合等各种格

式，而数据库则只支持基础类型。

4）扩展性：关系型数据库有类似join这样的多表查询机制的限制导致扩展很艰难。

缺点：

1）维护的工具和资料有限，因为nosql是属于新的技术，不能和关系型数据库10几年的技术同日而语。

2）不提供对sql的支持，如果不支持sql这样的工业标准，将产生一定用户的学习和使用成本。

3）不提供关系型数据库对事物的处理。

进程的状态

一个进程从创建而产生至撤销而消亡的整个生命期间，有时占有处理器执行，有时虽可运行但分不到处理器、有时虽有空闲处理器但因等待某个事件的发生而无法执行，这一切都说明进程和程序不相同，它是活动的且有状态变化的，这可以用一组状态加以刻画。为了便于管理进程，‘般来说，按进程在执行过程中的不同情况至少要定义三种不同的进程状态： [3] 

（1）运行(running)态：进程占有处理器正在运行。

（2）就绪(ready)态：进程具备运行条件，等待系统分配处理器以便运行。

（3）等待(wait)态：又称为阻塞(blocked)态或睡眠(sleep)态，指进程不具备运行条件，正在等待某个事件的完成。

通常，一个进程在创建后将处于就绪状态。每个进程在执行过程中，任意时刻当且仅当处于上述三种状态之一。同时，在一个进程执行过程中，它的状态将会发生改变。引起进程状态转换的具体原因如下：

（1）运行态一一等待态：等待使用资源或某事件发生，如等待外设传输;等待人工干预。

（2）等待态一一就绪态：资源得到满足或某事件己经发生，如外设传输结束；人工干预完成。

（3）运行态一一就绪态：运行时间片到，或出现有更高优先权进程。

（4）就绪态一一运行态：CPU空闲时被调度选中一个就绪进程执行。

[https加密解密过程详解](https://www.cnblogs.com/swaggyC/p/6117743.html)

要点：

　https协议对传输内容进行加密，具有更强的安全性，防止被抓包后解析出请求内容。

　https是建立在ssl之上的http协议。

　服务器支持https协议必须安装一套数字证书，所谓数字证书就是一对公钥和私钥，公钥用来加密，私钥用来解密。为了与下文中的私钥进行区分，这里的公钥和私钥称为公钥1和私钥1。

数字证书可以自己制作或者向组织申请，自己制作的会在客户端弹出提示框，手动验证通过，而申请的就无需客户端手动验证了。

请求过程：

　　1.客户端访问 https开头的url

　　2.服务端返回公钥1，客户端验证通过（如果不通过，则访问终断）。

　　3.客户端根据公钥1生成一个私钥2，这个私钥2用来加密和解密请求信息。使用公钥1对私钥2进行加密，回传给服务端。服务端用私钥1对该信息解密，得到私钥2。至此，客户端和服务端都已经有了私钥2。

4.客户端和服务端之间使用私钥2对信息进行加密后通信，这样即使第三方抓包，也无法轻易获取通信内容了。

5.断开的时候比传统的四次断开多了一项

Alert Message: Encrypted Alert

[【Linux日志】系统日志及分析](https://www.cnblogs.com/yingsong/p/6022181.html)

Linux系统拥有非常灵活和强大的日志功能，可以保存几乎所有的操作记录，并可以从中检索出我们需要的信息。

大部分Linux发行版默认的日志守护进程为 syslog，位于 /etc/syslog 或 /etc/syslogd 或/etc/rsyslog.d，默认配置文件为 /etc/syslog.conf 或 rsyslog.conf，任何希望生成日志的程序都可以向 syslog 发送信息。

Linux系统内核和许多程序会产生各种错误信息、警告信息和其他的提示信息，这些信息对管理员了解系统的运行状态是非常有用的，所以应该把它们写到日志文件中去。

完成这个过程的程序就是syslog。syslog可以根据日志的类别和优先级将日志保存到不同的文件中。

例如，为了方便查阅，可以把内核信息与其他信息分开，单独保存到一个独立的日志文件中。默认配置下，日志文件通常都保存在“/var/log”目录下。

日志类型

下面是常见的日志类型，但并不是所有的Linux发行版都包含这些类型：

|  |  |
| --- | --- |
| 类型 | 说明 |
| auth | 用户认证时产生的日志，如login命令、su命令。 |
| authpriv | 与 auth 类似，但是只能被特定用户查看。 |
| console | 针对系统控制台的消息。 |
| cron | 系统定期执行计划任务时产生的日志。 |
| daemon | 某些守护进程产生的日志。 |
| ftp | FTP服务。 |
| kern | 系统内核消息。 |
| local0.local7 | 由自定义程序使用。 |
| lpr | 与打印机活动有关。 |
| mail | 邮件日志。 |
| mark | 产生时间戳。系统每隔一段时间向日志文件中输出当前时间，每行的格式类似于 May 26 11:17:09 rs2 -- MARK --，可以由此推断系统发生故障的大概时间。 |
| news | 网络新闻传输协议(nntp)产生的消息。 |
| ntp | 网络时间协议(ntp)产生的消息。 |
| user | 用户进程。 |
| uucp | UUCP子系统。 |

日志优先级

常见的日志优先级请见下标：

|  |  |
| --- | --- |
| 优先级 | 说明 |
| emerg | 紧急情况，系统不可用（例如系统崩溃），一般会通知所有用户。 |
| alert | 需要立即修复，例如系统数据库损坏。 |
| crit | 危险情况，例如硬盘错误，可能会阻碍程序的部分功能。 |
| err | 一般错误消息。 |
| warning | 警告。 |
| notice | 不是错误，但是可能需要处理。 |
| info | 通用性消息，一般用来提供有用信息。 |
| debug | 调试程序产生的信息。 |
| none | 没有优先级，不记录任何日志消息。 |

常用日志文件

系统日志是由一个名为syslog的服务管理的，如以下日志文件都是由syslog日志服务驱动的：

/var/log/boot.log：录了系统在引导过程中发生的事件，就是Linux系统开机自检过程显示的信息

/var/log/lastlog ：记录最后一次用户成功登陆的时间、登陆IP等信息

/var/log/messages ：记录Linux操作系统常见的系统和服务错误信息

/var/log/secure ：Linux系统安全日志，记录用户和工作组变坏情况、用户登陆认证情况

/var/log/btmp ：记录Linux登陆失败的用户、时间以及远程IP地址

/var/log/syslog：只记录警告信息，常常是系统出问题的信息，使用lastlog查看

/var/log/wtmp：该日志文件永久记录每个用户登录、注销及系统的启动、停机的事件，使用last命令查看

/var/run/utmp：该日志文件记录有关当前登录的每个用户的信息。如 who、w、users、finger等就需要访问这个文件

/var/log/syslog 或 /var/log/messages 存储所有的全局系统活动数据，包括开机信息。基于 Debian 的系统如 Ubuntu 在 /var/log/syslog 中存储它们，而基于 RedHat 的系统如 RHEL 或 CentOS 则在 /var/log/messages 中存储它们。  
/var/log/auth.log 或 /var/log/secure 存储来自**可插拔认证模块(PAM)**的日志，包括成功的登录，失败的登录尝试和认证方式。Ubuntu 和 Debian 在 /var/log/auth.log 中存储认证信息，而 RedHat 和 CentOS 则在 /var/log/secure 中存储该信息。

**日志文件详细介绍：**

**/var/log/boot.log**

该文件记录了系统在引导过程中发生的事件，就是Linux系统开机自检过程显示的信息，如图1所示：

内存溢出解决方案

https://jingyan.baidu.com/article/948f5924f08643d80ef5f970.html

[**好好耕耘**](https://www.cnblogs.com/457248499-qq-com/)

## [redis和memcached的区别（总结）](https://www.cnblogs.com/457248499-qq-com/p/7392653.html)

观点一：

1、Redis和Memcache都是将数据存放在内存中，都是内存数据库。不过memcache还可用于缓存其他东西，例如图片、视频等等；

2、Redis不仅仅支持简单的k/v类型的数据，同时还提供list，set，hash等数据结构的存储；

3、[虚拟内存](https://www.baidu.com/s?wd=%E8%99%9A%E6%8B%9F%E5%86%85%E5%AD%98&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHcvrjTdrH00T1Y4rjDYnju-njb1nycLmW-b0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbzPHfYP10LPjDznWnkn1T3Ps" \t "_blank)--Redis当物理内存用完时，可以将一些很久没用到的value 交换到磁盘；

4、过期策略--memcache在set时就指定，例如set key1 0 0 8,即永不过期。Redis可以通过例如expire 设定，例如expire name 10；

5、分布式--设定memcache集群，利用magent做一主多从;redis可以做一主多从。都可以一主一从；

6、存储数据安全--memcache挂掉后，数据没了；redis可以定期保存到磁盘（持久化）；

7、[灾难恢复](https://www.baidu.com/s?wd=%E7%81%BE%E9%9A%BE%E6%81%A2%E5%A4%8D&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHcvrjTdrH00T1Y4rjDYnju-njb1nycLmW-b0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbzPHfYP10LPjDznWnkn1T3Ps" \t "_blank)--memcache挂掉后，数据不可恢复; redis数据丢失后可以通过aof恢复；

8、Redis支持数据的备份，即master-slave模式的数据备份；

 Haproxy和nginx代理的区别

Haproxy特点

1. 支持TCP与Http协议，工作在网络4层和7层
2. 支持Session共享、Cookies引导
3. 支持通过URL健康检测
4. 支持8种负载均衡策略
5. 支持心跳检测

Nginx特点

1. 支持Http协议，工作在网络7层
2. 支持通过端口健康检测
3. 支持强大的正则匹配规则
4. 支持WebSocket协议
5. 支持Http Cache

LVS：

1.抗负载能力强，性能高，能达到F5的60%，对内存和cpu资源消耗比较低

2.工作在网络4层，通过VRRP协议（仅做代理使用），具体的流量是由liunx内核来处理，因此没有流量的产生。

3.稳定，可靠性强，自身有完美的热备方案（Keepalived+LVS）

4.不支持正则处理，不能做动静分离

5.支持多种负载均衡算法：rr（轮询），wrr（带权轮询）、lc（最小连接）、wlc（带权最小连接）

6.配置相对复杂，对网络依赖比较大，稳定性很高。

7.LVS工作模式有4种：

（1）nat地址转换

     （2）dr直接路由

     （3）tun隧道

     （4）full-nat

Nginx：

{轮询方式，权重方式，ip\_hash依据ip分配方式，最少连接方式

1.工作在网络7层，可以针对http应用做一些分流的策略，比如针对域名，目录结构

2.Nginx对网络的依赖较小，理论上能ping通就能进行敷在功能

3.Nginx安装配置比较简单，测试起来很方便

4.也可以承担较高的负载压力且稳定，Nginx是为了解决c10k问题而诞生的

5.对后端服务器的健康检查，只支持通过端口来检测，不支持通过url来检测

6.Nginx对请求的异步处理可以帮助节点服务器减轻负载压力

7.Nginx仅能支持http、https和Email协议，这样就在适用范围较小

8.不支持Session的直接保持，但能通过ip\_hash来解决，对Big request header的支持不是很好

9.Nginx还能做Web服务器即Cache功能。

HAproxy：

1.支持两种代理模式：TCP（四层）和HTTP（七层），支持虚拟主机

2.能够补充Nginx的一些缺点比如Session的保持，Cookie引导等工作

3.支持url检测后端的服务器出问题的检测会有很好的帮助。

4.更多负载均衡策略比如：动态加权轮循，加权源地址哈希，加权URL哈希加权等参数哈希已经实现。

5.单纯从效率上来讲HAproxy更会比Nginx有更出色的负载均衡

6.HAproxy可以对MYsql进行负载均衡，对后端的DB节点进行检测和负载均衡

7.支持负载均衡算法：轮循、带权轮循、源地址保持、请求URL、根据Cookie

8.不能做Web服务器即Cache。

三大主流软件负载均衡器适用的生产场景：

1.网站建设初期，可以选用Nginx、HAproxy作为反向代理负载均衡（流量不大时可以选择不用负载均衡）因为其配置简单，性能也能满足一般业务场景。如果考虑到负载均衡器是有单点失败问题，可以采用Nginx+Keepalived避免负载均衡器自身单点问题。

2.网站并发达到一定程度后，为了提高稳定性和转发效率，可以使用LVS，毕竟LVS比Nginx/HAproxy要更稳定，转发效率也高

两台linux服务器里面安装keepalived，配置好后发现两台机器都是master状态，网上搜索发现是防火墙没有允许vrrp的组播

RAID 的含义及优势？RAID0、RAID1、RAID5 分别指什么、各自的特点？ 参考答案 1）RAID：廉价冗余磁盘阵列，指通过硬件/软件技术将多个较小/低速的磁盘整合成一个大磁盘使用的一种存储技 术，其不仅可存储数据，还可以实现一定程度的冗余保障，具有“速度快、安全性高”的优势。 2）RAID0、RAID1、RAID5 的含义及特点如下： RAID0：条带模式，由两个或两个以上的磁盘组成，同一份文档分散在不同的磁盘中，并行写入，提高写效率。 RAID1：镜像模式，由至少两个磁盘组成，同一份文件被分别写入到不同的磁盘中，每份磁盘数据一样，实现容错， 提高读效率。 RAID5：分布式奇偶校验的独立磁盘模式，结合 RAID0 和 RAID1 的好处，同时避免它们的缺点。由至少 3 块以上大 小相同的磁盘组成，实现冗余

Tomcat的最大并发数是可以配置的，实际运用中，最大并发数与硬件性能和CPU数量都有很大关系的。更好的硬件，更多的处理器都会使Tomcat支持更多的并发。

　　Tomcat默认的HTTP实现是采用阻塞式的Socket通信，每个请求都需要创建一个线程处理，当一个进程有500个线程在跑的话，那性能已经是很低很低了。Tomcat 默认配置的最大请求数是150，也就是说同时支持150个并发。具体能承载多少并发，需要看硬件的配置，CPU 越多性能越高，分配给JVM的内存越多性能也就越高，但也会加重GC的负担。当某个应用拥有 250 个以上并发的时候，应考虑应用服务器的集群。[操作系统](https://www.baidu.com/s?wd=%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)对于进程中的线程数有一定的限制：

1. Windows 每个进程中的线程数不允许超过 2000
2. Linux 每个进程中的线程数不允许超过 1000
3. 在Java中每开启一个线程需要耗用1MB的JVM内存空间用于作为线程栈之用，此处也应考虑。