Zabbix分布式监控

## 监控的目标

在IT环境中存在各种各样的设备，硬件设备，软件设备。都是由很多模型构成，底层网络环境、硬件，系统架构（centos,kvm,xen），上层应用（mysql,nginx,php,tomcat）。为了保证业务的正常运行，需要运维进行设计架构维护和调优，在这个过程中，为了及时掌握基础环境和业务的可用性，需要获取各组件的运行状态，这样便离不开监控系统。

一般来说监控系统可以分为两部分，数据采集部分（客户端）和数据存储分析告警展示部分（服务端）。 主要监控内容有：硬件，系统，应用，业务。

### 1.1 硬件监控

通常服务器上都会有远程控制卡，如DELL的IDRAC, HP的ILO和 IBM的IMM，可以通过web界面来进行硬件的监控和管理工作。在linux下，通过我们使用IPMI来完成物理设备的监控工作，温度，电压，电扇，电源供应状态，故障等。

### 1.2 系统监控

系统监控是监控体系的基础，系统监控主要对象有CPU,内存，IO

CPU： 运行队列，CPU使用率，上下文切换，常用的监控工具有：top, vmstat, mpstat

内存: Linux虚拟内存是一个庞大的东东，通过需要监控内在的使用率，SWAP使用率，同时可以通过内存使用率曲线来发现某些服务的内存溢出等，常用监控工具有，free, vmstat

IO: IO 可以分发磁盘IO，和网络IO，除了在性能调优我们要监控更详细的数据外，日常监控只关注磁盘使用率，io wait即可。 网络也是监控网卡流量即可。 工具有iostat, iotop, iftop

TCP监控： 很多监控下有必要监控TCP状态，可以使用netstat 或者ss 来获取所有TCP状态， 可以在大并发中及时发现TCP相关故障

### 1.3 应用服务监控

应用服务监控在监控体系上也是非常重要的内容：例如：

1. Apache： Apache提供了mod\_status 和mod\_info模块用来输出Apache状态，
2. Nginx： 编译安装时加上 with-http\_stub\_status\_module。便可以监控nginx状态
3. Memcached，redis监控： 使用nc命令便可以获取运行状态
4. JVM： 对于JAVA应用，JVM监控也是非常有必要的。
5. 接口监控：通过curl获取状态监控接口是否正常

### 1.4 业务监控

业务监控包括流量分析，PV,UV,IP,PR等指标。可以通过分析日志，开源软件PIWIK以及第三方 GA来完成分析。

## zabbix介绍

如果要把上面所有内容监控起来，并且通过邮件，短信报警，仅仅靠脚本会特别杂乱，这时候就需要使用到第三方监控工具，目前比较流行的监控软件有Nagios, Cacti, Zabbix。

Zabbix 通过C/S 模式采集数据，通过B/S模式在web端展示和配置，监控内容全面，支持自动发现，增加监控。

硬件监控： zabbix IPMI interface

系统监控: zabbix agent interface

JAVA监控： zabbix jmx interface

网络监控：zabbinx snmp interface

应用服务: zabbix agent userParameter

Mysql监控： percona-monitoring-plugins

URL监控： zabbix web moniter

多种告警(邮件，短信)方式，可以报警升级。可以使用proxy做分布式监控。

### 2.1 Zabbix-agent 介绍

Zabbix-agent 检测分为主动与被动两种形式，

**主动**： agent请求server获取主机监控项列表，并主动将监控项内需要的数据提交给server/proxy

**被动**： server 向agent请求获取监控项的数据，agent返回数据。。

备注：为了避免内存耗尽，server限制每个通信连接最多使用128MB内存。

#### 2.1.1 被动检测

supported items通信过程

* Server打开一个TCP连接
* Server发送请求agent.ping\n
* Agent接收到请求并且响应<HEADER><DATALEN>1
* Server处理接收到的数据1
* 关闭TCP连接

not supported items通信过程

* Server打开一个TCP连接
* Server发送请求vfs.fs.size[/nono]\n
* agent接收请求并且返回响应数据 <HEADER><DATALEN>ZBX\_NOTSUPPORTED\0Cannot obtain filesystem information: [2] No such file or directory
* Server接收并处理数据, 将item的状态改为“ not supported ”
* 关闭TCP连接

#### 2.1.2 主动检测

如前面所说，zabbix首先向ServerActive配置的IP请求获取active items，获取并提交active tiems数据值server或者proxy。很多同学会提出疑问：zabbix多久获取一次active items？它会根据配置文件中的RefreshActiveChecks的频率进行，如果获取失败，那么将会在60秒之后重试, 获取到的items列表中的所有item属性key, delay, lastlogsize ,mtime都必须存在，

获取列表的通信过程如下：

* Agent打开TCP连接（主动检测变成Agent打开）
* Agent请求items检测列表
* Server返回items列表
* Agent 处理响应
* 关闭TCP连接

Agent开始收集数据

提交active items数据 , 如果有些数据提交失败，比如host、item被删除或者禁用，agent不会尝试从新提交

主动检测提交数据过程如下：

* Agent建立TCP连接
* Agent提交items列表收集的数据
* Server处理数据，并返回响应状态
* 关闭TCP连接

#### 2.1.3 agent 安装与配置

安装agent

yum install zabbix-agent-2.4.3

配置agent也特别简单，下面是主动模式的配置，使用ansible或者saltstack批量部署时可以把Hostname改为变量获取主机名就OK

[yulei@localhost ~]$ grep '^[a-Z]' /etc/zabbix/zabbix\_agentd.conf

PidFile=/var/run/zabbix/zabbix\_agentd.pid

LogFile=/var/log/zabbix/zabbix\_agentd.log

LogFileSize=0

Server=127.0.0.1

ServerActive=127.0.0.1

Hostname=Zabbix server

HostMetadataItem= Zabbix server

Include=/etc/zabbix/zabbix\_agentd.d/

Server和ServerActive对应的是server或者proxy的IP，Hostname为该主机名

如果是被动模式 ServerActive 与Hostname可以注掉。

HostMetadataItem主要用于自动注册时对不同的主机做不同的处理，比如对带有nginx的主机分到nginx组，连接nginx模版，等。

自定义监控需要使用UserParameter这个参数，针对不同的应用做不同的监控，可见下文介绍。

最后启动agent并加到自启动中

chkconfig –add zabbix-agent

service zabbix-agent start

### 2.2 zabbix-proxy 介绍

使用zabbix-proxy针对不同机房，不同区域进行分布式监控， 他代理server来收集数据并定时发送给server。当服务器集群上千后也可以使用proxy减轻server端的压力。

Proxy与server是基于TCP通信，proxy先从server端获取server端的配置，把所需要的数据临时存储到本地数据库中，然后再把数据发送给server，在发送之前如果server端配置变了，那么对应的数据便会被丢弃。所以可以根据server配置变更的频率来修改proxy配置（ProxyLocalBuffer and ProxyOfflineBuffer parameters）来设置多长时间发送数据到server端。

Proxy仅是一个数据收集者，所以不会去计算triggers,处理事件。下面是proxy能做的一些事。

| Function | | Supported by proxy |
| --- | --- | --- |
| Items | |  |
|  | *Zabbix agent checks* | Yes |
| *Zabbix agent checks (active)* | Yes 1 |
| *Simple checks* | Yes |
| *Trapper items* | Yes |
| *SNMP checks* | Yes |
| *SNMP traps* | Yes |
| *IPMI checks* | Yes |
| *JMX checks* | Yes |
| *Log file monitoring* | Yes |
| *Internal checks* | Yes |
| *SSH checks* | Yes |
| *Telnet checks* | Yes |
| *External checks* | Yes |
| Built-in web monitoring | | Yes |
| Network discovery | | Yes |
| Low-level discovery | | Yes |
| Calculating triggers | | *No* |
| Processing events | | *No* |
| Sending alerts | | *No* |
| Remote commands | | *No* |

#### 2.2.1 proxy安装与配置

Proxy支持sqllit,db2,mysql,pgsql。这里以mysql为主安装。

yum install zabbix-proxy-2.4.3 zabbix-proxy-mysql-2.4.3 mysql-server mysql

在数据库是创建zabbix\_proxy库，并创建用户授权。

Mysql> create database zabbix\_proxy;

Mysql> grant all on zabbix\_proxy.\* to [zabbix\_proxy@’127.0.0.1](mailto:zabbix_proxy@'127.0.0.1)’ identified by ‘zabbix\_proxy’;

Mysql> flush privileges;

$ mysql –uzabbix\_proxy –p’zabbix\_proxy’ zabbix\_proxy < /usr/share/doc/zabbix-proxy-mysql-2.4.3/create/schema.sql

修改/etc/zabbix/zabbix\_proxy.cof配置文件

[yulei@zabbix\_proxy ~]$ sudo grep '^[a-Z]' /etc/zabbix/zabbix\_proxy.conf

Server=192.168.56.33

Hostname=zabbix\_proxy

LogFile=/var/log/zabbix/zabbix\_proxy.log

LogFileSize=0

PidFile=/var/run/zabbix/zabbix\_proxy.pid

DBName=zabbix\_proxy

DBUser=zabbix\_proxy

DBPassword=zabbix\_pass

DBSocket=/var/lib/mysql/mysql.sock

ConfigFrequency=10

DataSenderFrequency=10

ExternalScripts=/usr/lib/zabbix/externalscripts

这里要注意几点 Hostname必须与server端配置的proxy name一致，不然会增加失败的。

如果需要加密传输，需要再增加别的配置。这里建议使用主动模式，

启动zabbix proxy 并且增加到开机自启动中

chkconfig –add zabbix-proxy

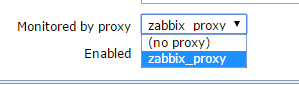
service zabbix-proxy start

然后在web界面增加proxy并且修改agent配置改为从proxy收集。

步骤为：Administration > proxies > create proxy



最后修改host



### 2.3 zabbix-server 介绍

Zabbix-server是zabbix集群的中心，它用于接收agent,proxy发送过来的数据，计算trigger,发送报警给用户，server端还存储所有配置和处理数据。

#### 2.3.1 zabbix-server 安装与配置

这里还是使用yum安装server端

yum install zabbix-server-mysql-2.4.3 zabbix-web-mysql-2.4.3 mysql-server mysql zabbix-get-2.4.3 zabbix-agent-2.4.3 zabbix-java-gatway-2.4.3

首先对数据库做处理

Mysql> create database zabbix;

Mysql> grant all on zabbix.\* to [zabbix@’127.0.0.1](mailto:zabbix@'127.0.0.1)’ identified by ‘zabbix’;

Mysql> flush privileges;

$ mysql –uzabbix –p’zabbix’ zabbix < /usr/share/doc/zabbix-server-mysql-2.4.3/create/schema.sql

mysql –uzabbix –p’zabbix’ zabbix < /usr/share/doc/zabbix-server-mysql-2.4.3/create/images.sql

mysql –uzabbix –p’zabbix’ zabbix < /usr/share/doc/zabbix-server-mysql-2.4.3/create/data.sql

然后配置zabbix\_server.conf

[yulei@zabbix\_server ~]$ grep '^[a-Z]' /etc/zabbix/zabbix\_server.conf

LogFile=/var/log/zabbix/zabbix\_server.log

LogFileSize=0

PidFile=/var/run/zabbix/zabbix\_server.pid

DBHost=127.0.0.1

DBName=zabbix

DBUser=zabbix

DBPassword=zabbix

DBSocket=/var/lib/mysql/mysql.sock

JavaGateway= 192.168.56.31

JavaGatewayPort=10052

StartJavaPollers=5

SNMPTrapperFile=/var/log/snmptt/snmptt.log

AlertScriptsPath=/usr/lib/zabbix/alertscripts

ExternalScripts=/usr/lib/zabbix/externalscripts

修改时区

sed -i '19c\\tphp\_value date.timezone Asia/Shanghai' /etc/httpd/conf.d/zabbix.conf

sed -i '946c\date.timezone = Asiz/Shanghai' /etc/php.ini  
sed -i '440s/30/300/; 729s/8M/16M/; 132c\max\_input\_time=300; 1586c\mbstring.func\_overload = 2' /etc/php.ini

zabbix.conf与php.ini仅需要修改其中一个便好，不用两处都修改。

最后增加一个配置文件

[root@zabbix\_server /etc/zabbix/web]# cat zabbix.conf.php

<?php

// Zabbix GUI configuration file.

global $DB;

$DB['TYPE'] = 'MYSQL';

$DB['SERVER'] = '127.0.0.1';

$DB['PORT'] = '0';

$DB['DATABASE'] = 'zabbix';

$DB['USER'] = 'zabbix';

$DB['PASSWORD'] = 'zabbix';

// Schema name. Used for IBM DB2 and PostgreSQL.

$DB['SCHEMA'] = '';

$ZBX\_SERVER = '127.0.0.1';

$ZBX\_SERVER\_PORT = '10051';

$ZBX\_SERVER\_NAME = 'zabbix\_server';

$IMAGE\_FORMAT\_DEFAULT = IMAGE\_FORMAT\_PNG;

?>

这里存储的便是所有server web界面的一些配置。增加此文件后便不会再在web界面看到install那一页了。

然后便可以启动zabbix-server了

chkconfig --add httpd

chkconfig --add zabbix-server

service zabbix-server start

service httpd start

最后便可以通过web界面配置zabbix-server了。现在所有的主机都没有增加，但是又不想一个个手动去增加，这时候就需要使用discovery或者auto regrestion来自动增加host以及链接模版了。

最后增加用户，增加报警，升级报警。一个zabbix-server集群便全部部署成功。

#### 2.3.2 报警

Sendmail需要curl 7.20以后，以及编译的时候加上 –with-libcurl 然后便可以使用server自带的sendmail来发邮件了。

yum install sendmail sendmail-devel

## 自定义监控

### 3.1 nginx 状态监控

在编译安装nginx的时候我们经常会增加这样一个参数 --with-http\_stub\_status\_module来开启状态模块。然后通过脚本来做监控。

首先修改nginx配置

server {

listen 80;

server\_name localhost ;

location /nginx\_status {

stub\_status on;

access\_log off;

allow 127.0.0.1;

deny all;

}

}

编写脚本

NGINX\_PORT=$1

NGINX\_COMMAND=$2

NGINX\_URL="http://127.0.0.1:"$NGINX\_PORT"/nginx\_status/"

# NGINX\_DATA=$(/usr/bin/curl $NGINX\_URL 2>/dev/null)

NGINX\_DATA=$(/usr/bin/wget -q http://127.0.0.1/nginx\_status -O - 2>/dev/null)

#echo "$NGINX\_DATA"

case $NGINX\_COMMAND in

active) echo "$NGINX\_DATA"| awk 'NR==1{print $NF}' ;;

reading) echo "$NGINX\_DATA"| awk 'NR==4{print $2}' ;;

writing) echo "$NGINX\_DATA"| awk 'NR==4{print $4}' ;;

waiting) echo "$NGINX\_DATA"| awk 'NR==4{print $6}' ;;

accepts) echo "$NGINX\_DATA"| awk 'NR==3{print $1}' ;;

handled) echo "$NGINX\_DATA"| awk 'NR==3{print $2}' ;;

requests) echo "$NGINX\_DATA"| awk 'NR==3{print $3}' ;;

esac

然后自定义监控

UserParameter=nginx\_status[\*],/bin/bash /etc/zabbix/scripts/nginx\_status.sh "$1"

### 3.2 mysql监控

Mysql监控可以使用第三方percona

### 3.3 php-fpm 状态监控

首先对php-fpm进行配置，开启状态

[root@linux-node1 ~]# vim /etc/php-fpm.d/www.conf

pm.status\_path = /phpfpm\_status

其次，修改nginx配置,通过nginx访问php-fpm状态

location ~ ^/(phpfpm\_status)$ {

include fastcgi\_params;

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

}

这时候便可以获取php-fpm状态了

# curl http://127.0.0.1/phpfpm\_status

pool: www

process manager: dynamic

start time: 05/Jul/2016:15:30:56 +0800

start since: 409

accepted conn: 22

listen queue: 0

max listen queue: 0

listen queue len: 128

idle processes: 4

active processes: 1

total processes: 5

max active processes: 2

max children reached: 0

#PHP-FPM状态解释：

pool #fpm池名称,大多数为www

process manager #进程管理方式dynamic或者static

start time #启动日志,如果reload了fpm，时间会更新

start since #运行时间

accepted conn #当前池接受的请求数

listen queue #请求等待队列,如果这个值不为0,那么需要增加FPM的进程数量

max listen queue #请求等待队列最高的数量

listen queue len #socket等待队列长度

idle processes #空闲进程数量

active processes #活跃进程数量

total processes #总进程数量

max active processes #最大的活跃进程数量（FPM启动开始计算）

max children reached #程最大数量限制的次数，如果这个数量不为0，那说明你的最大进程数量过小,可以适当调整。

最后编写脚本增加自定义key到agent。把脚本放到/etc/zabbix/scripts下面

#!/bin/bash

PHPFPM\_COMMAND=$1

PHPFPM\_PORT=80

PHPFPM\_URL="http://127.0.0.1:"$PHPFPM\_PORT"/phpfpm\_status"

PHPFPM\_DATA=$(/usr/bin/wget -q $PHPFPM\_URL -O - 2>/dev/null)

echo "$PHPFPM\_DATA"

case $PHPFPM\_COMMAND in

start\_since) echo "$PHPFPM\_DATA" |awk '/^start since:/ {print $NF}' ;;

accepted\_conn) echo "$PHPFPM\_DATA" |awk '/^accepted conn:/ {print $NF}' ;;

listen\_queue) echo "$PHPFPM\_DATA" |awk '/^listen queue:/ {print $NF}' ;;

max\_listen\_queue) echo "$PHPFPM\_DATA" |awk '/^max listen queue:/ {print $NF}' ;;

listen\_queue\_len) echo "$PHPFPM\_DATA" |awk '/^listen queue len:/ {print $NF}' ;;

idle\_processes) echo "$PHPFPM\_DATA" |awk '/^idle processes:/ {print $NF}' ;;

active\_processes) echo "$PHPFPM\_DATA" |awk '/^active processes:/ {print $NF}' ;;

total\_processes) echo "$PHPFPM\_DATA" |awk '/^total processes:/ {print $NF}';;

max\_active\_processes) echo "$PHPFPM\_DATA" |awk '/^max active processes:/ {print $NF}' ;;

max\_children\_reached) echo "$PHPFPM\_DATA" |awk '/^max children reached:/ {print $NF}' ;;

slow\_requests) echo "$PHPFPM\_DATA" |awk '/^slow requests:/ {print $NF}';;

\*) echo $"USAGE:$0 "

esac

然后自定义监控

UserParameter=phpfpm\_status[\*],/bin/bash /etc/zabbix/scripts/phpfpm\_status.sh "$1"

在server端便可以创建模版，增加item.

### 3.4 TCP状态监控

编写脚本

TCP\_STAT=$1

TCP\_DATA=$(ss -ant | awk 'NR>1 {++s[$1]} END {for(k in s) print k,s[k]}')

TCP\_STAT\_VALUE=$(echo $TCP\_DATA|grep "$TCP\_STAT" | cut -d ' ' -f2)

if [ -z $TCP\_STAT\_VALUE ];then

TCP\_STAT\_VALUE=0

fi

echo $TCP\_STAT\_VALUE

然后增加自定义监控

UserParameter=tcp\_status[\*],/bin/bash /etc/zabbix/scripts/tcp\_status.sh "$1"

### 3.5 url监控

Url监控可以使用zabbix web scenarios 来监控。

最后所有的这些自定义监控都可以使用一个脚本实现，所以需要合在一起处理。

## 安全

zabbix的jsrpc的profileIdx2参数存在insert方式的SQL注入漏洞，攻击者无需授权登陆即可登陆zabbix管理系统，也可通过script等功能轻易直接获取zabbix服务器的操作系统权限。

/jsrpc.php?sid=0bcd4ade648214dc&type=9&method=screen.get&tim

estamp=1471403798083&mode=2&screenid=&groupid=&hostid=0&pageFile=hi

story.php&profileIdx=web.item.graph&profileIdx2=2'3297&updateProfil

e=true&screenitemid=&period=3600&stime=20160817050632&resourcetype=

17&itemids%5B23297%5D=23297&action=showlatest&filter=&filter\_task=&

通过上面URL便可以得到下面信息

<div class="flickerfreescreen" id="flickerfreescreen\_1" data-timestamp="1471403798083" style="position: relative;"><table class="tableinfo" cellpadding="3" cellspacing="1" id="t58daedfc75c90"><tr class="header"><td>Timestamp</td><td>Value</td></tr><tr class="even\_row" origClass="even\_row"><td class="message" colspan="2">No values found.</td></tr></table></div><table class="msgerr" cellpadding="0" cellspacing="0" id="msg\_messages" style="width: 100%;"><tr><td class="msg" colspan="1"><ul class="messages"><li class="error">Undefined index: [jsrpc.php:208 &rarr; CScreenHistory-&gt;get() &rarr; CFrontendApiWrapper-&gt;get() &rarr; CApiWrapper-&gt;\_\_call() &rarr; CFrontendApiWrapper-&gt;callMethod() &rarr; CApiWrapper-&gt;callMethod() &rarr; CFrontendApiWrapper-&gt;callClientMethod() &rarr; CLocalApiClient-&gt;callMethod() &rarr; call\_user\_func\_array() &rarr; CHistory-&gt;get() &rarr; CHistoryManager::getTableName() in /usr/share/zabbix/include/classes/api/managers/CHistoryManager.php:72]</li><li class="error">Error in query [INSERT INTO profiles (profileid, userid, idx, value\_int, type, idx2) VALUES (96, 1, 'web.item.graph.period', '3600', 2, 2'3297)] [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''3297)' at line 1]</li><li class="error">Error in query [INSERT INTO profiles (profileid, userid, idx, value\_str, type, idx2) VALUES (97, 1, 'web.item.graph.stime', '20160817050632', 3, 2'3297)] [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''3297)' at line 1]</li><li class="error">Error in query [INSERT INTO profiles (profileid, userid, idx, value\_int, type, idx2) VALUES (98, 1, 'web.item.graph.isnow', '0', 2, 2'3297)] [You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''3297)' at line 1]</li></ul></td></tr></table

这时需要在nginx处做些访问限制，以避免被注入

        if ($request\_uri ~ ^(.+\.php)(.\*)$) {

           set $req $2;

        }

        if ($req ~\* "union[+|(%20)]") {

                return 503;

        }

        if ($req ~\* "and[+|(%20)]") {

                return 503;

        }

        if ($req ~\* "select[+|(%20)]") {

                return 503;

        }

原文链接

http://mp.weixin.qq.com/s?\_\_biz=MzA3MzYwNjQ3NA==&mid=2651297030&idx=1&sn=8cbcd3c5c9e0b27db0283999996a70f4&scene=1&srcid=0818N2Dl5YAomhMRsFU034nx#rd