由于之前已经部署了Zabbix监控系统，本次将结合Zabbix自带的IPMI，完成服务器温度及风扇转速等的监控。

1.环境说明

被监控端服务器型号：Dell PowerEdge R510

规划分配的IPMI地址: 10.103.1.100

2.Zabbix监控平台说明

Zabbix版本: 3.2.1，在安装时，未使用--with-openipmi

Zabbix网络接口可以连通10.103.1.100

3.前置学习

维基百科IPMI:

http://zh.wikipedia.org/wiki/IPMI

IBM DeveloperWorks --使用ipmitool实现Linux系统下对服务器的ipmi管理:

http://www.ibm.com/developerworks/cn/linux/l-ipmi/

Dell--Managing Dell PowerEdge Servers Using IPMItool:

http://www.dell.com/downloads/global/power/ps4q04-20040204-Murphy.pdf

ZabbixIPMIchecks:

https://www.zabbix.com/documentation/3.2/manual/config/items/itemtypes/ipmi

使用IPMITOOL实现终端重定向(课外读物):http://docs.linuxtone.org/ebooks/Dell/ipmitool.pdf

4.配置IPMI

4.1.配置IPMI地址

可以参考前置推荐中的《Managing Dell PowerEdge Servers Using IPMItool》在服务器启动时进行IPMI地址的配置，并开启IPMI Over LAN。

也可以使用Dell的iDRAC开启IPMI功能，具体可以查看文章最后的参考资料。

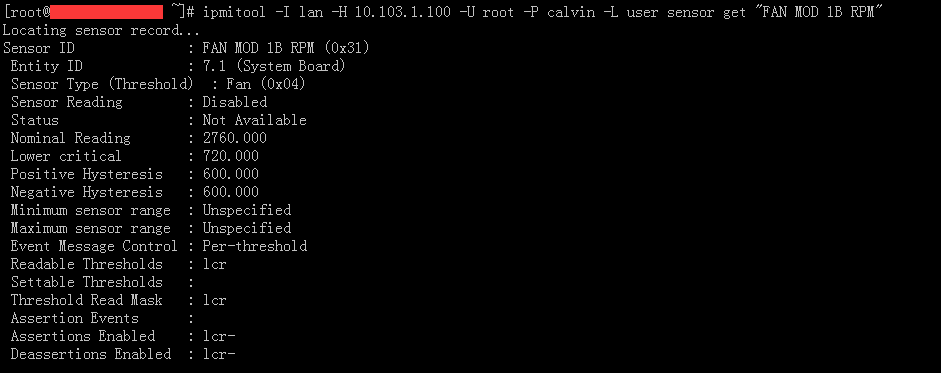


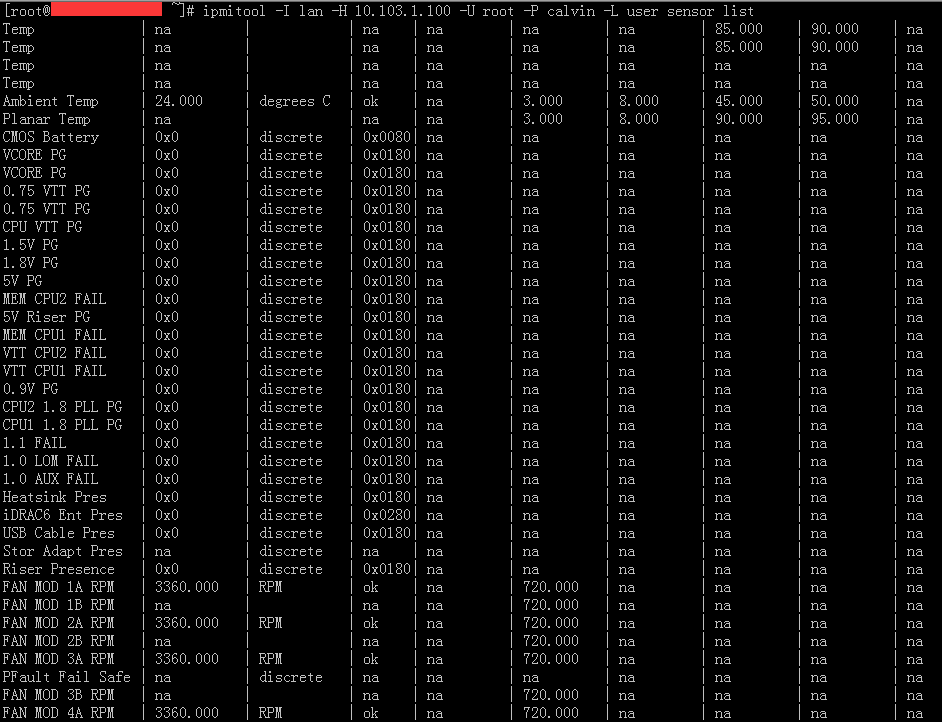
4.2.获取传感器信息

登录Zabbix服务器，通过ipmitool远程访问Dell服务器传感器信息

# ipmitool -I lan -H 10.103.1.100 -U root -P calvin -L user sensor list

# ipmitool -I lan -H 10.103.1.100 -U root -P calvin -L user sensor get "FAN MOD 1B RPM"





4.3.安装IPMItool软件包

# yum -y install OpenIPMI OpenIPMI-devel ipmitool freeipmi

4.4.配置Zabbix

注：为了支持IPMI,需要在zabbix server/proxy安装时增加--with-openipmi参数

服务器端配置zabbix IPMI pollers

zabbix\_server.conf/zabbix\_proxy.conf

# sed -i '/# StartIPMIPollers=0/aStartIPMIPollers=5' zabbix\_server.conf

# service zabbix-server restart

4.5.导入监控模板

下面提供DELL的2个型号的IPMI模板：

https://share.zabbix.com/cat-server-hardware/template-ipmi-dell-poweredge-2950

https://share.zabbix.com/cat-server-hardware/template-ipmi-dell-poweredge-r510

template-ipmi-dell-poweredge-r510

template-ipmi-dell-poweredge-2950

添加监控主机，关联上本模板，并在IPMI页面，设置Authentication algorithm为Default,Privilege level为User, Username为sensor, Password为sensor\_pass，保存即可。

使用此种方法获取数据的结果就是效率很差，基本没什么数据。

5.使用Zabbix External checks自定义IPMI

本来是选择nagios的IPMI插件：check\_ipmi\_sensor，文件是：check\_ipmi\_sensor\_v3-v3.9.tar.gz

具体使用方法详见：http://www.thomas-krenn.com/en/wiki/IPMI\_Sensor\_Monitoring\_Plugin

5.1.安装perl-IPC-Run模块

yum -y install perl-IPC-Run perl-Getopt-Long

5.2.使用check\_ipmi\_sensor查看效果

但是发现报错。

# ./check\_ipmi\_sensor -f ipmi.cfg -H 10.103.1.100 -vvv

------------- debug output for sel (-vvv is set): ------------

/usr/sbin/ipmi-sel was executed with the following parameters:

/usr/sbin/ipmi-sel -h 10.103.1.100 --config-file ipmi.cfg --driver-type=LAN\_2\_0 --output-event-state --interpret-oem-data --entity-sensor-names

output of FreeIPMI:

ID | Date | Time | Name | Type | State | Event

1 | Apr-08-2011 | 06:42:13 | System Board SEL | Event Logging Disabled | Nominal | Log Area Reset/Cleared

------------- debug output for sensors (-vvv is set): ------------

script was executed with the following parameters:

./check\_ipmi\_sensor -f ipmi.cfg -H 10.103.1.100 -vvv

check\_ipmi\_sensor version:

3.9

FreeIPMI version:

ipmi-sensors - 1.2.9

FreeIPMI was executed with the following parameters:

/usr/sbin/ipmi-sensors -h 10.103.1.100 --config-file ipmi.cfg --quiet-cache --sdr-cache-recreate --interpret-oem-data --output-sensor-state --ignore-not-available-sensors --driver-type=LAN\_2\_0 --output-sensor-thresholds

FreeIPMI return code: 0

output of FreeIPMI:

Record ID | Sensor Name | Sensor Group | Monitoring Status | Sensor Units | Sensor Reading

5 | Ambient Temp | Temperature | Nominal | C | 28.000000

不过根据它的提示（其实插件也是调用如下命令），可以使用

/usr/sbin/ipmi-sel -h 10.103.1.100 --config-file ipmi.cfg --driver-type=LAN\_2\_0 --output-event-state --interpret-oem-data --entity-sensor-names执行结果是：

# /usr/sbin/ipmi-sel -h 10.103.1.100 --config-file ipmi.cfg --driver-type=LAN\_2\_0 --output-event-state --interpret-oem-data --entity-sensor-names

ID | Date | Time | Name | Type | State | Event

1 | Apr-08-2011 | 06:42:13 | System Board SEL | Event Logging Disabled | Nominal | Log Area Reset/Cleared

5.3编写Zabbix外部检查（External checks）脚本

# pwd

/usr/local/zabbix/share/zabbix/externalscripts

# cat check\_ipmi

下面是脚本内容

#!/bin/bash

#用于检测ipmi相关信息

#Create on 2016-011-18

#@author: Chinge\_Yang

args="$\*"

echo $(date +%F-%T) $args >> /tmp/check\_ipmi.debug

check\_ipmi\_dir=/usr/local/zabbix/shell/check\_ipmi\_sensor

check\_ipmi\_bin=$check\_ipmi\_dir/check\_ipmi\_sensor

ipmi\_sensors=/usr/sbin/ipmi-sensors

ipmi\_cfg=$check\_ipmi\_dir/ipmi.cfg

#$check\_ipmi\_bin -f $ipmi\_cfg -v $args

#${ipmi\_sel} $args --config-file $ipmi\_cfg --driver-type=LAN\_2\_0 --output-event-state --interpret-oem-data --entity-sensor-names

options="--quiet-cache --sdr-cache-recreate --interpret-oem-data --output-sensor-state --ignore-not-available-sensors --driver-type=LAN\_2\_0 --output-sensor-thresholds"

function usage(){

echo "Usage: `basename $0` options (-h HOST|-n NAME)"

}

function check(){

result=$($ipmi\_sensors -h $host --config-file $ipmi\_cfg $options|grep "$name"|awk -F"| " '{print $NF}')

printf "%.4f\n" $result

}

if [ $# -lt 4 ]

then

usage

exit 55

fi

# 用法: scriptname -options

# 注意: 必须使用破折号 (-)

# 参数后接冒号，表示必须接值

while getopts ":h:n:" Option;do

case $Option in

h)

host=$OPTARG

;;

n)

name=$OPTARG

;;

\*)

usage

;; # 默认情况的处理

esac

done

shift $(($OPTIND - 1))

# (译者注: shift命令是可以带参数的, 参数就是移动的个数)

# 将参数指针减1, 这样它将指向下一个参数.

# $1 现在引用的是命令行上的第一个非选项参数,

#+ 如果有一个这样的参数存在的话.

check

exit 0

添加执行权限

chmod a+x check\_ipmi