

Zabbix库表结构

1. hosts表:

说明: 存放被监控主机相关信息的表。

相关主要字段含义:

hostid: 被监控主机在ZABBIX中的唯一标识。

proxy_hostid: 在zabbix-proxy中的标识。

host: 主机在ZABBIX中的hostname。

status: 主机目前的状态。“0”正常监控,“1”disable,“2”hostavailable,“3”Template

error: 报错信息

2. items表:

说明: 存放监控项相关信息的表

相关主要字段含义:

itemid: 监控项在ZABBIX中的唯一标识

type: 监控项类型

hostid: 监控项应用在host的hostid

name: 监控项名称

key_: item的key值

delay: 数据更新周期

history: 存储历史数据的保留时间

trends: 存储趋势数据保留时间

status: 监控项的状态。0: enable, 1: disable

value_type: 监控项返回值类型

trapper_hosts: 监控为trap时,记录了trapper的名称

error: 记录了监控项报错信息

state: 当前监控项情况。0: 正常 1: 不支持

3. triggers表:

说明: 存放触发器相关信息的表

相关主要字段含义:

triggerid: 触发器唯一标识ID

expression: 触发器表达式

description: 触发器名称

url: 链接信息

status: 启用状态。1: 启用, 2: 停用

value:

priority: 严重级别

lastchange: 最新更新时间

comments: 描述信息

error: 报错信息

templateid: 对应模板的ID

type:

state: 触发器是否被支持。0: 支持, 1: 不支持

flags:

4. events表:

说明: 每当收到数据,就会检查监控项目相关的触发器,无论触发器是否出发,就会生成一个event

相关主要字段含义:

eventid: event唯一标识

source: 生成event的源头。0: trigger, 1: discovery rule, 2: agent auto-registration, 3: internal

object: event生成的相关的对象。0: trigger, 1: discovered host, 2: discovered service, 3: auto-registration, 4: item, 5: low-level discovery

objectid: 与event生成相关的唯一标识

clock: 生成时间(unix时间戳形式)

value: 对不同类型source字段类型,值不同

对于trigger类型的event:“0”正常“1”出现问题

对于discovery类型的event:“0”正在工作 “1” 停止工作 “2”被侦测到“3”丢失了
对于internal类型的event:“0”正常状态“1”未知或不支持状态

5. history表：

说明：存放所有收集到的原始数据

（相关表：hsitory_str, history_text, history_uint）（表结构相同，数据类型不同）

相关主要字段含义：

itemid：数据的项目id

clock：生成时间

value：最新数据

ns：纳秒

6. trends表

说明：将history表的数据根据小时的唯独进行归档

相关主要字段含义：

itemid：数据设计的监控项id

clock：生成时间

num：每小时使用数据的数量

value_min：每小时最小值

value_avg：每小时平均值

value_max：每小时最大值

7. alerts表

说明：记录了报警相关的历史记录

相关主要字段含义：

alertid：报警唯一标识

actionid：动作唯一标识

eventid：事件唯一标识

userid：用户唯一标识

clock：生成事件

mediatypeid：未知

sendto：告警邮件发送地址

subject：告警主题

message：告警内容

status：确认状态。0：未确认，1：已确认

retries：告警通知重试次数

error：错误信息

esc_step：

alerttype：

8. actions表记录了当触发器触发时，需要采用的动作。

alerts 表保存了历史的告警事件，可以从这个表里面去做一些统计分析，例如某个部门、某人、某类时间的告警统计，以及更深入的故障发生、恢复时间，看你想怎么用了。

config表保存了全局的参数，前端包括后端也是，很多情况下会查询改表的参数的，例如用户的自定义主题、登陆认证类型等，非常重要。

function 表记录了trigger中使用的表达式，例如max、last、nodata等函数。

graphs 表包含了用户定义的图表信息。

graphs_items 保存了属于某个图表的所有的监控项信息。

groups 保存了组名和组的ID。

这部分表都差不多，唯一不同的是保存的数据类型，history_str保存的数据类型就算str即字符类型的。这个是和采集时设置的数据类型一致的。需要注意的时，因为history表有这么多的类型，那自己写报表系统等去查询数据时，就需要判断下数据的采集类型，如果查错了表，那肯定是没有数据的。

trends 也是保存了历史数据用的，和history不同的时，trends表仅仅保存了 小时平均的值，即你可以理解是为history表的数据压缩。所以trends表也有很多的类型，对应history。

值的注意的trends和history表这两类表数据量都非常大，我们一天大概就要有 40G 的数据。

所以注意定是去做压缩、删除。

hosts 非常重要，保存了每个agent、proxy等的IP、hostid、状态、IPMI等信息，几乎是记录了一台设备的所有的信息。

当然hostid是当中非常非常重要的信息，其他的表一般都时关联hostid的。

hosts_groups 保存了host（主机）与host groups（主机组）的关联关系。

这部分信息可以在我们自己做一些批量查询，例如查询关联到某个主机组的所有 设备的IP 、存活状态等，进一步去查询该批量设备的load、IO、mem等统计信息。

items 表保存了采集项的信息。

media 保存了某个用户的media配置项，即对应的告警方式。

media_type 表与media 表不同的是media_type 记录了某个告警方式对应的执行脚本，注意路径只是相对路径。

media 与media_type 通过mediatypeid 键关联。

profiles 表保存了用户的一些配置项。

rights 表保存了用户组的权限信息，zabbix的权限一直也是我理不太清的地方， 其实这个表里面有详细的记录。

screens 表保存了用户定义的图片。

screen_items 同graphs_items。

sessions 表保存了每个用户的sessions,在登陆、注销的时候均会操作 该张表的。

triggers 顾名思义保存了trigger的所有信息。

trigger_depends 保存了trigger的依赖关系。

users 部分用户配置会在该表中，例如autologin、autologout、 url、theme等信息。

zabbix 最重要三张表：

- **items**:存放最新一次采集项的值，存放采集项，模板信息等
- **history**: 这是一个系列的表如：**history_uint,history_str**,存放到哪张表是根据你 设置采集项选择数据类型的时候决定的，比如**uint**，存放的是整形的！
- **trends**: 每个小时根据对于**history**采集的值算出的平均值

手动查询数据库的基本逻辑：

1. 通过匹配hosts表中host字段查找hostid
2. 匹配items表中的hostid和key_字段查找itemid
3. 匹配history_uint表itemid字段查询clock、value字段既是我们需要的时间和数值。

```
select hostid from hosts where host = "app10-115";
select itemid from items where hostid = "10648" and key_ = "net.if.in[eth0]";
select from_unixtime(clock) as Date_Time,round(value/1024/1024,2) as Traffic_in from history_uint where itemid = "87488" order
by clock desc limit 10;
```

1. 查找模板id：

```
select hostid from hosts where host like "%ops_dba%" and status = 3 // status = 0 为主机
```

2. 查找itemid：

```
select itemid from items where hostid = templateid
```

3. 查找触发器id：

```
select triggerid from functions where itemid = itemid
```

4. 查找触发器：

```
select * from triggers where triggerid = triggerid and status = 1
```

```
// status = 1 触发器停用
```

```
// status = 0 触发器可用
```

5. 查找最新更新的一条采集项的值 查找特定键值的数据：

```
select *,from_unixtime(lastclock) from items where hostid = 10213 and key_ like "%lsi.raid%"G;
```

