# 1.功能基本信息

## 1.1概述

本程序为ganglia工程的组件gmetad的一个插件。Ganglia是UC Berkeley发起的一个开源集群监视项目，设计用于测量数以千计的节点。Ganglia的核心包含gmond、gmetad以及一个Web前端。主要是用来监控系统性能，如：cpu 、mem、硬盘利用率， I/O负载、网络流量情况等，通过曲线很容易见到每个节点的工作状态，对合理调整、分配[系统资源](http://baike.baidu.com/view/53557.htm)，提高系统整体性能起到重要作用。Gmond只有C语言实现版本，Gmetad有C语言和Python语言实现版本。本程序是基于Python语言版本。

本程序主要实现了监控项目定义，对监控项目细粒度报警，对报警人员细粒度的划分，多种报警方式，日志处理等功能。

## 1.2功能

### 1.2.1 报警项目

监控项目包括目标机器多种维度，大类分为CPU，MEM，DISK，NETIO，DISKIO和目标机器基本信息（只要监控中有的项目就可以添加到报警项目中去），在CPU大类中又可以分为cpu\_idle,cpu\_user,cpu\_sys等小类。我们设置报警项目就是以这些小类为主体构建的。

### 1.2.2 报警组

报警组和用户之间的关系是多对多。报警项目只和报警组有多对多的关系，不存在和用户之间的关系。我们可以对报警组和用户进行管理。

### 1.2.3 日志处理

日志模块为工程中的一些错误记录信息。同时也是一种报警信息的记录方式。对于报警的信息还设计了一个确认的功能，表示信息已经阅读。

# 2.程序设计

## 2.1 程序命名

包名：下划线命名法

文件名：下划线命名法

类名：驼峰命名发

文件名：下划线命名法

方法名：驼峰命名法

变量名：下划线命名法

## 2.2 原则

1.报警项目包含很多的报警规则。

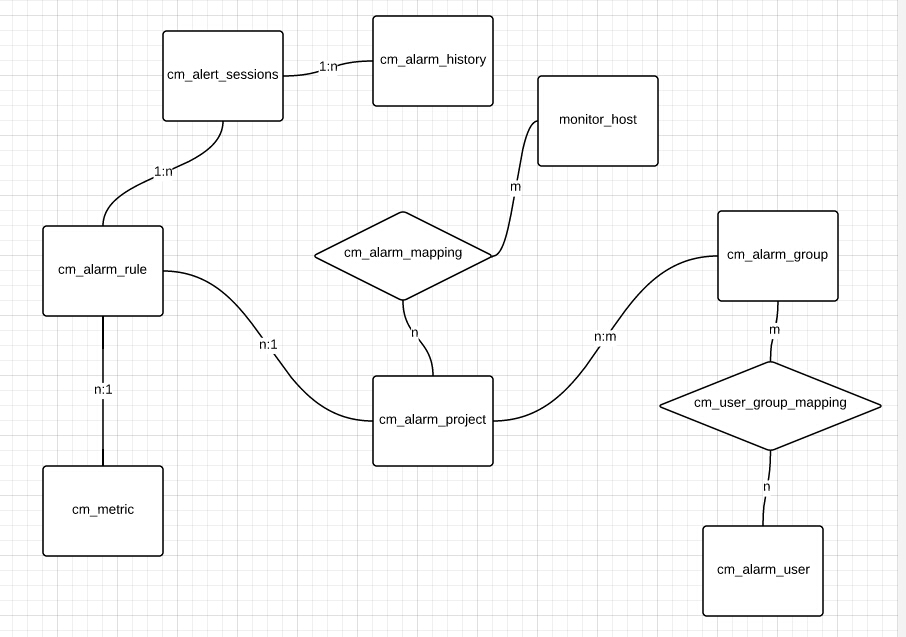
2.报警规则包含报警方式和报警等级，报警方式和报警等级属于多对多关系，因此报警规则显得特别的冗余。

3.报警项目和报警组的关系，没有使用关系表而是在报警项目表中增加一个字段来维护报警组的关系。

4.报警项目与监控对象之间的关系通过cm\_alarm\_mapping来记录，同时记录的还有报警组的信息，为了方便的查询。

5.报警工程里有一个session的概念，引入session的概念主要为了防止过多的报警。当触发一个规则时就创建一个对应的session并且session属性值计数加1，以后每次触发就执行加一操作；当session的计数器超过报警规则里字段alarm\_times时就报警；当超过一定数值时，session失效；当session超过一个过期时间并且没有接受到报警时也会失效。

## 2.3 相关表以及ER图



报警规则表cm\_alarm\_rule

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 报警规则Id |
| alarm\_times | int(11) | NOT NULL DEFAULT '0' | 连续报警多少次后开始报警 |
| metric\_id | bigint(20) | NOT NULL | 监控对象的id |
| metric\_name | varchar(255) | NOT NULL | 监控项名称 |
| metric\_desc | varchar(255) | NOT NULL | 监控项描述 |
| alarm\_project\_id | bigint(20) | NOT NULL | 报警项目id |
| alarm\_project\_name | varchar(255) | NOT NULL | 报警项目名称 |
| disabled | tinyint(2) | NOT NULL DEFAULT '0' | 是否生效 |
| alarm\_frequency | int(11) | NOT NULL DEFAULT '0' | 事件报警频率 |
| alarm\_method | int(11) | DEFAULT '1' | 报警配置 1日志 2邮件 3手机 |
| statistic | int(11) | NOT NULL | 报警判断的统计值 |
| comparison\_operator | varchar(255) | NOT NULL | 报警判断条件，如>=分号 |
| threshold | varchar(64) | NOT NULL | 报警阀值 |
| alarm\_level | int(11) | DEFAULT '1' | 1 warning 2 critical |

报警项目表cm\_alarm\_project

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 报警规则Id |
| alarm\_project\_name | varchar(255) | NOT NULL | 报警项目名称 |
| alarm\_project\_desc | varchar(255) | NOT NULL | 报警项目名称（中文） |
| alarm\_group\_ids | varchar(255) | NOT NULL DEFAULT '0' | 报警组id集合，以逗号隔开 |
| update\_time | datetime | DEFAULT NULL | 更新时间 |
| is\_auto | int(1) | NOT NULL DEFAULT '0' | 报警判断的统计值 |

报警组cm\_alarm\_group

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 报警组Id |
| alarm\_group\_name | varchar(255) | NOT NULL | 报警组名称 |
| alarm\_group\_desc | varchar(512) | NOT NULL | 报警组描述（中文） |

报警用户cm\_alarm\_user

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 无意义 |
| alarm\_user\_name | varchar(255) | NOT NULL | 用户名 |
| email | varchar(255) | NOT NULL | 邮箱 |
| phone | varchar(255) | NOT NULL | 手机 |
| office\_phone | varchar(255) | NOT NULL | qq |
| remark | varchar(255) | NOT NULL | 复述 |

报警会话cm\_alert\_sessions

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 无意义 |
| rule\_id | bigint(20) | NOT NULL | 报警规则id |
| fixed\_ip | varchar(39) | NOT NULL | 报警目标机器ip |
| last\_alert | datetime | DEFAULT NULL | 最后一次报警时间 |
| alert\_times | int(11) | DEFAULT '1' | 已报警次数 |
| is\_active | tinyint(4) | DEFAULT '1' | Session是否有效0为无效 |

报警种类cm\_metric

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 报警种类id |
| metric\_name | varchar(255) | NOT NULL | 报警种类名称 |
| metric\_desc | varchar(255) | NOT NULL | 报警种类名称(中文) |

报警用户与报警组关系表cm\_user\_group\_mapping

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 无意义 |
| alarm\_group\_id | varchar(255) | NOT NULL | 报警组Id |
| alarm\_user\_id | varchar(255) | NOT NULL | 用户Id |
| alarm\_group\_name | varchar(255) | DEFAULT NULL | 报警组名称 |
| alarm\_user\_name | varchar(255) | DEFAULT NULL | 用户名 |
| alarm\_config | int(11) | DEFAULT '0' | 废弃 |

报警项目与监控机器关系表cm\_alarm\_mapping

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 无意义 |
| alarm\_project\_id | bigint(20) | NOT NULL | 报警项目id |
| fixed\_ip | varchar(39) | NOT NULL | 报警目标机器唯一标示，目前定义是IP |
| is\_vm | int(1) | DEFAULT '1' | 是否是虚拟机 |
| alarm\_group\_ids | varchar(255) | NOT NULL | 报警组id集合，以逗号隔开 |

报警历史记录表cm\_alarm\_history

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 备注 |
| id | bigint(20) | NOT NULL | 无意义 |
| alarm\_rule\_id | bigint(20) | NOT NULL | 报警规则Id |
| metric\_name | varchar(255) | DEFAULT NULL | 监控项名称 |
| metric\_desc | varchar(255) | DEFAULT NULL | 监控项描述 |
| alarm\_time | datetime | DEFAULT NULL | 报警创建时间 |
| alarm\_content\_summary | varchar(1000) | NOT NULL | 简要报警内容（手机报警内容） |
| alarm\_content | text | NOT NULL | 报警内容 |
| alarm\_group\_ids | varchar(255) | NOT NULL | 报警组id 多个报警组用分号分割 |
| alarm\_group\_names | varchar(255) | NOT NULL | 报警组名称,多个用分号(,)分割 |
| fixed\_ip | varchar(255) | NOT NULL | 报警目标机器唯一标示，目前定义是IP |
| is\_verify | int(1) | NOT NULL | 报警确认（0表示未确认） |

# 3.源代码说明

## 3.1 gmetad-python源码

gmetad-python的功能为：1.从gmond端拉取数据(xml) 2.把xml数据转换成数据对象 3.聚合gmond提供的数据 4.提供查询数据的接口 5.提供丰富的插件接口 。

gmetad.py是程序的入口，gmetad.py主要干了2件事情，一件是初始化一些基本功能的对象比如创建了log信息，读取配置文件信息，创建了DataStore实例，接着创建了xmlsocket(返回所有信息)和interactivesocket实例(根据参数返回查询信息)。另外一件是创建了一个守护进程，该守护进程的功能就是不停的根据配置文件里的信息进行轮询。

gmetad\_data.py包含了2个类DataStoreGridSummary和DataStore，DataStore引用DataStoreGridSummary对象。Datastore主要做数据聚合以及触发notify。

gmetad\_notify.py定义一个线程类GmetadNotifier，功能是加载插件。

gmetad\_plugin.py定义一些控制插件的方法，加载插件，启动插件，停止插件，notify插件。定义一个插件模板类GmetadPlugin。

alert\_plugin.py是一个插件文件，继承自模板类GmetadPlugin，实现了notify方法，方法的功能是根据传入的数据对象clusternode来实现报警，报警的基础数据从数据源中获取，数据源获取的方式使用了反射来获取数据源的实现，目前只有一种实现mysql，而报警的具体实现是在alert方法中。

## 3.2报警插件源码

alert\_store.py是一个接口类，定义了基础数据获取的抽象类，继承自GmetadConfable，继承的方法是读取相关配置。

mysql\_store.py 是一个基础数据获取的具体实现，主要能获取报警规则getHMRules，获取联系人getContacts，存放报警信息storeAlertMsg，以及session的操作。

alert\_rule.py定义了2个类，AlertSession和AlertRule类，这2个类为数据库表的映射类，AlertRule类的功能还有是否应该报警和提供报警模板。

alert\_method.py 是一个接口类，定义了报警实现的基础类继承自GmetadConfable，继承的方法是读取相关配置。

log\_method.py 继承自alert\_method，实现了log报警。

mail\_method.py继承自alert\_method，实现了mail报警。