

Ganglia

搭建和介绍使用方法

quqinglei@icloud.com

October 23, 2013

目录

1	Ganglia 介绍	1
1.1	简介	1
1.2	工作原理	2
2	安装	2
2.1	gmond	3
2.2	gmetad	3
2.3	gweb	4
3	模块	5
4	配置	5
4.1	gmond	5
5	备注	9

1 Ganglia 介绍

1.1 简介

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，提高系统整体性能起到重要作用。

每台计算机都运行一个收集和发送度量数据的名为 gmond 的守护进程。接收所有度量数据的主机可以显示这些数据并且可以将这些数据的精简表单传递到层次结构中。正因为有这种层次结构模式，才使得 Ganglia 可以实现良好的扩展。gmond 带来的系统负载非常少，这使得它成为在集群中各台计

计算机上运行的一段代码，而不会影响用户性能。所有这些数据多次收集会影响节点性能。网络中的“抖动”发生在大量小消息同时出现时，可以通过将节点时钟保持一致，来避免这个问题。

gmetad 可以部署在集群内任一节点或者通过网络连接到集群的独立主机，它通过单播路由的方式与 gmond 通信，收集区域内节点的状态信息，并以 XML 数据的形式，保存在数据库中。

由 RRDTool 工具处理数据，并生成相应的图形显示，以 Web 方式直观的提供给客户端。

官网地址: <http://ganglia.sourceforge.net/>

1.2 工作原理

Ganglia 包括如下几个程序，他们之间通过 XDL(xml 的压缩格式) 或者 XML 格式传递监控数据，达到监控效果。集群内的节点，通过运行 gmond 收集发布节点状态信息，然后 gmetad 周期性的轮询 gmond 收集到的信息，然后存入 rrd 数据库，通过 web 服务器可以对其进行查询展示。

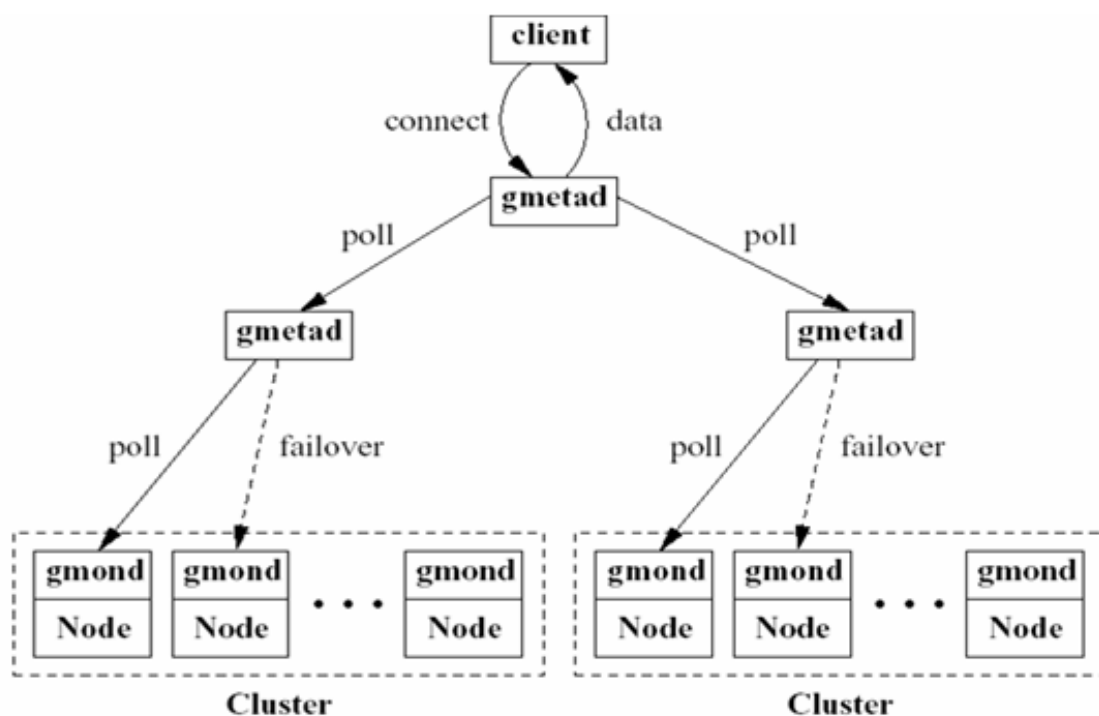


图 1: 工作原理

2 安装

我们刚才提到过 ganglia 是由 gmond、gmetad 以及 gweb 三部分组成的，安装较为简单，这三部分分别安装完后稍微配置一下就可以了，步骤如下：

2.1 gmond

gmond 意思是 Ganglia 的监控守护进程。它是一个轻量级服务，必须安装在每一个你需要采集数据的节点机上。

安装依赖 gmond 安装方法很简单，在现代 Linux 发行版上基本都有它需要的动态库，它需要 libconfuse, pkgconfig, PCRE, APR 库。在大多数发行版上都有其安装包，所以如果你使用包管理的 Linux 发行版解决依赖关系就不需要你操心了，简单的运行包管理工具自动安装就可以了。下面是不同发行版的安装方式。

```
1 # 基于 debian 的发行版，debian 基本都拥有此包
2 user@host:# apt-get install ganglia-monitor
3
4 # 基于 RPM 管理的发行版
5 user@host:# yum search ganglia-gmond
6
7 # 如果查不到可以使用一下命令添加软件仓库
8
9 # Red Hat 5.x:
10 user@host:# rpm -Uvh \ # 32 bit
11 http://mirror.ancl.hawaii.edu/linux/epel/5/i386/epel-release-5-4.noarch.rpm
12
13 user@host:# rpm -Uvh \ # 64 bit
14 http://mirror.ancl.hawaii.edu/linux/epel/5/x86_64/epel-release-5-4.noarch.rpm
15
16 # Red Hat 6.x
17 user@host:# rpm -Uvh \ # 32 bit
18 http://mirror.ancl.hawaii.edu/linux/epel/6/i386/epel-release-5-4.noarch.rpm
19
20 user@host:# rpm -Uvh \ # 64 bit
21 http://mirror.ancl.hawaii.edu/linux/epel/6/x86_64/epel-release-5-4.noarch.rpm
22
23 user@host:# yum install ganglia-gmond
24
25 # 如果没法通过网络安装，可以采用编译的方式，去官网下载源代码编译安装
```

2.2 gmetad

gmetad 从其他 gmetad 或者 gmond 获取数据，并存储数据到 rrd 格式的文件。它提供了一个简单的查询组用于搜集特定信息的机制，并支持分级授权，使人们可以建立联合监控域。

安装依赖 安装依赖了 gmond 基本一样，但需要安装 RRDtool，用来存放数据，使用包管理工具，可自动处理此依赖

```
1 # 基于 debian 的发行版，debian 基本都拥有此包
2 user@host:# apt-get install gmetad
3
```

```

4 # 基于 RPM 管理的发行版
5 user@host:# yum install ganglia-gmetad

```

2.3 gweb

Ganglia 需要 gweb 才能完整，它提供一个 PHP 为前端的 UI。

安装依赖 Ganglia 3.4.0 以后，前端就被分离出来了，且版本号和 gmond 以及 gmetd 都不再一致。gweb 3.4.0 支持所有版本的 gmond/gmetad 3.1.x 或者更高版本。具体还是看官方文档怎么写的，下面是依赖：

- Apache Web Server
- PHP 5.2 or later
- PHP JSON extension installed and enabled

```

1 # 基于 debian 的发行版，debian 基本都拥有此包
2 user@host:# apt-get install apache2 php5 php5-json
3
4 # 测试配置文件中是否启用了 json 模块
5 user@host:# grep ^extension=json.so /etc/php5/conf.d/json.ini
6 # 如果 json 模块没有启用，可以通过以下步骤
7 user@host:# echo 'extension=json.so' >> /etc/php5/conf.d/json.ini
8
9 # 执行完以上步骤，可以去官网下载 gweb 包，解压到 /var/www/ 路径，并重命名为 ganglia
10 root@host:# tar -xzf ganglia-web-major.minor.release.tar.gz
11 root@host:# mv ganglia-web-major.minor.release ganglia
12 root@host:# cd ganglia
13 root@host:# make install
14 root@host:# cd /var/www
15 root@host:# chown -R www-data:www-data ganglia
16
17 # 以上步骤就安装好了，重启 apache2 服务器，就能在 http://localhost/ganglia 里面看到页面了
18
19 # 基于 RPM 管理的发行版
20 user@host:# yum install httpd php
21
22 # 接下来安装 PHP5.2 并配置好 JSON 的支持，这个和 debian 差不多，具体问题具体分析吧，下面基本就一样了
23 # 注意 chown 的时候应该把权限赋给 apache 用户，查看/etc/passwd 文件看看，应该是 apache
24
25
26 # 注：在基于 debian 的系统上可以直接使用包管理工具安装 WEB 端
27 user@host:# apt-get install ganglia-webfrontend
28 user@host:# mv /usr/share/ganglia-webfrontend/ /var/www/ganglia
29 user@host:# chown -R www-data:www-data /var/www/ganglia

```

3 模块

ganglia 支持添加模块，并支持添加 C、Python、Perl、bash 的数据采集模块，可以理解成插件，它使用了 Apache 的 apr 来加载。具体编写方法每种语言是不同的，但我们使用的绝大部分插件它已经提供了，所以基本上不需要我们再另行编写，关于如何编写的具体方式可以查看：<http://sourceforge.net/apps/trac/ganglia/wiki/> 在这里就不做过多的介绍了。

4 配置

4.1 gmond

配置文件名：`gmond.conf`

该文件一般位于 `/etc/ganglia/gmond.conf` 路径。

描述

此文件为 gmond 所使用，是此守护进程的配置文件，包含要收集数据的一些配置，下面具体讲解一下：

具体配置讲解

- 配置中的配置项是大小写敏感的
- 有些节的配置是可以重复，有些不可以，你只能设置一个 `cluster` 但可以设置很多的 `module`

cluster

此配置文件中的 `cluster` 项只能配置一个，这个配置控制着 UI 上的报告，即此机器属于哪个集群。它有四个属性：`name`, `owner`, `latlong`, `url`。

```
1 cluster {  
2     name = "Millennium Cluster"  
3     owner = "UC Berkeley CS Dept."  
4     latlong = "N37.37 W122.23"  
5     url = "http://www.millennium.berkeley.edu/";  
6 }
```

name 定义集群名称，也就是此台机器属于哪个集群

owner 集群管理者的名字，`name` 和 `owner` 在所有集群中必须唯一

latlong GPS 经纬度，即机器的地理位置，当然可以不写，根据自己的情况来啦

url 集群的信息，具体查询位置，当然可以不写，必经不是每个集群都有网站说明

host

```

1 host {
2     location = "1,2,3"
3 }

```

机架位子，根据自己的情况来，可以不写

globals

```

1 globals {
2     daemonize = true
3     setuid = true
4     user = nobody
5     host_dmax = 3600
6 }

```

daemonize 是否以守护进程的形式启动，一般设为 true

setuid 为真时，将设置其有效的有用户指定的 uid，为假时不会改变他的有效用户，默认为真

user 这个和上面的一项有关系，即这个进程以哪个用户启动，如果使用包管理工具，它的默认值应该是 ganglia 当然可以是其他用户，一般来说在 Linux 下为了安全会把软件安装到不同的用户下，这样权限划比较合理。

host_dmax 这个一般设置为 0，它的单位是秒，如果设置了数，当此机器在此时间内不再报告，gmond 会删除它

udp_send_channel

可以把它理解成为转发，gmond 可以转发数据到其他 gmond 上，这样在收集数据的时候只需要读取那一个 gmond 上的数据。

```

1 udp_recv_channel {
2     port = 8666
3     family = inet4
4 }
5 udp_recv_channel {
6     port = 8666
7     family = inet6
8 }

```

udp_recv_channel

tcp_accept_channel

以上三个功能分别为转发、接收、TCP 协议，具体使用功能待测，没有完全理解 [tbd]

collection_group 这项可以设置很多，使用 `gmond -m` 能得到所有的内置测量工具，格式如下：

```

1 collection_group {
2     collect_once = yes
3     time_threshold = 1800
4     metric {
5         name = "cpu_num"
6         title = "Number of CPUs"
7     }
8 }
9
10 collection_group {
11     collect_every = 60
12     time_threshold = 300
13     metric {
14         name = "cpu_user"
15         value_threshold = 5.0
16         title = "CPU User"
17     }
18     metric {
19         name = "cpu_idle"
20         value_threshold = 10.0
21         title = "CPU Idle"
22     }
23 }

```

collect_once 上面的意思是在重启之前只搜集一次信息

time_threshold 上面的 1800 代表 1800 秒发送一次 cpu_num 的数据

name 内置的采集器名称，可以通过 `gmond -m` 获得所有的内置数据采集器

title 这个可以根据情况设了，就是一个描述字段

collect_every 数据采集间隔时间，单位为秒

modules

这个是插件的配置信息，ganglia 支持插件，也就是非内置的信息采集方式，用户可以根据它提供的格式写一些插件，基本上不用写了，在其官网上有我们能用到的几乎所有插件。

```

1 modules {
2     module {
3         name = "example_module"
4         enabled = yes
5         path = "modexample.so"
6         params = "An extra raw parameter"
7         param RandomMax {
8             value = 75
9         }
10        param ConstantValue {

```

```
11         value = 25
12     }
13 }
14 }
15
16 modules {
17     module {
18         name = "core_metrics"
19     }
20     module {
21         name = "cpu_module"
22         path = "/usr/lib/ganglia/modcpu.so"
23     }
24     module {
25         name = "disk_module"
26         path = "/usr/lib/ganglia/moddisk.so"
27     }
28     module {
29         name = "load_module"
30         path = "/usr/lib/ganglia/modload.so"
31     }
32     module {
33         name = "mem_module"
34         path = "/usr/lib/ganglia/modmem.so"
35     }
36     module {
37         name = "net_module"
38         path = "/usr/lib/ganglia/modnet.so"
39     }
40     module {
41         name = "proc_module"
42         path = "/usr/lib/ganglia/modproc.so"
43     }
44     module {
45         name = "sys_module"
46         path = "/usr/lib/ganglia/modsys.so"
47     }
48 }
```

name 模块名，或者说插件名称

path 模块位置，上面的例子主要是动态链接库，其实它还支持其他语言编写的插件

include

配置文件可以再包含其他配置文件，这个基本上是为了扩展其他语言而定义的此项

```
1 include ('/etc/ganglia/conf.d/*.conf')
```

access control

访问控制

```
1 acl {  
2     default = "deny"  
3     access {  
4         ip = 192.168.0.4  
5         mask = 32  
6         action = "allow"  
7     }  
8 }
```

udp_recv_channel 和 tcp_accept_channel 可以设置访问控制列表，第一个例子会拒绝来自 192.168.0.4 的数据包第二个例子默认允许接收来自 192.168.0.0 子网的所有数据包

5 备注

ganglia 的配置方式比较简单，可以采用其作为 hadoop 的集群监控软件和测试 Hadoop 性能的工具，或者其他集群的监控工具，也可以配合 nagios。在思考是否可以从中提取一些有效信息为 Hadoop 混合型集群的任务分配提供数据依据。