

目录

21.	典型大数据平台监控运维实战.....	2
21.1.	项目简介	2
21.1.1.	项目背景.....	2
21.1.2.	预备知识.....	2
21.1.3.	实验环境.....	2
21.1.3.1.	安装 ganglia 所需依赖	2
21.1.3.2.	监控端安装 gmeta,gmondganglia-web,nginx,php.....	3
21.1.3.3.	被监控端安装 gmond.....	8
21.1.4.	数据集.....	8
21.1.5.	实验任务.....	8
21.2.	开启 Ganglia 监控 hadoop 集群	9
21.2.1.	修改 ganglia-monitor 的配置文件.....	9
21.2.2.	主节点配置	9
21.2.3.	修改 Hadoop 的配置文件	9
21.2.4.	重启所有服务	10
21.2.5.	访问页面查看各机器节点信息	10
21.3.	本地数据上传到分布式文件系统 HDFS.....	11
21.3.1.	准备工作.....	11
21.3.2.	进行上传操作	13
21.4.	用数据仓库 Hive 查询数据	14
21.4.1.	准备工作.....	14
21.4.2.	进行查询操作	15
21.5.	上传和查询操作中 Ganglia 监控到的状态	16
21.5.1.	上传数据前后集群状态变化	16
21.5.2.	查询数据前后集群状态变化	17

21. 典型大数据平台监控运维实战

21.1. 项目简介

21.1.1. 项目背景

通过完成本章的典型大数据平台监控运维实战，您应该能够：

- 掌握大数据平台监控运维的基本流程
- 掌握 Ganglia 的安装及配置方法
- 掌握如何利用 Ganglia 分析 HDFS 上传操作
- 掌握如何利用 Ganglia 分析 Hive 查询操作

21.1.2. 预备知识

- 熟悉常用 Linux 操作系统命令
- 熟悉常用 Hadoop 集群搭建
- 熟悉常用 HDFS 操作命令
- 熟悉 MySQL 数据库常用操作
- 熟悉常用 Hive 操作命令

21.1.3. 实验环境

21.1.3.1. 安装 ganglia 所需依赖

步骤一：关闭 selinux

```
[root@master ~]# setenforce 0
```

步骤二：安装 CentOS 企业扩展 YUM 源（子节点也要）

```
[root@master ~]# rpm -ivh /opt/software/epel-release-latest-7.noarch.rpm
```

```
[root@master ~]# yum repolist
```

步骤三：安装依赖包

更新检索秘钥

```
[root@master ~]# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-7
```

```
[root@master ~]# wget -O /etc/yum.repos.d/CentOS-Base.repo
```

```
http://mirrors.aliyun.com/repo/Centos-7.repo
```

```
[root@master ~]# sed -i 's/$releasever/7/g' /etc/yum.repos.d/CentOS-Base.repo
```

```
[root@master ~]# yum repolist
```

这步要保证全部依赖安装完成，不然后面会出问题。

```
[root@master ~]# yum -y install gcc glibc glibc-common rrdtool rrdtool-devel apr apr-
devel expat expat-devel pcre pcre-devel dejavu-lgc-sans-mono-fonts dejavu-sans-mono-fonts
zlib zlib-devel libconfuse libconfuse-devel
```

21.1.3.2. 监控端安装 gmeta,gmond,ganglia-web,nginx,php

步骤一：监控端安装 gmond 及 gmeta

将下载后的 ganglia-3.7.2.tar.gz 放至 /root 目录下，然后执行以下操作

```
[root@master ~]# tar -zxvf /opt/software/ganglia-3.7.2.tar.gz
[root@master ~]# mv ganglia-3.7.2 /usr/local/src/ganglia
[root@master ~]# cd /usr/local/src/ganglia
[root@master ganglia]# ./configure --prefix=/usr/local/src/ganglia_make --with-gmetad
--enable-gexec
[root@master ganglia]# # make && make install
```

步骤二：安装 nginx

```
[root@master ganglia]# cd
[root@master ~]# yum install nginx -y
[root@master ~]# chkconfig nginx on
```

启动时有可能出现 80 端口冲突导致无法启动 nginx 服务，解决方法：查看哪个服务占用了

80 端口

```
[root@master ~]# netstat -ntlp
```

关闭占用 80 端口的服务

```
[root@master ~]# systemctl stop httpd.service
```

启动 nginx

```
[root@master ~]# systemctl start nginx
```

步骤三：安装 php

```
[root@master ~]# yum --enablerepo=remi,remi-php55 install php-fpm php-common
php-devel php-mysqlnd php-mbstring php-mcrypt
[root@master ~]# chkconfig php-fpm on
[root@master ~]# systemctl start php-fpm
```

步骤四：配置 nginx 代理访问 php

```
[root@master ~]# vim /etc/nginx/nginx.conf
```

server 处增加 location 配置块：

```
location ~ /\.php$ {
    root          /var/www;
    fastcgi_pass  127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root/$fastcgi_script_name;
    include       fastcgi_params;
}
```

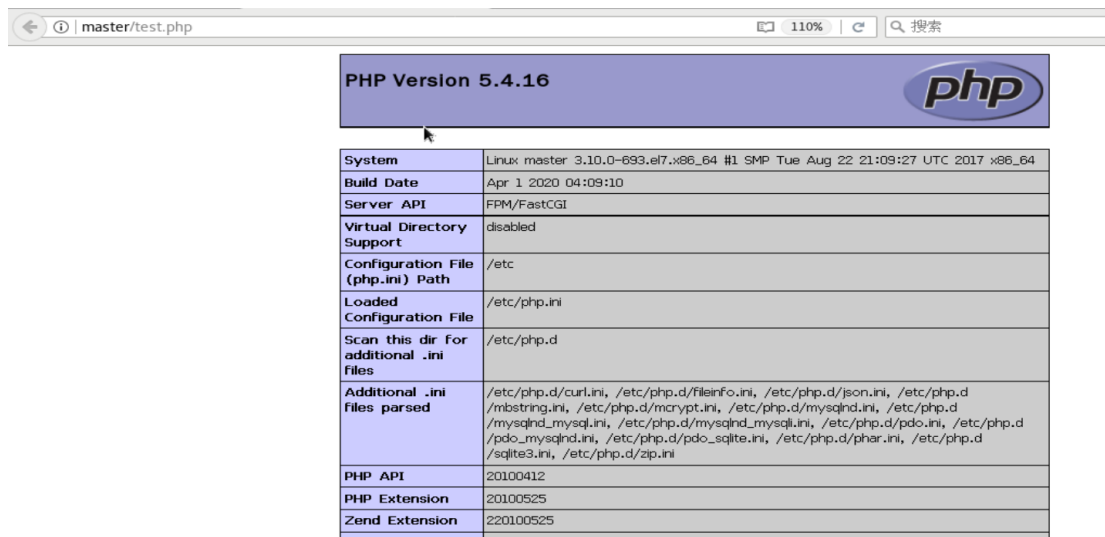
##重启 nginx

```
[root@master ~]# systemctl restart nginx
```

步骤五：测试 PHP+Nginx

```
[root@master ~]# mkdir /var/www
[root@master ~]# cd /var/www
[root@master www]# vim test.php
<?php
phpinfo();
?>
```

访问：master/test.php，出现如下界面即为调试成功



PHP Version 5.4.16	
System	Linux master 3.10.0-693.el7.x86_64 #1 SMP Tue Aug 22 21:09:27 UTC 2017 x86_64
Build Date	Apr 1 2020 04:09:10
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/mbstring.ini, /etc/php.d/mcrypt.ini, /etc/php.d/mysqli.ini, /etc/php.d/mysqli_mysql.ini, /etc/php.d/mysqli_sqlite.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/sqlite3.ini, /etc/php.d/zip.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extensions	apcu, gd, gettext, iconv, intl, json, ldap, libxml, mbstring, openssl, pcre, pdo_mysql, pdo_sqlite, readline, soap, sockets, sysvmsg, sysvshm, tokenizer, xsl, xmlwriter, zip

步骤六：配置 gmeta

```
[root@master www]# cd
[root@master ~]# mkdir -p /var/lib/ganglia/rrds
[root@master ~]# chown nobody:nobody /var/lib/ganglia/rrds
[root@master ~]# cd /usr/local/src/ganglia
[root@master ganglia]# cp ./gmetad/gmetad.init /etc/init.d/gmetad
```

修改 gmetad，具体值通过 “find / -name 'gmetad' -print” 查

```
[root@master ganglia]# vim /etc/init.d/gmetad
GMETAD=/usr/local/src/ganglia_make/sbin/gmetad
```

修改 gmetad.conf 配置文件

如果文件不存在：cp ./gmetad/gmetad.conf /usr/local/src/ganglia_make/etc

```
[root@master ganglia]# vim /usr/local/src/ganglia_make/etc/gmetad.conf
data_source "my grid" 192.168.87.128
xml_port 8651
interactive_port 8652
rrd_rootdir "/var/lib/ganglia/rrds"
case_sensitive_hostnames 0
```

```
[root@master ganglia]# chkconfig --add gmetad
```

```
[root@master ganglia]# mkdir -p /usr/local/src/ganglia_make/var/run/
```

```
[root@master ganglia]# cd /usr/local/src/ganglia_make/var/run/
```

新建 gmetad.pid 文件

```
[root@master run]# vim gmetad.pid
```

```
[root@master run]# service gmetad restart
```

可以通过日志 `tail -f /var/log/messages` 查看启动情况

步骤七：配置 gmond

```
[root@master run]# cd /usr/local/src/ganglia
[root@master ganglia]# cp ./gmond/gmond.init /etc/init.d/gmond
[root@master ganglia]# ./gmond/gmond -t >
/usr/local/src/ganglia_make/etc/gmond.conf
```

修改 gmond 配置

```
[root@master ganglia]# vim /etc/init.d/gmond
GMOND=/usr/local/src/ganglia_make/etc/gmond.conf
```

修改 gmond.conf 配置

```
[root@master ganglia]# vim /usr/local/src/ganglia_make/etc/gmond.conf
cluster {
    name = "my grid" #要与 gmated.conf 中 data_source 的名称相同
    owner = "nobody"
    latlong = "unspecified"
    url = "unspecified"
}

##配置网络（多播，单播）
udp_send_channel {
    #bind_hostname = yes # Highly recommended, soon to be default.
    # This option tells gmond to use a source address
    # that resolves to the machine's hostname. Without
    # this, the metrics may appear to come from any
    # interface and the DNS names associated with
    # those IPs will be used to create the RRDs.

    mcast_join = master
    port = 8649
    ttl = 1
}

udp_rcv_channel {
    #mcast_join = 239.2.11.71
    port = 8649
    #bind = 239.2.11.71
    retry_bind = true
    # Size of the UDP buffer. If you are handling lots of metrics you really
    # should bump it up to e.g. 10MB or even higher.
    # buffer = 10485760
}

tcp_accept_channel {
    port = 8649
    # If you want to gzip XML output
```

```
gzip_output = no
}
```

重启 gmond

```
[root@master ganglia]# service gmond restart
```

步骤八：安装 Ganglia Web

```
[root@master ganglia]# cd /usr/local/src
```

```
[root@master src]# tar -zxvf /opt/software/ganglia-web-3.7.2.tar.gz
```

```
[root@master src]# cd ganglia-web-3.7.2
```

```
[root@master ganglia-web-3.7.2]# vim Makefile
```

```
GDESTDIR = /var/www/ganglia
```

```
APACHE_USER = apache # 与 /etc/php-fpm.d/www.conf 中 user 保持一致
```

```
[root@master ganglia-web-3.7.2]# make install
```

步骤九：配置 nginx 访问 ganglia

Nginx 新增 ganglia 文件目录访问配置

```
[root@master ganglia-web-3.7.2]# vim /etc/nginx/nginx.conf
```

```
location /ganglia {
```

```
    root    /var/www;
```

```
    index   index.html index.htm index.php;
```

```
}
```

```
[root@master ganglia-web-3.7.2]# cd /var/www
```

```
[root@master www]# chown -R apache:apache ganglia/
```

```
[root@master www]# mkdir /var/www/ganglia/dwoo/compiled
```

```
[root@master www]# mkdir /var/www/ganglia/dwoo/cache
```

```
[root@master www]# chmod 777 /var/www/ganglia/dwoo/compiled
```

```
[root@master www]# chmod 777 /var/www/ganglia/dwoo/cache
```

步骤十：配置 Ganglia Web

```
[root@master www]# cd /var/www/ganglia
```

```
[root@master ganglia]# cp conf_default.php conf.php
```

```
[root@master ganglia]# vim conf.php
```

conf. php 中有些默认配置和以上设置不一样的需要进行修改：

```
=====
```

```
$conf['gweb_root'] = "/var/www/ganglia";
```

```
$conf['gweb_confdir'] = "/var/www/ganglia";
```

```
include_once $conf['gweb_root'] . "/version.php";
```

```
#
```

```
# 'readonly': No authentication is required. All users may view all resources. No edits
are allowed.
```

```
# 'enabled': Guest users may view public clusters. Login is required to make changes.
```

```
# An administrator must configure an authentication scheme and ACL
rules.
```

```
# 'disabled': Guest users may perform any actions, including edits. No authentication is
required.
```

```

$conf['auth_system'] = 'readonly';

#
# The name of the directory in "./templates" which contains the
# templates that you want to use. Templates are like a skin for the
# site that can alter its look and feel.
#
$conf['template_name'] = "default";

#
# If you installed gmetad in a directory other than the default
# make sure you change it here.
#

# Where gmetad stores the rrd archives.
$conf['gmetad_root'] = "/var/lib/ganglia";
$conf['rrds'] = "${conf['gmetad_root']}/rrds";

# Where Dwoo (PHP templating engine) store compiled templates
$conf['dwoo_compiled_dir'] = "${conf['gweb_confdir']}/dwoo/compiled"; ##如果不存
在可以手动创建并注意权限
$conf['dwoo_cache_dir'] = "${conf['gweb_confdir']}/dwoo/cache";

# Where to store web-based configuration
$conf['views_dir'] = $conf['gweb_confdir'] . '/conf';
$conf['conf_dir'] = $conf['gweb_confdir'] . '/conf';

# Where to find filter configuration files, if not set filtering
# will be disabled
#$conf['filter_dir'] = "${conf['gweb_confdir']}/filters";

# Leave this alone if rrdtool is installed in $conf['gmetad_root'],
# otherwise, change it if it is installed elsewhere (like /usr/bin)
$conf['rrdtool'] = "/bin/rrdtool"; ##通过命令 which rrdtool 查看

```

步骤十一：重启服务并查看结果

```

[root@master ganglia]# cd
[root@master ~]# service gmond start
[root@master ~]# service gmetad start
[root@master ~]# systemctl restart php-fpm
[root@master ~]# systemctl restart nginx

```

访问页面 <http://master/ganglia/>

21.1.3.3. 被监控端安装 gmond

```
[root@slave1 ~]# yum -y install ganglia-gmond
```

Master 复制配置文件进被监控机器

```
[root@master ~]# scp /usr/local/src/ganglia_make/etc/gmond.conf slave1:/etc/ganglia/
```

```
[root@master ~]# scp /usr/local/src/ganglia_make/etc/gmond.conf slave2:/etc/ganglia/
```

```
[root@slave1 ~]# service gmond start
```

```
[root@slave2 ~]# service gmond start
```

至此，ganglia 安装完成

21.1.4. 数据集

本章实验提供一个包含 30 万条记录的网站用户行为数据集。数据集内容如下：

序号	字段	含义
1	user_id	用户 id
2	item_id	商品 id
3	behaviour_type	用户行为类型(包括浏览 1、收藏 2、加购物车 3、购买 4)
4	user_geohash	用户地理位置哈希值
5	item_category	商品分类
6	time	该记录产生时间

21.1.5. 实验任务

本章实验需要完成以下任务：

- 开启 Ganglia 监控 hadoop 集群
- 本地数据上传到分布式文件系统 HDFS
- 用数据仓库 Hive 查询数据
- 两个操作中 Ganglia 监控到的状态

实验流程如图所示：



图 21-1 实验流程

21.2. 开启 Ganglia 监控 hadoop 集群

21.2.1. 修改 ganglia-monitor 的配置文件

每台机器上都进行如下配置

```
[root@master ~]# vim /usr/local/src/ganglia_make/etc/gmond.conf
```

```
[root@slave1 ~]# vim /etc/ganglia/gmond.conf
```

```
[root@slave2 ~]# vim /etc/ganglia/gmond.conf
```

```
cluster {
    name = "hadoop"
    owner = "nobody"
    latlong = "unspecified"
    url = "unspecified"
}

udp_send_channel {
    #the host who gather this cluster's monitoring data and send these data to gmetad node
    host = master
    port = 8649
}

udp_recv_channel {
    port = 8649
}

tcp_accept_channel {
    port = 8649
```

21.2.2. 主节点配置

```
[root@master ~]# vim /usr/local/src/ganglia_make/etc/gmetad.conf
```

#需要在原来的 data_source 前加上#注释掉

```
data_source "hadoop" 3 master:8649 slave1:8649 slave2:8649
```

21.2.3. 修改 Hadoop 的配置文件

```
[root@master src]# vim /usr/local/src/hadoop/etc/hadoop/hadoop-metrics2.properties
```

```
namenode.sink.ganglia.servers=master:8649
resourcemanager.sink.ganglia.servers=master:8649
mrappmaster.sink.ganglia.servers=master:8649
jobhistoryserver.sink.ganglia.servers=master:8649
*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31
*.sink.ganglia.period=10
```

```
*.sink.ganglia.supportsparse=true
*.sink.ganglia.slope=jvm.metrics.gcCount=zero,jvm.metrics.memHeapUsedM=both
*.sink.ganglia.dmax=jvm.metrics.threadsBlocked=70,jvm.metrics.memHeapUsedM=40
```

```
[root@slave1 ~]# vim /usr/local/src/hadoop/etc/hadoop/hadoop-metrics2.properties
[root@slave2 ~]# vim /usr/local/src/hadoop/etc/hadoop/hadoop-metrics2.properties
```

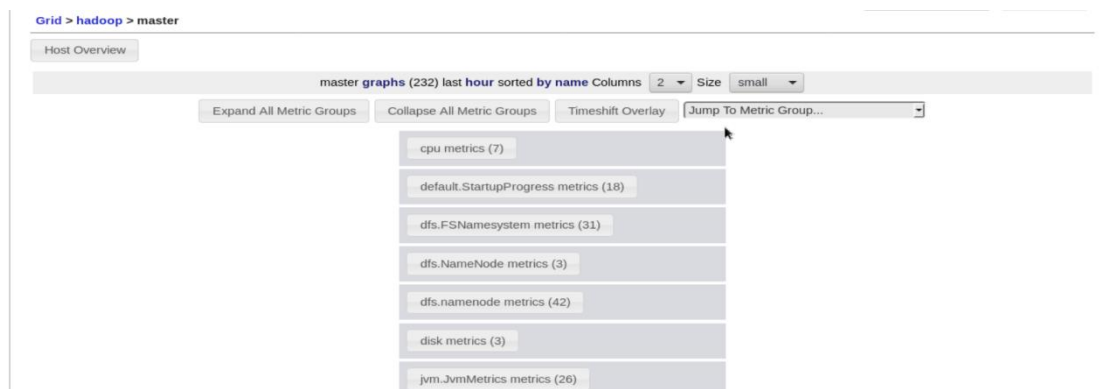
```
datanode.sink.ganglia.servers=master:8649
nodemanager.sink.ganglia.servers=master:8649
*.sink.ganglia.class=org.apache.hadoop.metrics2.sink.ganglia.GangliaSink31
*.sink.ganglia.period=10
*.sink.ganglia.supportsparse=true
*.sink.ganglia.slope=jvm.metrics.gcCount=zero,jvm.metrics.memHeapUsedM=both
*.sink.ganglia.dmax=jvm.metrics.threadsBlocked=70,jvm.metrics.memHeapUsedM=40
```

21.2.4. 重启所有服务

```
[root@slave1 ~]# systemctl stop firewalld
[root@slave2 ~]# systemctl stop firewalld
[root@master ~]# systemctl stop firewalld
[root@slave1 ~]# service gmond restart
[root@slave2 ~]# service gmond restart
[root@master ~]# service gmond restart
[root@master ~]# service gmetad restart
[root@master ~]# service nginx restart
重启 hadoop
[root@master ~]# su - hadoop
[hadoop@master ~]$ cd /usr/local/src/hadoop/sbin/
[hadoop@master sbin]$ ./stop-all.sh
[hadoop@master sbin]$ ./start-all.sh
```

21.2.5. 访问页面查看各机器节点信息

在浏览器中打开页面 <http://master/ganglia/>





21.3. 本地数据上传到分布式文件系统 HDFS

21.3.1. 准备工作

步骤一：创建目录，将数据集放入目录

首先在/opt/software 建立一个用于运行本案例的目录 bigdatacase

```
[root@master ~]# cd /opt/software/
```

```
[root@master software]# mkdir bigdatacase
```

给 hadoop 用户赋予针对 bigdatacase 目录的各种操作权限

```
[root@master software]# cd /usr/local/src
```

```
[root@master src]# chown -R hadoop:hadoop ./bigdatacase
```

```
[root@master src]# cd bigdatacase
```

在 bigdatacase 下创建一个 dataset 目录，用于保存数据集

```
[root@master bigdatacase]# mkdir dataset
```

将下载好的数据集 small_user.csv 放到 dataset 目录下

```
[root@master bigdatacase]# cp /opt/software/small_user.csv
/opt/software/bigdatacase/dataset
```

取出前面 5 条记录看一下

```
[root@master bigdatacase]# cd dataset
```

```
[root@master dataset]# head -5 small_user.csv
```

```
user_id,item_id,behavior_type,user_geohash,item_category,time
10001082,285259775,1,971k14c,4076,2014-12-08 18
10001082,4368907,1,,5503,2014-12-12 12
10001082,4368907,1,,5503,2014-12-12 12
10001082,53616768,1,,9762,2014-12-02 15
```

可以看出, 每行记录都包含 5 个字段, 数据集中的字段及其含义如下:

user_id (用户 id)

item_id(商品 id)

behaviour_type (包括浏览、收藏、加购物车、购买, 对应取值分别是 1、2、3、4)

user_geohash(用户地理位置哈希值, 有些记录中没有这个字段值, 所以后面我们会用脚本做数据预处理时把这个字段全部删除)

item_category (商品分类)

time (该记录产生时间)

删除文件第一行记录, 即字段名称

步骤二: 数据预处理

删除文件第一行记录, 即字段名称

small_user 中的第一行都是字段名称, 我们在文件中的数据导入到数据仓库 Hive 中时, 不需要第一行字段名称, 因此, 这里在做数据预处理时, 删除第一行

```
[root@master dataset]# cd /opt/software/bigdatacase/dataset
```

下面删除 small_user 中的第 1 行

```
[root@master dataset]# sed -i '1d' small_user.csv
```

下面再用 head 命令去查看文件的前 5 行记录, 就看不到字段名称这一行了

```
[root@master dataset]# head -5 small_user.csv
```

```
10001082,285259775,1,971k14c,4076,2014-12-08 18
10001082,4368907,1,,5503,2014-12-12 12
10001082,4368907,1,,5503,2014-12-12 12
10001082,53616768,1,,9762,2014-12-02 15
10001082,151466952,1,,5232,2014-12-12 11
```

对字段进行预处理

下面对数据集进行一些预处理, 包括为每行记录增加一个 id 字段 (让记录具有唯一性)、增加一个省份字段 (用来后续进行可视化分析), 并且丢弃 user_geohash 字段 (后面分析不需要这个字段)。

下面我们要建一个脚本文件 pre_deal.sh, 请把这个脚本文件放在 dataset 目录下, 和数据集 small_user.csv 放在同一个目录下:

```
[root@master dataset]# vim pre_deal.sh
```

在这个脚本文件中加入下面代码:

```
#!/bin/bash
```

#下面设置输入文件, 把用户执行 pre_deal.sh 命令时提供的第一个参数作为输入文件名称

```
infile=$1
```

#下面设置输出文件, 把用户执行 pre_deal.sh 命令时提供的第二个参数作为输出文件名称

```
outfile=$2
```

#注意!! 最后的\$infile > \$outfile 必须跟在}' 这两个字符的后面

```
awk -F "," 'BEGIN{
    srand();
    id=0;
    Province[0]="山东";Province[1]="山西";Province[2]="河南";Province[3]="河北";
    Province[4]="陕西";Province[5]="内蒙古";Province[6]="上海市";
    Province[7]="北京市";Province[8]="重庆市";Province[9]="天津市";Province[10]="福建";
    Province[11]="广东";Province[12]="广西";Province[13]="云南";
    Province[14]="浙江";Province[15]="贵州";Province[16]="新疆";Province[17]="西藏";
    Province[18]="江西";Province[19]="湖南";Province[20]="湖北";
    Province[21]="黑龙江";Province[22]="吉林";Province[23]="辽宁";Province[24]="江苏";
    Province[25]="甘肃";Province[26]="青海";Province[27]="四川";
    Province[28]="安徽";Province[29]="宁夏";Province[30]="海南";Province[31]="香港";
    Province[32]="澳门";Province[33]="台湾";
}
{
    id=id+1;
    value=int(rand()*34);
    print id"\t"$1"\t"$2"\t"$3"\t"$5"\t"substr($6,1,10)"\t"Province[value]
}' $infile > $outfile
```

最后, 保存 pre_deal.sh 代码文件, 退出 vim 编辑器。

下面就可以执行 pre_deal.sh 脚本文件, 来对 small_user.csv 进行数据预处理

```
[root@master dataset]# bash ./pre_deal.sh small_user.csv user_table.txt
```

查看处理后的前 10 行数据(输出结果不唯一)

```
[root@master dataset]# head -10 user_table.txt
```

1	10001082	285259775	1	4076	2014-12-08	河北
2	10001082	4368907	1	5503	2014-12-12	四川
3	10001082	4368907	1	5503	2014-12-12	新疆
4	10001082	53616768	1	9762	2014-12-02	山东
5	10001082	151466952	1	5232	2014-12-12	香港
6	10001082	53616768	4	9762	2014-12-02	江苏
7	10001082	290088061	1	5503	2014-12-12	宁夏
8	10001082	298397524	1	10894	2014-12-12	重庆市
9	10001082	32104252	1	6513	2014-12-12	广西
10	10001082	323339743	1	10894	2014-12-12	云南

21.3.2. 进行上传操作

user_table.txt 中的数据最终导入到数据仓库 Hive 中。为了完成这个操作, 我们会首先把 user_table.txt 上传到分布式文件系统 HDFS 中, 然后, 在 Hive 中创建一个外部表, 完成数据的导入。

启动 Hadoop 集群

```
[root@master dataset]# su - hadoop
```

```
[hadoop@master dataset]$ /usr/local/src/hadoop/sbin/start-dfs.sh
```

```
[hadoop@master dataset]$ /usr/local/src/hadoop/sbin/start-yarn.sh
```

在 HDFS 的根目录下创建一个新的目录 bigdatacase，并在这个目录下创建一个子目录 dataset

```
[hadoop@master dataset]$ cd /usr/local/src/hadoop
```

```
[hadoop@master hadoop]$ ./bin/hdfs dfs -mkdir -p /bigdatacase/dataset
```

把 Linux 本地文件系统中的 user_table.txt 上传到分布式文件系统 HDFS 中，存放在 HDFS 中的 “/bigdatacase/dataset” 目录下。

```
[hadoop@master hadoop]$ ./bin/hdfs dfs -put /opt/software/bigdatacase/dataset/user_table.txt /bigdatacase/dataset
```

下面可以查看一下 HDFS 中的 user_table.txt 的前 10 条记录

```
[hadoop@master hadoop]$ ./bin/hdfs dfs -cat /bigdatacase/dataset/user_table.txt | head -10
```

```
1  10001082    285259775    1   4076    2014-12-08  河北
2  10001082    4368907    1   5503    2014-12-12  四川
3  10001082    4368907    1   5503    2014-12-12  新疆
4  10001082    53616768    1   9762    2014-12-02  山东
5  10001082    151466952    1   5232    2014-12-12  香港
6  10001082    53616768    4   9762    2014-12-02  江苏
7  10001082    290088061    1   5503    2014-12-12  宁夏
8  10001082    298397524    1  10894    2014-12-12  重庆市
9  10001082    32104252    1   6513    2014-12-12  广西
10 10001082    323339743    1  10894    2014-12-12  云南
```

21.4. 用数据仓库 Hive 查询数据

21.4.1. 准备工作

在 Hive 上创建数据库

启动 MySQL 数据库

```
[hadoop@master hadoop]$ service mysql start
```

```
[hadoop@master hadoop]$ cd /usr/local/src/hive
```

```
[hadoop@master hive]$ ./bin/hive #启动 Hive
```

启动成功以后，就进入了 “hive>” 命令提示符状态，可以输入类似 SQL 语句的 HiveQL 语句。

下面，我们要在 Hive 中创建一个数据库 dblab，命令如下：

```
hive> create database dblab;
```

```
OK
```

```
Time taken: 1.471 seconds
```

```
hive> use dblab;
```

```
OK
```

```
Time taken: 0.119 seconds
```

21.4.2. 进行查询操作

在数据库 dlab 中创建一个外部表 bigdata_user，它包含字段 (id, uid, item_id, behavior_type, item_category, date, province)，在 hive 命令提示符下输入如下命令：

```
hive> CREATE EXTERNAL TABLE dlab.bigdata_user(id INT,uid STRING,item_id
STRING,behavior_type INT,item_category STRING,visit_date DATE,province STRING) COMMENT
'Welcome to xmu dlab!' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS
TEXTFILE LOCATION '/bigdatacase/dataset';
```

OK

Time taken: 5.606 seconds

上面已经成功把 HDFS 中的“/bigdatacase/dataset”目录下的数据加载到了数据仓库 Hive 中，我们现在可以使用下面命令查询：

```
hive> select * from bigdata_user limit 10;
```

OK

1	10001082	285259775	1	4076	2014-12-08	河北
2	10001082	4368907	1	5503	2014-12-12	四川
3	10001082	4368907	1	5503	2014-12-12	新疆
4	10001082	53616768	1	9762	2014-12-02	山东
5	10001082	151466952	1	5232	2014-12-12	香港
6	10001082	53616768	4	9762	2014-12-02	江苏
7	10001082	290088061	1	5503	2014-12-12	宁夏
8	10001082	298397524	1	10894	2014-12-12	重庆市
9	10001082	32104252	1	6513	2014-12-12	广西
10	10001082	323339743	1	10894	2014-12-12	云南

Time taken: 8.347 seconds, Fetched: 10 row(s)

```
hive> select behavior_type from bigdata_user limit 10;
```

OK

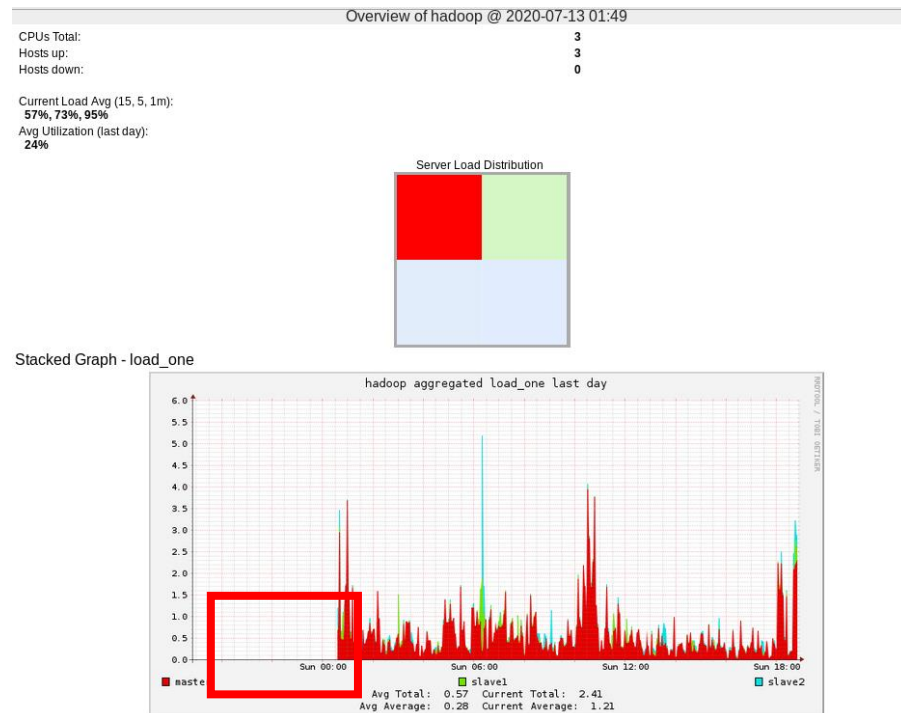
1
1
1
1
1
1
4
1
1
1
1

Time taken: 0.486 seconds, Fetched: 10 row(s)

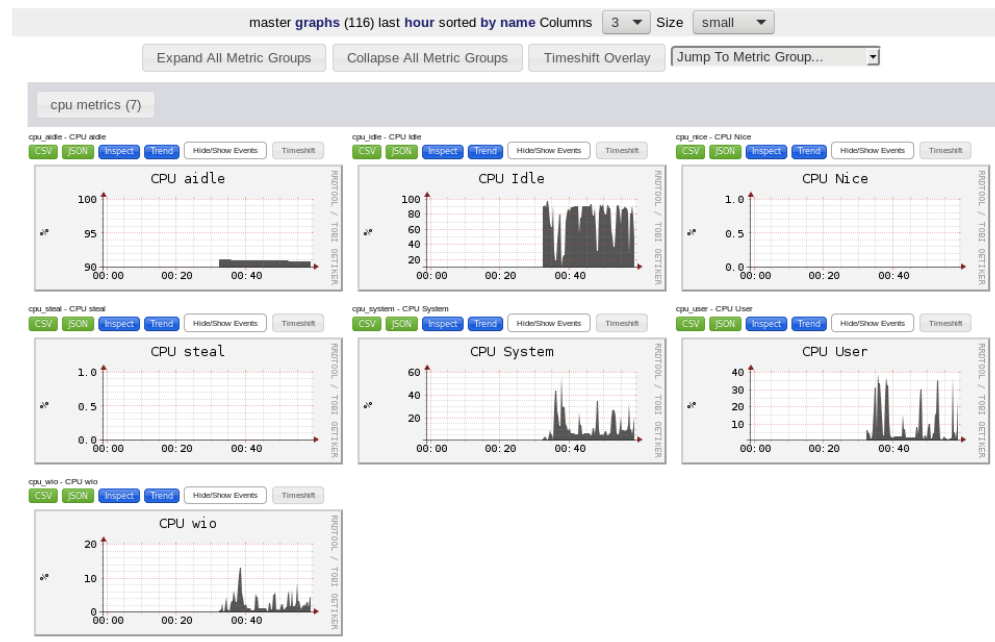
21.5. 上传和查询操作中 Ganglia 监控到的状态

21.5.1. 上传数据前后集群状态变化

进行上传操作前 ganglia 监控到的 hadoop 的整体状态



进行上传操作前 ganglia 监控到的 hadoop 中 master 节点的状态

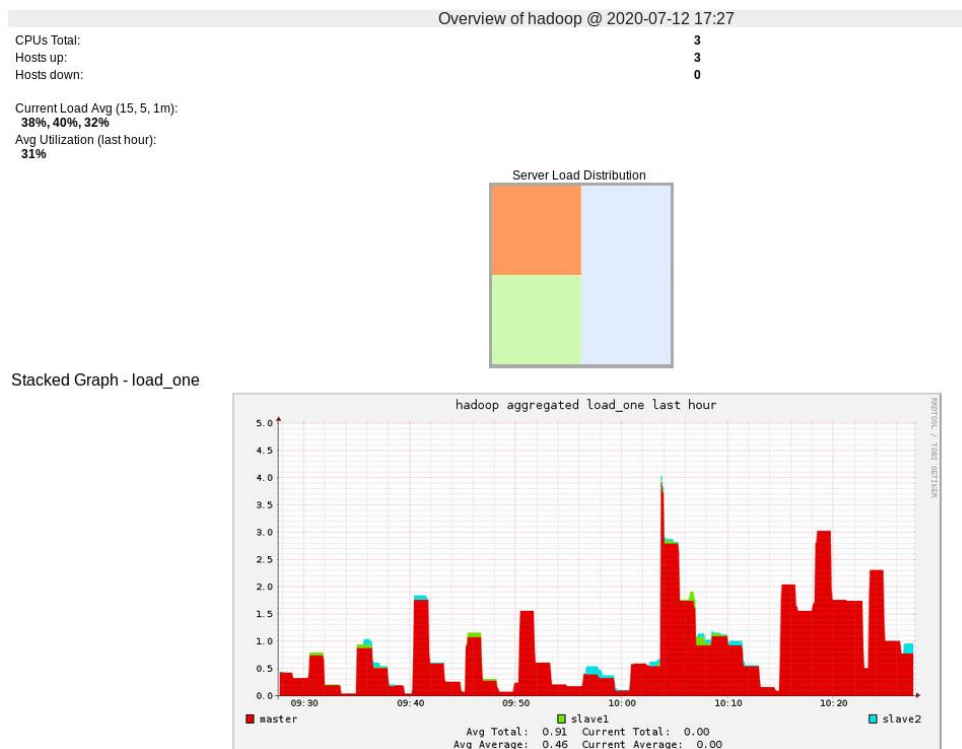


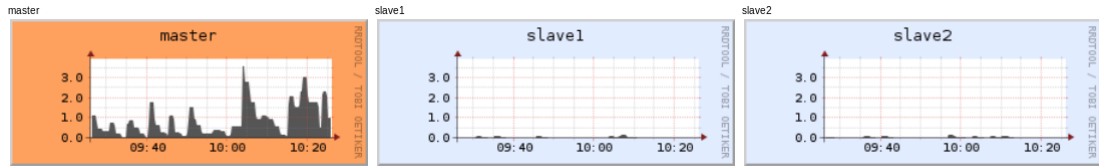
进行上传操作后 ganglia 监控到的 hadoop 中 master 节点的状态



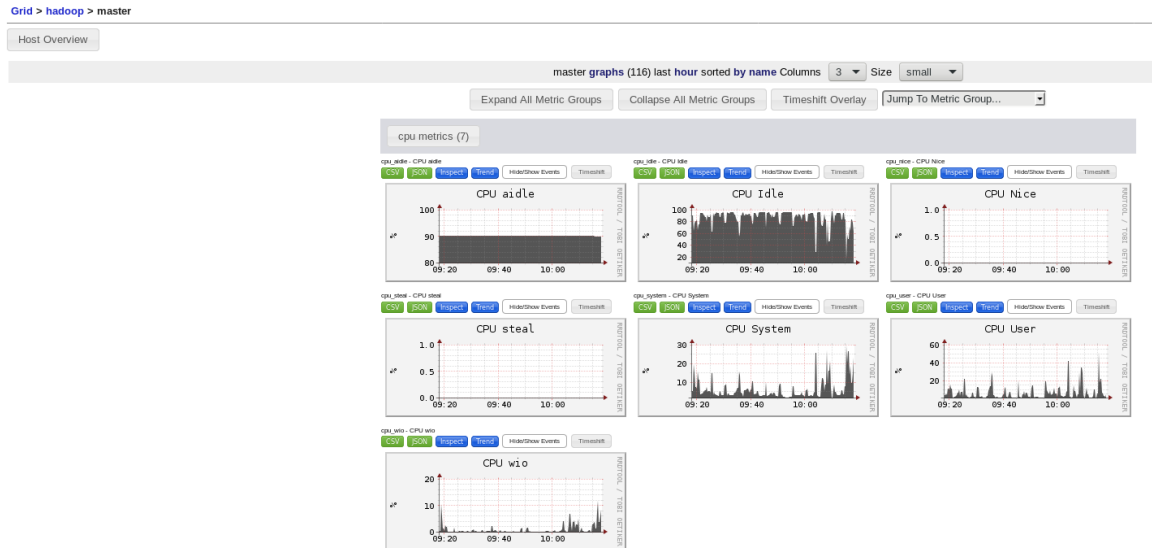
21.5.2. 查询数据前后集群状态变化

进行查询操作前 ganglia 监控到的 hadoop 的整体状态





进行查询操作前 ganglia 监控到的 hadoop 中 master 节点的状态



进行查询操作前 ganglia 监控到的 hadoop 中 slave1 节点的状态

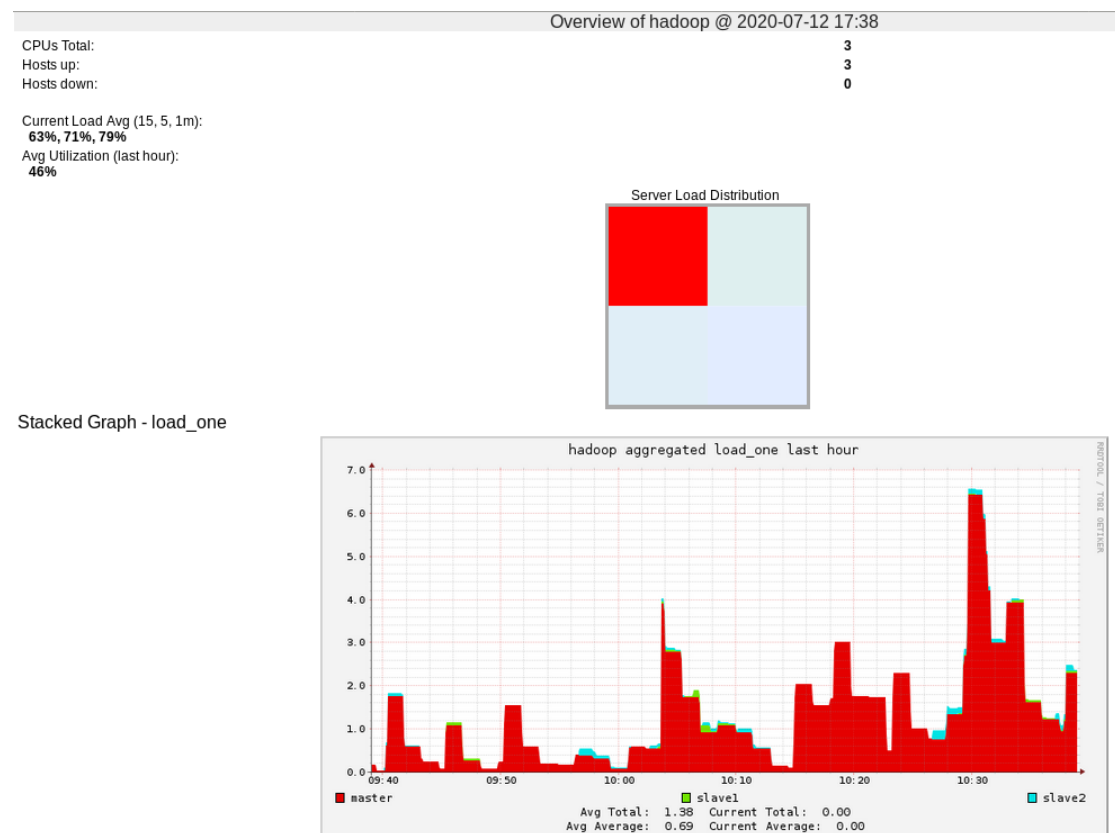


进行查询操作前 ganglia 监控到的 hadoop 中 slave2 节点的状态

Grid > hadoop > slave2



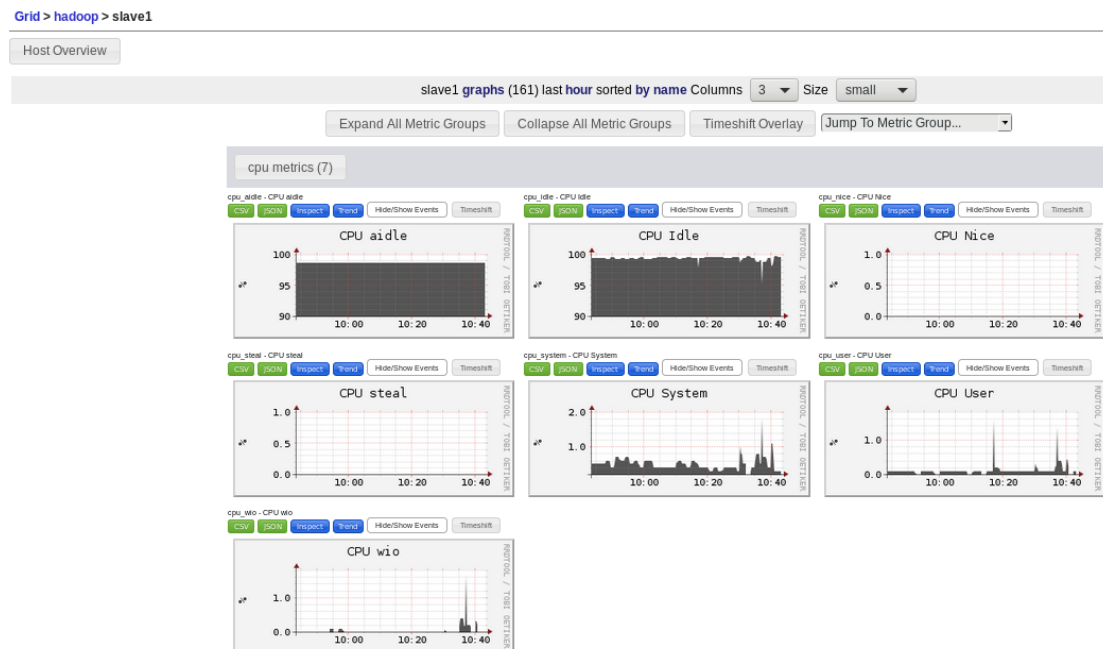
进行查询操作后 ganglia 监控到的 hadoop 的整体状态



进行查询操作后 ganglia 监控到的 hadoop 中 master 节点的状态



进行查询操作后 ganglia 监控到的 hadoop 中 slave1 节点的状态



进行查询操作后 ganglia 监控到的 hadoop 中 slave2 节点的状态

