

讓HPC管理效率與安全

國立臺灣師範大學物理學系 陳俊明

chunming@ntnu.edu.tw

幫助管理工具

- Ganglia Monitoring System
- pdsh
- Environment Modules
- Shell Scripts

Ganglia Monitoring System

Ganglia是一套OpenSource的Cluster監控系統，可幫助管理人員迅速且簡單的瞭解Cluster中各個節點的狀態，並可查詢記錄

Monitoring clusters and Grids since the year 2000

Ganglia Monitoring System

[Home](#)[Demos](#)[Download](#)[Community](#)[Support](#)[Contributors](#)

What is Ganglia?

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization. It uses carefully engineered data structures and algorithms to achieve very low per-node overheads and high concurrency. The implementation is robust, has been ported to an extensive set of operating systems and processor architectures, and is currently in use on thousands of clusters around the world. It has been used to link clusters across university campuses and around the world and can scale to handle clusters with 2000 nodes.

Ganglia is a [BSD-licensed](#) open-source project that grew out of the [University of California, Berkeley Millennium Project](#) which was initially funded in large part by the [National Partnership for Advanced Computational Infrastructure](#) (NPACI) and [National Science Foundation](#) RI Award EIA-9802069. NPACI is funded by the [National Science Foundation](#) and strives to advance science by creating a ubiquitous, continuous, and pervasive national computational infrastructure: the Grid. Current support comes from [Planet Lab](#): an open platform for developing, deploying, and accessing planetary-scale services.

No Comments

We're back!

Posted by [Matt Massie](#) in [Uncategorized](#) on March 7, 2018

SourceForge updated their infrastructure and we missed the email explaining how to prevent downtime. Everything should be back to normal now. Sorry for the inconvenience.

No Comments

Ganglia Web 3.7.2 released

Posted by [vuksan](#) in [Uncategorized](#) on June 14, 2016

Ganglia Web 3.7.2 has been released. Changes in this release are

- Fix for a reflected XSS issue in the metrics API
- Other minor improvements and fixes

Thanks to Lior Adar from Palantir Security for reporting the XSS issue.

Download the release from

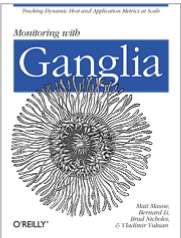
<https://sourceforge.net/projects/ganglia/files/ganglia-web/3.7.2/>

No Comments

Ganglia

SEARCH BLOG, MAILING LISTS, WIKI

ENHANCED BY Google



Buy the book today!

■ ANNOUNCEMENTS (33)

■ FUN (3)

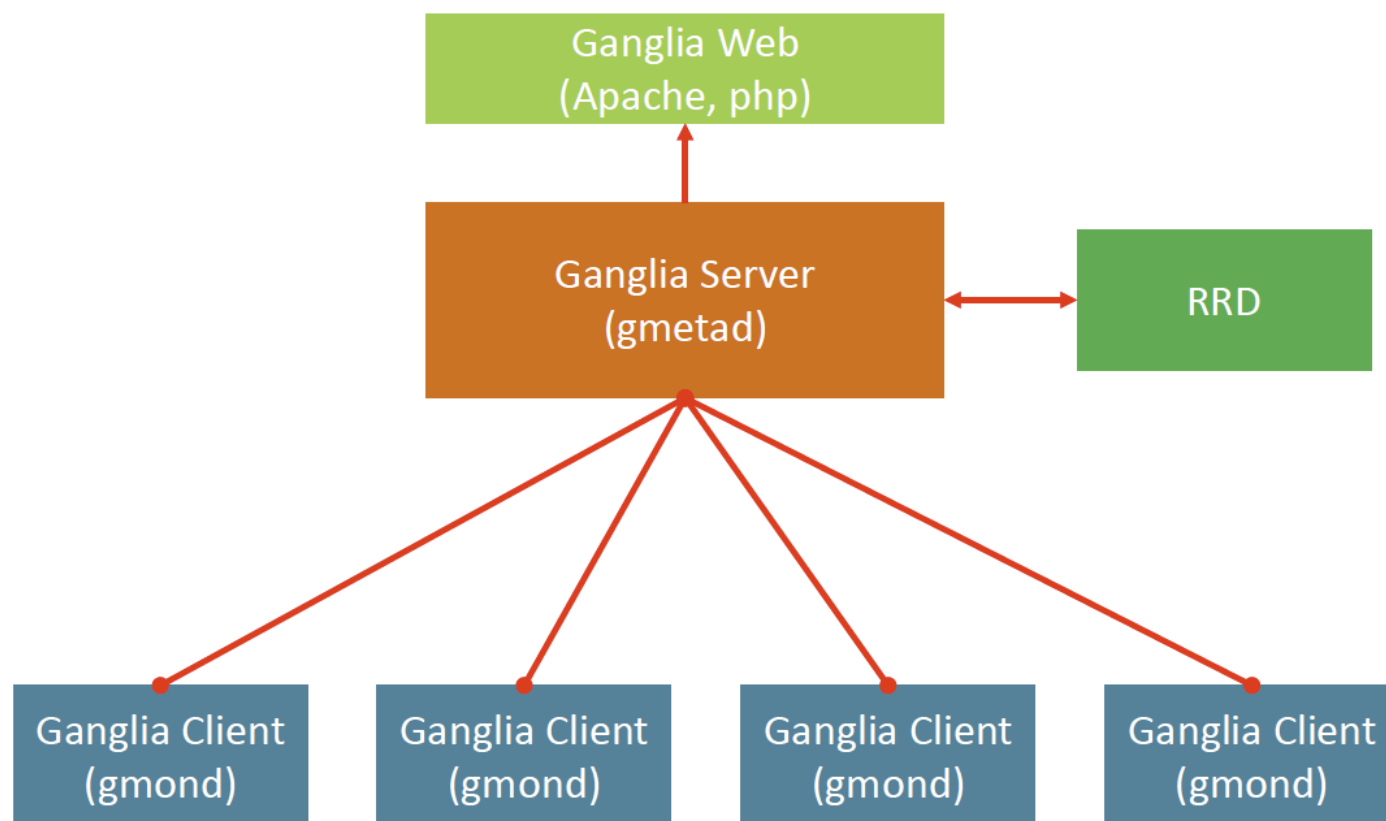
■ FYI (10)

■ RELEASES (45)

■ UNCATEGORIZED (25)

WHO USES GANGLIA?

Ganglia的架構



RRDtool (Round Robin Database Tool) 是開源工具，是用時間序列資料為業界標準、高效能資料紀錄的繪圖系統

Ganglia 伺服器端安裝與設定

- 安裝 epel 基本套件後，再安裝 ganglia 套件

```
[root@master ~]# dnf install -y epel-release  
[root@master ~]# dnf install -y ganglia-gmetad ganglia-gmond ganglia-web
```

- 設定伺服器端 gmetad 設定檔

```
[root@master ~]# vi /etc/ganglia/gmetad.conf  
gridname "HPC"  
data_source "HPC cluster" master cn1 cn2
```

- 設定伺服器端網頁設定檔

```
[root@master ~]# vi /etc/httpd/conf.d/ganglia.conf  
<Location /ganglia>  
    Require local  
    Require ip 192.168.XXX.0/24  
</Location>
```

Ganglia 伺服器端安裝與設定

- 設定伺服器端 gmond 設定檔

```
[root@master ~]# vi /etc/ganglia/gmond.conf
cluster {
  name = "HPC cluster"           #對應到gmetad的data source
}
udp_send_channel {
  #mcast_join = 239.2.11.71      #註解掉
  host = 192.168.1.254          #新增一行，並對應到Server的IP或hostname
  port = 8649
  ttl = 1
}
udp_recv_channel {
  #mcast_join = 239.2.11.71      #註解掉
  port = 8649
  #bind = 239.2.11.71            #註解掉
  retry_bind = true
}
```

Ganglia 伺服器端安裝與設定

- 啟動 Ganglia 及網頁伺服器，順便設定開機啟動

```
[root@master ~]# systemctl enable --now httpd  
[root@master ~]# systemctl enable --now gmond  
[root@master ~]# systemctl enable --now gmetad
```

- 用瀏覽器輸入網址
<http://192.168.xxx.xxx/ganglia/>

Ganglia Client安裝與設定

- 安裝 epel 基本套件後，再安裝 ganglia 套件

```
[root@cn1 ~]# dnf install -y epel-release  
[root@cn1 ~]# dnf install -y ganglia-gmond
```

- 複製伺服器端 gmond 設定檔到運算節點

```
[root@cn1 ~]# scp master:/etc/ganglia/gmond.conf /etc/ganglia/
```

- 啟動 Ganglia 服務，順便設定開機啟動

```
[root@cn1 ~]# systemctl enable --now gmond
```

- 重新啟動 master Ganglia 伺服器

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

Parallel Distributed Shell - pdsh

- pdsh (parallel distributed shell)可以以平行的方式對指定的節點送出指令操作
- 下載pdsh source code

```
[root@master ~]# git clone https://github.com/chaos/pdsh.git
```

- 編譯並安裝

```
[root@master ~]# cd pdsh  
[root@master ~]# ./bootstrap  
[root@master ~]# ./configure --with-ssh  
[root@master ~]# make && make install
```

- 透過 pdsh 在計算節點執行校時指令

```
[root@master ~]# pdsh -w ssh:cn[1-2] "chronyc makestep"
```

Environment Modules

- 藉由設定好環境模組檔案，以快速的載入或清除系統環境

<https://modules.sourceforge.net/>

- 下載 Modules source code

```
[root@master ~]# wget https://github.com/envmodules/modules/archive/refs/tags/v5.6.0.tar.gz
```

- 解壓縮、編譯及安裝

```
[root@master ~]# tar zxvf modules-5.6.0.tar.gz  
[root@master ~]# cd modules-5.6.0  
[root@master ~]# dnf install -y tcl-devel  
[root@master ~]# ./configure --prefix=/opt/Modules-5.6.0  
[root@master ~]# make && make install
```

需事先安裝 tcl-devel

Environment Modules

- 環境模組檔案 (位於Modules安裝目錄下的modulefiles)

```
[root@master ~]# cd /opt/modules-5.6.0/modulefiles
[root@master ~]# vi openmpi-5.0.8-GNU
#%Module*- tcl *-
module-whatis "OPENMPI-5.0.8 with GNU Compiler"
setenv      CC icc
setenv      CXX icpc
setenv      FC ifort

prepend-path PATH /opt/openmpi/openmpi-5.0.8-GNU/bin
prepend-path LD_LIBRARY_PATH /opt/openmpi/openmpi-5.0.8-GNU/lib
```

Environment Modules

- 載入Environment Modules

```
[root@master ~]# source /opt/Modules-5.6.0/init/bash
```

- Environment Modules 常用指令

```
module COMMAND
```

COMMAND	說明	範例
purge	清除上一次載入的環境	module purge
load <modulefile>	載入modulefile已設定環境	module load openmpi-4.1.4-intel
avail	查詢可用的modulefile	module avail
whatis <modulefile>	顯示modulefile說明 (modulefile中的module-whatis)	module whatis openmpi-4.1.4-intel

管理上常用的指令

- echo
- date
- find
- xargs
- Filter commands
 - cat, tac, grep, cut, head, tail, nl, join, split, sort, tr, uniq, wc, sed, awk
- test
- kill
- ps

其他管理指令及工具

- lscpu, lspci, lsscsi
- blkid
- parted, fdisk
- mkfs
- fsck
- htop
- iftop
- wireshark

Shell Scripts

電腦程式使用的文字檔案，內容由一連串的 `shell` 命令組成，經由 `Unix Shell` 直譯其內容後運作。被當成是一種手稿語言來設計，其運作方式與直譯語言相當，由 `Unix shell` 扮演命令行直譯器的角色，在讀取 `shell` 指令碼之後，依序執行其中的 `shell` 命令，之後輸出結果。利用 `shell` 指令碼可以進行系統管理，檔案操作等。

六個字「執行程式列表」

Shell Script 範例

- 檢查使用者家目錄使用多少空間

```
[root@master ~]# vi chkhome.sh
#!/bin/bash
## display home size
echo "check home size"
date
du -chs /home/*
echo "done"
```

- 執行 script

```
[root@master ~]# bash chkhome.sh
or
[root@master ~]# chmod +x chkhome.sh
[root@master ~]# ./chkhome.sh
```

Shell Script 迴圈範例

```
for i in $(ls)
do
    echo "item: $i"
done
```

```
for i in `seq 1 10`
do
    echo "$i"
done
```

```
for i in `seq 1 10`
do
    echo ${i}time
done
```

怎麼寫好Shell Scripts

- https://linux.vbird.org/linux_basic/centos7/0340bashshell-scripts.php
- 不停的寫寫寫
- 跟別人討論
- Google search (找需要的指令)

Crontab 讓Shell Scripts定時執行

```
# vi /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
```

HPC安全性管理 - 防火牆設定

- 啟用對外連線node防火牆

```
[root@master ~]# systemctl start firewalld  
[root@master ~]# systemctl enable firewalld
```

- Firewalld 防火牆相關指令

- | | |
|------------------|---------------------------------|
| ● 查詢版本 | firewall-cmd --version |
| ● 檢視防火牆規則 | firewall-cmd --list-all |
| ● 查詢那協正在運作的 zone | firewall-cmd --get-active-zones |
| ● 重新讀取設定 | firewall-cmd --reload |
| ● 永久設置參數 | --permanent |

HPC安全性管理 - 防火牆設定

Zone – 區域

public	顧名思義就是公眾區域，好比家裡的大門是通往外面的世界，只會允許公開的服務接口通過，如：22
external	公開區域，適用於 NAT 網路環境
dmz	非軍事區域 (demilitarized zone，有點像是放在外頭的危險區域)，允許外部的連線進入，但其對內的連線則有限制，只有被允許的連線才能連進內部網路。
work	公司內部等工作區域，在此區域中不應該會有惡意的攻擊者。只有被允許的連線可以進入。
home	家中的網路區域，在此區域中不應該會有惡意的攻擊者。只有被允許的連線可以進入。
internal	內部網路區域，在此區域中不應該會有惡意的攻擊者。只有被允許的連線可以進入。
trusted	完全信任的區域，可以允許所有的連線連接主機裡面的服務。 # firewall-cmd --zone=trusted --permanent --add-source=192.168.1.0/24
drop	任何往內的封包都會被丟棄，只允許往外傳送的封包。
block	任何來自於外部的連線都會被阻擋，只允許自己系統主動建立的連線。

HPC安全性管理 - 防火牆設定

- 查詢網路卡預設區域

```
[root@master ~]# firewall-cmd --get-default-zone
```

- 更改網路卡預設區域

```
[root@master ~]# firewall-cmd --set-default-zone=work
```

- 查詢目前運作中的區域及網路卡

```
[root@master ~]# firewall-cmd --get-active-zones
```

- 查詢網路卡介面所屬的區域

```
[root@master ~]# firewall-cmd --get-zone-of-interface=ens160
```

- 改變網路卡介面所屬的區域

```
[root@master ~]# firewall-cmd --zone=trusted --change-interface=ens224
```

HPC安全性管理 - 防火牆設定

新增服務	<code>firewall-cmd --permanent -zone=public --add-service=service</code>
移除服務	<code>firewall-cmd --permanent --zone=public --remove-service=service</code>
新增port	<code>firewall-cmd --permanent --zone=public --add-port=port/tcp</code>
移除port	<code>firewall-cmd --permanent --zone=public --remove-port=port/tcp</code>
允許特定網段、IP、port 通過(rich-rule)	<code>firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.1" port protocol="tcp" port="33899" accept'</code>
	<code>firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="192.168.56.0/24" port protocol="tcp" port="33899" accept'</code>
移除特定網段、IP、通過 (rich-rule)	<code>firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source address="192.168.56.1" port protocol="tcp" port="33899" accept'</code>
	<code>firewall-cmd --permanent --remove-rich-rule='rule family="ipv4" source address="192.168.56.0/24" port protocol="tcp" port="33899" accept'</code>

HPC安全性管理 - SSH設定

- 修改SSH設定檔

```
[root@master ~]# vi /etc/ssh/sshd_config
```

1. 禁止外網root登入

PermitRootLogin **no**

Match Address 192.168.1.0/24

PermitRootLogin yes

2. 改變登入port

Port **7749**

- 重啟SSH服務

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

HPC安全性管理 - 密碼設定原則

- 密碼設定原則
 - 每6個月要變更一次密碼
 - 密碼長度要8碼以上
 - 加強密碼複雜度，密碼應包含英文大寫、小寫、特殊符號或數字三種
- 設定密碼有效期（新帳號預設）

```
[root@master ~]# vi /etc/login.defs
```

```
PASS_MAX_DAYS 180 # 最長使用天數 = 180 天 ( 6 個月 )  
PASS_MIN_DAYS 1 # 兩次變更至少間隔 1 天 ( 可按需調整 )  
PASS_WARN_AGE 14 # 到期前 14 天提醒
```

- 把期限政策套用到「現有帳號」

```
[user1@master ~]$ chage --maxdays 180 --mindays 1 --warndays 14 <username>
```

HPC安全性管理 - 密碼設定原則

- 設定「長度 ≥ 8 且四類中至少三類」

```
[root@master ~]# vi /etc/security/pwquality.conf

minlen = 8      # 最小密碼長度
minclass = 3    # 大/小/數/特殊 四類中至少三類
ucredit = 0     # -1 密碼中至少包含1個大寫字母
lcredit = 0     # -1 密碼中至少包含1個小寫字母
dcredit = 0     # -1 密碼中至少包含1個數字
ocredit = 0     # -1 密碼中至少包含1個特殊字符
retry = 3       # 改密碼時最多重試3次
difok = 3       # 新密碼與舊密碼至少有3個字符不同
# enforce_for_root # (可選) 對 root 也強制
```

- 禁止重複使用最近 N 次舊密碼

```
[root@master ~]# vi /etc/security/pwhistory.conf

remember = 5
file = /etc/security/opasswd
```

HPC安全性管理 - SSH 啟用 OTP 驗證功能

- 安裝必要套件

```
[root@master ~]# dnf install epel-release  
[root@master ~]# dnf install google-authenticator qrencode
```

- PAM 設定

```
[root@master ~]# cp /etc/pam.d/sshd /etc/pam.d/sshd.bak  
[root@master ~]# vi /etc/pam.d/sshd  
第3插入 auth      required  pam_google_authenticator.so nullok
```

HPC安全性管理 - SSH 啟用 OTP 驗證功能

- SSH 設定

```
[root@master ~]# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak  
[root@master ~]# vi /etc/ssh/sshd_config  
第75行 ChallengeResponseAuthentication yes
```

- 重新啟動SSH

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

- 測試

```
[chunming@master ~]$ google-authenticator
```

回家練習

- 將閒置超過 30 分鐘的使用者踢掉
- 每 30 分鐘檢查一次，若機器負載小於 0.5 時清除 /tmp 下的資料
- 取得機器上所有 account、uid、group、gid、home及shell，並且格式化排列

account	uid	group	gid	home	shell
pbsadmin	1002	pbsadmin	1002	/home/pbsadmin	/bin/bash
sam	1000	sam,test	1000,1005	/home/sam	/bin/bash
user1	1001	user1	1001	/home/user1	/bin/bash
user2	1003	user2	1003	/home/user2	/bin/bash
user3	1004	user3	1004	/home/user3	/bin/bash