

# RRDtool 簡介

- MRTG & RRDtool 的比較
- RRDtool 的基本架構
- RRDtool 學習過程中遇到的困難
- RRDtool 的簡單寫法
- RRDtool 的實際應用

# MRTG & RRDtool 的比較

- MRTG 的優點
  - 簡單 好用 網路上資料多 (中文)
  - 最普及的流量統計圖

# MRTG & RRDtool 的比較

- MRTG 的最大缺點 處理效率問題
  - 監控太多台設備機器會跑不動
  - 無法繪出特定時間需求
  - 數值過大的問題

# MRTG & RRDtool 的比較

- 比MRTG強勢的地方在於
  - 可閱讀舊有的資料 並選擇時段畫圖
  - 若是對全校所有的PORT進行監測 可能有上千個點 那五分鐘 MRTG SERVER就必須畫上 上千張圖 持續執行對於 SERVER來說 造成負擔 如同第一點 RRDtool第一層只做提取資料 如需閱讀圖形才進行選擇畫圖
  - 對於繪圖無自由度可言
  - 可解決數值過大的問題

# RRDtool的基本架構

- rrdtool create (第一層)
  - 紀錄檔 設定格式
- rrdtool update (第二層)
  - 可利用撰寫 bash + snmp 執行
- rrdtool graph (第三層)
  - 各式圖形上的表示法

# 遇到困難 & 解決方式

*2005 / 04 / 25*

完成一個 rrdtool 程式基本上有三個階段

1. 製作 \*.rrd 檔 ---> 類似 MRTG 的 LOG 檔
2. 製作 提取資料更新檔案程式 ---> 用來定時提取資料來更新 \*.rrd 檔案
3. 製作 繪圖程式 ---> 讀取 \*.rrd 檔案進行繪圖

目前執行的進度

完成校內大部分 SWITCH UPLINK PORT 的流量圖

但是只有如此的話 ....

事實上 .. 只有 .....

" 停留 " 在 mrtg 基本功能 .....

目前可以自動執行的結果 和 MRTG 相似度 幾乎 百分之八十

而距離終極目標 " 隨選 "

可自動執行的部分到完成一個 rrdtool 程式的第二階段 而對於第三階段的畫圖部分

雖然可以 用手寫程式的方式完成第三階段

但是 總不能要看一張 寫一個程式

預計用 在網頁上用選擇的方式 用選單指定 我要哪一個時段 哪些 PORT 然後自動畫圖

因為牽涉到動態網頁的部分 我不太會 所以想辦法解決中

# 遇到困難 & 解決方式

2005 / 05 / 12

寫了一個 \*.sh 檔案在執行的時候 ( \*.sh包括 snmp , rrdtool update , rrdtool graph )  
發現 week 以上 圖中的文字會變成亂碼

-----  
我發現一個很詭異的狀況

如果我用crontab -e 設定我要五分鐘執行一次 \*.sh 檔案 ( \*.sh 檔很多個 )

這樣 在rrdtool圖上呈現的 week的圖 就是 " 正常英文 "

但是為了分散CPU在同一個時間點過於忙碌

**並且必免 如果 SWITCH掛掉 透過SNMP抓資料時必須等到TIMEOUT之後才能繼續下去**

-----  
為了解決以上的問題

我寫了第一組JAVA程式 讓每一個 rrdtool程式 五分鐘自動執行一次

之後 在寫一個程式 讓 rrdtool " 個別 " 的執行

而且 " **起始點都不同** "

主要原因是 分散在同一時間內 CPU 的 LOADING 不至於太高

也可避免在單一個 \*.sh 程式 透過snmp提取資料時 時間過久

導致 對於整體資料的正確性造成影響

-----  
可是這樣的結果在 week以上的圖 在文字部分就會產生亂碼

-----  
這是不解的地方

照樣看來 反推 不是java的問題

# 遇到困難 & 解決方式

*2005 / 05 / 16*

今天測試 修改 i18n的內容  
看是不是顯示語系的問題 修改程 zh\_US 結果出乎意料的 Orz... 並未達到目的

原本i18n的內容為

```
LANG="zh_TW.Big5"  
SUPPORTED="zh_TW.UTF-8:zh_TW:zh"  
SYSFONT="latacyrheb-sun16"
```

修改成

```
LANG=en_US.iso88591  
SUPPORTED="en_US.iso88591:en_US:en"
```

結果一樣  
之後在修改環境變數測試 LANG=en\_US.iso88591  
發現一樣沒有改善  
當兩個設定一起改的時候 照樣沒有改善 oox



# 遇到困難 & 解決方式

增加基本知識的網頁 <http://xcin.linux.org.tw/i18n/pc2000/p2/node4.html>  
主要是locale 的指令

LC\_CTYPE: 字元分類及處理方式。  
LC\_COLLATE: 字元順序與字串比較。  
LC\_MESSAGES: 程式中用何種語言來顯示訊息。  
LC\_MONETARY: 貨幣顯式格式。  
LC\_NUMERIC: 數字顯式格式。  
LC\_TIME: 日期與時間的顯式格式。  
LC\_ALL: 此類別可以一次設定以上所有的類別。

指令模式 舉例 修改日期時間 `LC_TIME=zh_TW.Big5; export LC_TIME`  
依照這樣的說法進行測試

測試結果在 \*.sh 第一行加入 `LC_TIME="en_US"; export LC_TIME`  
結果相當滿意 muhahaha 蠢問題解決!!!!!!!

下午寫出了可以提供一目了然版 switch 是否被斷網程式  
等我把crontab 裡面的所有需要執行的程式 移植到 透過java下面之後  
在進行是否被斷網程式的執行 將會優先選擇宿舍區來做

# 遇到困難 & 解決方式

這是抓取的資料 ----> 宿舍斷孔統計資料 [http://163.22.4.126/~ycc/switch\\_port.html](http://163.22.4.126/~ycc/switch_port.html)

備註

這兩天寫了 可以紀錄各 switch 上每一個 port 的狀態  
包括 宿舍可以使用的port有多少個 開通的有多少個 被斷網的有多少個  
然後計算出 " 違規使用率 " 為多少

這樣可以了解宿舍網路的使用狀況  
目前已經讓 提取的資料進資料庫了

但是 動態網頁還不會做 所以先以這樣的方式呈現  
最終的目的是讓 同學能夠依照 自己的房間號碼 或是 牆壁上網路孔的孔號  
查詢房間的所有網路孔 或是 自己所屬的網路孔是否有被斷網

我想這是學生在不能上網的時候 急迫想要知道的訊息

Computer	D.C.	CCNET
Center	DataBank	215
DataBank / 4/126	ycc	

紀錄時間 2005-06-27 23:08:20 約 每兩個小時更新一次

## 共有 147 孔被斷網 被斷網的比率為 千分之 65

10.10.48.241-22 Disconnect  
10.10.48.242-16 Disconnect  
10.10.48.243-6 Disconnect  
10.10.48.246-6 Disconnect  
10.10.48.246-10 Disconnect  
10.10.48.247-5 Disconnect  
10.10.48.249-2 Disconnect  
10.10.49.242-22 Disconnect  
10.10.49.243-11 Disconnect  
10.10.49.244-14 Disconnect  
10.10.49.244-16 Disconnect  
10.10.49.245-17 Disconnect  
10.10.49.246-16 Disconnect  
10.10.49.246-17 Disconnect  
10.10.49.247-3 Disconnect  
10.10.49.249-4 Disconnect  
10.10.49.250-17 Disconnect  
10.10.50.245-5 Disconnect  
10.10.50.245-9 Disconnect  
10.10.50.246-19 Disconnect  
10.10.50.248-9 Disconnect  
10.10.50.248-24 Disconnect  
10.10.51.237-7 Disconnect  
10.10.51.237-13 Disconnect  
10.10.51.237-18 Disconnect  
10.10.51.238-9 Disconnect  
10.10.51.238-19 Disconnect  
10.10.51.243-24 Disconnect  
10.10.51.245-2 Disconnect  
10.10.51.245-19 Disconnect  
10.10.52.242-9 Disconnect  
10.10.52.242-11 Disconnect  
10.10.52.244-24 Disconnect  
10.10.53.231-5 Disconnect  
10.10.53.231-17 Disconnect  
10.10.53.231-18 Disconnect  
10.10.53.235-2 Disconnect  
10.10.53.235-4 Disconnect  
10.10.53.235-20 Disconnect  
10.10.53.235-23 Disconnect  
10.10.53.236-17 Disconnect  
10.10.53.236-19 Disconnect  
10.10.53.236-20 Disconnect  
10.10.53.236-23 Disconnect  
10.10.53.237-2 Disconnect  
10.10.53.237-15 Disconnect  
10.10.53.237-18 Disconnect  
10.10.53.237-20 Disconnect  
10.10.53.237-21 Disconnect  
10.10.53.237-24 Disconnect  
10.10.53.238-10 Disconnect  
10.10.53.238-16 Disconnect

# 遇到困難 & 解決方式

**2005 / 06 / 03**

今天發現 每次都去手動看流量頗麻煩 因為網頁做的不是很好  
所以要跳進跳出的 如果能自動通知斷線那就會更好處理  
那可以在程式裡面加入判斷式 如果沒資料的時候即刻發送一封mail給管理者  
但是 如果晚上十點發生到隔天早上八點上班才看到 經過10小時 那一共會有120封相同的mail  
如果是做成網頁呢 應該是會比較好一點  
目前已經有統計斷網數量的程式  
應該稍微改一下程式內容就可以show出如果沒有回應時的紀錄  
還有就是無線基地台目前是否正常運作的網頁  
如果可以用顏色標明出是否正常運作 那我也就不用每次在那邊跳來跳去了

**2005 / 05 / 30**

今天早上才發現 原來教學大樓的AP都沒做rrd 所以... .. 已經完成了 ...  
完成 重開24.237 另外星期五重開的另外兩台目前正常工作中

**2005 / 05 / 28**

發現 24.237 又掛點了

**2005 / 05 / 27**

發現 科一館有三台AP無法遠端連線 分別是 24.235 24.237 24.236 因此重新開機這三台AP  
在重開之後就恢復正常狀態 持續觀察

# 遇到困難 & 解決方式

*2005/06/10*

在流量圖的程式裡面加了幾行程式碼進去  
主要目的在讓我可以一目了然知道  
我所 選取區域地區AP的健康狀況  
目前只有製作 科技學院 的部分 以每五分鐘進行一次  
測試網址

<http://163.22.4.126/~rrdtool/test.html>

*2005/06/14*

獲報科一館 五樓 AP 出現異狀 無法由此AP 拿到IP  
查看上次寫的了解地區性AP的健康狀況網頁  
發現並無異狀 然後去ping AP 一樣沒有回應 然後在進行小測試  
最後確定是 是絕對路徑的問題  
所以 第一次 就出錯了 很好!! =.=" 天阿!!  
然後看流量圖 發現 不止五樓 連四樓都掛點 上帝保佑!!

### 科技大樓無線網路基地台 健康狀況 網頁

最近一次更新時間 2005-06-22 22:42:14

24-234  斷網	23-233  正常	19-233  正常	16-235  正常
24-235  正常	23-234  正常	19-234  正常	16-236  正常
24-236  正常	23-235  正常	19-235  正常	16-237  正常
24-237  正常	23-236  正常	19-236  正常	16-238  正常
24-238  正常	23-237  正常	19-237  正常	16-239  正常
	23-238  正常	19-238  正常	16-240  正常
	23-239  正常	19-239  斷網	
	23-240  正常	19-240  正常	

# RRDtool的簡單寫法

共有三個檔案

1. rrd.sh (第一步驟)
2. wlan.sh (第二和第三步驟)
3. list (預定測試設備的IP)

# rrdtool create

```
#!/bin/bash
rrdtool create $1_$4_$5.rrd \
--start `date +%s` \
--step 300 \
DS:$1_$4_$5_in:COUNTER:600:U:U \
DS:$1_$4_$5_out:COUNTER:600:U:U \
RRA:AVERAGE:0.5:1:603 \
RRA:AVERAGE:0.5:6:603 \
RRA:AVERAGE:0.5:24:603 \
RRA:AVERAGE:0.5:288:800 \
RRA:MAX:0.5:1:603 \
RRA:MAX:0.5:6:603 \
RRA:MAX:0.5:24:603 \
RRA:MAX:0.5:288:800 \
RRA:MIN:0.5:1:603 \
RRA:MIN:0.5:6:603 \
RRA:MIN:0.5:24:603 \
RRA:MIN:0.5:288:800 \
RRA:LAST:0.5:1:603 \
RRA:LAST:0.5:6:603 \
RRA:LAST:0.5:24:603 \
RRA:LAST:0.5:288:800
```



# rrdtool update

```
#!/bin/bash
LC_TIME="en_US"; export LC_TIME
now=`date +%s`
In1=`snmpwalk -v 1 -c admin $2.$3.$4.$5 ifInOctets.1 | gawk '{ print $4 }'`$In1
In2=`snmpwalk -v 1 -c admin $2.$3.$4.$5 ifOutOctets.1 | gawk '{ print $4 }'`$In2
echo "$now:$In1" >> /home/rrdtool/test/$1_$4_$5_in.cmd
echo "$now:$In2" >> /home/rrdtool/test/$1_$4_$5_out.cmd
if [ "$In1" = "" ]&&[ "$In2" = "" ]; then
    echo "0" > /home/rrdtool/test/$4-$5
else
    echo "1" > /home/rrdtool/test/$4-$5
fi
A=`snmpwalk -v 1 -c admin $2.$3.$4.$5 mib-2.17.4.3.1.1 | gawk '{ print $4 $5 $6 $7 $8 $9}'`$A
echo "`date "+%Y/%m/%d %H:%M:%S"`
$A" >> /home/rrdtool/test/$1_$4_$5.MAC

rrdtool update /home/rrdtool/test/$1_$4_$5.rrd $now:${In1}:${In2}
```

# rrdtool graph

