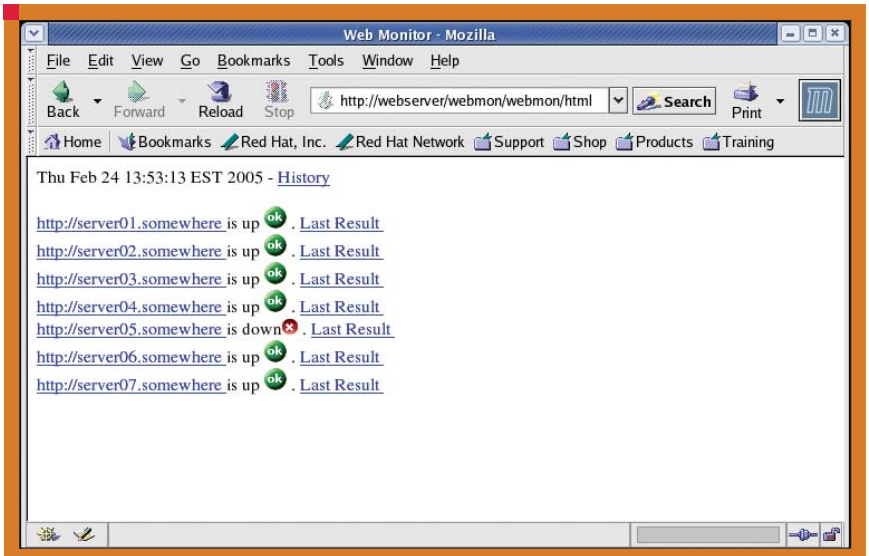
Rendszerműszerfal

Ebben a cikkben néhány olyan egyszerű héjprogramot mutatunk be, amelyekkel folyamatosan megfigyelés alatt tarthatjuk a betelésre hajlamos lemezeket, a processzort jelentősen terhelő folyamatokat, vagy a web illetve a levelezés problémáit.

A cégem körülbelül egy éven keresztül szenvedett egy megfelelő rendszermegfigyelő szoftver hiányától. A téves riasztások, a tévesen kiállított túlóralapok komolyan veszélyeztették a munkahelyi morált és pocsékolták a rendszergazdák idejét. Miután megbeszéltem a kollégákkal, hogy pontosan milyen monitorozó eszközre lenne szükségünk, az derült ki, hogy a megfigyelendő dolgok listája nem is olyan hosszú:

- Tudnunk kell a teljes *HTTP* forgalomról, vagyis nem csak a fizikai kiszolgálókról.
- Figyelnünk kell a lemezterület felhasználását.
- Tudnunk kell, hogy *SMPT* protokollon keresztül rendelkezésre állnak-e az *SMTP* kiszolgálók, vagyis megint nem elég a kiszolgáló egyszerű fizikai megfigyelése.
- Rögzítenünk kell az ezekkel kapcsolatos eseményeket, hogy kiszűrhessük az esetleges problémákat.

Ebben a cikkben bemutatom az általam kifejlesztett megoldást, illetve azt, hogy miként valósítottam meg könnyen és gyorsan a lemezek, web- és levélkiszolgálók megfigyelését. Ahhoz, hogy a monitoring rendszer egyszerű maradhasson, minden felhasznált eszköznek jelen kellett lennie valamelyik aktuális *Linux* terjesztésben, illetve nem használhattam sem összetett adatbázisokat, sem olyan kifinomult protokollokat mint teszem azt az *SNMP*. Ami azt illeti, a teljes rendszerem kizárólag a *Bash* szolgáltatásait, alapvető *HTML* megoldásokat, néhány *Perl* szkriptet és a *wget* segédprogramot használja.



1. ábra A monitor_web.sh működés közben. Futtassuk a szkriptet a cron segítségével a nekünk megfelelő időközönként. Minden futás újragenerálja a weblapot.

Valamennyi szkript felépítése ugyanazt az alapvető vázat követi, ugyanúgy telepíthetők, és valamennyi letölthető a *Linux Journal FTP*-helyéről (lásd a kapcsolódó címeket). A programok telepítését a következőképpen végezhetjük el. Először is másoljuk fel őket egy webkiszolgálóra, majd a `chmod` segítségével tegyük valamennyi mindenki számára futtathatóvá. Hozzunk létre a webkiszolgáló gyökerében egy olyan könyvtárat, amelyben a szkriptek a naplóikat tárolhatják. A `monitor_web.sh` megvalósítása során a `webmon` képességeit használtam fel. Ehhez teljesen hasonlóan a `monitor_smtp.sh` az `smtpmon`-ra, míg a `monitor_stats.pl` szkript a `stats` parancsra támaszkodik. A `monitor_disk.sh` program ezektől annyiban különbözik, hogy ezt helyileg kell telepíteni minden olyan kiszolgálóra, amit meg akarunk figyelni.

A következő lépésben a szkriptek futását időzítenünk kell a *cron* segítségével. A programok bármely olyan felhasználó nevében futhatnak, akinek megvannak a jogosultságai a `wget`, a `df -k` és a `top` futtatásához. Az illető felhasználónak ezen kívül írási joggal kell rendelkeznie a szkriptek saját könyvtárára.

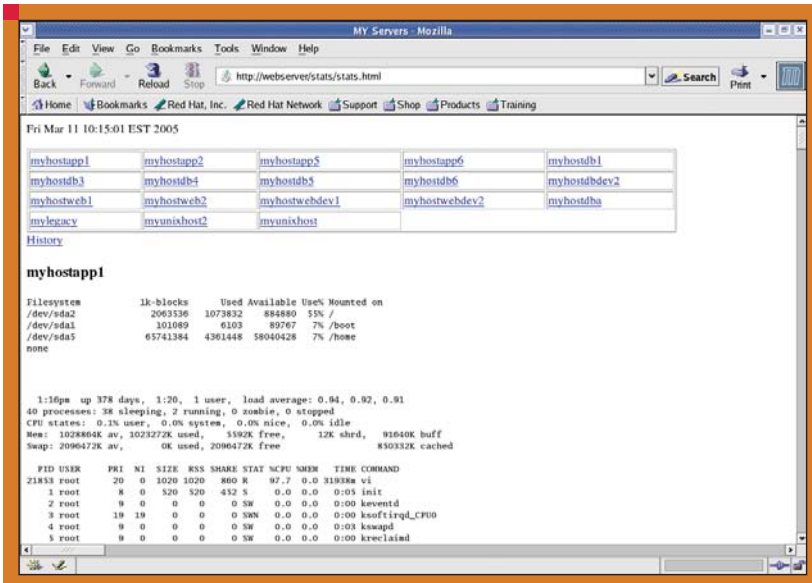
A legegyszerűbb talán az, ha létrehozunk egy monitor nevű virtuális felhasználót, és minden programot ennek a nevében futtatunk. Végül ha telepített rendszerünknek nem része a wget, ezt is telepítenünk kell.
Az első próbálkozásom a webkiszolgálók megfigyelése volt. „Motornak” a már említett `wget`-et használtam és tulajdonképpen e köré írtam egy megfelelően kialakított szkriptet. Ez lett a `monitor_web.sh`. Annak, aki esetleg nem ismerné a `wget`-et, leírom, hogy mit állít róla a program szerzője: ***„a wget olyan szabad szoftver, amely fájlokat tud letölteni HTTP, HTTPS és FTP segítségével, vagyis az interneten legelterjedtebben használt protokollokkal”*** (lásd a kapcsolódó címeket).
Telepítés után a `monitor_web.sh`-nak mindössze két információra van szüksége: hova küldhet e-mailt és mely webkiszolgálókat tartsa megfigyelés alatt. A megadott ***URL***-eknek igazodniuk kell a ***HTTP*** szabványokhoz, a kiszolgálóknak pedig érvényes `http 200 OK` karakterláncot kell visszaküldeniük ahhoz, hogy a szkript helyesen működjön. A címekben ***HTTP*** és ***HTTPS*** protokoll egyaránt szerepelhet, mivel a szkript is és a ***wget*** is támogatja mindkettőt. Ha telepítettük és futtatni kezdtük a megfigyelőprogramot, az erre feljogosított felhasználó a ***http://localhost/webmon/webmon.html URL***-en jelenítheti meg a statisztikákat és eseményeket egy webböngésző segítségével.
Vizsgáljuk most meg közelebbről a `monitor_web.sh` szkript belső szerkezetét. Először is beállítjuk az összes olyan változó értékét, amit a rendszer segédprogramjai, illetve a `wget` használ. Ezek természetesen helyről helyre változhatnak, tehát beállításuk a telepítés része. A következő lépésben ellenőrizzük, hogy él-e a hálózati kapcsolat. Ez megakadályozza, hogy túlterheljük a ***Sendmail*** rengeteg téves riasztás kiküldésével abban az esetben ha a monitorozó gép valamiért leválik a hálózatról.
Aztán egy ciklussal végigmegyünk az összes megadott ***URL***-en úgy, hogy a `wget` öt másodperces késleltetéssel kétszer kísérel meg kapcsolódni valamennyi kiszolgálóhoz. Ha a kapcsolatfelvétel sikertelen, a szkript elküld egy levelet a beállítása során megadott címre, és frissíti a statisztikai weblapot. A program ezután folyamatosan figyeli, hogy mikor éled fel újra a kérdéses kiszolgáló, és erről az eseményről is küld egy levelet. Mivel minden váltáskor csak egy ilyen levél megy ki, nem terheljük túl a címzett gépét. Mindezt a következő kód valósítja meg:

```
wget $URL  2>>   $WLOG
if (( $? != 0 ));then
  echo \<A HREF\="$URL"\>$URL\
  ➥<\/A\> is down\
          $RTAG $EF.\
          $LINK Last Result
          ➥$LTAG  >> $WPAGE
  if [[ ! -a down.$ENV ]];then
        touch /tmp/down.$ENV
        mail_alert down
  else
        echo Alert already
        ➥sent for $ENV \
        - waiting | tee -a
        ➥$WLOG
  fi
fi
```

Miután elintéztük a webkiszolgálókat, ideje, hogy a lemezhasználatot is górcső alá vegyük. Az imént meghirdetett egyszerűség szent nevében itt sem történik semmi egyéb, mint hogy a `df -k` parancs kimenete alapján értesítjük az érdekelteket a kialakult helyzetről, az időzítést pedig most is a `cron` segítségével végezzük. A kérdéses program a `monitor_disk.sh`, amelynek első, igazán lényeges részlete feltérképezi, hogy milyen fájlrendszerek vannak rendszerünkben:

```
FILESYSTEMS=$(mount | grep -iv
➥proc |\
grep -iv cdrom | awk'{print
➥$3}')
```

Természetesen kihagyjuk a dologból a */proc* fájlrendszert, és arra is ügyelünk, hogy a ***CD-ROM*** se kerüljön bele a statisztikába, ha esetleg valamelyik kolléga berakott volna egy lemezt. A százalékban kifejezett lemeztelítettségeket (`Use%`) a szkript két értékkel, `THRESHOLD_MAX`-szal és `THRESHOLD_WARN`-nal hasonlítja össze.



2. ábra A monitor_stats.pl akció közben

Ha bármelyiket túllépte valamelyik fájlrendszer, a program összeállít egy levelet, és elküldi a megadott címre. Figyeljük meg, hogy az itt látható kódsorral mindenütt gondoskodtam róla, hogy a százalékos foglaltságot a program egész számként kezelje:

```
typeset -i  UTILIZED=$(df -k
➥$FS | tail -1 | \
awk '{print $5}' | cut -d"%"
➥-f1)
```

A generált üzeneteknek felállítottunk egy levelezési listát, amelynek tagja minden kolléga, illetve az a külső személy is, akit probléma esetén riasztani kell. A dolognak ez a részét természetesen máshogy is meg lehet oldani.

Telepítve van a rendszerem ez a Perl modul?

Ha ellenőrizni akarjuk, hogy egy adott Perl modul telepítve van-e saját rendszerünkön, adjuk ki a következő parancsot:

```
$ perl -e "use Net::SMTP";
```

Ha semmi nem jelenik meg, akkor minden a legngyobb rendben, a modul telepítve van. Ha viszont hiányzik, akkor ilyesféle hibaüzenetet kapunk:

```
$perl -e "use Net::OTHER";
Can't locate Net/OTHER.pm in @INC (@INC contains:
/usr/lib/perl5/5.8.3/i386-linux-thread-multi /usr/lib/perl5/5.8.3
/usr/lib/perl5/vendor_perl/5.8.0 /usr/lib/perl5/vendor_perl .) at -e line
1.
BEGIN failed-compilation aborted at -e line 1.
```

Küldhetünk például értesítést egy adott csoport valamennyi tagjának, illetve használhatnak a kollégák személyhívót is.
Mivel az általunk használt fájlrendszerek viszonylag nagyok (72-140 GB körüliek), a kritikus értesítési szintet 95 százalékos telítettségre állítottam, hogy adott esetben legyen még időnk beavatkozni. A `THRESHOLD_MAX` és `THRESHOLD_WARN` változóknak természetesen ettől eltérő értékeket is adhatunk, saját igényeinknek megfelelően. Nálunk tovább árnyalja a helyzetet, hogy adatbázis kiszolgálóink is rendszeresen futtatnak olyan folyamatokat, amelyek viszonylag sok lemezterületet igényelnek, illetve tetemes mennyiségű naplóbejegyzést generálnak. Ennek megfelelően a megfigyelési időközt 15 percre állítottam. Ez a dolog természetesen megint hely- és alkalmazásfüggő, hiszen egy kevésbé terhelt rendszeren az egy órás időköz is tökéletesen megfelelő lehet.
A harmadik szkript (`monitor_smtp.sh`) a levélkiszolgálók levélküldési képességét vizsgálja. Szerkezetét tekintve rendkívül hasonló a két mér ismertetett programhoz, hiszen semmi egyebet nem tesz, mint ellenőrzi, hogy képes-e kapcsolódni egy megadott *SMTP* kiszolgálóhoz, és el tud-e küldeni a segítségével egy levelet. A kód tehát a szokásos módon végigmegy a felhasználó által beállított listán, és minden levélkiszolgálónak elküld egy tesztlevelet.
Ez az a pont, ahol belép a képbe az `smtp.pl` szkript. Ez egy *Perl* szkript *(1. Lista)*, amely a *NET::SMTP* modul segítségével küld levelet egy *SMTP* címre. A legtöbb mai *Linux* terjesztés amúgy eleve tartalmazza ezt a modult. Attól függően, hogy az `smtp.pl` sikeresen lefutott-e, a monitorozó szkript frissíti a naplót tartalmazó weblapot. Itt most persze nem küldünk levelet akkor sem, ha gond van, hiszen – érthető módon – az nem biztos hogy célba érne. Ez a program tehát nem is annyira megfigyelésre, inkább hibakeresésre való. Tervezem egy olyan, fejlettebb változat megírását is, amely hiba esetén egy másik, biztosan működő *SMTP* kapcsolaton keresztül mégis kiküld egy értesítést.
Végül elérkeztünk a `monitor_stats.pl` szkripthez. Ez a program minden gazdagépre bejelentkezik, és végrehajtja a következő parancsokat:

1. Lista Az smtp.pl a kimenő levelek kézbeeíthetőségét ellenőrzi az SMTP kiszolgáló lekérdezésével

```
#!/usr/bin/perl -w
#
# Title  :   smtp.pl
# Author :   John_Ouellette@yahoo.com
# Files  :   smtp.pl
# Pupose :   Send email through SMTP server
#            Called from monitor_smtp.sh
#
# Submit as
use Net::SMTP;
my $rcpt   = $ARGV[2] || 'mygroup@somewhere';
my $sender = $ARGV[1] || 'root@host01';
my $host   = $ARGV[0];
#Start Script
my $smtp =Net::SMTP->new($host, Debug => 1);
my $input="test msg for server $host";
$smtp->mail("$sender");
$smtp->to("$rcpt");
$smtp->data();
$smtp->datasend("To: $rcpt\n") ;
$smtp->datasend("From: $sender\n") ;
$smtp->datasend("Subject: $host test\n") ;
$smtp->datasend("$input");
$smtp->dataend();
$smtp->quit;
```

```
df -k
swapon -s
top -n 1 | head -n 20
hostname
uptime
```

Az eredményeket egyrészt megjeleníti egy böngészőben *(2. ábra)*, másrészt dátum szerint sorba rakva beírja egy naplóba. Ez a program tehát amolyan központi műszerfalként funkcionál, amely lehetővé teszi, hogy villámgyorsan információt szerezzünk a számunkra érdekes gépekről.
A rendszermegfigyelés ilyen kialakításának rögtön három haszna is van:

1. Teljes áttekintésünk van a processzorok, a lemezek és a csereterületek használatáról, így könnyen kiszúrhatjuk, ha valahol hiba van.
2. Könnyebben férünk hozzá egy adott kiszolgáló adataihoz. Így egyrészt nem kell hosszas lekérdezéseket fogalmazgatni, ha éppen kutatunk valami után, másrészt eleve könnyebben buknak ki a hibák, ami csökkenti annak az esélyét, hogy az éjszaka kellős közepén kapjunk egy kellemetlen telefont vagy riasztást.
3. A naplókat a vállalat vezetősége is figyelemmel kísérheti, így pontosan láthatják, mennyi jók vagyunk.

Összefoglalás

A cikkeben vázolt egyszerű rendszermegfigyelő szoftvernek köszönhetően az általam vezetett csoport termelékenysége jelentősen megnőtt, másrészt az emberek végre hinni kezdtek a megfigyelt eredményekben, mert nem kell az idejüket a téves riasztásokra vesztegetniük. A telepítménnyel kapcsolatos problémák kiszűrésében is nagy segítségnek bizonyult az, hogy szükség esetén a rendszer teljes története rendelkezésre áll. Végezetül a bemutatott programok egyszerű telepíthetősége a kevésbé képzett kollégáknak is lehetővé teszi, hogy akár egyetlen munkanap alatt beletanuljanak a rendszerfelügyeletbe.

Linux Journal 2005. 137. szám

John Ouellette

Rendszergazdaként dolgozik. Kilenc év tapasztalata van Microsoft Windows NT és UNIX rendszerekkel kapcsolatban. Hiszi, hogy a parancssor nagyszerű dolog.
Johnt a john_ouellette@yahoo.com címen érhetjük el.

KAPCSOLÓDÓ CÍMEK

- ftp.ssc.com/pub/lj/listings/issue137/8216.tgz
- www.gnu.org/software/wget/wget.html
- www.gnu.org/software/bash/manual/bashref.html
- search.cpan.org/~gbarr/libnet-1.19/Net/SMTP.pm
- perl.about.com/b/a/007039.htm