

Network Monitoring System

Source: <http://linux.derkeiler.com/Newsgroups/comp.os.linux.misc/2004-05/0438.html>

From: Graham Nicholls (graham_at_rockcons.co.uk)

Date: 05/05/04

Date: Wed, 05 May 2004 16:39:58 +0100

I Posted this earlier to c.o.l.answers, but perhaps here is better:

I want to monitor various things on client networks, such as status of backup, disk space, etc.

Big brother and its siblings are a big pain, IMO, so I've written my own stuff. Its simple.

Basically there are 3 programs and a database;

a sender on each machine which wants to send messages to the monitor, a receiver on the logging machine, and a database loader on the receiving machine. Each is written in python, and is only 100-200 lines of code (IIRC).

What happens is this:

an event happens - perhaps a daemon starts. AS part of the startup script, the admin puts in something like this:

```
if [ $? -eq 0 ]
then
    echo "~myservername~service~2~Service plokij started ok
$(date)~ ..... etc
>> /var/log/log.send
else
    echo "~myservername~service~6~Service plokij failed to start
$(date) ...
etc >> /var/log/log.send
fi
```

What this does is append to a file a message according to the status of a process. The message is in the format [delimiter]field[delimiter]field ... etc, where the 1st char in the line is the delimiter.

The sender process on that machine is quietly waiting for data in the file "/var/log/log.send", and when it sees data, it sends it across the network to the receiving program on the logging machine, where it is written to a file - both files are I suppose, FIFOS. On the receiving machine, a loader process reads the file which is constantly being appended to, and uses the

data to populate a new record in a mysql database. Both the sender & the loader logically AND the 1st character of the line in the file to indicate that the line has been successfully sent, in case either process has to be restarted.

We now have a mysql database with potentially all sorts of data – disk space, process info, snmp derived stuff, etc, which can be queried to generate a nice html page using simple php scripts.

It seems to me that this is a nice way of logging all sorts of info about networks, machines, routers etc. Simple shell scripts just have to write data to a file in an agreed format. These scripts could be written in any language – perl (ugh!), python, ruby, shell, etc, they could even run on windows (ugh, again!). All data is archived on the sender & receiver. Recovery is simple – just send all lines whos 1st char is below 128. Reports could be nice – who logged on to hp_server_1 on 21st jan would be a simple sql query which would work from a simple web interface. Above all, the system is SIMPLE. Simple to install (you need python), simple to setup (a startup script in /etc/init.d, with appropriate links to the runlevel), simple to extent – you just write scripts either on the machine to be monitored, OR on the logger – if on the logger, then data need just be appended to the receive file as the listener program does (or use a sender to localhost).

The database format I have already designed, but it may need to be changed.

Now, my question is (or are :-)

1. Am I wasting my time 'cause theres already something better out there?
2. Is this of any use to anyone else?
3. Anyone else fancy writing some agents to get info – eg in the style of top, etc.
4. Anyone want to help with some php to query the database?
5. Is anyone interested, full stop?
6. Is my database design right for the future(you'll have to email me for it!

Thanks, sorry its a long message
Graham Nicholls

--
#include <wit>