Munin FAQ

Table of Contents

1. Installation

Q: What architectures are supported? What operating systems?

Munin is programmed in Perl, which can be installed on most operating systems. In addition, it needs some perl modules, which you can fetch from CPAN.

However, the "plugins" used by the client are often OS or application spesific. E.g., many of the Linux−plugins use /proc to gather the data. There are some plugins that should work on most architectures. The architectures which have their own plugins as well, are

Linux
FreeBSD
SunOS (Solaris)
AIX

If you port/create/improve plugins, please contribute them to the project.

2. Configuration

Q: What should a minimal configuration look like?

```
dbdir     /var/lib/munin
htmldir   /var/www/munin
logdir    /var/log/munin
rundir    /var/run/munin

[bing.foo.bar]
        address 10.232.33.259
```

The server will expand the node to contain all the services it offers.

Q: How du I use the "fieldname.cdef"−thingie?

The cdef is fed to "rrdtool graph", so the man−page for "rrdgraph" should give you a bit more info. I'll try to explain it briefly here, though;

The cdef is defined using Reverse Polish Notation (RPN), which means that you would say "4,5,+" instead of "4+5". When you create a cdef−field, be sure to use the fieldname at least once in the definition, or rrdgraph will croak with an error.

For a more thorough definition of RPN, take a look at the man−page for "rrdgraph".

Q: Can I get a graph to show percentages instead of regular values?

You can modify a graph with two data fields to a percentage view with the following cdef (gotten from Nicolai Langfeldt). It uses two field names, "hit" and "all", with "hit" being the field you want in percentages of "all".

```
hit,all,1,all,0,EQ,IF,/,100,*,UNKN,all,0,EQ,IF,UNKN,all,UNKN,EQ,IF
```

Q: How do I define aliases with graph_order?

Normally, graph_order looks something like this:

```
graph_order apps buffers cache unused swap
```

It can, however, be used to define alises from other graphs. E.g. if you want to incorporate the number of http−connections in the "processes"−graph (i.e. together with the number of processes on the machine), you'd say:

```
graph_order processes connections=port_http.count
```

This would first draw the "proesses" data−source (as normal), then draw the "count" data−source from the "port_http"−graph. "connections" would then be a field−name in the same way as "processes", so you could give it a cdef, draw, negative, etc.

Example graph_orders:

- `graph_order bing bofh` (will draw two normal values)
- `graph_order bing bang=bing` (will draw two normal values, the second an alias to the first)

`graph_order bing bang=graph.bing` ("loan" a data−source from another graph)
- `graph_order bing bang=host:graph.bing` ("loan" a data−source from another host's graph)
- `graph_order bing bang=domain;host:graph.bing` ("loan" a data−source from another domain's graph)

Q: How do I use fieldname.stack?

The format of "special_stack" is the same as the graph_order−arguments above. You cannot, however, specify extra arguments for the fields. If you want to specify a cdef for the 'whole stack', you can use the fieldname defining the special stack. E.g.

```
[some.machine.boo]
    total_mail.graph_order total_received
        total_mail.graph_title Mail received by machine1 and machine2
        total_mail.graph_vlabel mails/min
        total_mail.total_received.label not_used
        total_mail.total_received.stack \
            machine1=machine1.your.dom:exim_mailstats.received \
            machine2=machine2.your.dom:exim_mailstats.received
        total_mail.total_received.cdef total_received,60,*
```

Q: How do I use fieldname.sum?

Same as special_stack, with one exception: drop the "alias"−bit in the field definition. I.e., the above would become:

```
[some.machine.boo]
    total_mail.graph_order total_received
        total_mail.graph_title Mail received by machine1 and machine2
        total_mail.graph_vlabel mails/min
        total_mail.total_received.label mails per minute # Now used
        total_mail.total_received.sum \
            machine1.your.dom:exim_mailstats:received \
            machine2.your.dom:exim_mailstats:received
        total_mail.total_received.cdef total_received,60,*
```

Q: Can I change the order of the domains?

Yes, use the "domain_order" option at the topmost level. To specify that you want the topmost level, either put the option before any host/domain definitions, or reset the host/domain definition with "[]".

```
        []
                domain_order foo.bar goo.bar alpha.bar
```

The default domain order is alphabetically.

Q: Can I change the order of the nodes under a domain?

Yes, use the "node_order" option under a domain. To specify that we want to modify a variable under the domain, supply an empty host:

```
        [bing.foo.bar]
                address 10.232.33.259
```

```
[fii.foo.bar]
        address 10.232.33.259

[goo.foo.bar]
        address 10.232.33.259

[foo.bar;]
        node_order fii.foo.bar bing.foo.bar goo.foo.bar
```

The default node order is alphabetically.

Q: How can I use an SSH tunnel to connect to a node?

You have to configure the node thusly:

```
[ssh-node] address 127.0.0.1 port 5050
```

Then use ssh to establish the tunnel:

```
ssh -L 5050:localhost:4949 -f -N -i keyfile user@ssh-node
```

This will establish a tunnel between TCP ports 5050 on the calling machine to 4949 on the called machine. It will also send ssh in the background after possibly asking for a passphrase, a password or something like that. Since we are using a key made for this purpose, we have to specify that file with this key.

You should protect against misuse of ssh by creating a special key (and possibly also a special user). On the node, put something like this in ~user/.ssh/authorized_keys:

```
from="192.168.1.35",command="/bin/false",no-pty,no-X11-forwarding,no-agent-forwa
ssh-dss AAAAB3......
```

Thus, we are restricting the key to a forced command "/bin/false" that is run independent of the request from the calling side. We are also restricting a few options:

from="192.168.1.35"
: accept the key only from this IP address

command="/bin/false"
: always run this command

no-pty
: never allocate a PTY for interactivity

no-X11-forwarding
: do not forward X11 client connections

no-agent-forwarding
: prevent ssh-agent usage

no-port-forwarding
: prevent ssh -R ...

permitopen="localhost:4949"
: only allow this for ssh -L ...

This FAQ entry was developed by Lupe Christoph, with the help of Jim Cheetham.

Q: I don't like the bandwith usage graph, how to make it look like MRTG or Cricket graphs?

For interface eth0 on host foo.bar.example,

```
if_eth0.graph no

my_ifeth0.graph_args --base 1000 --lower-limit 0
my_ifeth0.graph_title Traffic on interface eth0
my_ifeth0.graph_order received=foo.bar.example:if_eth0.down sent=foo.bar.example:if_eth0.
my_ifeth0.graph_vlabel Bits per second
my_ifeth0.received.cdef received,8,*
my_ifeth0.received.draw AREA
my_ifeth0.sent.cdef sent,8,*
my_ifeth0.sent.draw LINE1
```

(Answer supplied by Jacques Caruso.)

Q: I added the graph_sums option, bug I only get one new graph (instead of two).

The graph_sums option require that rrdtool version 1.0.39 or above is installed.

3. Client plugins

Q: What is the minimum requirements of a client plugin?

- Without any parameters it should print "<fieldname>.value <value>"
- It must support the "config"-parameter
- With the config-parameter, it should print at least:
    - <fieldname>.label <label_of_field>

An example plugin (on a linux-system, using /proc/loadavg to graph the load average):

```
#!/bin/sh

if [ "$1" = "config" ]; then
   echo "load.label load"          # Field label (legend)

   # These are not really nedded, but makes the graph prettier.
   echo "graph_title Load average" # Set main title
   echo 'graph_args -l 0'          # Y-axis starts at 0
   echo 'graph_scale no'           # Say 0.04 load, not 40 milliload
   echo 'graph_vlabel load'        # Y-axis label

   exit 0
fi

# The real data-gathering
printf "load.value "
cut -f1 -d' ' < /proc/loadavg
```

Q: Can I make the plugin run as another user/group than nobody/nogroup?

Yes, you can create a file in the plugin configuration directory (client−conf.d). The file should contain the username and group to run the plugin as. E.g., on linux, the exim_mailqueue−plugin need access to the exim mail spool (to count the messages in the queue). It needs "mail" group permissions to do this, so the "exim_mailqueue.auth" contents look like:

```
[exim_mailqueue]
        group mail
```

Similarly, you could use the "user" option to run the plugin as a user.

If more than one group is needed, or some of the groups only exist on certain hosts (and you want a common config file), the syntax supports this:

```
[some_plugin]
        group mail,adm,group1,group2,(group3_that_might_not_exist)
```