

# Measuring Samba Performance

Douglas Bagnall

2017-05-03

# Measuring Samba performance

Douglas Bagnall  
douglas@catalyst.net.nz

# Measuring Samba performance

- ▶ A *huge* topic
- ▶ I will talk about a couple of specific things

Specific thing 1: `make perftest`

Samba has a well-developed test system  
irreplaceable

```
make test
```

To test everything<sup>1</sup>:

```
make test
```

To run some tests:

```
make test TESTS=foo
```

1: not actually *everything*

make test under the hood 1

```
make test TESTS=foo
```

is converted to

```
./buildtools/bin/waf --test TESTS=foo
```

## make test under the hood 2

```
./buildtools/bin/waf --test TESTS=foo
```

calls

```
(/usr/bin/perl
```

```
/home/douglas/src/samba/selftest/selftest.pl
```

```
--target=samba --prefix=./st
```

```
--srcdir=/home/douglas/src/samba
```

```
--exclude=/home/douglas/src/samba/selftest/skip
```

```
--testlist="/usr/bin/python
```

```
/home/douglas/src/samba/selftest/tests.py|"
```

```
--testlist="/usr/bin/python
```

```
/home/douglas/src/samba/source3/selftest/tests.py|"
```

```
--testlist="/usr/bin/python
```

```
/home/douglas/src/samba/source4/selftest/tests.py|"
```

```
--exclude=/home/douglas/src/samba/selftest/slow
```

```
--nss_wrapper_so_path=/home/douglas/src/samba/bin/default/
```

```
--resolv_wrapper_so_path=/home/douglas/src/samba/bin/default/
```

```
--socket_wrapper_so_path=/home/douglas/src/samba/bin/default/
```

## make test under the hood 3

replacing \$HOME/src/samba and /usr/bin, and reformatting:

```
(perl ./selftest/selftest.pl --target=samba \  
  --prefix=./st \  
  --srcdir=. \  
  --exclude=./selftest/skip \  
  --testlist="python ./selftest/tests.py|" \  
  --testlist="python ./source3/selftest/tests.py|" \  
  --testlist="python ./source4/selftest/tests.py|" \  
  --exclude=./selftest/slow \  
  --nss_wrapper_so_path=./bin/default/lib/nss_wrapper/libnss_wrapper.so \  
  --resolv_wrapper_so_path=./bin/default/lib/resolv_wrapper/libresolv_wrapper.so \  
  --socket_wrapper_so_path=./bin/default/lib/socket_wrapper/libsocket_wrapper.so \  
  --uid_wrapper_so_path=./bin/default/lib/uid_wrapper/libuid_wrapper.so \  
  --use-dns-faking \  
  --socket-wrapper \  
  foo \  
  && touch ./st/st_done \  
)
```



## make test continued

```
(perl ./selftest/selftest.pl [...]\  
  --exclude=./selftest/skip \  
  --testlist="python ./selftest/tests.py| " \  
  --testlist="python ./source3/selftest/tests.py| " \  
  --testlist="python ./source4/selftest/tests.py| " \  
  [...] \  
) \  

```

test are defined in these lists

```
--testlist="a-list-of-tests.txt"
```

```
--testlist="a-script-that-generates-a-list-of-tests.sh|"
```

## test list format

XXX highlight foo

-- TEST --

samba3.blackbox.smbclient\_posix\_large -e NTLM(nt4\_dc:local)

nt4\_dc:local

/home/douglas/src/samba/source3/script/tests/test\_smbclient

WORD --configfile=\$SMB\_CONF\_PATH 2>&1 | /home/douglas/src/

.smbclient\_posix\_large -e NTLM." --suffix="(nt4\_dc:local)"

-- TEST --

samba3.blackbox.smbclient\_netbios\_aliases [foo](ad\_member:)

ad\_member:local

/home/douglas/src/samba/source3/script/tests/test\_smbclient

-- TEST --

samba3.blackbox.smbclient\_netbios\_aliases [bar](ad\_member:)

ad\_member:local

/home/douglas/src/samba/source3/script/tests/test\_smbclient

-- TEST-LOADLIST --

generated by:

tests are run in a fake environment

XXX doesn't go here

using wrappers with LD\_PRELOAD magic

make test TESTS=foo

```
./source3/script/tests/test_smbclient_netbios_aliases.sh \  
    ./bin/smbclient foo Administrator locDCpass1 \  
    ./st -e --configfile=./st/ad_member/lib/server.conf 2>&  
| \  
./selftest/filter-subunit --fail-on-empty \  
    --prefix="samba3.blackbox.smbclient_netbios_aliases [f  
    --suffix="(ad_member:local)"
```

the actual script can produce all manner of output but `filter-subunit` only pays attention to lines beginning with certain words (like “success” and “failure”).

## through filter-subunit again

and into format-subunit:

```
( [the bit just discussed] \  
  && touch ./st/st_done \  
  ) \  
| python -u ./selftest/filter-subunit \  
  --expected-failures=./selftest/knownfail \  
  --flapping=./selftest/flapping \  
| tee ./st/subunit \  
| python -u ./selftest/format-subunit \  
  --prefix=./st \  
  --immediate
```

## format-subunit

- ▶ takes out details
- ▶ adds summary

```
make perf-test TESTS=foo TEST_LIST="python
./source3/selftest/tests.py|"
```

This does different things in filter-subunit:

```
elapsed-time: kinit: 0.084216
```

```
elapsed-time: smbclient: 0.43144
```

and format-subunit-json:

```
{
  "kinit": 0.084216,
  "smbclient": 0.43144
}
```

## performance specific tests

Most tests make bad performance tests  
they set up barely realistic edge cases  
and are too quick  
(the times are jittery)



## performance specific tests

Supposing you want this to work:

```
make perftest TESTS='foo'
```

add the test to selftest/perf\_tests.py:

```
+for alias in ["foo", "bar"]:  
+    plantestsuite("samba3.blackbox.smbclient_netbios_alias",  
+                  "ad_member:local",  
+                  [os.path.join(samba3srcdir,  
+                                "script/tests/test_smbclient",  
+                                "smbclient3", alias, "$DC_USERNAME", "$DC_PASSWORD",  
+                                "-e", configuration)])
```

(copy-paste-reformat from ./source3/selftest/tests.py)

backporting make perftest TESTS='foo'

XXX this needs to happen outside of samba XXX in  
samba-cloud-autobuild

## samba-cloud-autobuild

Catalysts repository of miscellaneous Samba testing stuff.

<http://git.catalyst.net.nz/gw?p=samba-cloud-autobuild.git;a=tree>

## Generating realistic loads

few Samba tests reflect realistic load

Realist performance tests must be based on observed patterns but  
real-world Samba use is always secret

## Load modelling (work in progress)

Take a pcap file and throw away sensitive information

To make Samba faster we need to understand why it is slow. To help with this we have developed tools that rest on top of the Samba self-test framework and Linux perf and tracing frameworks. Using a variety of munging and visualization scripts, we can see Samba performance changes across versions and under various workloads. These tools can be turned to any part of Samba, testing workloads of varying degrees of artifice, by writing new tests or borrowing existing ones from the self-test framework. ,