NFStest and NFSometer

The next generation test tools

Steve Dickson, Red Hat
Dros Adamson, Netapp
Jorge Mora, Netapp

NFSTest

- Author: Jorge Mora
  - Jorge.Mora@netapp.com
- Function: Provides a set of tools for testing either the NFS client or the NFS server
- Website:
  - http://wiki.linux-nfs.org/wiki/index.php/NFStest
- Available: Fedora 19/RHEL 7

NFSometer

- Author: Dros Adamson
  - Weston.Adamson@netapp.com
- Function: A framework for running workloads and reporting results across NFS protocol versions
- Website:
  - http://nfsometer.linux-nfs.org/
- Availability: Fedora 18 and RHEL6 (epel)
  - http://koji.fedoraproject.org/koji/packageinfo?packageID=
  - http://fedoraproject.org/wiki/EPEL

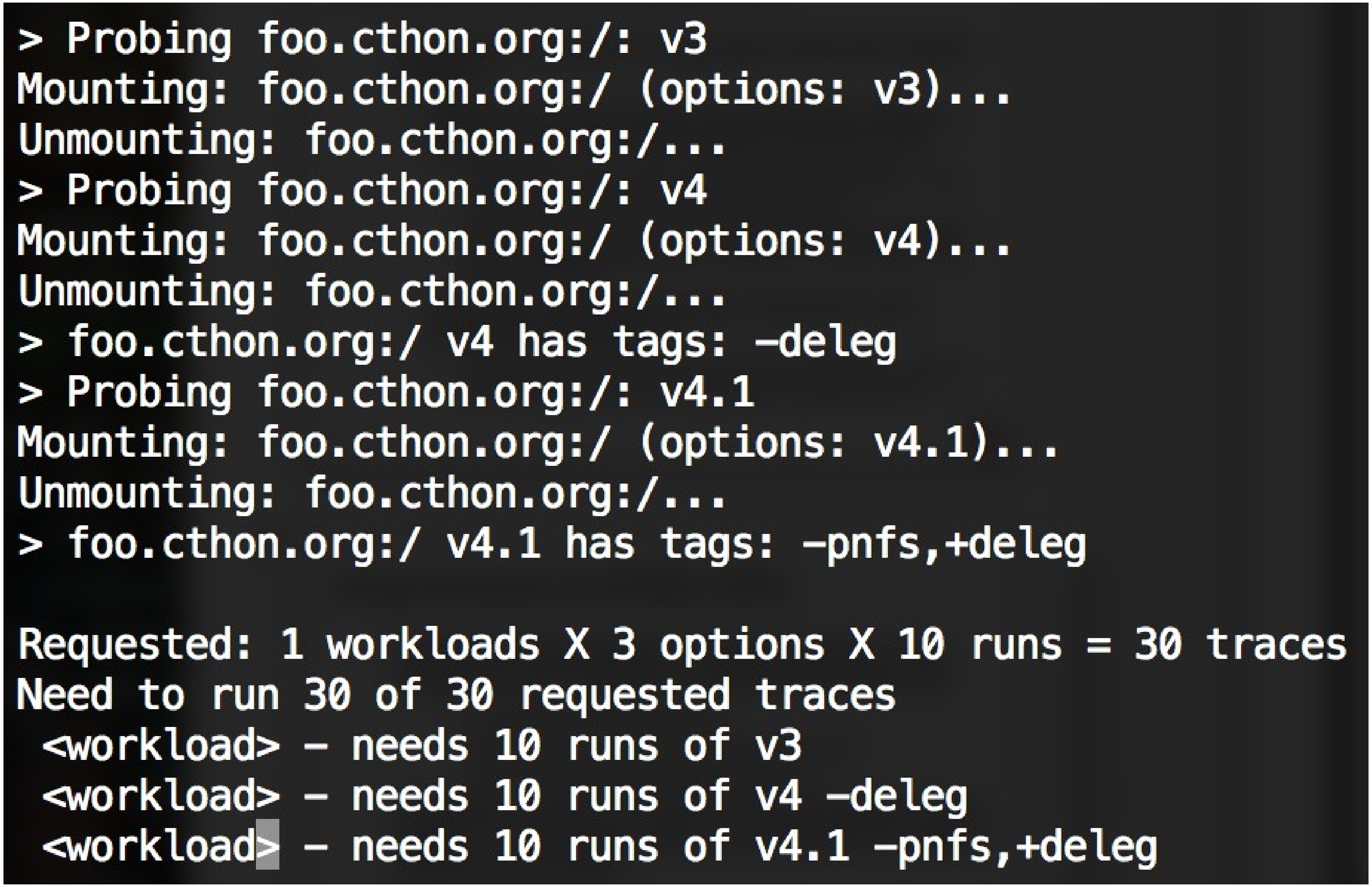
Using nfsometer at CTHONs / BATs

- nfsometer provides an easy way to kick off a series of tests:
  - Iterate over mount options by specifying multiple -o options, use -a for mount options common to all:
    - “nfsometer -o v3 -o v4 -o v4.1 server:/ <workload>” - will run each workload for each protocol version
    - “nfsometer -o v4 -o v4.1 -a sec=krb5 server:/ <workload>” - will run each workload with mountopts “v4,sec=krb5” and “v4.1,sec=krb5”
  - Run each workload a number of times for each requested configuration
    - “nfsometer -o v3 -o v4 -n 10 server:/ <workload> - will run each workload 10 times with v3 and 10 times with v4

Using nfsometer at CTHONs / BATs

- nfsometer will complain if something goes wrong and point to the tracedir in /tmp that has logs, etc:
  - The workload exits uncleanly
  - There is still an entry in /proc/fs/nfsfs/servers
  - “NFS:” lines in dmesg output during run
- nfsometer probes the server for all requested mount options (-o) to check access and determine if certain features are present:
  - pnfs
  - delegations

Using nfsometer at CTHONs / BATs



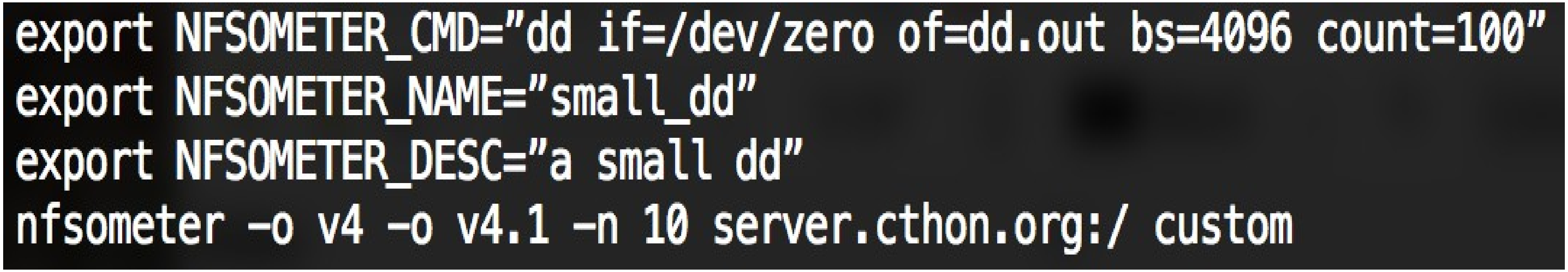
```
> Probing foo.cthon.org:/: v3
Mounting: foo.cthon.org:/ (options: v3)...
Unmounting: foo.cthon.org:/...
> Probing foo.cthon.org:/: v4
Mounting: foo.cthon.org:/ (options: v4)...
Unmounting: foo.cthon.org:/...
> foo.cthon.org:/ v4 has tags: -deleg
> Probing foo.cthon.org:/: v4.1
Mounting: foo.cthon.org:/ (options: v4.1)...
Unmounting: foo.cthon.org:/...
> foo.cthon.org:/ v4.1 has tags: -pnfs,+deleg

Requested: 1 workloads X 3 options X 10 runs = 30 traces
Need to run 30 of 30 requested traces
 <workload> - needs 10 runs of v3
 <workload> - needs 10 runs of v4 -deleg
 <workload> - needs 10 runs of v4.1 -pnfs,+deleg
```

Using nfsometer at CTHONs / BATs

- nfsometer ships with useful workloads defined:
  - cthon, iozone, filebench, etc
  - ie “nfsometer -o v4.1 server:/ cthon”
- nfsometer also has a “custom” workload that allows the user to set env variables:
  - NFSOMETER_CMD – the command to run, cwd is the runroot on the nfs server
  - NFSOMETER_NAME – a name for the workload
  - NFSOMETER_DESC – a description of the workload

Using nfsometer at CTHONs / BATs

- An example use of the custom workload that runs “dd if=/dev/zero bs=4096 count=100” to a file on server.cthon.org:/



```
export NFSOMETER_CMD=”dd if=/dev/zero of=dd.out bs=4096 count=100”
export NFSOMETER_NAME=”small_dd”
export NFSOMETER_DESC=”a small dd”
nfsometer -o v4 -o v4.1 -n 10 server.cthon.org:/ custom
```

Using nfsometer at CTHONs / BATs

- Other useful commands:
  - “nfsometer –help”
  - “nfsometer examples” - lists several examples of how to use nfsometer (same as in manpage)
  - “nfsometer workloads” - list available workloads, also shows unavailable workloads and why they're unavailable (ie “iozone not installed”)
  - “nfsometer -t <user defined tag>” - use tags to differentiate configurations that aren't mountopts