RECENT ADVANCES IN PYNFS

Andy Adamson

Alexandros Batsakis

this talk is about

- communicating how **pynfs** can be used as a **development tool**
  - this is mostly client-developer centric but it can be easily applied to the server

connectathon-oriented programming

```
while True:
        run cthon test
        if not found_bug:
                add feature
        else:
                fix bug
        continue
```

beyong cthon

- what about corner cases / error paths ?
- artifact of new protocol features
  - state recovery
  - unusual network conditions (reordering, CB down)
  - races
  - resource-constraint servers
  - unusual layouts
- solution: “hack” bad servers/clients :(

the pynfs story

- an effort led by the University of Michigan
  - RPC, NFS client and server in Python
    - relatively slow but functional and stable
    - Fred Isaman is the maintainer
- newpynfs supports v4.1
- more than an “easy to hack” server

pynfs and pNFS

- **pynfs** supports **pNFS** (files and blocks)
  - for files:
    - double personality
      - acting as a metadata & data server
    - fake multiple DSs via virtual interfaces
    - bonus: supports Linux data servers
      - a small patch is required (3 lines)
      - pretty fast

old test framework

- pynfs server creates various control files
  - config/ops/<operation>
- each operation can inspect its file
- writing an error to the file would cause the error to be returned the next time the operation is executed
- requires custom tests

new test framework

- uses an extensible instruction language
  - does more things besides error returns
  - set error, run workload, unset error

```
testclient.py /mnt --userparams=sequence:ERROR:NFS4ERR_BADSESSION:50
TWO_VALUE_SETUP_OR_CLEANUP

run cthon tests

testclient.py /mnt --userparams=sequence:ERROR:NFS4_OK:0
TWO_VALUE_SETUP_OR_CLEANUP
```

more than error codes

- NFS4SIG_CB_RECALL_TRIPLES_RACE
- NFS4SIG_RANDOM_REBOOT

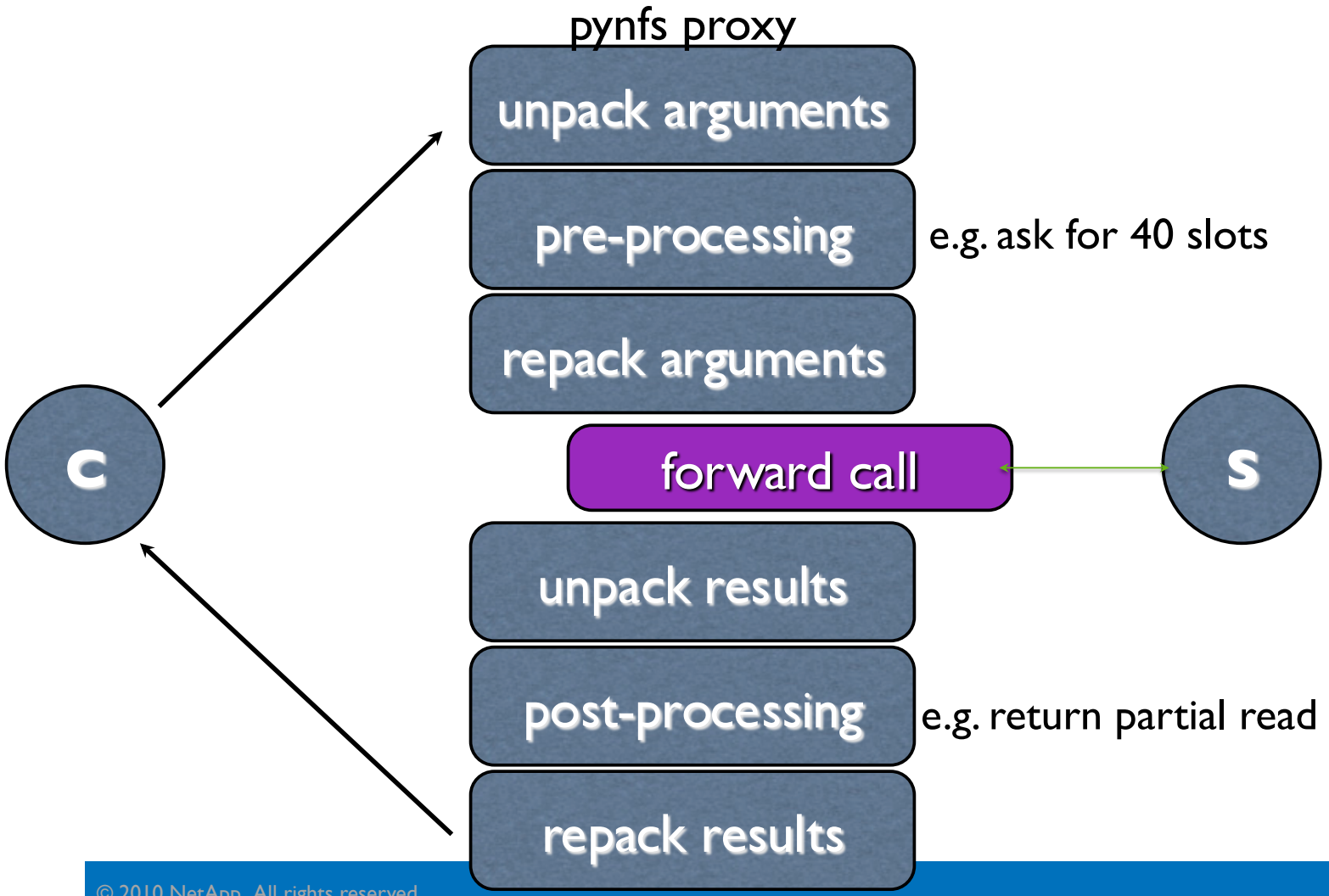
still not good enough ?

- pynfs helped us found > 20 client bugs
- suffers from a major drawback
  - is **not a "real" server**

pynfs proxy

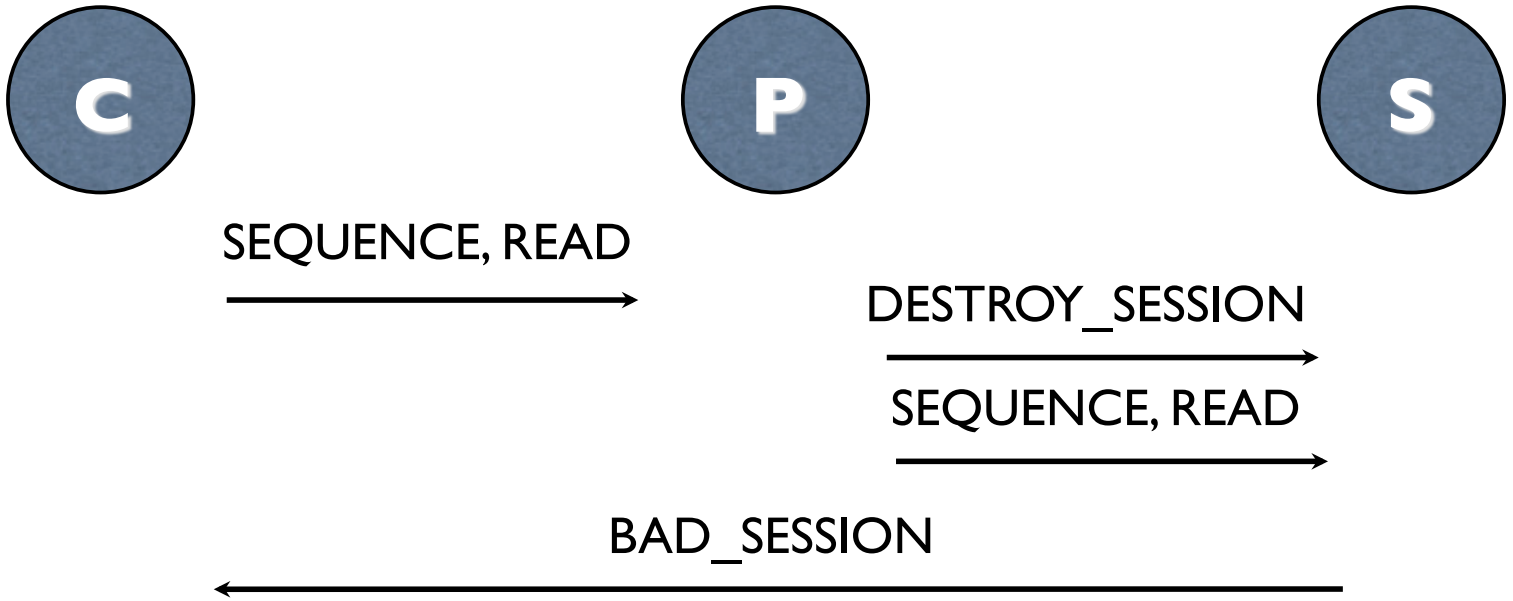
- intercepts requests from the client and forwards them to the destination server
  - NFS level -- **NFS payload-aware**
  - client - server independent
  - passes cthon !!!
    - Linux client ⟶ pynfs proxy ⟶ Linux server

proxy overview



faking state

- putting the server into an error state



proxy error injection

- error behavior described in an xml file
  - functional errors
    - return BAD_SEQUENCE every 50th operation
    - for each operation return random error
  - network errors
    - delay operation X by Y secs
    - reorder layout returns / gets
  - state errors
    - loose clientid / state / locks / layout etc.

please contribute

- ideally this is something that should be useful to every NFS developer
- we coded it in our “spare” time
- let us know if something is missing
  - or better do it yourself
  - Python is easy :)

download

- git://linux-pnfs.org/~iisaman/newpynfs.git
  - not everything there yet
  - email me (batsakis@netapp.com) if you are interested in trying