Op: Styx batching for High Latency Links

Francisco J. Ballesteros

Gorka Guardiola

Enrique Soriano

The problem

ramfs

cd /tmp/

mkdir bla

ls

cd

19 RPCs x 120ms = 2.28s

The problem

```
1.      ramfs 152684:<-Tversion tag 65535 msize 8216 version '9P2000'
2.      ramfs 152684:->Rversion tag 65535 msize 8216 version '9P2000'
3.      ramfs 152684:<-Tattach tag 7 fid 933 afid -1 uname paurea aname
4.      ramfs 152684:->Rattach tag 7 qid (0000000000000000 0 d)
5.      ramfs 152684:<-Twalk tag 7 fid 933 newfid 1060 nwname 1 0:mkdir
6.      ramfs 152684:->Rerror tag 7 ename file does not exist
7.      ramfs 152684:<-Twalk tag 7 fid 933 newfid 1060 nwname 1 0:bla
8.      ramfs 152684:->Rerror tag 7 ename file does not exist
9.      ramfs 152684:<-Twalk tag 7 fid 933 newfid 826 nwname 1 0:bla
10.     ramfs 152684:->Rerror tag 7 ename file does not exist
11.     ramfs 152684:<-Twalk tag 7 fid 933 newfid 826 nwname 0
12.     ramfs 152684:->Rwalk tag 7 nwqid 0
13.     ramfs 152684:<-Tcreate tag 7 fid 826 name bla perm %M% mode -2147483137
14.     ramfs 152684:->Rcreate tag 7 qid (0000000000000001 0 d) iounit 8192
15.     ramfs 152684:<-Tclunk tag 7 fid 826
16.     ramfs 152684:->Rclunk tag 7
17.     ramfs 152684:<-Twalk tag 7 fid 933 newfid 1060 nwname 1 0:bla
18.     ramfs 152684:->Rwalk tag 7 nwqid 1 0:(0000000000000001 0 d)
19.     ramfs 152684:<-Twalk tag 7 fid 1060 newfid 826 nwname 1 0:ls
20.     ramfs 152684:->Rerror tag 7 ename file does not exist
21.     ramfs 152684:<-Twalk tag 7 fid 1060 newfid 975 nwname 0
22.     ramfs 152684:->Rwalk tag 7 nwqid 0
23.     ramfs 152684:<-Tstat tag 7 fid 975
24.     ramfs 152684:->Rstat tag 7  stat 'bla' 'paurea' 'paurea' 'paurea' q (0000000000000001 0 d) m 020000000775 at
        1196337178 mt 1196337176 l 0 t 65535 d -256
25.     ramfs 152684:<-Tclunk tag 7 fid 975
26.     ramfs 152684:->Rclunk tag 7
27.     ramfs 152684:<-Twalk tag 7 fid 1060 newfid 975 nwname 0
28.     ramfs 152684:->Rwalk tag 7 nwqid 0
29.     ramfs 152684:<-Topen tag 7 fid 975 mode 0
30.     ramfs 152684:->Ropen tag 7 qid (0000000000000001 0 d) iounit 128
31.     ramfs 152684:<-Tread tag 7 fid 975 offset 0 count 8192
32.     ramfs 152684:->Rread tag 7 count 0 ''
33.     ramfs 152684:<-Tclunk tag 7 fid 975
34.     ramfs 152684:->Rclunk tag 7
35.     ramfs 152684:<-Tclunk tag 7 fid 1060
36.     ramfs 152684:->Rclunk tag 7
37.     ramfs 152684:<-Tclunk tag 7 fid 933
38.     ramfs 152684:->Rclunk tag 7
```

The problem

- To be fair, the dot is set there and path starts with dot.
- The clunks could be done asynchronously.
- Without the dot, less RPCs: 23
- Without the clunks and the dot, 17

17*120ms = 2s

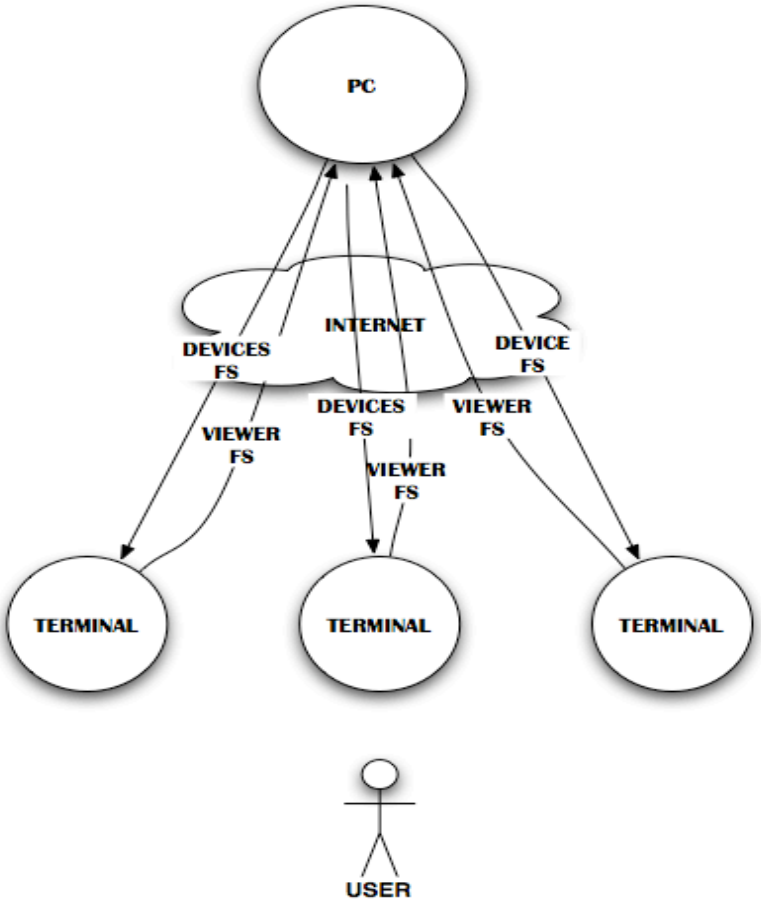
just for 3 commands!!

But the www works so?.

Octopus

- Central PC
- Terminals
- The PC mounts the terminal devices
- The terminals use the Interface FS.
- Everything goes through the internet using an FS
- Latency with Styx/9P is unbearable

The octopus:
the latency makes it unusable

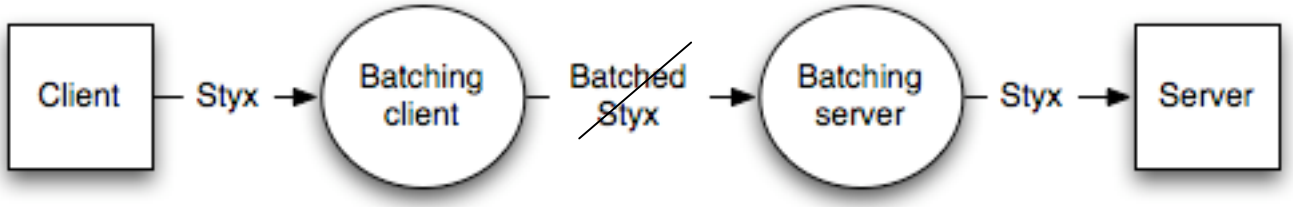


Altering Styx (proposal)

- Batch RPCs with the same tag
- We need a Batching client how do we batch? Who batches, who knows how? The kernel?.
- A Batching server for backwards compatibility (or modify all the servers)

Batching:

- We end up tunneling anyways.
- Why not use a maybe **easier**, **simpler**, **better suited** protocol?



OP

OP

size[4] Tattach tag[2] uname[s] path[s]
size[4] Rattach tag[2]
size[4] Tflush tag[2] oldtag[2]
size[4] Rflush tag[2]
size[4] **Tget** tag[2] path[s] fd[2] mode[2] nmsgs[2] offset[8] count[4]
size[4] **Rget** tag[2] fd[2] mode[2] stat[n] count[4] data[count]
size[4] **Tput** tag[2] path[s] fd[2] mode[2] stat[n] offset[8] count[4] data[count]
size[4] **Rput** tag[2] fd[2] count[4] qid[13] mtime[4]
size[4] Tremove tag[2] path[s]
size[4] Rremove tag[2]
size[4] Rerror tag[2] ename[s]

FDs

- **They are not fids**
- They cache paths already walked
- Set/chosen by the server, who knows if they are valid
- OMORE set to reuse a valid fd
- Special NOFD for the first and last request (when there is no fd to use)
- OMORE in the request not set, it gets freed
- Implicit open. If chosen by client, would need to wait for open anyway to check errors. Also the servers can reboot and nothing bad happens (the fds just get invalid, some operations get errors).

Transactions

- There can be more than one response to one request, OMORE flag for responses:
- Tget ->
- Rget <- OMORE set
- Rget <- OMORE set
- Rget <- OMORE not set (means EOF)
- Count also set, so OMORE and

count == 0 means a 0 sized read

Put

size[4] **Tput** tag[2] path[s] fd[2] mode[2] stat[n] offset[8] count[4] data[count]
size[4] **Rput** tag[2] fd[2] count[4] qid[13] mtime[4]

- Walk + Wstat + Write + Create
- Mode means which ones or this are done
- ODATA means Write
- OSTAT means Wstat
- OCREATE means Create
- You can create, write and change permissions in one RPC
- Some parts of the RPC works in directories, others not:
  - If CREATE and DMDIR, ODATA is not allowed

Get

size[4] **Tget** tag[2] path[s] fd[2] mode[2] nmsgs[2] offset[8] count[4]

size[4] **Rget** tag[2] fd[2] mode[2] stat[n] count[4] data[count]

- Walk + Rstat
- OMORE when sent means “Keep the fd”
- OMORE in Rget means not EOF yet
- Nmsgs means the maximum number of messages I allow you to respond me with for one request. Ignored for directories (a reason why it is important to have OMORE on replies.

Remove

- Walk + remove
- As expected

Implementation

- Ofs: Styx server - Op client with caching
  - 3209 lines, complicated because of cache
- Oxport: Op server - Fscalls
  - 589 lines
- Op library to serialize calls and so
  - 726 lines

Cache

- Cache important for metadata (brings whole directories)
- Cache can be told to bring whole files (for BW optimization)
- Invents intermediate directories
- Files with zero length and first read not cached.
- Cached data is valid for coherency window then checked
- Writes
  - Directories are write-through
  - Not offset zero and filling an entire msg are done async
  - Others, sync to report errors to application

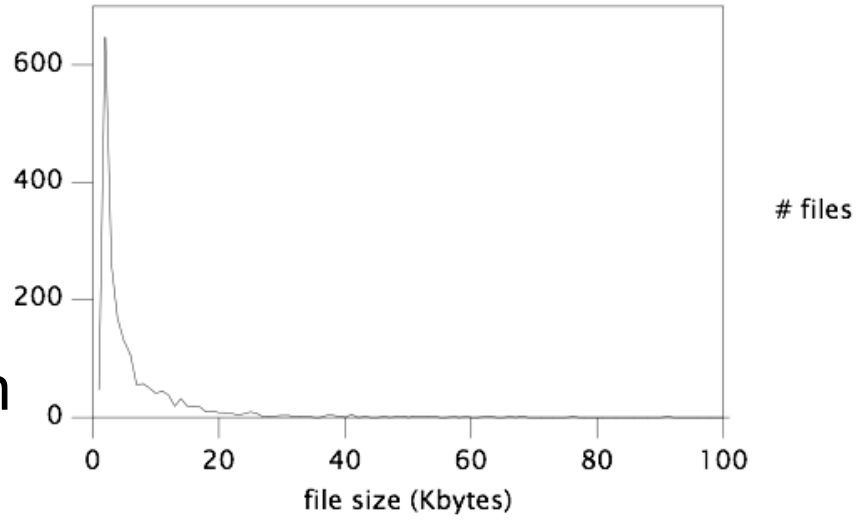
Fids, fd

- Multiple fids are mapped to one fd in the server, one fd for reading another for writing
- Tclunk closes an fd (guaranteed)
- No clone files
- The cache is not aware of OEXCL, does not work

Measurements

Protocol	lc	mk clean	mk
Styx	2.314	30.6	87.5
Op (1s)	0.76	2.93	34.02
Op (2s)	0.142	2.58	30.37

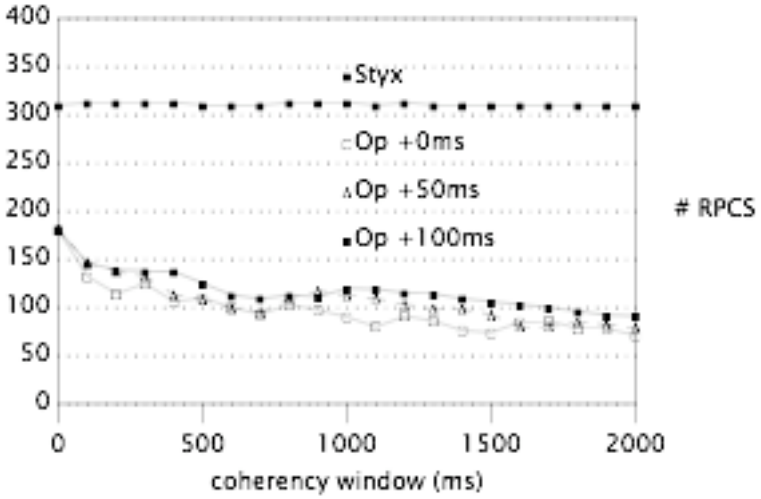
Times (s) for completion
85ms RTT 1s and 2s coherency win



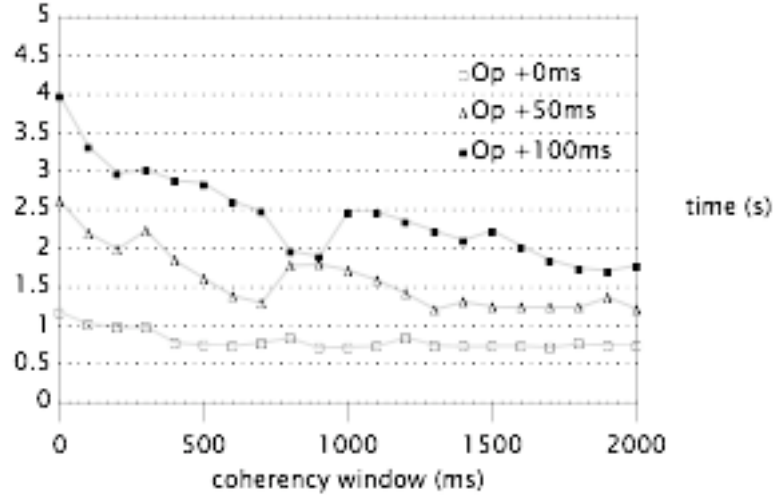
Filesize in inferno

Protocol	direction	lc	mk clean	mk
Styx	*read*	97633.19	8677.66	104143.81
Styx	*write*	na	2928.37	5933.42
Op (1s)	*read*	4353804.98	50380.33	1163658.94
Op (1s)	*write*	na	2747.05	16228.07

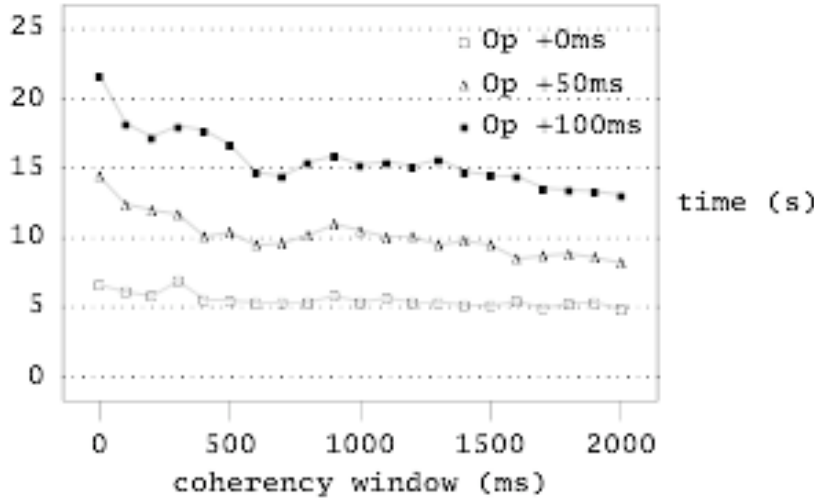
Transfer rate (Kb/s)
85ms RTT, 1s and 2s
Coherency win



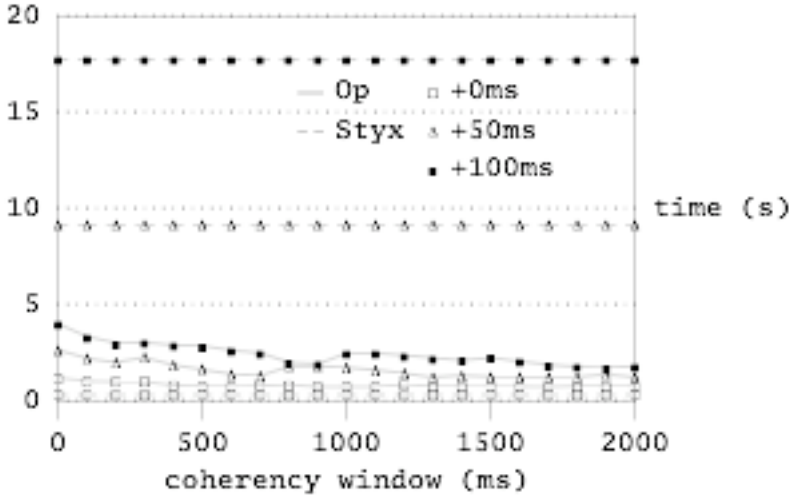
#RPCs for different coherency win



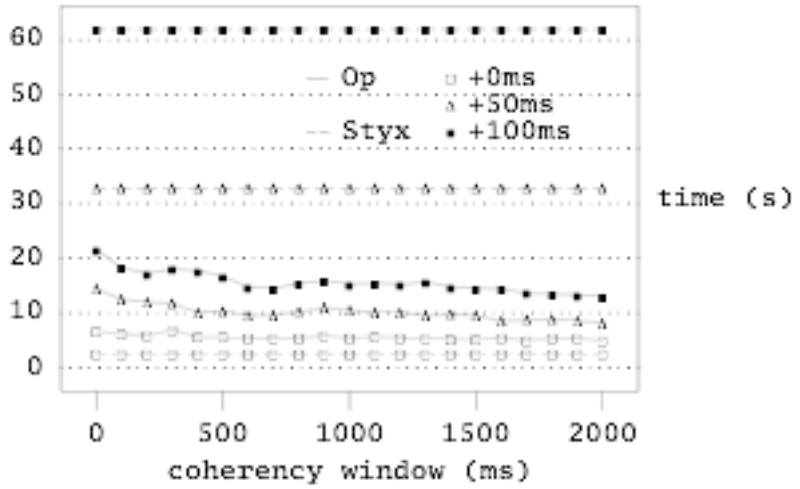
Time for mk clean on different Coherency win



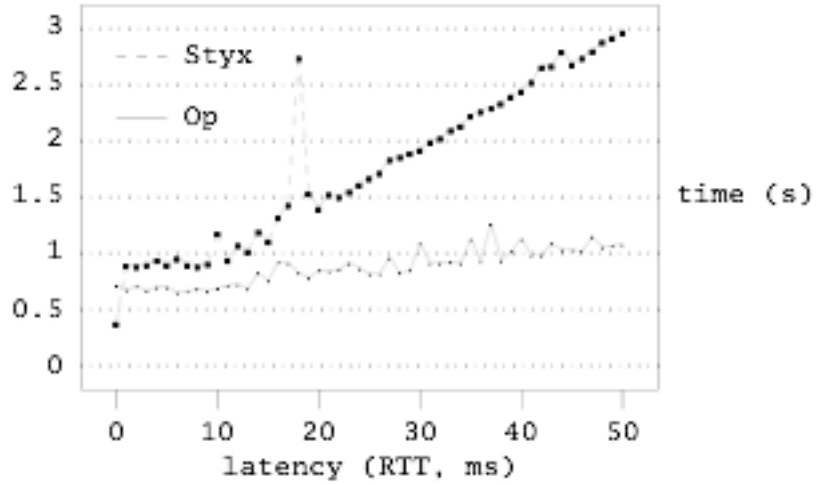
Time for mk , different coherency win



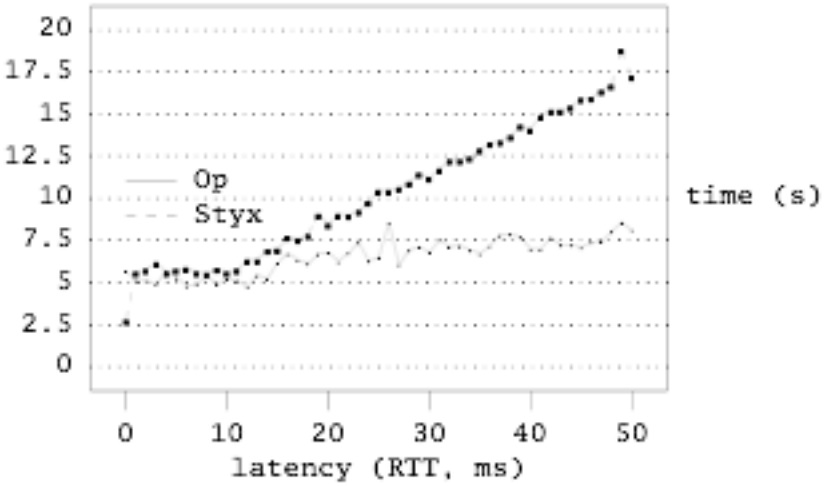
Compared time for mk On different coherency networks



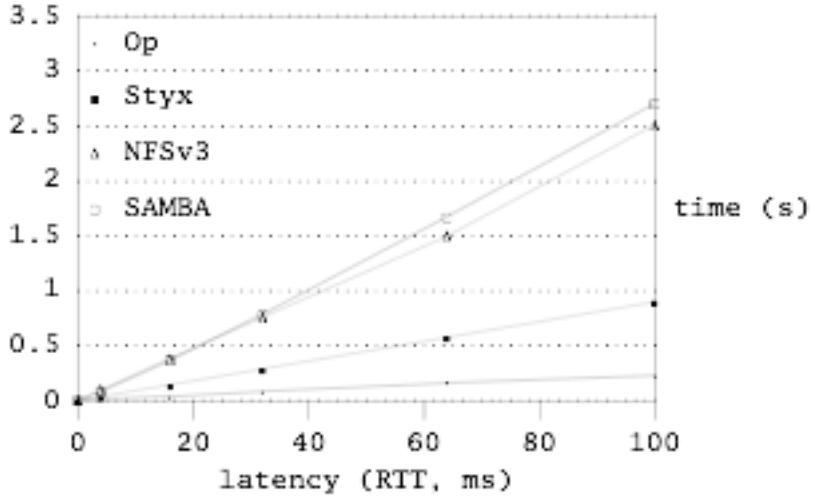
Compared time for mk, different windows and latencies



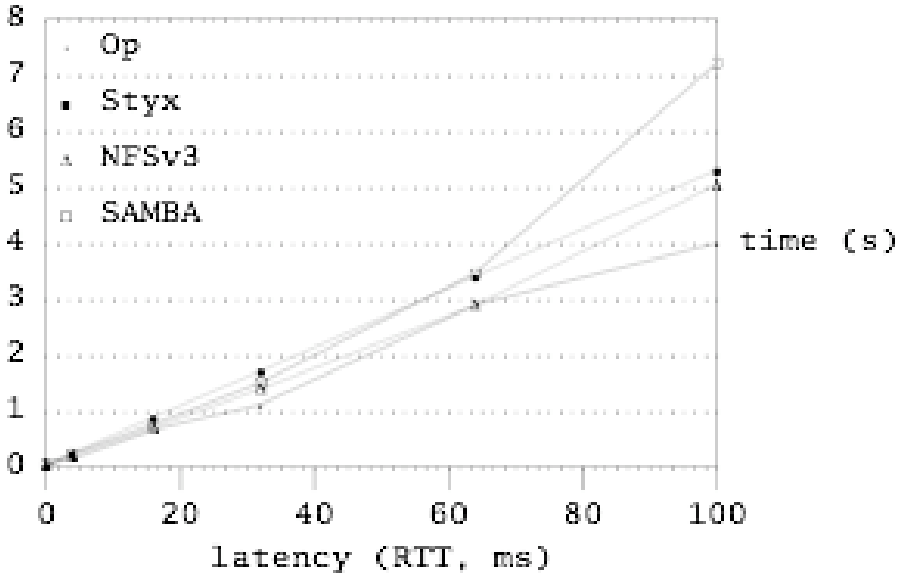
Compared time Mk clean



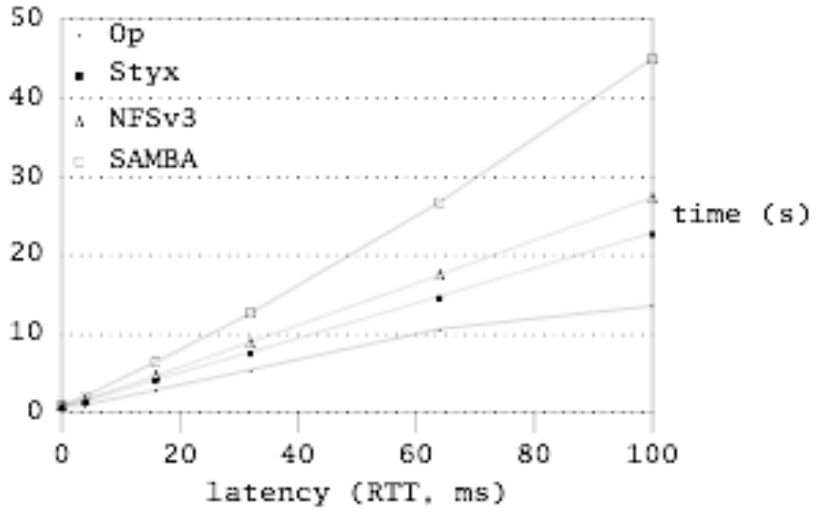
Compared time mk



Compared times To run lc



Compared mk clean



Compared running mk

Related work

- Require too much modification (TCP/IP…):
  - Riverbed (also doesn´t work with synth).
- Optimized for BW (not latency)
  - LBFS: hashes, maintaining them is too much latency
  - CFS: stats for each file RTT for each file
- Don´t work with synth:
  - CIFS: Cisco WAFS, Packeteer
  - NFS v4
- Sync fs, no point in exporting fs then (tar?).
  - Disksites, Availl
- Rangboom?
  - Used to have problems when we tried it
  - Apparently caches metadata

Q/A

- Can be downloaded separately from the octopus at http://www.lsub.org/ls/octopus.html