afsload

Real-world AFS testing

Andrew Deason <adeason@sinenomine.net>
Sine Nomine Associates

2011 AFS & Kerberos Best Practices Workshop

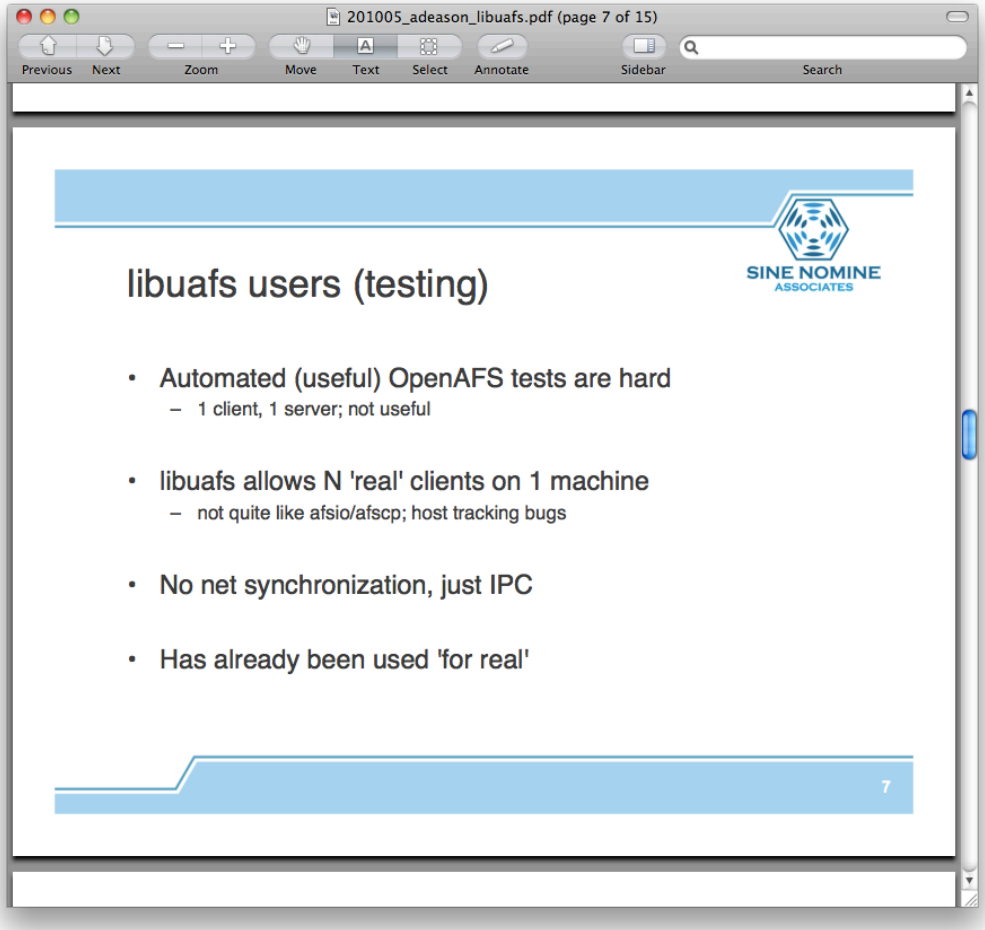
Legal Notices

Background

Let's go back to 2010 for a moment...

Background

- Remember this?
- I wasn't lying; this is afsload!



afsload, what is it?

SINE NOMINE ASSOCIATES

- Perl, libuafs, MPI
  - Parallel::MPI::Simple
  - OpenMPI
- Controls a bunch of libuafs pseudo-clients
- Miniature test framework

afsload, what is it NOT?

SINE NOMINE ASSOCIATES

- An actual test framework
- Use it inside another framework
  - like $srcdir/tests/

afsload, why?

- Single client testing is of limited value
- "Real" multiple-client testing is hard / costly
- N clients on 1 machine
- Expose races
- Simulate real-world scenarios
  - Moving IPs / NATs
  - IP conflicts

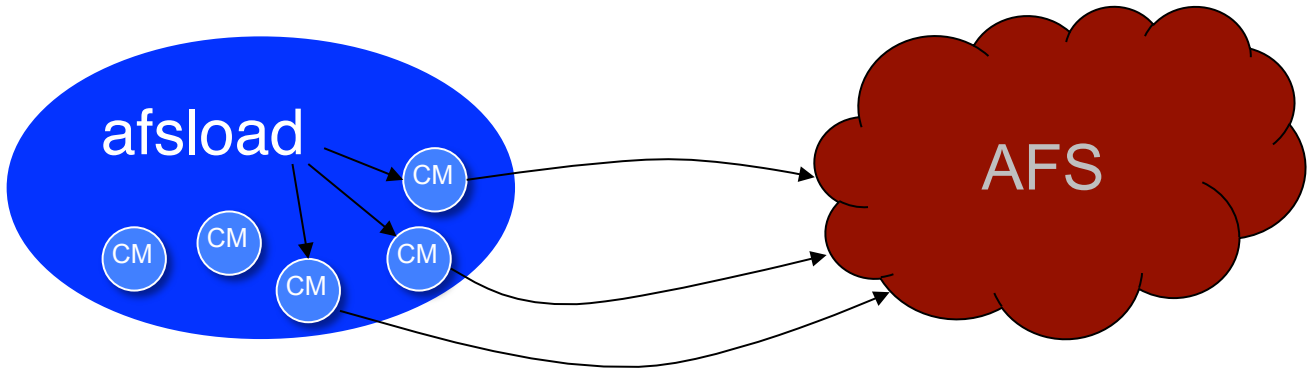
How many is N?

SINE NOMINE ASSOCIATES

- How many nodes?
  - hundreds
- Can scale as high as the machine
- Maybe multiple machines

afsload, how?

- Spawn N libuafs-using processes
- Config file says do X, Y, Z
- "Director" node directs nodes via MPI



afsload, how?

SINE NOMINE ASSOCIATES

- Results gathered via MPI, "director" checks results

```
$ afsload -p 20 -t test.conf
1..7
ok 1 - Step 1: chdir
ok 2 - Step 2: create file
ok 3 - Step 3: read newly created file
ok 4 - Step 4: truncwrite
ok 5 - Step 5: read after overwrite
ok 6 - Step 6: unlink
ok 7 - Step 7: fail opening after unlink
$
```

Configuration

- Parallel actions specified between serial "steps"

```
nodeconfig
 node * afsconfig "-fakestat -cachedir /tmp/afsload/cache.$RANK"
 node * logfile "/tmp/afsload/log.$RANK"
step name chdir
 node * chdir "/afs/.localcell/afsload"
step name "create file"
 node 0 creat foo "foo contents"
step name "read newly created file"
 node * read foo "foo contents"
step name truncwrite
 node 1 truncwrite foo "different contents"
step name "read after overwrite"
 node * read foo "different contents"
step name unlink
 node 0 unlink foo
step name "fail opening after unlink"
 node * fail ENOENT access_r foo
```

Configuration

SINE NOMINE ASSOCIATES

- Cache and logs need set up
  - Each node gets its own cache, its own (optional) log
- Most configuration same as afsd (afsconfig)
- Existing AFS server-side infrastructure

Configuration

SINE NOMINE ASSOCIATES

- Why not just plain Perl?
- Syntax is a little inflexible BUT...
- Config file "actions" simple Perl modules
  - Easy to add more, easy to chain together

Current Limitations

- Unauthenticated (libuafs)
- No IP changes, administrative actions, etc.
- No timing data or "speed test" capabilities

Availability

- Not anywhere yet
- Requires libuafs Perl bindings (AFS::ukernel)
- Watch the OpenAFS newsletter
- May go in src/libuafs/afsload ?

Questions?

Thanks!

Andrew Deason
<adeason@sinenomine.net>