



NRAOrsync (Transfer Plugin)

K. Scott Rowe

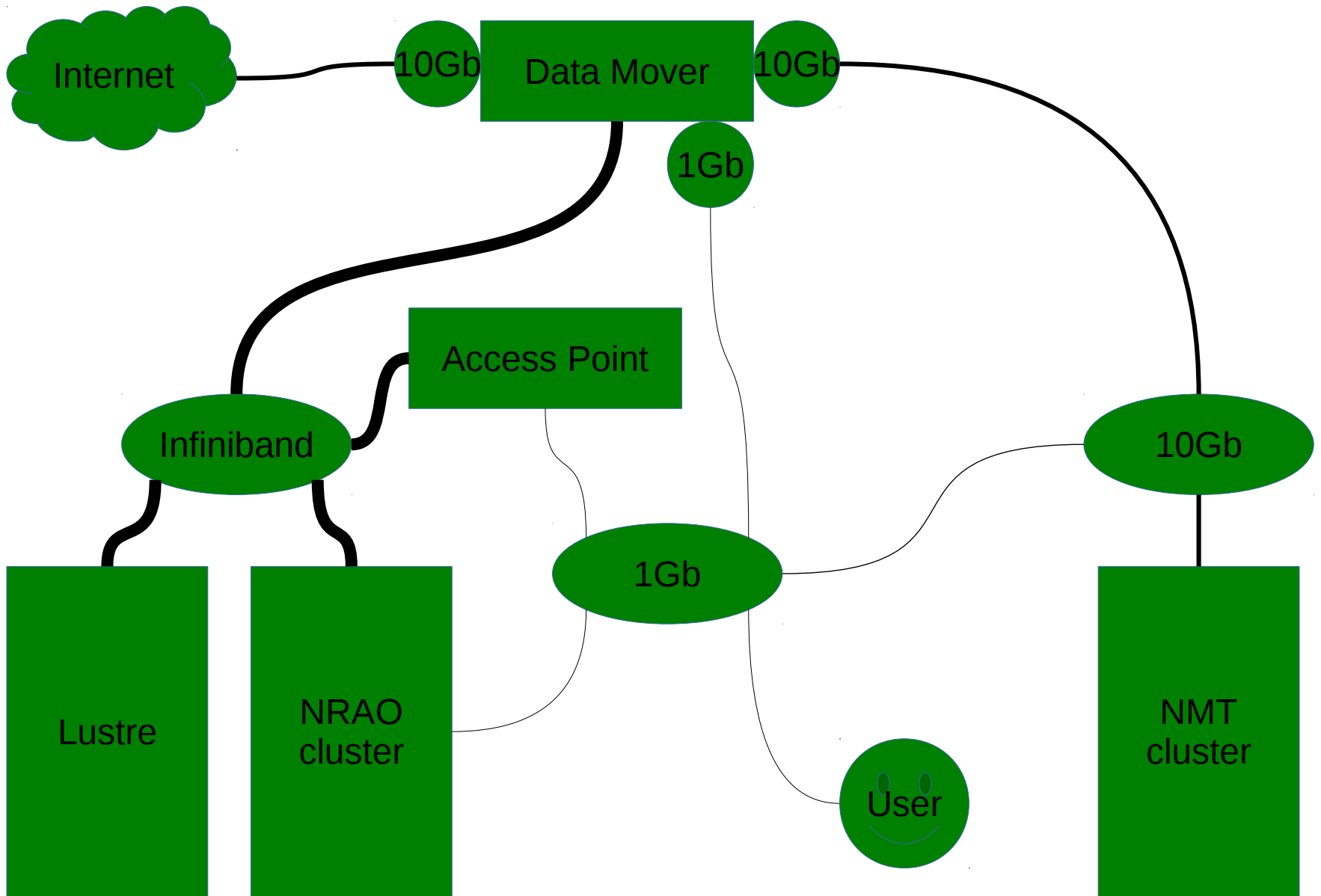


Abstract

- File Transfer Mechanism (FTM)
 - Copy files for you. Downloads input data to and uploads output data from execution host. Allows for local storage.
 - `should_transfer_files = YES`
 - `transfer_input_files`, `transfer_output_files`
- File Transfer Plugins
 - You can write a plugin to copy files the way you want
 - `/usr/libexec/condor/*_plugin*` like `box`, `gdrive`, `onedrive`
 - HTCondor runs all plugins with `-classad` to see what `SupportedMethods` they handle (e.g. `nraorsync`) which equate to schemes (e.g. `nraorsync://`). See example later.

Why a plugin?

- Want to process against our local NVMe drives
- Our input data can be 100s of GB. Output larger
- We tend to launch jobs in batches
- FTM doesn't support choosing a NIC
 - We have 1Gb, 40Gb Infiniband, 10Gb
- FTM uses the Access Point for data transfer
 - All of this means our Access Point NIC can get flooded
- Want rsync so only what has changed is uploaded
- So we created a plugin and a Data Mover host



How does nraorsync work?

- `transfer_input_files = $ENV(HOME)/.ssh/condor_transfer,`
`nraorsync://$ENV(PWD)/inputdata`
- **NRAO_TRANSFER_HOST** (custom attribute)
 - LOCAL (NRAO cluster)
 - `rsync /src/path /dst/path`
 - gibson-10g2 (NMT cluster)
 - `rsync` using gibson-10g2
 - Undefined (site outside our control)
 - `rsync` using gibson over Internet
- `output_destination = nraorsync://$ENV(PWD)`

Implementation problems

- `transfer_output_files`
 - If `transfer_output_files` includes a directory
 - FTM walks the directory and uploads each file
 - This can break `rsync` because destination directory may not exist
 - E.g. `rsync output/file gibbon:/output/file` (`/output` does not exist)
 - So we created `+nrao_output_files` to replace `transfer_output_files`
 - If `transfer_output_files` is undefined, FTM uploads all new files, not dirs, since the job started
 - Not what we want
 - So we set `transfer_output_files = _condor_stdout`
 - Setting output in submit description file creates `_condor_stdout` on Execute Point

More problems

- `_condor_stdout` and `_condor_stderr`
 - FTM normally remaps both, but not if using a plugin
 - Plugin remaps by asking schedd for Out and Err
- `NOT_RESPONDING_TIMEOUT = 86400`
 - `condor_starter` forks on download but blocks on upload
 - `condor_shadow` kills job after 3,600 seconds by default
- `Condor_chirp` doesn't work on upload
 - Requires `+WantIOProxy = True`
 - `condor_starter` handles chirp calls but blocks on upload, so deadlock

Appendix

```
executable = small.sh
arguments = "27"
output = stdout.$(ClusterId).log
error = stderr.$(ClusterId).log
log = condor.$(ClusterId).log
should_transfer_files = YES
transfer_input_files = $ENV(HOME)/.ssh/condor_transfer, nraorsync://$ENV(PWD)/input
transfer_output_files = _condor_stdout
+nrao_output_files = "software output"
output_destination = nraorsync://$ENV(PWD)
+WantIOProxy = True
queue
```

```
#!/bin/sh
date > date
echo "error" > /dev/stderr
mv input output
sleep ${1}
```

https://github.com/htcondor/htcondor/blob/master/src/condor_examples/

`git clone https://gitlab.nrao.edu/krowe/condor_transfer_plugin.git`



www.nrao.edu
science.nrao.edu
public.nrao.edu

*The National Radio Astronomy Observatory is a facility of the
National Science Foundation
operated under cooperative agreement by Associated Universities,
Inc.*