A photograph of Earth's horizon from space, showing a vibrant blue atmosphere, white clouds, and dark landmasses below.

THE FRAMEWORK PULSAR PIPELINE

**LOFAR**  
Low Frequency Array

pulp.py

# FRAMEWORK PULSAR PIPELINE

- “PULP” is the framework pulsar pipeline
  - › Current release under SVN
    - @ \$LOFARSOFT/src/Pulsar/pipeline:

```
drwxr-xr-x  9 <username> staff  306 Nov 1 11:17 .svn
-rw-r--r--  1 <username> staff 1726 Nov 1 11:15 pipeline.cfg
-rwxr-xr-x  1 <username> staff 2341 Nov 1 11:15 pulp.py*
drwxr-xr-x  5 <username> staff  170 Nov 1 11:15 recipes
drwxr-xr-x 14 <username> staff  476 Nov 1 11:15 support
-rw-r--r--  1 <username> staff  658 Nov 1 11:15 tasks.cfg
```

# FRAMEWORK PULSAR PIPELINE

- Current Pulp package: 9 recipes, 8 support modules, 3400+ loc

Head Node Recipes	Compute Node Recipes	Support Modules
buildPulsArch		buildRSPS, RSPList
bf2presto	bf2presto	bf2Pars
buildRSPAll		fullRSP
prepareInf		prepareInfFiles
prefold	prefold	prefoldPars
rfiplot	rfiplot	rfiDirectories
In dev:		pulpEnv
flyseye, coherentstokes, tarBall, rfiAssemble	...	?

# FRAMEWORK PULSAR PIPELINE

## ○ Configuration

### › Environment:

- Define \$LOFARSOFT, \$LOFARROOT (=/opt/LofIm/daily/lofar)
- Make “pipeline\_runtime/” directory (arbitrary location)
- source \${LOFARSOFT}/devel\_common/scripts/init.sh
- Use LofIm
  - Define \$TEMPO, \$PRESTO
- PYTHONPATH include
  - PYTHONPATH=/opt/pipeline/dependencies/lib/python2.5/site-packages:\
  - /opt/pipeline/framework/lib/python2.5/site-packages:\
  - /opt/LofIm/daily/pyrap/lib:\
  - \${LOFARROOT}/lib/python2.5/site-packages:\
  - /opt/pythonlibs/lib/python/site-packages:\
  - \${LOFARSOFT}/src/Pulsar/pipeline/recipes/master:\
  - \${LOFARSOFT}/src/Pulsar/pipeline/recipes/nodes:\
  - \${LOFARSOFT}/src/Pulsar/pipeline/support
- PATH include
  - \${LOFARSOFT}/release/bin:\
  - \${LOFARSOFT}/release/share/pulsar/bin:\
  - \${LOFARSOFT}/src/Pulsar/pipeline:\
  - /opt/LofIm/daily/casarest/bin:\
  - /opt/pipeline/dependencies/bin:\
  - /opt/LofIm/daily/askapsoft/bin:\
  - /opt/scripts:\${PATH}

# FRAMEWORK PULSAR PIPELINE

- Configuration ...

- › Framework configuration files (delivered with svn download)
  - Pipeline.cfg
    - Framework paths definition file
  - Tasks.cfg
    - Task definitions file
  - Sub[n].clusterdesc
    - Sub-cluster defintion files
    - Templates in \$LOFARROOT/share
  - task.furl
  - multiengine.furl

N.B. \* No shell interpolation \*

# FRAMEWORK PULSAR PIPELINE

## ○ Configuration cont'd ...

**Pipeline.cfg**, example

```
[DEFAULT]
runtime_directory = /home/kanderson/LOFAR/pipeline_runtime
recipe_directories= [.., /home/kanderson/LOFAR/lofarsoft/src/Pulsar/pipeline/recipes, /opt/pipeline/recipes]
lofarroot          = /opt/LofIm/daily/Thu/lofar
task_files         = [/home/kanderson/LOFAR/lofarsoft/src/Pulsar/pipeline/tasks.cfg]           ← tasks.cfg
default_working_directory = /data/scratch/kanderson

[layout]
job_directory      = %(runtime_directory)s/jobs/%(job_name)s
log_directory      = %(job_directory)s/logs/%(start_time)s
vds_directory      = %(job_directory)s/vds
parset_directory   = %(job_directory)s/parses
results_directory  = %(job_directory)s/results/%(start_time)s
[cluster]
clustername        = pulsar
clusterdesc        = /home/kanderson/LOFAR/pipeline_runtime/sub5.clusterdesc           ← cluster definition
#clusterdesc       = /home/kanderson/LOFAR/pipeline_runtime/sub6.clusterdesc
task_furl          = %(runtime_directory)s/task.furl
multiengine_furl   = %(runtime_directory)s/multiengine.furl                           ← task.furl file
← multiengine.furl file
[deploy]
script_path         = /opt/pipeline/framework/bin
controller_ppath   = /home/kanderson/LOFAR/lofarsoft/src/Pulsar/pipeline/support:/opt/pipeline/dependencies/lib/
                  python2.5/site-packages:/opt/pipeline/framework/lib/python2.5/site-packages
engine_ppath        = /home/kanderson/LOFAR/lofarsoft/src/Pulsar/pipeline/recipes/master:/home/kanderson/LOFAR/
                  lofarsoft/src/Pulsar/pipeline/recipes/nodes:/home/kanderson/LOFAR/lofarsoft/src/Pulsar/pipeline/support:/opt/
                  pipeline/dependencies/lib/python2.5/site-packages:/opt/pipeline/framework/lib/python2.5/site-packages:/opt/
                  LofIm/daily/pyrap/lib:/opt/LofIm/daily/lofar/lib/python2.5/site-packages:/opt/pythonlibs/lib/python/site-
                  packages:
engine_lpath        = /opt/pipeline/dependencies/lib:/opt/LofIm/daily/pyrap/lib:/opt/LofIm/daily/casacore/lib:/opt/
                  LofIm/daily/lofar/lib:/opt/wcslib/lib:/opt/hdf5/lib
```

# FRAMEWORK PULSAR PIPELINE

- Tasks.cfg:

```
[buildPulsArch]
recipe      = buildPulsArch
filefactor  = 8

[bf2presto]
recipe      = bf2presto
executable  = /home/kanderson/LOFAR/lofarsoft/release/share/pulsar/bin/bf2presto8
filefactor  = 8
collapse    = False
nsigmas     = 7

[buildRSPAll]
recipe      = buildRSPAll
filefactor  = 8

[prepareInf]
recipe      = prepareInf
filefactor  = 8

[prepfold]
recipe      = prepfold
executable  = /home/kanderson/LOFAR/lofarsoft/release/share/pulsar/bin/prepfold
filefactor  = 8
nopolsearch = True
nperstokes  = 256
noxwin      = True
fine        = True

[rfiplot]
recipe      = rfiplot
executable  = /home/kanderson/LOFAR/lofarsoft/release/share/pulsar/bin/subdyn.py
filefactor  = 8
```

# FRAMEWORK PULSAR PIPELINE

- Clusterdesc

- › Clusterdesc template files in \$LOFARROOT/share

---

```
ClusterName = sub5

# Storage nodes.
Storage.Nodes = [ lse013..15 ]
Storage.LocalDisks = [ /data1..4 ]

# Compute nodes.
Compute.Nodes = [ lce037..45 ]
Compute.RemoteDisks = [ /net/sub1/lse013..15/data1..4 ]
Compute.RemoteFileSys = [ /lse013..15:/data1..4 ]
Compute.LocalDisks = [ /data ]

# Head nodes.
Head.Nodes = [ lfe001..2 ]
Head.LocalDisks = [ /data ]
```

# FRAMEWORK PULSAR PIPELINE

- Pulsar group has a legacy alias clusterdesc file,  
pulsar.clusterdesc:

```
ClusterName = Pulsar
# Directory is the same as the parent.
SubClusters = [sub5.clusterdesc]
```

# FRAMEWORK PULSAR PIPELINE

- The parallel pipeline framework

- › A series, possibly complex, of “recipes” are executed, much like a job queue.
- › Executed recipes create push-pull connections between the controlling “head node” and designated “compute nodes” of the cluster
- › The recipes create jobs based upon the number of input subbands, definition, and argument parameters
- › Assigns recipe jobs to available compute nodes
- › Compute node recipe receives control, computes
- › Completion of compute node recipe returns control to head node
- › Next recipe
- › Nominal processing:
  - 248 subband incoherentstokes
  - 4 or 8 “RSP” splits
  - 4 jobs, 62 subbands / job or
  - 8 jobs, 31 subbands/job

# FRAMEWORK PULSAR PIPELINE

- The pulsar pipeline definition:

```
def pipeline_logic(self):  
    obsid      = self.inputs['obsid']  
    pulsar     = self.inputs['pulsar']  
    filefactor = self.inputs['filefactor']  
    arch       = self.inputs['arch']  
  
    with log_time(self.logger):  
        with ipython_cluster(self.config, self.logger):  
            if filefactor == 1:  
                self.run_task("buildPulsArch", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("bf2presto", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("prepareInf", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("prefold", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("rfiplot", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
            else:  
                self.run_task("buildPulsArch", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("bf2presto", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("buildRSPAll", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("prepareInf", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("prefold", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
                self.run_task("rfiplot", obsid=obsid, pulsar=pulsar, filefactor=filefactor, arch=arch)  
...  
...
```

# FRAMEWORK PULSAR PIPELINE

- Built-in processing:
  - Parallel processing of all subband splits, (i.e. +RSPA)
    - Capacity for parallel processing n splits for m obsids
  - “all” subbands processed automatically
    - “all” subband group handled as just another job
    - Except: <filefactor> == 1
  - Implements script switch “-rfi”
  - Current full cycle “incoherentstokes” processing
    - Build output path, [make or] read subband data locator list, subband split directories
    - bf2presto
      - Converts “incoherentstokes” [others] to data fold (prepfold) PRESTO format
      - Performance IO limited on single output path
    - Data folding preparation
      - Prepfold .inf files
    - Prepfold processing
    - Now, a simple, straight pipe
      - Recipe 1 → recipe2 → recipe3 →...

# FRAMEWORK PULSAR PIPELINE

(pulp.py, beta, delivered 13.10.2010)

- ◎ Current usage:

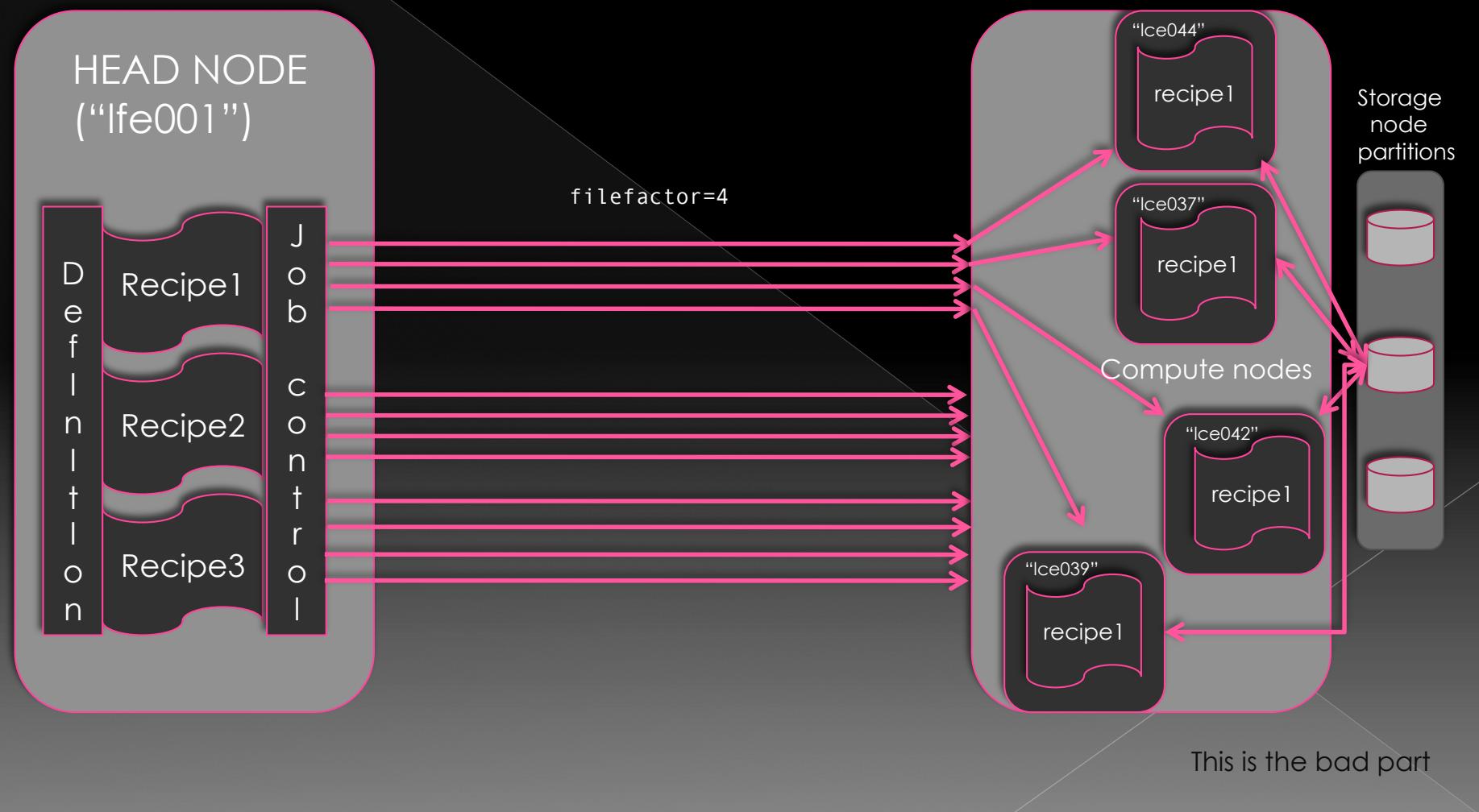
```
$ pulp.py -d --obsid=L<yyyy>_<nnnnn[n...]> --job-name=<job_name> --arch=<archSelection> \
--pulsar=<pulsar> [--filefactor=<m>]
```

where,

--obsid	= observation identifier <str>
	eg., --obsid=L2010_06160
--arch	= selected pulsar archive <str>
	eg., --arch=arch134 (PULP_ARCHIVE @ lse013/data4)
--pulsar	= name of pulsar (soon: or csv list of pulsar names).
	--pulsar=B0919+06
--filefactor	= subband splitting factor, <int>
	eg., --filefaactor=4
	optional user specification (range, 1-248)
	default = 8
-d	= debug level, pipeline verbosity switch, full logging -- framework argument
-j,--job-name=	arbitrary user-defined name for the processing job -- framework argument

# FRAMEWORK PULSAR PIPELINE

Nominal:



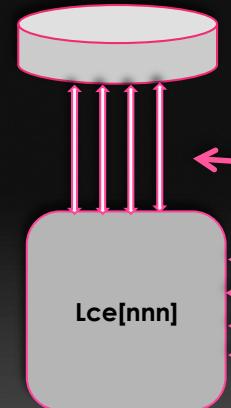
# FRAMEWORK PULSAR PIPELINE

Single node operation:

- All paths are common and shared.
- all jobs see the same 'scratch disk'

Local mount

/data/scratch



Four jobs / single node  
Either path, okay



Read only  
.../lse015/data1/



Read only  
.../lse015/data2/



Read only  
.../lse015/data3/



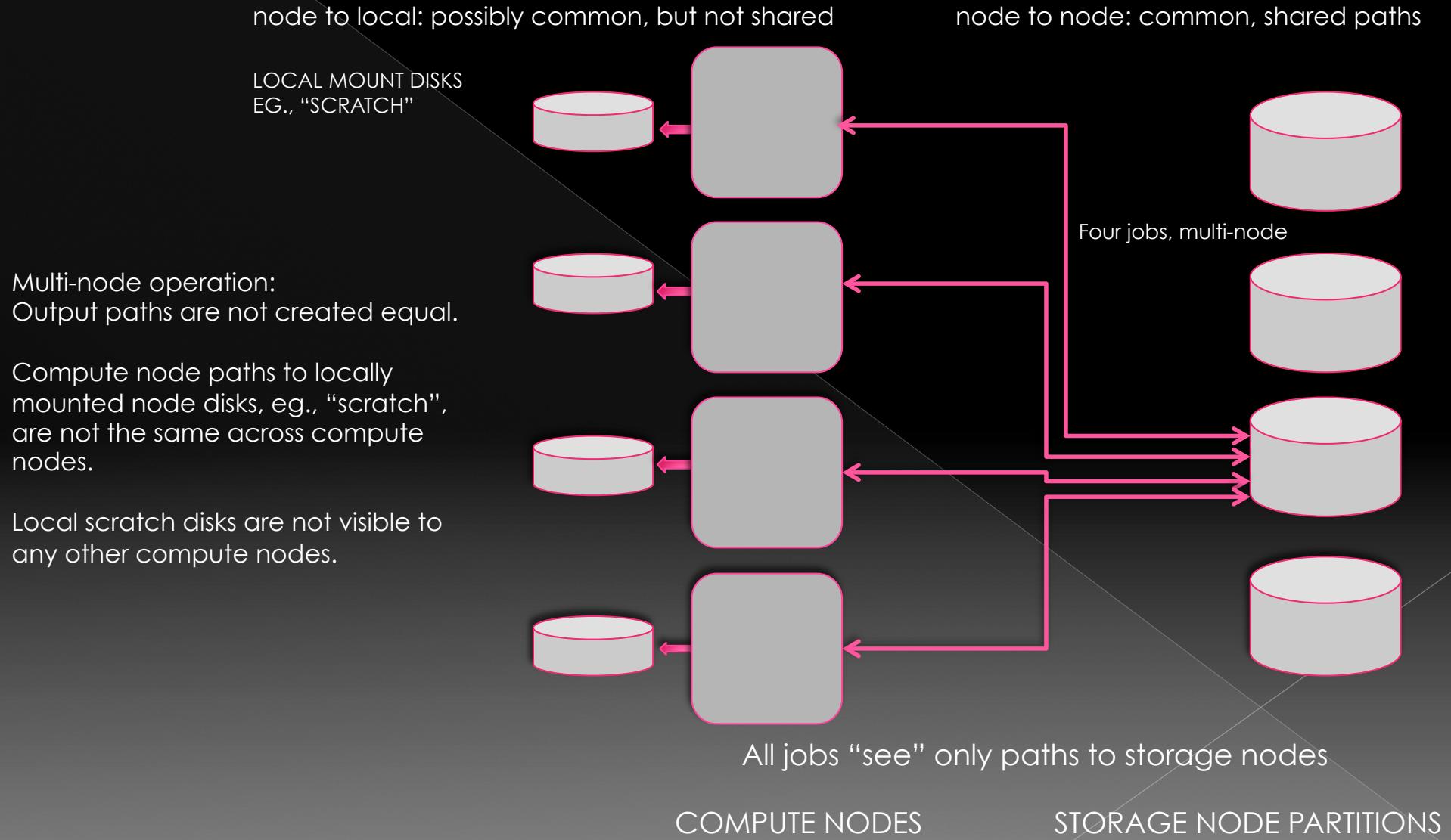
Writable! drwxrwxr--  
.../lse015/data4/

All jobs can see paths to storage nodes and local scratch

COMPUTE NODE

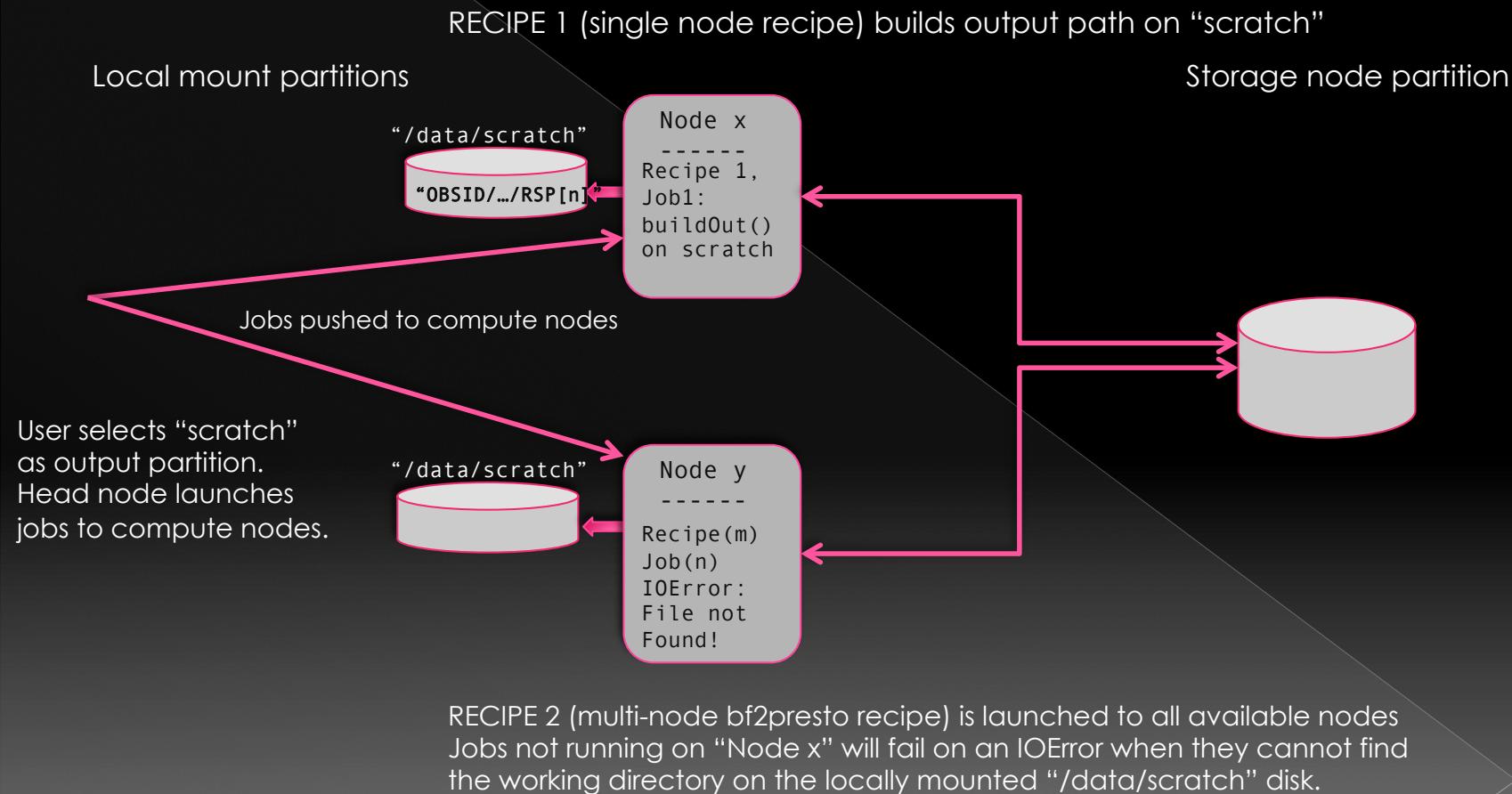
STORAGE NODE PARTITIONS

# FRAMEWORK PULSAR PIPELINE



# FRAMEWORK PULSAR PIPELINE

Example,



# FRAMEWORK PULSAR PIPELINE

- What does this mean?
  - Storage node paths to all compute nodes are common and shared.
  - Local mount paths to all compute nodes are NOT common and shared.
    - Duh
  - Only writable shared paths (storage nodes) can be specified as “path to output”
    - Unless (!) the framework pipeline operation is **restricted to one node**
  - Therefore:
    - Locally mounted disks are not usable for the pipeline framework.
    - A user selected output path of ‘scratch’ or any other locally mounted disk will break the pipeline.

# FRAMEWORK PULSAR PIPELINE

- What does that leave for selectable output paths?
  - > Currently, the pulsar group (possibly others) have write permission only on data4 partitions of storage nodes attached to subnet 5 and subnet 6.
  - > Pre-defined, user-selectable Pulsar Archives (--arch=) on subnet5 and subnet6:
    - 'arch134' = /net/sub5/lse013/data4/PULP\_ARCHIVE
    - 'arch144' = /net/sub5/lse014/data4/PULP\_ARCHIVE
    - 'arch154' = /net/sub5/lse015/data4/PULP\_ARCHIVE
    - 'arch164' = /net/sub6/lse016/data4/PULP\_ARCHIVE
    - 'arch174' = /net/sub6/lse017/data4/PULP\_ARCHIVE
    - 'arch184' = /net/sub6/lse018/data4/PULP\_ARCHIVE
  - > Once a node is selected, a “user” can set an output path beyond this
    - But do they really need to do this?
    - No known examples of execution with extended output paths
  - > Nominal operation here:
    - Output written on node's “/data4” partition + user selected “archive” directory.
    - i.e.
      - Output → /net/sub5/lse015/data4/PULP\_ARCHIVE/OBSID

# FRAMEWORK PULSAR PIPELINE

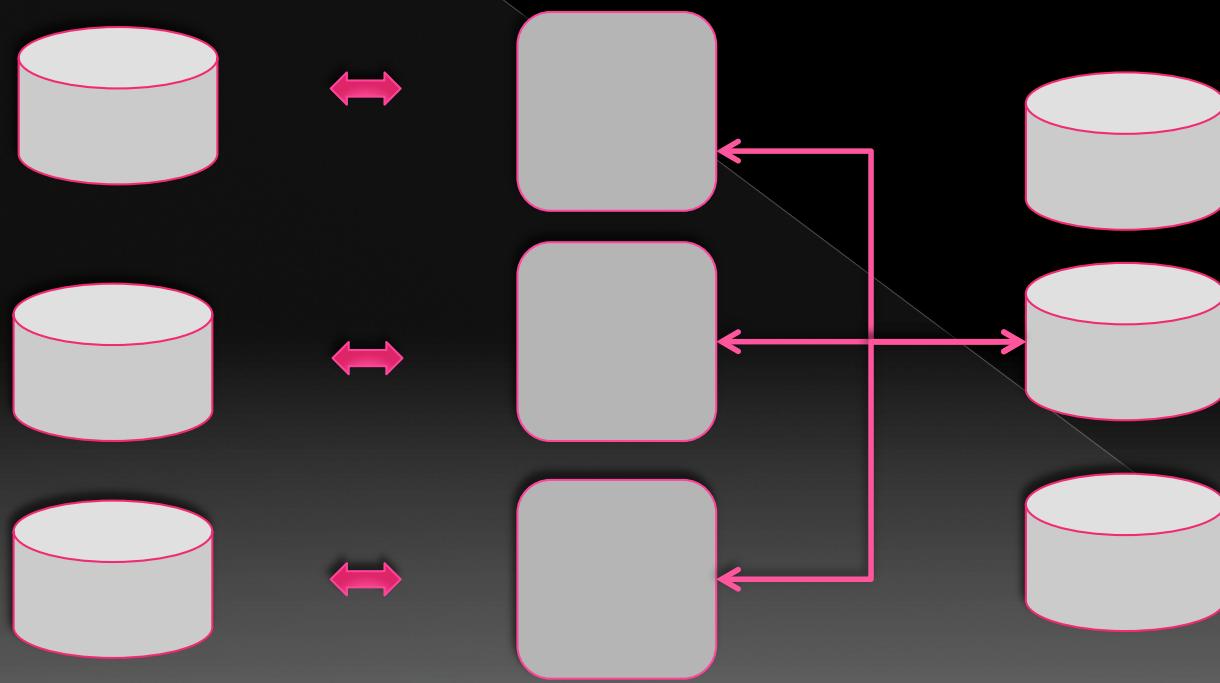
- Bottom line:
  - Presently, output paths available to the framework pipeline are limited to the writable 'data4' partitions of subnet 5 and subnet 6, which total six.
  - Scratch disks, or any local disk, will not function under current "mimic script" requirements.

# FRAMEWORK PULSAR PIPELINE

LOCAL MOUNT DISKS  
EG., "SCRATCH"

PATHS TO SCRATCH ARE NOT THE SAME  
ON COMPUTE NODES.

A user selected output path of /data/scratch, or **any** locally mounted disk, will break the pipeline.

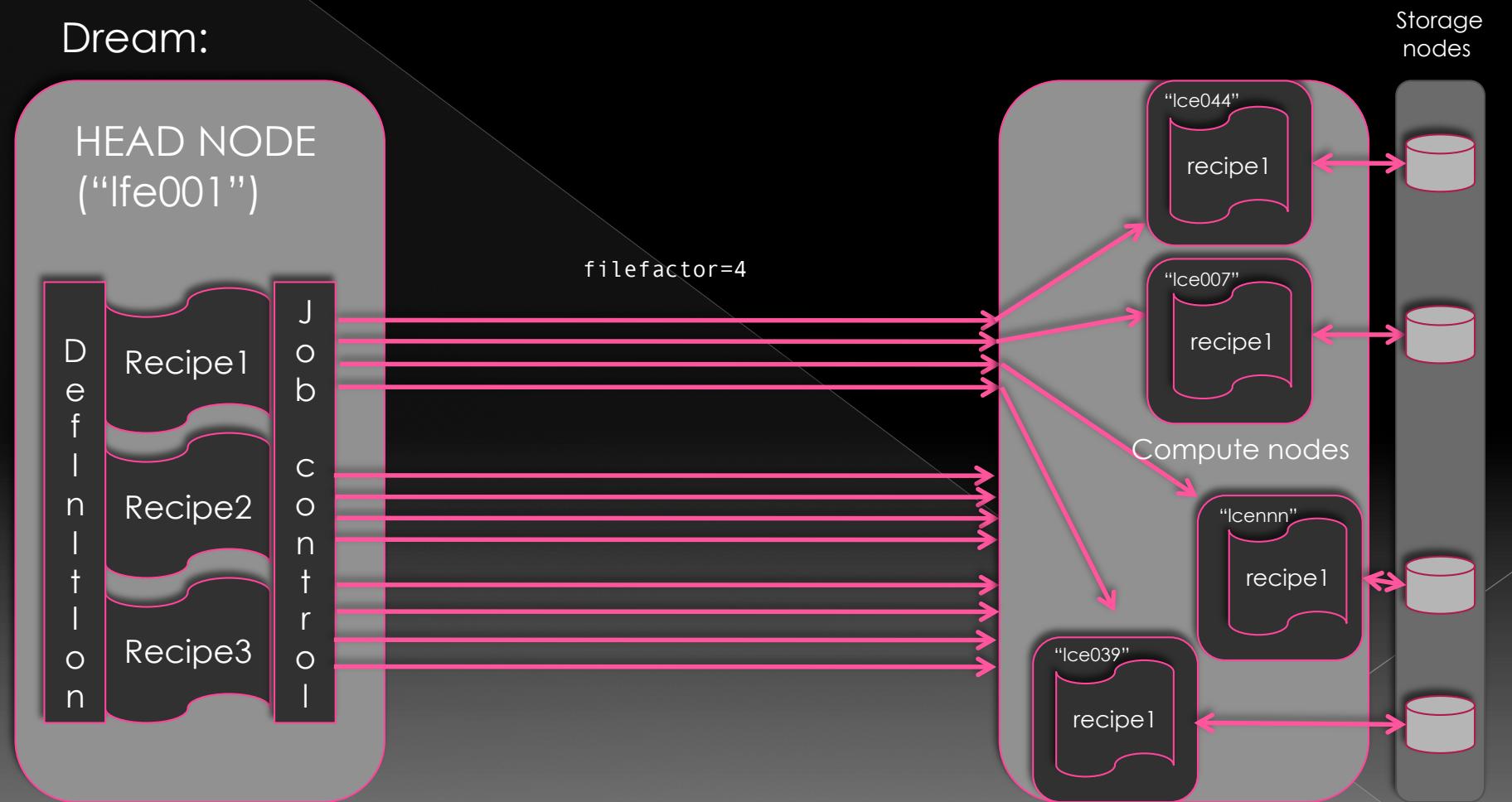


COMUTE NODES

STORAGE NODE PARTITIONS

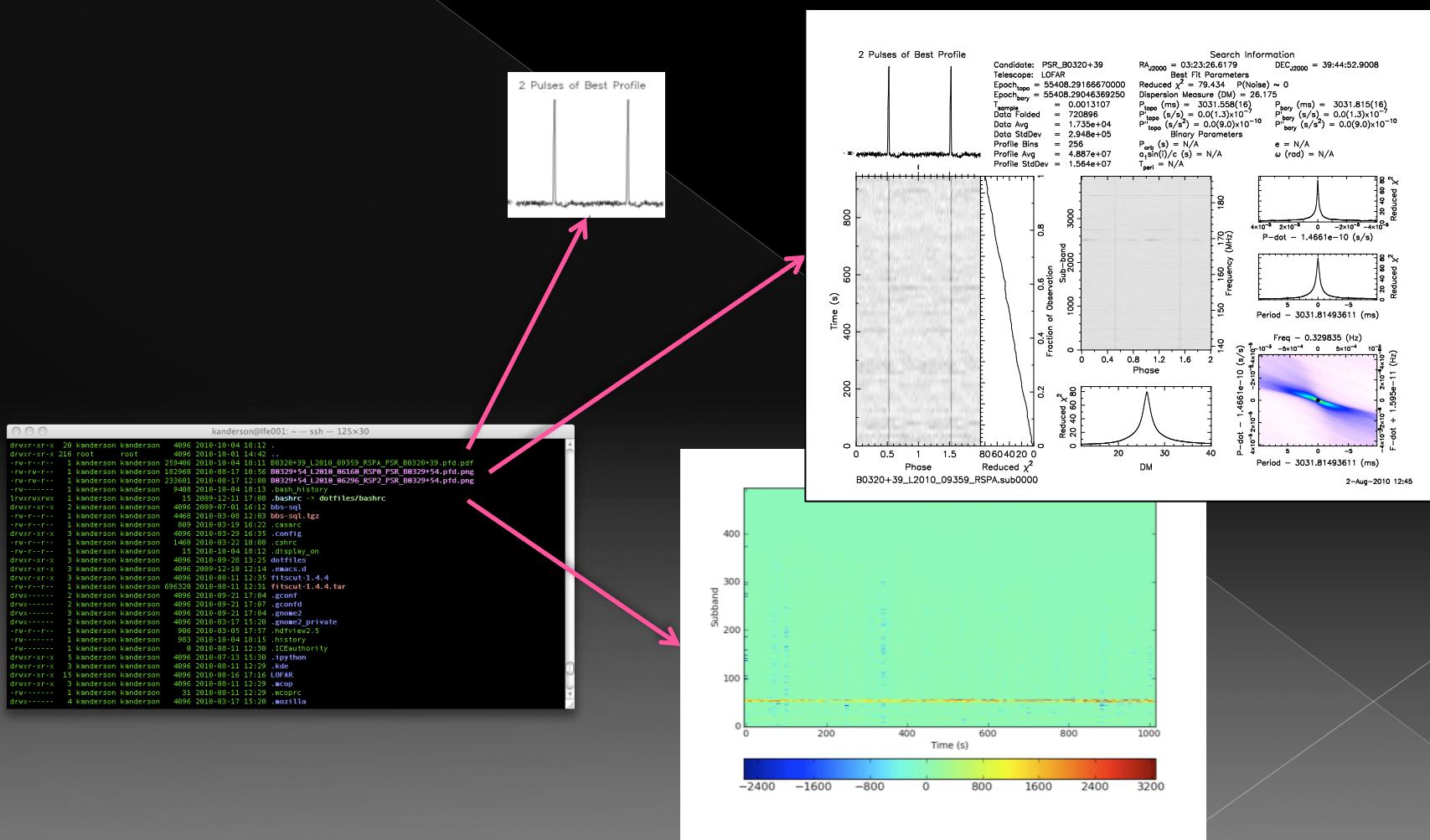
# FRAMEWORK PULSAR PIPELINE

Dream:



# FRAMEWORK PULSAR PIPELINE

pulp prepfold and supplemental  
plots



# FRAMEWORK PULSAR PIPELINE

Near term implementation

- › Fly's eye mode processing
- › Multi-beam observations
- › Coherentstokes processing
  - Pending beam-formed data in HDF5 format
- › FOV position pulsar name(s) determination
  - Implement "pulsars\_at\_location.sh" vis-a-vis pulp shell script

# FRAMEWORK PULSAR PIPELINE

- Future work, enhancements
  - > Documentation (!)
    - User's guide
    - Developer guide to recipe writing
  - > Robust Parameter set handler
  - > Robust data lists
    - Pipeline dependent on data lists
    - RTCP Storage log files have served
      - Whereabouts currently unknown
  - > Complex pipeline definition
    - implement other observational modes,
    - new modes, requirements,
      - fly's eye,
      - coherentstokes,
      - more plotting, charts, ... .
    - Parameter and/or argument based recipe selection
  - > Distributed writing on bf2presto process
    - Currently IO bound

# FRAMEWORK PULSAR PIPELINE

A existing gui control panel, usefulness under investigation

