

Wide-field Infrared Survey Explorer (WISE)

Data Processing Operations Procedures

Version [0.1]

02-October-2009

Prepared by: Ron Beck



**Infrared Processing and Analysis Center
California Institute of Technology**

WSDC D-C005

Approved By:

Ned Wright, WISE Principal Investigator

Donald Royer, WISE Mission Operations Center Manager

Roc Cutri, WISE Science Data Center Manager

[Other Appropriate Names], WISE Science Data Center [Title]

[Other Appropriate Names], WISE Science Data Center [Title]

Revision History

Date	Version	Author	Description
			Initial Draft

Table of Contents

1 INTRODUCTION

1.1 Document Scope

This document will explain the procedures for Data Processing Operations. Once the raw data has been ingested, we can run the scan pipeline processing on it. We expect to receive about 32 scans per day containing around 250 frames worth of raw images in four deliveries. The scan pipeline processing runs the frame pipeline on all frames and then will run a number of modules on the entire scan. Once all the scans for a region of the sky have been processed, we can run the one degree square coadds that cover that region including the source lists. When the coadds and source lists are created, we will have scheduled deliveries of the scans, coadds and lists to IRSA so the data can be made public. We are responsible for the raw data archive which means multiple backups and sending copies offsite. We need to manage resources - disk space, cpus, condor (the pipeline distribution system) and troubleshoot network issues.

1.2 Applicable Documents

http://wise.ipac.caltech.edu/wiki/index.php/Nodes_command
http://wise.ipac.caltech.edu/wiki/index.php/Node_clean_command
http://wise.ipac.caltech.edu/wiki/index.php/Scan_summary_command
http://wise.ipac.caltech.edu/wiki/index.php/Compile_mframe_logs_command
http://wise.ipac.caltech.edu/wiki/index.php/Qlook_summary_command
http://wise.ipac.caltech.edu/wiki/index.php/Broken_links_command
http://wise.ipac.caltech.edu/wiki/index.php/Add2scans_command
http://wise.ipac.caltech.edu/wiki/index.php/Run_scans_command
http://wise.ipac.caltech.edu/wiki/index.php/Showme_running_command
http://www.cs.wisc.edu/condor/manual/v6.2/8_Command_Reference.html#command-reference

1.3 Acronyms

IPAC - Infrared Processing and Analysis Center, California Institute of Technology
MOS - Mission Operations System
WSDC - WISE Science Data Center (IPAC)
WSDS - WISE Science Data System
2MASS - The Two Micron All-Sky Survey

2 Scan Pipeline Procedure

2.1 run_scans Command

Command `/home/beck/bin/run_scans` is the scan distribution script to be run on `wcnode01`. This script will run continuously in operations. In dir `/wise/fops/operations`, there is a file "scan". `run_scans` will continuously (every 10 seconds) check this file for scans to run. The scan file can take two different forms. There can be entries with just the scan identifier or a scan identifier with the `wsspipe` command to be run.

Here is an example of both flavors ...

```
beck@caustic;rhe4():operations[0]% cat scan
01248a
01248a pass "wsspipe -ind . -cluster -preclean all -steps @all,+dynacal"
beck@caustic;rhe4():operations[0]%
```

Any entries with just the scan identifier will be run with the following exec.

```
wsspipe -v -cluster -preclean all -mkdir -ind . -frnums -1
```

When the `run_scans` script sees the `pass` parameter, it knows to run the attached `wsspipe` command verbatim instead of the vanilla exec.

The `run_scans` script will monitor the condor job queue using the `condor_q` command. When the number of jobs returned from this command dips below 300, the script will look to start another scan from the scan file. This is an attempt to keep the 248 node slots busy continuously.

`run_scans` has an optional parameter which is the max number of scans to run at once and defaults to 8.

2.2 run_scans Miscellaneous files

The `run_scans` script creates or updates numerous files. These files are located in the `/wise/fops/operations` dir. File `exec.status` keeps track of the scan start and stop times. Here is a sample.

```
beck@caustic;rhe4():operations[0]% tail exec.status
Jul 13 16:37:39 01248a start
Jul 13 16:45:37 01248a complete
Jul 16 11:35:28 01248a start
Jul 16 11:59:30 01248a complete
Jul 16 13:11:51 01248a start
Jul 16 13:42:17 01248a complete
Jul 16 15:19:24 01250a start
Jul 16 15:28:54 01250a complete
beck@caustic;rhe4():operations[0]%
```

This comes in handy if you want to know when a scan was run or how many times.

There is a gostop file where you can control whether more scans are started from the scan file. 0 means don't start any more, 1 means let em rip. This is queried every 10 seconds by run_scans.

2.3 Scan Summary Script

When a scan completes, the run_scans script fires off another script that goes through all the log files for the scan including all the frame pipe logs and outputs them to files in the /wise/fops/operations/problems dir. the files take the form SCAN.MM-D-HH:MM:SS.PROBS where PROBS is either ok or errors.

Below is a directory listing with scan summary files.

```
beck@caustic;rhe4():problems[0]% pwd
/wise/fops/operations/problems
beck@caustic;rhe4():problems[0]% \ls -l | head
00433a.06-30-14:55:14.errors
00434a.06-30-15:48:44.errors
00434a.07-1-08:12:05.ok
00435a.06-30-15:49:13.errors
00435a.07-1-08:13:16.ok
00436a.06-30-15:55:21.errors
00436a.07-1-08:18:39.ok
00437a.06-30-19:46:30.ok
00438a.06-30-19:40:56.ok
00439a.06-30-19:51:41.ok
beck@caustic;rhe4():problems[0]%
```


The *.ok scans return '0' code for all pipeline modules whereas the *.errors files had some non-zero codes returned.

Below is a sample ok listing.

beck@caustic;rhe4():problems[1]% m 12345a.06-13-18:37:18.ok

FRAME	START	ELAP	STAT	SIG	CODE	HOST	PROGRAM
	09/06/14_01:24:44	12:13	0 0 0			caustic	WSSPipe
001	09/06/14_01:24:57	05:18	0 0 0	10		WSFPipe	
001	09/06/14_01:35:53	00:04	0 0 0	20		WSFPipePost	
002	09/06/14_01:24:57	05:23	0 0 0	10		WSFPipe	
002	09/06/14_01:35:53	00:10	0 0 0	04		WSFPipePost	
003	09/06/14_01:24:57	10:20	0 0 0	02		WSFPipe	
003	09/06/14_01:35:54	00:09	0 0 0	11		WSFPipePost	
004	09/06/14_01:24:57	05:21	0 0 0	11		WSFPipe	
004	09/06/14_01:35:55	00:08	0 0 0	12		WSFPipePost	
005	09/06/14_01:24:57	04:15	0 0 0	20		WSFPipe	
005	09/06/14_01:35:55	00:11	0 0 0	08		WSFPipePost	
007	09/06/14_01:24:57	05:25	0 0 0	10		WSFPipe	
007	09/06/14_01:35:55	00:10	0 0 0	08		WSFPipePost	
008	09/06/14_01:24:57	10:31	0 0 0	02		WSFPipe	
008	09/06/14_01:35:54	00:10	0 0 0	04		WSFPipePost	
009	09/06/14_01:24:57	05:19	0 0 0	11		WSFPipe	
009	09/06/14_01:35:55	00:10	0 0 0	08		WSFPipePost	
010	09/06/14_01:24:57	04:14	0 0 0	20		WSFPipe	
010	09/06/14_01:35:55	00:10	0 0 0	08		WSFPipePost	
011	09/06/14_01:24:57	05:28	0 0 0	10		WSFPipe	
011	09/06/14_01:35:54	00:07	0 0 0	30		WSFPipePost	
012	09/06/14_01:24:57	10:34	0 0 0	02		WSFPipe	
012	09/06/14_01:35:54	00:09	0 0 0	20		WSFPipePost	
	09/06/14_01:35:33	00:00	0 0 0			caustic	Spawn_rotmeta
	09/06/14_01:35:33	00:13	0 0 0			caustic	Spawn_wssflag
	09/06/14_01:35:33	00:13	0 0 0			caustic	WSSFlag
	09/06/14_01:35:34	00:00	0 0 0			caustic	Spawn_dumptbl
	09/06/14_01:35:48	00:02	0 0 0	30		WSSPCal	
	09/06/14_01:35:49	00:00	0 0 0	30		Spawn_spcal	
	09/06/14_01:35:51	00:17	0 0 0			caustic	Spawn_wsspipelinepost
	09/06/14_01:35:51	00:17	0 0 0			caustic	WSSPipePost
	09/06/14_01:36:19	00:27	0 0 0	10		Spawn_scansync	

```

09/06/14_01:36:19 00:37 0 0 0 10 ScanQA
09/06/14_01:36:46 00:01 0 0 0 10 Spawn_rotmeta
09/06/14_01:36:47 00:03 0 0 0 10 Spawn_rotmeta
09/06/14_01:36:51 00:01 0 0 0 10 Spawn_rotmeta
09/06/14_01:36:52 00:01 0 0 0 10 Spawn_rotmeta
09/06/14_01:36:53 00:01 0 0 0 10 Spawn_rotmeta
09/06/14_01:36:54 00:01 0 0 0 10 Spawn_rotmeta
09/06/14_01:36:55 00:00 0 0 0 10 Spawn_rotmeta
09/06/14_01:36:55 00:01 0 0 0 10 Spawn_dumptbl

```

The summary is sorted by the module start time and lists the return codes and elapsed wall clock time by module. An error file will have non-zero return code to be followed by the offending error messages.

2.4 run_scans Summary Output

The window where run_scans is running updates every 10 seconds and monitors the running scans by sifting through the WSSPipe.log file. The display shows the running scans, the currently running module and the scan elapsed time. If the scan is in the WSFPIPE step, the run_scans script breaks that down by total frames, running, complete and failed frames. Below is a sample of the screen.

```

          scan  elapsed  step
of 251 frames total  01248a 00:21:10 SFPIPE      Jul 17 14:40:38
    217 frames running
      33 frames finished ok
>>>  1 frames failed

```

sleeping 10 ...

2.5 showme_running Script

Additionally, run_scans outputs a exec.update file that looks similar to the above screen sample every 10 seconds. This works in conjunction with the showme_running script that anyone can run and see the scans progress just like the ops people.

Here is a sample output from showme_running ...

```

                scan  elapsed  step
of 251 frames total 01248a 00:22:14 SFPIPE      Jul 17 14:41:42
    7 frames running
    243 frames finished ok
>>> 1 frames failed

```

3 Coadd Pipeline Procedure

4 IRSA Delivery Procedure

Probably, weekly we will get a request for IRSA database deliveries. This entails the source lists generated by the frame pipelines and their images. We will also deliver the coadded images and their source lists. This is accomplished with the dbprep and imgprep routines.

4.1 DBPREP Script

The dbprep output data currently goes to /wise/data/irsa_in directory. Since the deliveries will cover a number of scans, we will use the wsdc_submit command to submit them to the cluster to run. Basically, the command you want to run just follows verbatim the wsdc_submit command. The following command was run to create the IRSA delivery for scan 00433a.

```

wsdc_submit dbprep -frame_dirs "/wise/fops/scans/3a/00433a/fr/*" -outdir \
/wise/data/irsa_in >& /wise/data/irsa_in/dbprep.out.00433a &

```

The -frame_dirs parm points to the frame dirs to be preped. Entering command "dbprep -help" will display all the available parms. Below are the resulting created files.

```

beck@caustic;rhe4(dev):irsa_in[0]% pwd
/wise/data/irsa_in
beck@caustic;rhe4(dev):irsa_in[0]% ls -al *00433a*
-rw-rw-r--+ 1 beck wise 121654 Aug 7 14:44 dbprep.out.00433a
-rw-rw-r--+ 1 beck wise 12733 Aug 7 14:44 wise_i1ba_frm_3.bartbl.00433a
-rw-rw-r--+ 1 beck wise 455405 Aug 7 14:44 wise_i1ba_mch_3.bartbl.00433a
-rw-rw-r--+ 1 beck wise 10663 Aug 7 14:43 wise_i1bc_frm_3.bartbl.00433a
-rw-rw-r--+ 1 beck wise 1577 Aug 7 14:43 wise_i1bc_mch_3.bartbl.00433a
-rw-rw-r--+ 1 beck wise 157 Aug 7 14:43 wise_i1bc_scn_3.bartbl.00433a

```

```
-rw-rw-r--+ 1 beck wise 401500 Aug 7 14:43 wise_i1bs_frm_3.bartbl.00433a
-rw-rw-r--+ 1 beck wise 991839862 Aug 7 14:43 wise_i1bs_psd_3.bartbl.00433a
beck@caustic;rhe4(dev):irsa_in[0]%
```

Seven files are output from dbprep. The **s_psd** is the point source data for the scan. **s_frm** is per frame metadata whereas **c_scn** is the calibration scan level metadata. The **a_frm** and **a_mch** files are the asteroids and their associations. The **c_frm** is calibration frame level metadata and **c_mch** are calibration matches.

The dbprep output files should be checked for return codes after running.

```
beck@caustic;rhe4(dev):irsa_in[0]% grep "End of DBPrep" dbprep.out.00433a
<< End of DBPrep >> status=0
beck@caustic;rhe4(dev):irsa_in[0]%
```

Problems need to be addressed with Tracey Evans. Sherry Wheelock is Tracey's backup.

When running a number of scans which will be the case in operations, we need to concatenate all output files except the **s_psd** file by file type. For instance, all **c_scn** files can be put together into one as below.

```
beck@caustic;rhe4(dev):irsa_in[0]% cat wise_i1bc_scn_3.bartbl.004* >
wise_i1bc_scn_3.bartbl
beck@caustic;rhe4(dev):irsa_in[1]% wc wise_i1bc_scn_3.bartbl.004*
wise_i1bc_scn_3.bartbl
  1  1 157 wise_i1bc_scn_3.bartbl.00433a
  1  1 163 wise_i1bc_scn_3.bartbl.00434a
  1  1 165 wise_i1bc_scn_3.bartbl.00435a
  1  1 168 wise_i1bc_scn_3.bartbl.00436a
  1  1 157 wise_i1bc_scn_3.bartbl.00437a
  1  1 176 wise_i1bc_scn_3.bartbl.00438a
  1  1 164 wise_i1bc_scn_3.bartbl.00439a
  1  1 176 wise_i1bc_scn_3.bartbl.00440a
  1  1 164 wise_i1bc_scn_3.bartbl.00441a
  1  1 163 wise_i1bc_scn_3.bartbl.00442a
  1  1 156 wise_i1bc_scn_3.bartbl.00443a
  1  1 161 wise_i1bc_scn_3.bartbl.00444a
  1  1 165 wise_i1bc_scn_3.bartbl.00445a
```

1 1 148 wise_ilbc_scn_3.bartbl.00446a
1 1 164 wise_ilbc_scn_3.bartbl.00447a
1 1 176 wise_ilbc_scn_3.bartbl.00448a
1 1 173 wise_ilbc_scn_3.bartbl.00449a
1 1 173 wise_ilbc_scn_3.bartbl.00450a
1 1 164 wise_ilbc_scn_3.bartbl.00451a
1 1 167 wise_ilbc_scn_3.bartbl.00452a
1 1 157 wise_ilbc_scn_3.bartbl.00453a
1 1 173 wise_ilbc_scn_3.bartbl.00454a
1 1 164 wise_ilbc_scn_3.bartbl.00455a
1 1 164 wise_ilbc_scn_3.bartbl.00456a
1 1 173 wise_ilbc_scn_3.bartbl.00457a
1 1 172 wise_ilbc_scn_3.bartbl.00458a
1 1 173 wise_ilbc_scn_3.bartbl.00459a
1 1 173 wise_ilbc_scn_3.bartbl.00460a
1 1 156 wise_ilbc_scn_3.bartbl.00461a
1 1 123 wise_ilbc_scn_3.bartbl.00462a
1 1 155 wise_ilbc_scn_3.bartbl.00463a
1 1 119 wise_ilbc_scn_3.bartbl.00464a
1 1 157 wise_ilbc_scn_3.bartbl.00465a
1 1 119 wise_ilbc_scn_3.bartbl.00466a
1 1 173 wise_ilbc_scn_3.bartbl.00467a
1 1 156 wise_ilbc_scn_3.bartbl.00468a
1 1 155 wise_ilbc_scn_3.bartbl.00469a
1 1 173 wise_ilbc_scn_3.bartbl.00470a
1 1 156 wise_ilbc_scn_3.bartbl.00471a
1 1 119 wise_ilbc_scn_3.bartbl.00472a
1 1 165 wise_ilbc_scn_3.bartbl.00473a
1 1 119 wise_ilbc_scn_3.bartbl.00474a
1 1 157 wise_ilbc_scn_3.bartbl.00475a
1 1 119 wise_ilbc_scn_3.bartbl.00476a
1 1 173 wise_ilbc_scn_3.bartbl.00477a
1 1 173 wise_ilbc_scn_3.bartbl.00478a
1 1 165 wise_ilbc_scn_3.bartbl.00479a
1 1 156 wise_ilbc_scn_3.bartbl.00480a
1 1 165 wise_ilbc_scn_3.bartbl.00481a
1 1 173 wise_ilbc_scn_3.bartbl.00482a
1 1 156 wise_ilbc_scn_3.bartbl.00483a
1 1 173 wise_ilbc_scn_3.bartbl.00484a
1 1 157 wise_ilbc_scn_3.bartbl.00485a

```

1 1 164 wise_ilbc_scn_3.bartbl.00486a
1 1 155 wise_ilbc_scn_3.bartbl.00487a
1 1 164 wise_ilbc_scn_3.bartbl.00488a
1 1 164 wise_ilbc_scn_3.bartbl.00489a
1 1 164 wise_ilbc_scn_3.bartbl.00490a
1 1 156 wise_ilbc_scn_3.bartbl.00491a
1 1 173 wise_ilbc_scn_3.bartbl.00492a
60 60 9601 wise_ilbc_scn_3.bartbl
120 120 19202 total
beck@caustic;rhe4(dev):irsa_in[0]%

```

loading 1 file instead of 60 makes life easier on the IRSA people when they load the database. The `*s_psd*` files are pretty big and should be packaged up by 20 scans per file.

4.2 IMGPREP Script

The `imgprep` command runs basically the same as `dbprep` only `imgprep` gleans information from the images created by scan pipelines. It creates a pipe-delimited file containing position info and filenames for loading into the IRSA image server database. Again we submit the `imgprep` job to the cluster using the `wscd_submit` command.

```
wscd_submit imgprep -frame_dirs "/wise/fops/scans/3a/00433a/fr/*" -outdir \
/wise/data/irsa_in > & /wise/data/irsa_in/imgprep.out.00433a &
```

The `-frame_dirs` parm points to the frame dirs to be preped. Entering command "`imgprep -help`" will display all the available parms. Below are the resulting created files.

```

beck@caustic;rhe4(dev):irsa_in[0]% pwd
/wise/data/irsa_in
beck@caustic;rhe4(dev):irsa_in[0]% ls -al | grep Sep
drwxrwxr-x  2 beck wise    1759 Sep  2 09:31 .
-rw-rw-r--+ 1 beck wise    4682 Sep  2 09:24 imgprep.out.00433a
-rw-rw-r--+ 1 beck wise   723385 Sep  2 09:24 wise_ilbm_frm_3.bartbl.00433a
beck@caustic;rhe4(dev):irsa_in[0]%

```

`imgprep` creates only the one pipe-delimited file describing the images for the scan.

The imgprep output files should be checked for return codes after running.

```
beck@caustic;rhe4(dev):irsa_in[0]% grep "End of ImgPrep" imgprep.out.00433a
<< End of ImgPrep >> status=0
beck@caustic;rhe4(dev):irsa_in[0]%
```

Problems need to be addressed with Tracey Evans. Sherry Wheelock is Tracey's backup.

When running a number of scans which will be the case in operations, we need to concatenate all output files into one for delivery to IRSA.

```
beck@caustic;rhe4(dev):irsa_in[0]% cat wise_i1bm_frm_3.bartbl.004* >
wise_i1bm_frm_3.bartbl
beck@caustic;rhe4(dev):irsa_in[1]%
```

5 WISE Raw Data Backup Procedure

6 Pipeline Machines Maintenance

6.1 nodes Script

The nodes command is handy when you want to run the same command on all node machines. The IFILE parm is optional and would contain a list of machines to run on. If no IFILE parm is supplied, file /wise/fops/operations/nodes will be used. You have to create a public key prior to running this command unless you want to be prompted for a password for each machine. Instructions below.

```
beck@caustic;rhe4(ops):~[0]% nodes
```

command usage: nodes COMMANDS BACKGROUND IFILE

where COMMANDS is list of commands separated by ';' to
run on all pipeline nodes
BACKGROUND is y or n to put the command in background
IFILE is file containing list of machines to run
commands on (optional and defaults to wcnod01-32)

Here is an example.

```
beck@caustic;rhe4(ops):~[1]% nodes "uptime" n
ssh wcnode01 uptime ...
09:13:58 up 20 days, 18:03, 1 user, load average: 0.28, 0.21, 0.12
ssh wcnode02 uptime ...
09:13:58 up 20 days, 17:57, 1 user, load average: 0.00, 0.01, 0.00
ssh wcnode03 uptime ...
09:13:58 up 20 days, 17:57, 0 users, load average: 0.03, 0.02, 0.00
ssh wcnode04 uptime ...
09:13:59 up 20 days, 17:58, 0 users, load average: 0.00, 0.02, 0.00
ssh wcnode05 uptime ...
09:13:59 up 20 days, 17:58, 0 users, load average: 0.00, 0.00, 0.00
ssh wcnode06 uptime ...
09:14:00 up 20 days, 17:58, 0 users, load average: 0.00, 0.00, 0.00
ssh wcnode07 uptime ...
09:14:01 up 20 days, 17:58, 0 users, load average: 0.01, 0.01, 0.00
ssh wcnode08 uptime ...
09:14:01 up 20 days, 17:58, 0 users, load average: 0.01, 0.01, 0.00
ssh wcnode09 uptime ...
09:14:02 up 20 days, 17:57, 0 users, load average: 0.00, 0.00, 0.00
ssh wcnode10 uptime ...
09:14:03 up 20 days, 17:58, 1 user, load average: 0.72, 0.88, 0.81
ssh wcnode11 uptime ...
09:14:03 up 20 days, 17:58, 1 user, load average: 0.00, 0.00, 0.00
ssh wcnode12 uptime ...
09:14:04 up 20 days, 17:58, 0 users, load average: 0.00, 0.00, 0.00
ssh wcnode13 uptime ...
09:14:04 up 20 days, 17:57, 0 users, load average: 0.00, 0.02, 0.01
ssh wcnode14 uptime ...
09:14:05 up 20 days, 17:58, 1 user, load average: 0.00, 0.00, 0.00
ssh wcnode15 uptime ...
09:14:05 up 20 days, 17:58, 0 users, load average: 0.00, 0.00, 0.00
ssh wcnode16 uptime ...
09:14:06 up 20 days, 17:58, 0 users, load average: 0.00, 0.00, 0.00
ssh wcnode17 uptime ...
09:14:06 up 20 days, 17:58, 0 users, load average: 0.00, 0.05, 0.03
ssh wcnode18 uptime ...
09:14:07 up 20 days, 17:58, 1 user, load average: 0.32, 0.11, 0.04
ssh wcnode19 uptime ...
09:14:07 up 20 days, 17:58, 0 users, load average: 0.00, 0.00, 0.00
```



```
ssh wcnode20 uptime ...
09:14:08 up 20 days, 17:58, 0 users, load average: 0.00, 0.05, 0.07
ssh wcnode21 uptime ...
09:14:09 up 20 days, 17:58, 0 users, load average: 0.04, 0.03, 0.00
ssh wcnode22 uptime ...
09:14:09 up 20 days, 17:58, 0 users, load average: 0.00, 0.06, 0.07
ssh wcnode23 uptime ...
09:14:10 up 20 days, 17:58, 0 users, load average: 0.00, 0.03, 0.01
ssh wcnode24 uptime ...
09:14:10 up 20 days, 17:58, 0 users, load average: 0.00, 0.02, 0.05
ssh wcnode25 uptime ...
09:14:11 up 20 days, 17:58, 0 users, load average: 0.00, 0.07, 0.04
ssh wcnode26 uptime ...
09:14:11 up 20 days, 17:58, 0 users, load average: 0.00, 0.02, 0.04
ssh wcnode27 uptime ...
09:14:12 up 20 days, 17:58, 0 users, load average: 0.08, 0.24, 0.13
ssh wcnode28 uptime ...
09:14:13 up 20 days, 17:58, 0 users, load average: 0.15, 0.03, 0.01
ssh wcnode29 uptime ...
09:14:13 up 20 days, 17:58, 0 users, load average: 0.00, 0.00, 0.00
ssh wcnode30 uptime ...
09:14:14 up 20 days, 17:59, 1 user, load average: 0.00, 0.00, 0.00
ssh wcnode31 uptime ...
09:14:14 up 20 days, 17:59, 1 user, load average: 0.57, 0.21, 0.07
ssh wcnode32 uptime ...
09:14:15 up 20 days, 17:58, 1 user, load average: 0.58, 0.28, 0.10
beck@caustic;rhe4(ops):~[0]%
```

6.2 Creating a Public Key

Instructions for creating a public key.

```
sirius:.ssh tungn$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/tungn/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/tungn/.ssh/id_rsa.
Your public key has been saved in /Users/tungn/.ssh/id_rsa.pub.
```

copy the id_rsa.pub to the remote machine \$HOME/.ssh/authorized_keys

In this case, the remote machine would be caustic.

6.3 Node Machines Disk Usage

The scan pipelines create work directories on the node machines in the /local/wise directories fops, rtb and tops. These work directories contain intermediate pipeline processed files which would be useful for debug purposes and are linked to the /wise/[fops|rtb|tops]/scans dirs. There are two scripts to help us maintain enough space in these directories to continuously keep scan pipelines running.

6.3.1 broken_links Script

Command broken_links is to be run on the wcnodes machines and will check the dirs on the wcnodes machine versus the work link name in /wise/[fops|rtb|tops]/scans/*/*fr/*. Output displays discrepancies with the option of deleting the broken link dirs thereby saving the node /local space.

```
beck@caustic;rhe4(ops):~[0]% /home/beck/bin/broken_links
command usage: broken_links DELETE
      where DELETE is y or n to delete or just display
beck@caustic;rhe4(ops):~[0]%
```

Below is an example run on wcnodes10.

```
/home/beck/bin/broken_links n
```

Creates following output.

```
/compute/wcnodes13/wise/fops/scans/5a/00435a/fr/051/work ...
/compute/wcnodes21/wise/fops/scans/5a/00435a/fr/100/work ...
/compute/wcnodes02/wise/fops/scans/5a/00435a/fr/002/work ...
/compute/wcnodes11/wise/fops/scans/5a/00435a/fr/020/work ...
/compute/wcnodes26/wise/fops/scans/5a/00445a/fr/144/work ...
/compute/wcnodes05/wise/fops/scans/5a/00445a/fr/117/work ...
/compute/wcnodes16/wise/fops/scans/5a/00445a/fr/059/work ...
/compute/wcnodes14/wise/fops/scans/5a/00445a/fr/237/work ...
/local/wise/fops/scans/5a/00445a/fr/090/work link broken - dir does not exist ...
```

/compute/wcnode11/wise/fops/scans/2a/00442a/fr/159/work ...
/local/wise/fops/scans/2a/00442a/fr/147/work link broken - dir does not exist ...

All the lines that start with /compute are the link names that are in the /wise/[fops|rtb|tops]/scans/**/fr/* dirs or where the frame work dir actually is for that frame. The "link broken" lines mean that there is a dir on wcnode10 with no corresponding link in /wise/[fops|rtb|tops]/scans at all and therefore orphaned and a prime candidate for deletion.

6.3.2 node_clean Script

Command node_clean runs continuously on the wcnode machines 2 - 32 making sure we do not run out of pipeline work space on the nodes. Log files are located in /wise/fops/operations/node_clean dir.

command usage: node_clean DISKPER CLEANPER SLEEP

where DISKPER is minimum percent used on /local to start deletes
CLEANPER is percentage to clean down to
SLEEP is minutes before checking

notes: this command will awaken every SLEEP minutes and check the /local disk used percentage versus DISKPER parm. if percent used is greater than DISKPER, all /local/wise/[fops|rtb|tops]/scans work dir names are gathered by modified date. the oldest work dirs are deleted until /local used percentage is below CLEANPER and then back to sleep. the pointer to the /local node dir in /wise/fops/scans is also deleted first checking that it indeed is pointing to the /local work directory.

node_clean also cleans up all condor log files older than 3 days that are located in the /local directory.

6.4 Node Machines Messages Files

The wcnode machines messages files are located in /var/log. The messages files are rotated weekly with the messages file being current and messages.1, messages.2 and so on being older versions. The messages file can be read with the "dmesg" command as in the example below.

```

beck@wcnod01;rhe4():log[0]% dmesg messages | tail -15
microcode: No new microdata for cpu 6
microcode: No new microdata for cpu 7
IA-32 Microcode Update Driver v1.14 unregistered
Linux Kernel Card Services
  options: [pci] [cardbus] [pm]
ip_tables: (C) 2000-2002 Netfilter core team
ip_tables: (C) 2000-2002 Netfilter core team
MSI INIT SUCCESS
bnx2: eth0: using MSI
bnx2: eth0 NIC Copper Link is Up, 1000 Mbps full duplex
ip_tables: (C) 2000-2002 Netfilter core team
i2c /dev entries driver
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
flatcal[28746]: segfault at 0000000000771958 rip 000000000042f200 rsp
0000007fbfff9468 error 4
Losing some ticks... checking if CPU frequency changed.
beck@wcnod01;rhe4():log[0]%

```

This can clue us in should the node machine start having problems such as a disk going bad, disk out of space, memory problems etc.

7 Condor commands

Condor is the software package that we use for distributing background jobs to the node machines. The condor master runs on wcnod01 while the machines wcnod02 - 32 do the actual processing as jobs are farmed out to them by the master. The condor master and clients start up automatically when rebooted and otherwise run continuously.

7.1 condor_status Command

The condor_status command displays all the nodes status and a summary of claimed and unclaimed nodes. See below.

```
beck@caustic;rhe4(ops):~[0]% condor_status
```

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@wcnod02.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+08:57:43

slot2@wcnode02.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+06:38:56
slot3@wcnode02.ipa	LINUX	X86_64	Claimed	Busy	0.000	4020	0+10:46:07
slot4@wcnode02.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+09:50:28
slot5@wcnode02.ipa	LINUX	X86_64	Claimed	Busy	0.000	4020	0+10:46:09
slot6@wcnode02.ipa	LINUX	X86_64	Unclaimed	Idle	0.080	4020	0+01:14:24
slot7@wcnode02.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+14:59:32
slot8@wcnode02.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+13:18:06
slot1@wcnode03.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+10:37:16
slot2@wcnode03.ipa	LINUX	X86_64	Claimed	Busy	0.000	4020	0+10:46:56
slot3@wcnode03.ipa	LINUX	X86_64	Claimed	Busy	0.000	4020	0+10:46:57
slot4@wcnode03.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+14:12:37
slot5@wcnode03.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+15:03:53
slot6@wcnode03.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+15:02:43
slot7@wcnode03.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	4020	0+15:04:30
slot8@wcnode03.ipa	LINUX	X86_64	Unclaimed	Idle	0.040	4020	0+00:55:03
slot1@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+07:31:21
slot2@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:05:22
slot3@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:05:58
slot4@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:06:30
slot5@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:17:26
slot6@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:14:32
slot7@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:38:54
slot8@wcnode04.ipa	LINUX	X86_64	Unclaimed	Idle	0.030	1501	0+00:55:03
slot1@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:05:45
slot2@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:05:51
slot3@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:14:19
slot4@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:13:17
slot5@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:39:04
slot6@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+17:09:42
slot7@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+17:08:08
slot8@wcnode05.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+00:55:04
slot1@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:16:16
slot2@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:17:47
slot3@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:36:55
slot4@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:46:15
slot5@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+17:13:15
slot6@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+17:08:05
slot7@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+00:55:10
slot8@wcnode06.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+19:05:23
slot1@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:38:24

slot2@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:45:59
slot3@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+17:09:57
slot4@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+19:04:57
slot5@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+19:05:04
slot6@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+19:11:14
slot7@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+20:29:32
slot8@wcnode07.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+00:55:03
slot1@wcnode08.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+05:16:48
slot2@wcnode08.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+10:45:41
slot3@wcnode08.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+11:44:04
slot4@wcnode08.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:04:39
slot5@wcnode08.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:01:14
slot6@wcnode08.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:04:03
slot7@wcnode08.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:17:21
slot8@wcnode08.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+00:55:03
slot1@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:03:48
slot2@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+08:21:49
slot3@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:06:47
slot4@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:04:23
slot5@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:19:25
slot6@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:15:07
slot7@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:39:58
slot8@wcnode09.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+00:55:03
slot1@wcnode10.ipa	LINUX	X86_64	Claimed	Busy	0.060	1501	0+10:45:09
slot2@wcnode10.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+10:45:10
slot3@wcnode10.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+00:32:01
slot4@wcnode10.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+10:45:16
slot5@wcnode10.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+10:45:17
slot6@wcnode10.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+02:35:22
slot7@wcnode10.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+05:42:25
slot8@wcnode10.ipa	LINUX	X86_64	Unclaimed	Idle	0.700	1501	0+00:55:03
slot1@wcnode11.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+16:25:20
slot2@wcnode11.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+06:38:48
slot3@wcnode11.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+07:38:32
slot4@wcnode11.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+10:49:16
slot5@wcnode11.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+04:20:58
slot6@wcnode11.ipa	LINUX	X86_64	Unclaimed	Idle	0.010	1501	0+01:13:35
slot7@wcnode11.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:07:52
slot8@wcnode11.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:07:19
slot1@wcnode12.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+03:46:11

slot2@wcnode12.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+10:47:29
slot3@wcnode12.ipa	LINUX	X86_64	Claimed	Busy	0.000	1501	0+10:47:30
slot4@wcnode12.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:07:44
slot5@wcnode12.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:03:24
slot6@wcnode12.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:06:06
slot7@wcnode12.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+15:06:42
slot8@wcnode12.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	1501	0+02:25:05
slot1@wcnode13.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:48:40
slot2@wcnode13.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+14:43:16
slot3@wcnode13.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:08:37
slot4@wcnode13.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:03
slot5@wcnode13.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:06:08
slot6@wcnode13.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:19:11
slot7@wcnode13.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:16:59
slot8@wcnode13.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+03:55:03
slot1@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:06
slot2@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:06:59
slot3@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:01
slot4@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:18:42
slot5@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:28:06
slot6@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:36:38
slot7@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.530	2005	0+00:55:10
slot8@wcnode14.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:09:55
slot1@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:46
slot2@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:19:10
slot3@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:24:14
slot4@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:41:08
slot5@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:10:16
slot6@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:10:14
slot7@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+03:55:10
slot8@wcnode15.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:06:56
slot1@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:29:25
slot2@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:41:09
slot3@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:15:51
slot4@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:11:22
slot5@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:06
slot6@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:14
slot7@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+20:34:25
slot8@wcnode16.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+00:55:04
slot1@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:10:37

slot2@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:12:27
slot3@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:14
slot4@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:10
slot5@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+20:33:37
slot6@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+20:34:39
slot7@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.050	2005	0+03:55:10
slot8@wcnode17.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+21:18:12
slot1@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+12:24:06
slot2@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:08
slot3@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:01
slot4@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:06:48
slot5@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:19:03
slot6@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:27:52
slot7@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:37:46
slot8@wcnode18.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+00:55:03
slot1@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:40
slot2@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:41
slot3@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:19:06
slot4@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:27:20
slot5@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:40:41
slot6@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:11:50
slot7@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:04
slot8@wcnode19.ipa	LINUX	X86_64	Unclaimed	Idle	0.050	2005	0+00:55:03
slot1@wcnode20.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+01:47:31
slot2@wcnode20.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:46:06
slot3@wcnode20.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:46:07
slot4@wcnode20.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+10:28:45
slot5@wcnode20.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:46:09
slot6@wcnode20.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:00:06
slot7@wcnode20.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+01:07:11
slot8@wcnode20.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+12:56:07
slot1@wcnode21.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+04:46:21
slot2@wcnode21.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:49:17
slot3@wcnode21.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:49:18
slot4@wcnode21.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:09:31
slot5@wcnode21.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:08:57
slot6@wcnode21.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:09:34
slot7@wcnode21.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:08:34
slot8@wcnode21.ipa	LINUX	X86_64	Unclaimed	Idle	0.570	2005	0+01:00:03
slot1@wcnode22.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:47:37

slot2@wcnode22.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:51
slot3@wcnode22.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:54
slot4@wcnode22.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:40
slot5@wcnode22.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:53
slot6@wcnode22.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:19:00
slot7@wcnode22.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:28:13
slot8@wcnode22.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+00:50:03
slot1@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:43
slot2@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:59
slot3@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:08:56
slot4@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:18:51
slot5@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:28:14
slot6@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:40:13
slot7@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+00:55:10
slot8@wcnode23.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:08:48
slot1@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:06:42
slot2@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:19:40
slot3@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:29:44
slot4@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:38:07
slot5@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:10:56
slot6@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:11:59
slot7@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+00:55:10
slot8@wcnode24.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:10
slot1@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:28:20
slot2@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:40:37
slot3@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:15:23
slot4@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:10:41
slot5@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:12
slot6@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:13
slot7@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+20:34:22
slot8@wcnode25.ipa	LINUX	X86_64	Unclaimed	Idle	0.030	2005	0+00:50:03
slot1@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:09:48
slot2@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:12:41
slot3@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:24
slot4@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:21
slot5@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:13:27
slot6@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+20:34:52
slot7@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.640	2005	0+00:55:11
slot8@wcnode26.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+21:18:24
slot1@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:21

slot2@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:26
slot3@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+20:32:40
slot4@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+20:32:40
slot5@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+21:18:34
slot6@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.040	2005	0+03:10:09
slot7@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+21:18:33
slot8@wcnode27.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+21:18:26
slot1@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+14:46:06
slot2@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:15
slot3@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:11:11
slot4@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:28:51
slot5@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:38:25
slot6@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:14:28
slot7@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:14:51
slot8@wcnode28.ipa	LINUX	X86_64	Unclaimed	Idle	0.090	2005	0+03:55:04
slot1@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:19:16
slot2@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:26:44
slot3@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:41:04
slot4@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:17:47
slot5@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+17:17:31
slot6@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:31
slot7@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+19:07:41
slot8@wcnode29.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+00:55:04
slot1@wcnode30.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+08:55:09
slot2@wcnode30.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:47:17
slot3@wcnode30.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:47:18
slot4@wcnode30.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:32
slot5@wcnode30.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:46
slot6@wcnode30.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:05:22
slot7@wcnode30.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:49
slot8@wcnode30.ipa	LINUX	X86_64	Unclaimed	Idle	0.490	2005	0+03:10:05
slot1@wcnode31.ipa	LINUX	X86_64	Claimed	Busy	0.000	2005	0+10:47:46
slot2@wcnode31.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:59
slot3@wcnode31.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:04:15
slot4@wcnode31.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:06:06
slot5@wcnode31.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:07:51
slot6@wcnode31.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:18:46
slot7@wcnode31.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:25:23
slot8@wcnode31.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+00:55:03
slot1@wcnode32.ipa	LINUX	X86_64	Unclaimed	Idle	0.000	2005	0+15:06:12

```

slot2@wcnode32.ipa LINUX      X86_64 Unclaimed Idle  0.000 2005 0+15:08:21
slot3@wcnode32.ipa LINUX      X86_64 Unclaimed Idle  0.000 2005 0+15:15:06
slot4@wcnode32.ipa LINUX      X86_64 Unclaimed Idle  0.000 2005 0+15:18:38
slot5@wcnode32.ipa LINUX      X86_64 Unclaimed Idle  0.000 2005 0+15:27:58
slot6@wcnode32.ipa LINUX      X86_64 Unclaimed Idle  0.000 2005 0+15:50:36
slot7@wcnode32.ipa LINUX      X86_64 Unclaimed Idle  0.000 2005 0+00:55:10
slot8@wcnode32.ipa LINUX      X86_64 Unclaimed Idle  0.000 2005 0+17:05:23

```

Total Owner Claimed Unclaimed Matched Preempting Backfill

```

X86_64/LINUX 248 0 23 225 0 0 0
Total 248 0 23 225 0 0 0
beck@caustic;rhe4(ops):~[0]%

```

7.2 condor_q Command

The condor_q command displays the actual jobs status and owners.

```
beck@caustic;rhe4(ops):~[0]% condor_q -g
```

```

-- Schedd: wcnode11.ipac.caltech.edu : <134.4.142.213:32774>
ID   OWNER      SUBMITTED  RUN_TIME ST PRI SIZE CMD
54.0 bauer      8/4 23:17 0+10:52:04 R 0 1220.7 procGlob
54.1 bauer      8/4 23:17 0+10:52:04 R 0 732.4 procGlob
54.4 bauer      8/4 23:17 0+10:52:07 R 0 1220.7 procGlob
54.6 bauer      8/4 23:17 0+10:52:07 R 0 1220.7 procGlob
54.9 bauer      8/4 23:17 0+10:52:07 R 0 1953.1 procGlob
54.10 bauer     8/4 23:17 0+10:52:07 R 0 976.6 procGlob
54.13 bauer     8/4 23:17 0+10:52:07 R 0 1464.8 procGlob
54.14 bauer     8/4 23:17 0+10:52:07 R 0 976.6 procGlob
54.15 bauer     8/4 23:17 0+10:52:07 R 0 2685.5 procGlob
54.17 bauer     8/4 23:17 0+10:52:07 R 0 317.4 procGlob
54.20 bauer     8/4 23:17 0+10:52:07 R 0 244.1 procGlob
54.21 bauer     8/4 23:17 0+10:52:07 R 0 1464.8 procGlob
54.23 bauer     8/4 23:17 0+10:52:07 R 0 1464.8 procGlob
54.24 bauer     8/4 23:17 0+10:52:07 R 0 1464.8 procGlob
54.25 bauer     8/4 23:17 0+10:52:07 R 0 732.4 procGlob
54.26 bauer     8/4 23:17 0+10:52:07 R 0 1220.7 procGlob

```

```
54.27 bauer      8/4 23:17 0+10:52:07 R 0 7324.2 procGlob
54.29 bauer      8/4 23:17 0+10:52:07 R 0 1709.0 procGlob
54.30 bauer      8/4 23:17 0+10:52:07 R 0 1464.8 procGlob
54.31 bauer      8/4 23:17 0+10:52:07 R 0 244.1 procGlob
54.32 bauer      8/4 23:17 0+10:52:07 R 0 1709.0 procGlob
54.33 bauer      8/4 23:17 0+10:52:07 R 0 1709.0 procGlob
```

22 jobs; 0 idle, 22 running, 0 held

-- Schedd: caustic.ipac.caltech.edu : <134.4.141.175:32785>

```
ID   OWNER      SUBMITTED  RUN_TIME ST PRI SIZE CMD
84506.0 bauer      8/4 17:40 0+16:28:14 R 0 122.1 hashGlobs -i 30orb
```

1 jobs; 0 idle, 1 running, 0 held

beck@caustic;rhe4(ops):~[0]%

Condor is a very sophisticated software package with lots of bells and whistles that we will not need. We will be using it to basically control running of the batch pipelines. The following link explains all of the available commands.

http://www.cs.wisc.edu/condor/manual/v6.2/8_Command_Reference.html#command-reference