Troubleshooting

Tuesday, July 26

Showmic Islam
What Can Go Wrong?

• Jobs can go wrong “internally”:
  • the executable experiences an error

• Jobs can go wrong *logistically*, from HTCondor’s perspective:
  • a job can’t be matched
  • files not found for transfer
  • job used too much memory
  • badly-formatted executable
  • and more...
Reviewing Failed Jobs

• Job log, output and error files can provide valuable troubleshooting details:

<table>
<thead>
<tr>
<th>Log</th>
<th>Output</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>• when jobs were submitted, started, held, or stopped</td>
<td>• stdout (or other output files) may contain errors from the executable</td>
<td>• stderr captures errors from the operating system, or reported by the executable, itself.</td>
</tr>
<tr>
<td>• where job ran</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• resources used</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• interruption reasons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• exit status</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Job Holds

• HTCondor will *hold* your job if there’s a *logistical* issue that YOU (or maybe an admin) need to fix.
  • files not found for transfer, over memory, etc.

• A job that goes on hold is interrupted (all progress is lost) but remains in the queue in the “H” state until removed, or (fixed and) released.

$ condor_q

<table>
<thead>
<tr>
<th>OWNER</th>
<th>BATCH_NAME</th>
<th>SUBMITTED</th>
<th>DONE</th>
<th>RUN</th>
<th>IDLE</th>
<th>HOLD</th>
<th>TOTAL JOB_IDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>ID: 123456</td>
<td>7/11 11:23</td>
<td>_</td>
<td>_</td>
<td>_</td>
<td>1</td>
<td>1 123456.0</td>
</tr>
</tbody>
</table>
Common Hold Reasons

- Incorrect path to files that need to be transferred
- Badly formatted executables (e.g. Windows line endings on Linux)
- Job has used more memory or disk than requested.
- Job has run longer than allowed. (e.g. 20-hour default in OSPool)
- Submit directory is over quota.
- The admin has put your job on hold.
Failed to initialize user log to /path or /dev/null
   ▶ Could not create log file, check /path carefully
Error from …: Job has gone over memory limit of AAA megabytes.
   ▶ Job used too much memory
   ▶ Request more – at least BBB megabytes!
Error from …: STARTER at … failed to send file(s) to <…>: error reading from /path: (errno 2) No such file or directory; SHADOW failed to receive file(s) from <…>
   ▶ Job specified transfer_output_files
   ▶ But /path on remote server was not found
   ▶ Jargon: SHADOW is Access Point, STARTER is Execute Point
Diagnosing Holds

• If HTCondor puts a job on hold, it provides a hold reason, which can be viewed in the log file, with `condor_q –hold <Job.ID>`, or with `<username>`:

```
$ condor_q -hold -af HoldReason
Error from slot1_1@wid-003.chtc.wisc.edu: Job has gone over memory limit of 2048 megabytes.
Error from slot1_20@e098.chtc.wisc.edu: SHADOW at 128.104.101.92 failed to send file(s) to <128.104.101.98:35110>: error reading from /home/alice/script.py: (errno 2) No such file or directory;
STARTER failed to receive file(s) from <128.104.101.92:9618>
Error from slot1_11@e138.chtc.wisc.edu: STARTER at 128.104.101.138 failed to send file(s) to <128.104.101.92:9618>; SHADOW at 128.104.101.92 failed to write to file /home/alice/Test_18925319_16.err: (errno 122) Disk quota exceeded
```
What To Do About Held Jobs

1. If the situation can be fixed while job is held (e.g., you forgot to create directory for output):
   a. Fix the situation
   b. Release the job(s): `condor_release JOB_IDs`

2. Otherwise (and this is common):
   a. Remove the held jobs: `condor_rm JOB_IDs`
   b. Fix the problems
   c. Re-submit
General Troubleshooting Tips

- Comparing expectations vs. what happened: Either might be wrong!
- Read messages carefully — even if some parts make no sense, what hints can you get?
- Search online … but evaluate what you find
- Collect links and other resources that help
- Ask for help! And provide key details: versions, commands, files, messages, logs, etc.
- Always keep the log, error and condor output file
$ condor_submit job.sh
Submitting job(s)
ERROR: on Line 6 of submit file:
ERROR: Failed to parse command file (line 6).

• Completely failed to submit!
• **Notice:** Failed to parse
• **Why:** You tried to submit your executable (or other file), not an HTCondor submit file
• **Fix:** Submit an HTCondor submit file (e.g., .sub)
Issue: Typos in Submit File

$ condor_submit sleep.sub
Submitting job(s)

- ERROR: No 'executable' parameter was provided
- ERROR: Parse error in expression: RequestMemory = 1BG
- ERROR: Executable file /bin/slep does not exist

- Also failed to submit (missing job(s) submitted)
- Why: Typos in your submit file (e.g., BG for GB)
- Fix: Correct typos!
### Issue: Jobs Idle for a Long Time

Jobs are *idle* for a *long* time – *can be hard to judge!*

```
$ condor_q -analyze <JobId>
$ condor_q -better-analyze <JobId>
```

```
$ condor_q -better-analyze 123456.0
...
<table>
<thead>
<tr>
<th>Step</th>
<th>Slots</th>
<th>Matched</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0]</td>
<td>13033</td>
<td></td>
<td>TARGET.PoolName == &quot;OSPool&quot;</td>
</tr>
<tr>
<td>[9]</td>
<td>13656</td>
<td></td>
<td>TARGET.Disk &gt;= RequestDisk</td>
</tr>
<tr>
<td>[11]</td>
<td>0</td>
<td></td>
<td>TARGET.Memory &gt;= RequestMemory</td>
</tr>
</tbody>
</table>
```
Issue: Missing or Unexpected Results

- Job runs … but something does not seem right
  - Short or zero-length output file(s)
  - Very short runtime (almost instant)
- May be problems with app, inputs, arguments, …
  - Check log files for **unexpected exit codes**, etc.
  - Check output and error files for messages from app
  - Can’t find anything? Add more debugging output
Issue: Badput

• What is *badput*?
  • Basically, wasted computing
    • Job runs for 97 *minutes*, gets kicked off, starts over on another server
    • Job runs for 97 *minutes*, is removed
  • Not jobs that must be re-run due to code changes! (that’s just part of science, right?)
• Badput uses resources that others could have used
• If contacted, help us help you and others!
Tips for Avoiding Badput

• Always test with a **small set of jobs** before scaling up. (This practice applies to any modifications made to a **tried and tested** code as well)

• Monitor your jobs memory and disk usage
  
  ```bash
  condor_q <jobid> -af RequestMemory MemoryUsage |sort |uniq --c
  condor_q <jobid> -af RequestDisk DiskUsage |sort |uniq --c
  ```

• Have an idea/expectation about the software/code’s limit- e.g. **Segfault**.

• Have a general idea about the inner workings of the software and libraries.
More Troubleshooting Resources

• Lauren Michael & Tim Cartwright’s OSG User School 2021 talk
• OSG Connect Documentation
• support@opensciencegrid.org
Acknowledgements
Acknowledgements

• OSG team, especially Tim Cartwright, Lauren Michael & Christina Koch

• This work was supported by NSF grants MPS-1148698, OAC-1836650, and OAC-2030508
You Can Acknowledge OSG!

If you publish or present results that benefitted from using OSG, please acknowledge us!

https://osg-htc.org/acknowledging