



# Self-Checkpointing

**Tim Cartwright**

*OSG User School Director & OSG Special Projects Manager  
University of Wisconsin–Madison*



# Why?

---

- Suppose your job will run for a long time
  - Reminder: Look at the “Ideal Jobs” table
  - But let’s say more than about 8 hours
- May be kicked off of execute point before done: HTCondor will restart job somewhere else
- But! It starts over and loses all progress (*badput*)



# How?

---

- **Ideal solution:** Break up job into shorter pieces
  - Try to get back into that “Ideal Jobs” column
- But this does not always work; for example, when one iteration depends on the previous one
- **Another solution — self-checkpointing:**
  - Periodically write state (checkpoint) to disk & *restart*
  - State must be sufficient to restart job *at that point*
  - Code itself must know to look for checkpoint data
  - May need a wrapper script to accomplish



# When?

---

- Balance overhead versus (risk of) wasted compute
  - Writing to disk is slow (relatively) and restarts take time
  - Test early! Collect metrics (checkpoint & restart times)
- Look for natural checkpoint times
  - Generally, when there is the least data to write
  - Often between outermost iterations
  - Could use iteration count, time, ...
- Save only what you need!  
(Must be transferred back to access point each time)



# HTCondor Has 2 Ways to Checkpoint

- **Exit-driven self-checkpointing**
  - Newer: Need HTCondor  $\geq 8.9.7$  (CHTC & OS Pool do!)
  - *Waaaay* better for most use cases, esp. in OSG
  - What is shown here
- Eviction-driven self-checkpointing
  - Not even worth talking about for OSG!
  - Documented in the HTCondor Manual
  - But don't use it 😊



---

# Technical Details



# HTCondor Submit File Changes

- Tell HTCondor what special exit code your software will use when checkpointing (85 is suggested):

```
checkpoint_exit_code = 85
```

- Tell HTCondor what files (on the execute point) to save (on access point) and restore *if moved to new execute point* — list files and directories, maybe including output file(s) (if cumulative):

```
transfer_checkpoint_files = foo.txt, ...
```



# Example Submit File

---

```
executable = my_software
transfer_input_files = my_input.txt
transfer_checkpoint_files = my_output.txt, temp_dir, temp_file.txt
transfer_output_files = my_output.txt
```

```
request_cpus = 1
request_memory = 1GB
request_disk = 1GB
```

```
log = example.log
output = example.out
error = example.err
```

```
checkpoint_exit_code = 85
```

```
queue
```





# Code Changes: Writing a Checkpoint

- Simple example – one-variable parameter sweep
  - Save function *overwrites* its output each iteration
  - Designed to save checkpoint every 1000th iteration

```
def save_checkpoint(iteration):  
    cp_file = open(checkpoint_path, 'w')  
    cp_file.write('%d\n' % (iteration))  
    sys.exit(85)  
  
# ...  
for iteration in xrange(start, end + 1):  
    if ((iteration - start + 1) % 1000) == 0:  
        save_checkpoint(iteration)  
    do_science(iteration)  
sys.exit(0)
```



# Code Changes: Using a Checkpoint

- Continuation of previous example... reading command-line arguments and using the checkpoint file

```
start, end = map(int, sys.argv[1:])  
  
if os.path.exists(checkpoint_path):  
    cp_file = open(checkpoint_path, 'r')  
    cp_data = cp_file.readlines().strip()  
    cp_file.close()  
    checkpoint_iteration = int(cp_data)  
    if checkpoint_iteration >= start:  
        start = checkpoint_iteration  
    else:  
        # Potential problem?
```



# Other Details

---

- You may be able to view your checkpoint files on the access point — see the HTCondor Manual (using the last 4 digits of your Cluster ID?)
- After transferring your checkpoint files, HTCondor immediately tries to restart your job *in place* — without changing anything
- If evicted and restarted elsewhere, the remote job directory will contain:
  - executable
  - transfer\_input\_files
  - transfer\_checkpoint\_files
- Today, need to explicitly checkpoint `_condor_stdout` and `_condor_stderr`



---

# Step-by-Step Example



# Example Step 1: Before Submit

## Submit Directory

```
my_software  
my_input.txt  
my_submit.sub
```

```
executable = my_software  
transfer_input_files = my_input.txt  
transfer_checkpoint_files = my_output.txt, temp_dir,  
                           temp_file.txt  
transfer_output_files = my_output.txt  
  
request_cpus    = 1  
request_memory  = 1GB  
request_disk    = 1GB  
  
log             = zzz.log  
output          = zzz.out  
error          = zzz.err  
  
checkpoint_exit_code = 85  
  
queue
```



# Example Step 2: Just Before Execute

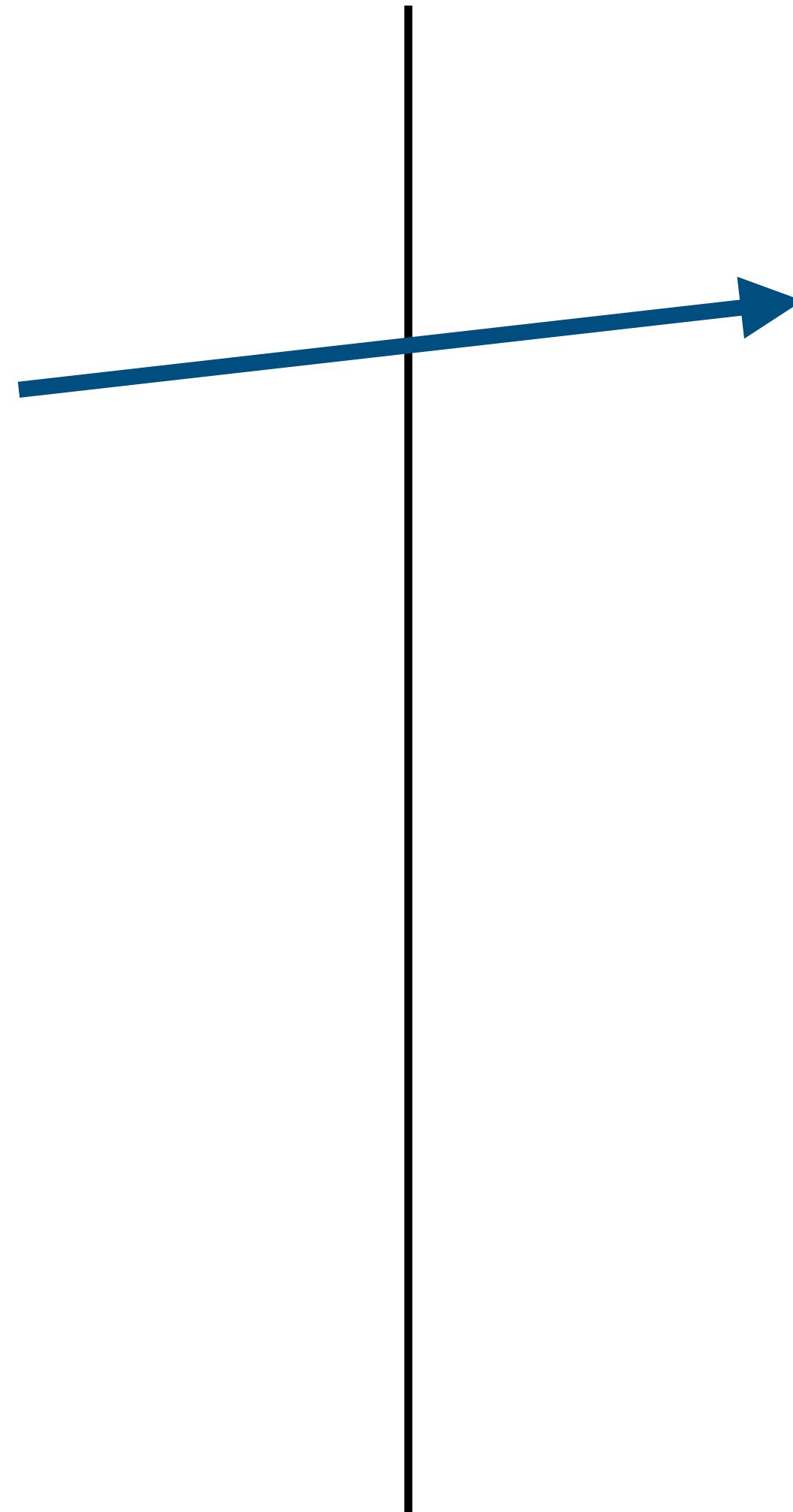
## Submit Directory

```
my_software  
my_input.txt  
my_submit.sub  
zzz.log
```

## Spool Directory

## Execute Directory

```
my_input.txt  
my_software
```





# Example Step 3: After 1 Minute

## Submit Directory

```
my_software  
my_input.txt  
my_submit.sub  
zzz.log
```

## Spool Directory

## Execute Directory

```
my_input.txt  
my_output.txt  
my_software  
_condor_stderr  
_condor_stdout  
temp-dir/1.txt  
temp-dir/2.txt  
temp-file.txt  
trash.txt
```



# Example Step 4: After 1 Hour – exit(85)

## Submit Directory

```
my_software  
my_input.txt  
my_submit.sub  
zzz.log
```

## Spool Directory

## Execute Directory

```
my_input.txt  
my_output.txt  
my_software  
_condor_stderr  
_condor_stdout  
temp-dir/42.txt  
temp-dir/43.txt  
temp-file.txt  
trash.txt
```





# Example Step 5: Checkpoint Complete

```
transfer_checkpoint_files = my_output.txt, temp-dir, temp-file.txt
```

## Submit Directory

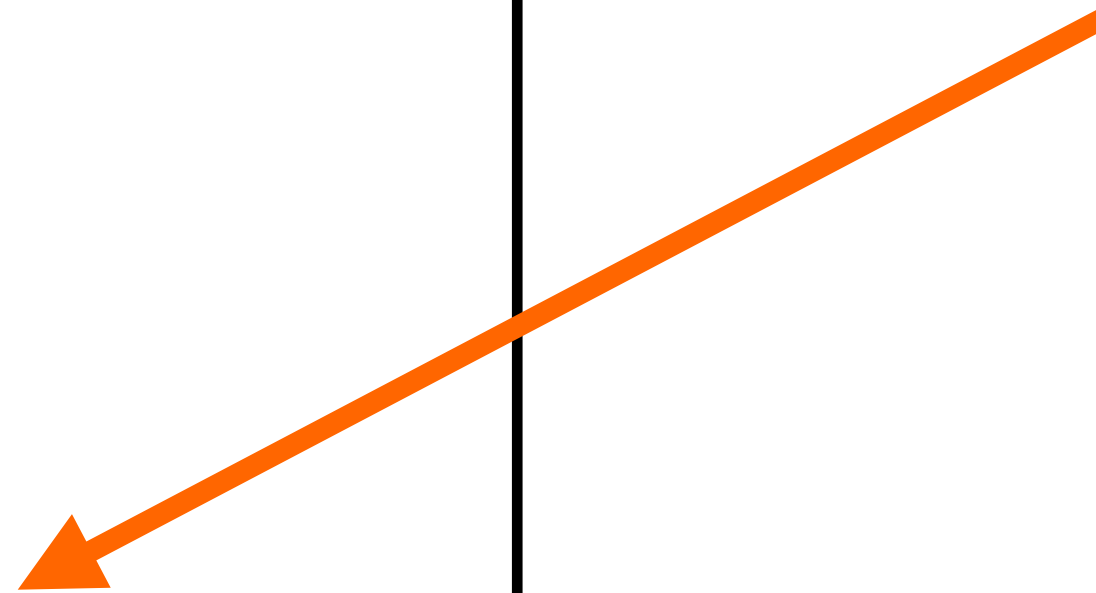
```
my_software  
my_input.txt  
my_submit.sub  
zzz.log
```

## Spool Directory

```
my_output.txt  
temp-dir/42.txt  
temp-dir/43.txt  
temp-file.txt
```

## Execute Directory

```
my_input.txt  
my_output.txt  
my_software  
_condor_stderr  
_condor_stdout  
temp-dir/42.txt  
temp-dir/43.txt  
temp-file.txt  
trash.txt
```



**Job execute directory is not changed before restart.**



# Example Step 6: 10 Min. Later – Eviction!

## Submit Directory

```
my_software  
my_input.txt  
my_submit.sub  
zzz.log
```

## Spool Directory

```
my_output.txt  
temp-dir/42.txt  
temp-dir/43.txt  
temp-file.txt
```

## Execute Directory

```
my_input.txt  
my_output.txt  
my_software  
_condor_stderr  
_condor_stdout  
temp-dir/51.txt  
temp-dir/52.txt  
temp-file.txt  
trash.txt
```

**Lose changes since last checkpoint**



# Example Step 7: Restart on New Execute

## Submit Directory

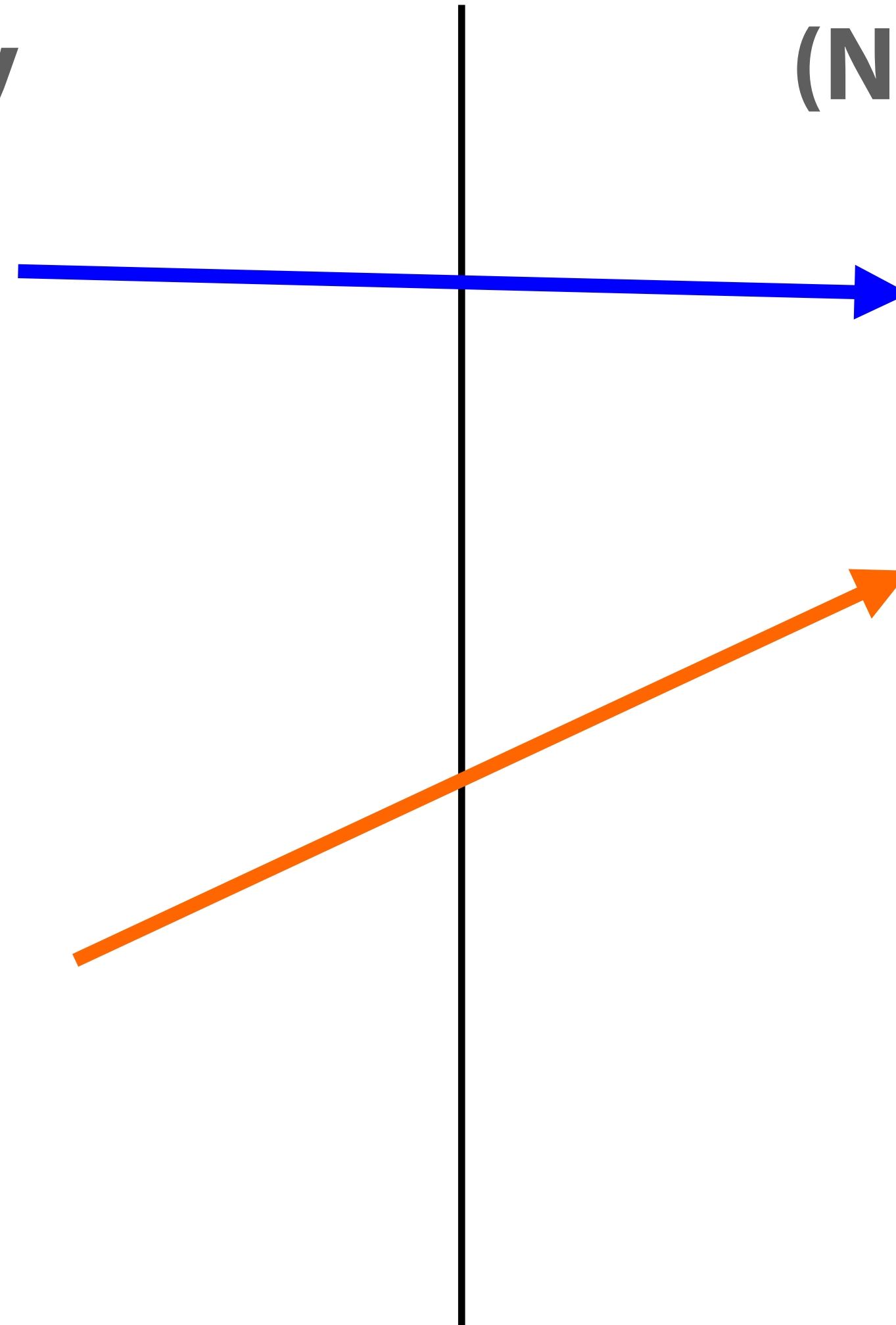
```
my_software  
my_input.txt  
my_submit.sub  
zzz.log
```

## Spool Directory

```
my_output.txt  
temp-dir/42.txt  
temp-dir/43.txt  
temp-file.txt
```

## (New) Execute Directory

```
my_input.txt  
my_output.txt  
my_software  
temp-dir/42.txt  
temp-dir/43.txt  
temp-file.txt
```





# Example Step 8: Job Completes Normally

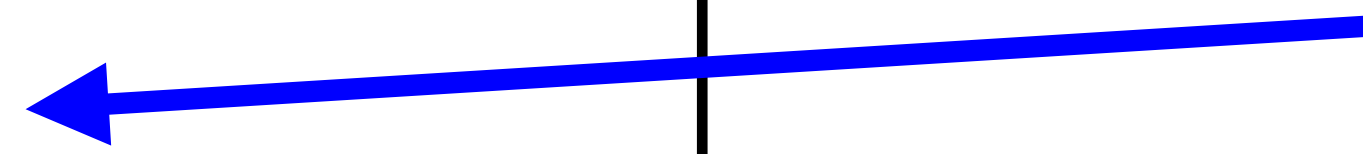
```
transfer_output_files = my_output.txt
```

## Submit Directory

```
my_software  
my_input.txt  
my_output.txt  
my_submit.sub  
zzz.err  
zzz.log  
zzz.out
```

## (New) Execute Directory

```
my_input.txt  
my_output.txt  
my_software  
_condor_stderr  
_condor_stdout  
temp-dir/98.txt  
temp-dir/99.txt  
temp-file.txt  
trash.txt
```





# Notes & Acknowledgements

---

- Official documentation:
  - <https://htcondor.readthedocs.io/en/latest/users-manual/self-checkpointing-applications.html>
  - Includes full working example (Python + submit)
  - The exercise is derived from that example
- Many thanks to Todd Miller, Christina Koch, and Jason Patton for their help!
- This work was supported by NSF grants MPS-1148698, OAC-1836650, and OAC-2030508