

Mini-Topic: Self-Checkpointing

Tim Cartwright

OSG Project Manager

University of Wisconsin–Madison



Why and How?

- Suppose your job will run for a long time (> 8 h?)
- May be preempted
- HTCondor will re-run job
- But that means it starts over — lose all progress
- **One solution:**
 - 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 vs. (risk of) wasted compute
 - Writing to disk is slow (relatively) and restarts take time
 - If checkpoints are small and restarts fast, code can checkpoint more often
- Look for natural checkpoint times
 - Generally, when there is the least data to write
 - Often between outermost iterations
 - Could base on iteration count, time, ...
- Save only what you need



HTCondor Tweaks

- Must tell HTCondor what special exit code your software will use when checkpointing

```
checkpoint_exit_code = 77
```

- When your executable – maybe wrapper – exits:
 - HTCondor transfers checkpoint file to submit
 - Immediately tries to restart job in place
- If using **transfer_output_files**, include checkpoint!



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)) # See Notes  
    sys.exit(77)  
  
# ...  
for iteration in xrange(start, end + 1):  
    do_science(iteration)  
    if ((iteration - start + 1) % 1000) == 0:  
        save_checkpoint(iteration)
```



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()  
    cp_start = int(cp_data)  
    if cp_start >= start:  
        start = cp_start  
    else:  
        # Potential problem?
```



- Depends on HTCondor version 8.9.7
 - CHTC pool (**Learn**) has this already
 - OSG Open Science pool pilots (OSG Connect) still on version 8.8.8 — so, coming soon!
- Official documentation:
 - <https://htcondor.readthedocs.io/en/latest/users-manual/self-checkpointing-applications.html>
 - Includes full working example (Python + submit)