PU!
Setting up parallel universe in your pool and when (not!) to use it

HTCondor Week 2018 – Madison, WI

Jason Patton (jpatton@cs.wisc.edu)

Center for High Throughput Computing
Department of Computer Sciences
University of Wisconsin-Madison

Imagine some software…

- Requires more resources than a single execute machine can provide, or
- Needs a list of machines prior to runtime, or
- Assumes child processes will run (and exit) on all machines at the same time

Examples:

- MPI
- Master-Worker frameworks (some, not all)
- Server-Client testing (networking, database)

What is parallel universe?

- All slots for a job are claimed by the “dedicated scheduler” before the job runs
- Each slot is given a node number (`$(NODE)`)
- Execution begins simultaneously
- By default, all slots terminate when the executable on the "Node 0” slot exits
- Slots share a single job ad and a spool directory on the submit machine (for `condor_chirp`)

Use parallel universe when a job…

- Cannot be made to fit on a single machine
- Needs a list of machines prior to runtime
- Needs simultaneous execution on slots

Classic example: You have a MPI job that cannot fit on one machine, and you don't have a HPC cluster.

Example helper script for Open MPI: `openmpiscript`

Don't use parallel universe…

- **When submitting MPI jobs that could be made to fit on a single machine**
- Break these up in to multicore vanilla universe jobs… MPI works well on single machines (core binding, shared memory, single fs, etc.)

Example parallel universe job life cycle

1. `machine_count = 8`
2. Dedicated scheduler claims idle slots (slots become **Claimed/Idle**) until it has 8 slots that match job requirements
3. Job execution begins on all slots simultaneously
4. Processes on all slots terminate when the process on node 0 exits
5. Slots return to **Claimed/Idle** state

Example parallel universe job

```
universe = parallel

executable = setup.sh
arguments = $(NODE)
transfer_input_files = master.sh,worker.sh

output = out.$(CLUSTER).$(NODE)
error = err.$(CLUSTER).$(NODE)
log = log.$(CLUSTER)

request_cpus = 1
request_memory = 1G

machine_count = 8

queue
```

queue 2?

setup.sh

```
#!/usr/bin/env bash

node=$1

# check if on node 0
if (( $node == 0 )); then
   # run master program
   ./master.sh
else
   # run worker program
   ./worker.sh
fi
```

Example parallel universe job life cycle

```
$ condor_status
Name               State     Activity
slot1@execute1 Claimed   Busy
slot2@execute1 Claimed   Busy
slot3@execute1 Unclaimed Idle
slot4@execute1 Claimed   Busy
slot1@execute2 Unclaimed Idle
slot2@execute2 Unclaimed Idle
slot3@execute2 Claimed   Busy
slot4@execute2 Unclaimed Idle
slot1@execute3 Unclaimed Idle
slot2@execute3 Unclaimed Idle
```

Job Submitted

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Unclaimed Idle
slot4@execute1 Claimed    Busy
slot1@execute2 Unclaimed Idle
slot2@execute2 Unclaimed Idle
slot3@execute2 Claimed    Busy
slot4@execute2 Unclaimed Idle
slot1@execute3 Unclaimed Idle
slot2@execute3 Unclaimed Idle
```

Job Submitted

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Claimed    Idle
slot4@execute1 Claimed    Busy
slot1@execute2 Claimed    Idle
slot2@execute2 Claimed    Idle
slot3@execute2 Claimed    Busy
slot4@execute2 Claimed    Idle
slot1@execute3 Claimed    Idle
slot2@execute3 Claimed    Idle
```

Negotiation Cycle #1

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Claimed    Idle
slot4@execute1 Claimed    Busy
slot1@execute2 Claimed    Idle
slot2@execute2 Claimed    Idle
slot3@execute2 Claimed    Busy
slot4@execute2 Claimed    Idle
slot1@execute3 Claimed    Idle
slot2@execute3 Claimed    Idle
```

Negotiation Cycle #2

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Claimed    Idle
slot4@execute1 Unclaimed  Idle
slot1@execute2 Claimed    Idle
slot2@execute2 Claimed    Idle
slot3@execute2 Claimed    Busy
slot4@execute2 Claimed    Idle
slot1@execute3 Claimed    Idle
slot2@execute3 Claimed    Idle
```

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Claimed    Idle
slot4@execute1 Claimed    Idle
slot1@execute2 Claimed    Idle
slot2@execute2 Claimed    Idle
slot3@execute2 Claimed    Busy
slot4@execute2 Claimed    Idle
slot1@execute3 Claimed    Idle
slot2@execute3 Claimed    Idle
```

Negotiation Cycle #3

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Claimed    Idle
slot4@execute1 Claimed    Idle
slot1@execute2 Claimed    Idle
slot2@execute2 Claimed    Idle
slot3@execute2 Claimed    Busy
slot4@execute2 Claimed    Idle
slot1@execute3 Claimed    Idle
slot2@execute3 Claimed    Idle
```

Negotiation Cycle #4

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Claimed    Idle
slot4@execute1 Claimed    Idle
slot1@execute2 Claimed    Idle
slot2@execute2 Claimed    Idle
slot3@execute2 Claimed    Busy
slot4@execute2 Claimed    Idle
slot1@execute3 Claimed    Idle
slot2@execute3 Claimed    Idle
```

Negotiation Cycle #5

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1 Unclaimed Idle
slot2@execute1 Claimed   Busy
slot3@execute1 Claimed   Idle
slot4@execute1 Claimed   Idle
slot1@execute2 Claimed   Idle
slot2@execute2 Claimed   Idle
slot3@execute2 Claimed   Busy
slot4@execute2 Claimed   Idle
slot1@execute3 Claimed   Idle
slot2@execute3 Claimed   Idle
```

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1      Claimed    Idle
slot2@execute1      Claimed    Busy
slot3@execute1      Claimed    Idle
slot4@execute1      Claimed    Idle
slot1@execute2      Claimed    Idle
slot2@execute2      Claimed    Idle
slot3@execute2      Claimed    Busy
slot4@execute2      Claimed    Idle
slot1@execute3      Claimed    Idle
slot2@execute3      Claimed    Idle
```

Negotiation Cycle #6

Example parallel universe job life cycle

```
$ condor_status
Name                 State      Activity
slot1@execute1 Claimed    Busy
slot2@execute1 Claimed    Busy
slot3@execute1 Claimed    Busy
slot4@execute1 Claimed    Busy
slot1@execute2 Claimed    Busy
slot2@execute2 Claimed    Busy
slot3@execute2 Claimed    Busy
slot4@execute2 Claimed    Busy
slot1@execute3 Claimed    Busy
slot2@execute3 Claimed    Busy
```

Job Starts

Example parallel universe job life cycle

```
$ condor_status
Name                State      Activity
slot1@execute1      Claimed    Idle
slot2@execute1      Claimed    Busy
slot3@execute1      Claimed    Idle
slot4@execute1      Claimed    Idle
slot1@execute2      Claimed    Idle
slot2@execute2      Claimed    Idle
slot3@execute2      Claimed    Busy
slot4@execute2      Claimed    Idle
slot1@execute3      Claimed    Idle
slot2@execute3      Claimed    Idle
```

Job Completes

Example parallel universe job life cycle

```
$ condor_status
Name               State      Activity
slot1@execute1 Unclaimed Idle
slot2@execute1 Claimed   Busy
slot3@execute1 Unclaimed Idle
slot4@execute1 Unclaimed Idle
slot1@execute2 Unclaimed Idle
slot2@execute2 Unclaimed Idle
slot3@execute2 Claimed   Busy
slot4@execute2 Unclaimed Idle
slot1@execute3 Unclaimed Idle
slot2@execute3 Unclaimed Idle
```

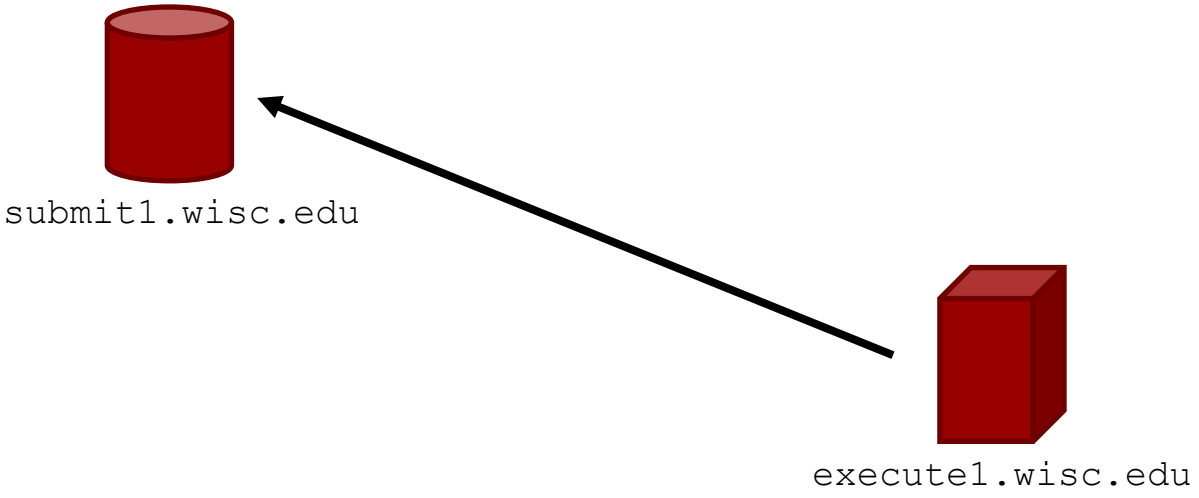
10 minutes later

Enabling parallel universe in your pool

1. Choose a submit machine to host the "dedicated scheduler"
2. Set `DedicatedScheduler` on participating execute machines
3. Adjust other settings (`START`, `RANK`, `PREEMPT`, etc.) to taste
4. Easy way – modify the example config:

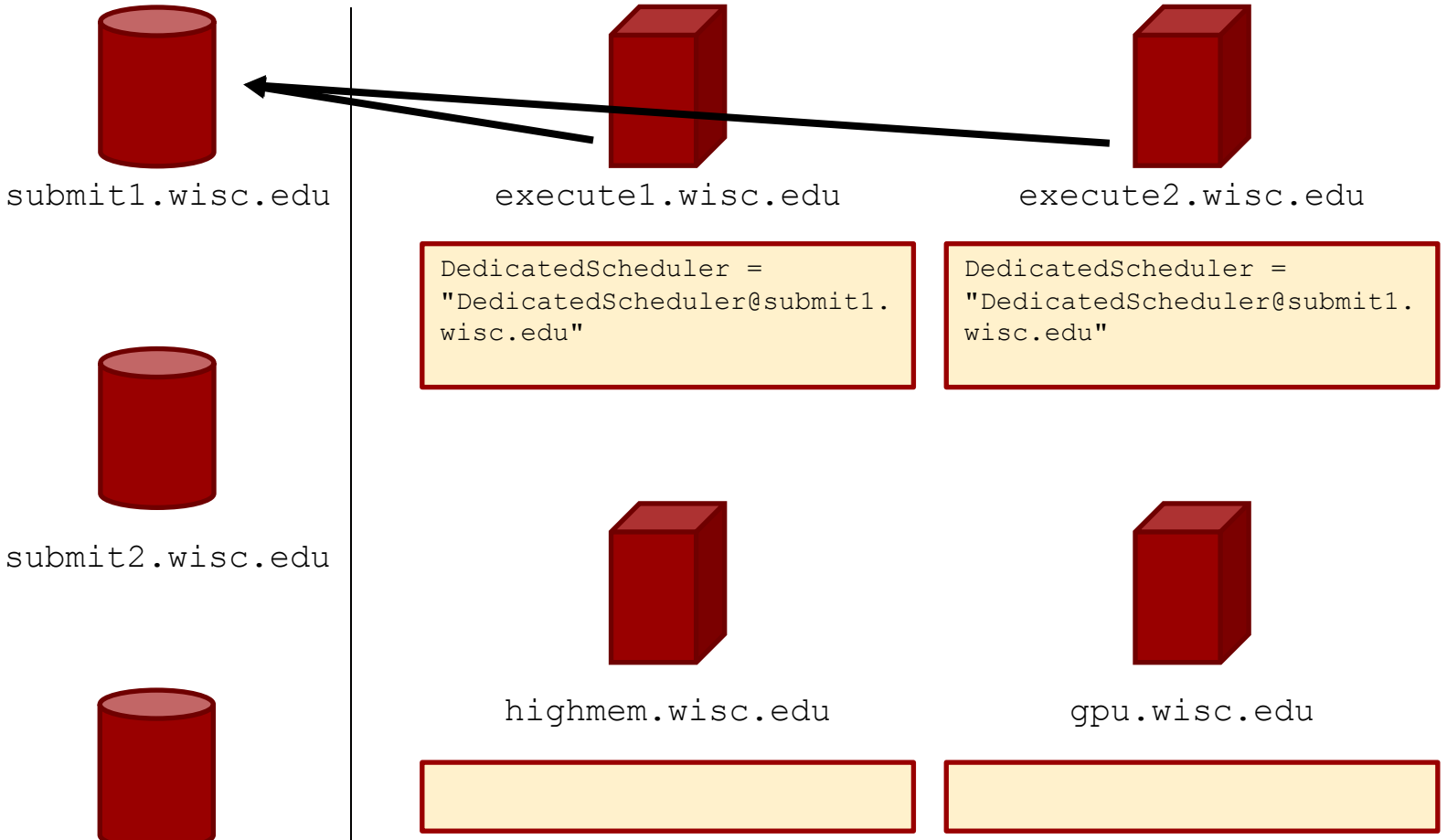
`condor_config.local.dedicated.resource`

Example config




```
DedicatedScheduler = "DedicatedScheduler@submit1.wisc.edu"
START = (Scheduler =?= $(DedicatedScheduler)) || ($(START))
PREEMPT = Scheduler =!= $(DedicatedScheduler) && ($(PREEMPT))
SUSPEND = Scheduler =!= $(DedicatedScheduler) && ($(SUSPEND))
RANK = Scheduler =?= $(DedicatedScheduler)
```

Example config



Don't enable parallel universe…

- If you are particularly concerned about reduced throughput in your pool
  - Claimed/Idle slots when PU jobs are being scheduled and completed
  - The dedicated scheduler may not schedule dynamic slot claims efficiently
  - If you're not careful about where PU jobs can land, slow networks can hurt performance, see `ParallelSchedulingGroup` in manual
  - Preemption hurts total throughput if enabled

Other config notes

- Can adjust how long dedicated scheduler holds on to Claimed/Idle slots
  - `UNUSED_CLAIM_TIMEOUT`, see example `condor_config.local.dedicated.submit`
- PU jobs usually talk between slots, check firewall settings
- PU jobs may be sensitive to shared filesystems and user names

Parallel Universe Trivia

- Can you submit PU jobs without your admin having configured your pool for them?
  - No. (Yes, will sit idle while dedicated scheduler searches for nonexistent dedicated resources.)
- Should all MPI jobs use PU?
  - No, only if they cannot fit on a single machine.
- Can you submit Docker jobs using PU?
  - Yes!

```
universe = docker
WantParallelScheduling = true
```

CENTER FOR HIGH THROUGHPUT COMPUTING

HTCondor

Questions?

Example `examples/` location:

`/usr/share/doc/condor-8.7.8/examples`