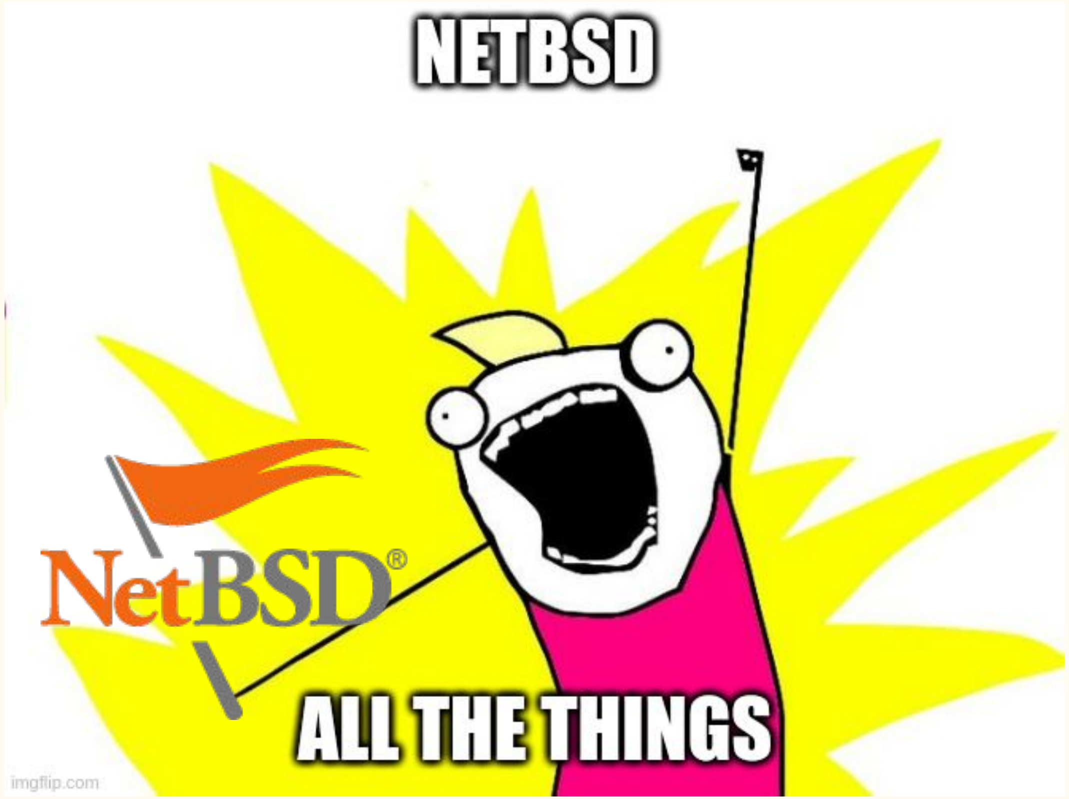
HTCondor + OpenMPI

Zach McGrew

Introductions

- I’m Zach McGrew
- "Computational Science Administrator"*
    - *Naming is one of those two really hard computer science problems…
    - Fun way of saying Cluster Admin
- WWU BS CS 2017
- WWU MS CS 2018
    - Phil Nelson advisor - NetBSD RISC-V port

Side Note



Getting Help

- https://cluster.cs.wwu.edu
- Email: csaw.support@wwu.edu
- In Person : CF412

Cluster Overview

The Pools (or… Clusters)

- CSE Cluster (College)
- CSCI Cluster (us)
- CSCI-Lab (including the lab you're sitting in)
  - Extra rules required here, and no parallel universe *(foreshadowing!)*

Head Nodes (APs — Access Points)

- cse-head.cluster.cs.wwu.edu
- csci-head.cluster.cs.wwu.edu
- csci-lab-head.cs.wwu.edu
    - Note not part of the cluster.cs subdomain!

CSE Cluster (EPs — Execute Points)

- Comprised of 20 compute nodes
  - many connected via InfiniBand (40Gb/s)
  - Hardware ages vary, but it’s getting a bit old
  - Avoid running here, prefer the CSCI Cluster
- + 3 researcher purchased boxes (access allowed with preemption)

CSCI Cluster (EPs — Execute Points)

- Comprised of 8 compute nodes
  - All connected via InfiniBand (100Gb/s)
  - Dual socket, 16 cores / 32 threads per processor
    - Linux treats this as "64 CPUs", HTCondor agrees
  - 192 GB RAM
- + 3 4-way GPU nodes (2080 Ti)
- + 3 8-way GPU nodes (2080 Ti)
- + 1 3-way H100 GPU node

CSCI Lab ~~Cluster~~ Pool (EPs — Execute Points)

- Comprised of all academic lab systems in the department
  - CF162
  - CF164
  - CF165
  - CF167 (when it reopens)
  - CF405
  - CF420
  - KB307
  - KB311

HTCondor

HTCondor

- HTCondor controls everything in the cluster.
  - More on this next
- You can only log into an AP (csci-head), not any of the EPs.
  - You can get an interactive session via HTCondor though

WARNING

Warning

- Do not run things directly on the head node
    - Non-abusive text editors allowed
- No development tools installed here on purpose
    - No `git`, `gcc`, etc.
- ***<u>VSCode Remote not allowed here</u>***

Connecting

- We'll be using csci-head.cluster.cs.wwu.edu
    - Login with your WWU Universal Account

```
$ ssh csci-head.cluster.cs.wwu.edu
```

Building MPI code

Building MPI Code

- Get an interactive session on an EP

```
condor_submit -i request_cpus=2 request_memory=2048 getenv=true
```

- `-i` - Interactive job
- `request_cpus` - Yes please
- `request_memory` - In MB, suffix no longer allowed =(
    - Don't worry; Phil has you covered. Do math in `bc`
- `getenv=true` - Keep all my environment variables
    - I need to know `$USER, $HOME, $PATH,` etc.
    - Set this for interactive jobs, avoid for normal jobs

HTCondor job environment overview

- HTCondor will setup a cgroup to contain all of your processes
- Memory limits are "soft"
  - You can use more than you request, until someone else actually requests it and uses it
- It will also assign CPU affinity to limit you to the number of CPUs you requested.
  - Ex: You request 2 CPUs, but create 64 threads. You're gonna have a bad time.
  - Do not set CPU affinity via mpirun or your OpenMP code
- HTCondor will give you hints about your environment by setting environment variables automagically
  - Sometimes helpful, sometimes not
- HTCondor will lie about some things.
  - Namely, `/tmp` and `/var/tmp`

Building MPI Code

Copy the example to your home directory

```
cp -r /cluster/academic/CSCI415/202520/mpi ~/
```

Building MPI Code

- MPI is available as module in our environment
  - `module avail` — list the modules
  - `module load X` — load module X

- `module load mpi`
  - Now you have access to all the mpi* and ompi* commands and libraries

Building MPI Code

- This demo has a `Makefile`
  - And a `build.sh` which loads the mpi module, and then invokes make
  - For the lazy and forgetful, like myself

- `make`
  - `*wait a sec or two while it compiles*`

Test run

- `Makefile` also includes a `run` target

- `make run`

Back to the AP!

Submitting HTCondor Jobs

HTCondor Universes

- Vanilla
- Container
- Docker
- VM
- Java
- Parallel
- etc.

HTCondor Universes for MPI - Vanilla

- Copy executable to EP
- Start executable on EP
- Good luck!
  - You got this
  - Probably?

- Runs on exactly one node

- Advantages
  - FAST!
  - For MPI, communication between processes happens via shared memory

HTCondor Universes for MPI - Parallel

- Allocate X number of requested slots matching your resource request
    - May or may not be on multiple EPs
- Copy executable to allocated EPs
- Run same executable on all EPs at once
    - Wait… This seems… Bad?
- Good luck?!?!

HTCondor Universes for MPI - Parallel

- Advantages
  - Access to waaaaaay more cores and memory.
  - 100Gb/s transfers is pretty fast still…

- Disadvantages
  - Requires coordination between various nodes, and some planning
  - MPI’s got your back on sending messages
  - But who has MPI’s back on knowing how and where to talk?

Wrapper scripts to save the day!

- In reality, neither of these universes really matter.
  - Choose the universe that fits your needs
    - Need quicker compute? Vanilla
    - Need more cores and memory? Parallel

- I took an awesome wrapper script written by Jason Patton of the HTCondor project and extended it to work in both universes.

- It also sets all the magic MPI settings you need to use the InfiniBand or shared memory when running on the local node in our cluster.

Vanilla Job Demo

Parallel Job Demo

HTCondor Cheatsheet

HTCondor Commands

- `condor_submit`

- `condor_q`
- `condor_q -all`
- `condor_q -hold`

- `condor_rm`

- `condor_status`

Logs

- STDERR from each EP is separate (`logs/err.X`)
- STDOUT is combined (`logs/out.0` — only useful one)
- Condor log is more useful for me, but you can read it too!
    - Keeps track of resource usage
    - Where your job ran

Getting Help

- https://cluster.cs.wwu.edu
- Email: csaw.support@wwu.edu
- In Person : CF412