GPU on WCWC and SDCC

Brett Viren

March 6, 2026

WCWC - Wire-Cell Workstation Cluster

Two main "workstations" with GPUs: `wcgpu0` and `wcgpu1`, each has:

- GPUs: $2\times$ RTX 4090, 24 GB VRAM
- CPU: AMD Ryzen Threadripper 7970X 32-Cores
- 256 GB CPU RAM, $\mathcal{O}(1)$ TB fast nVME SSD storage.

Caveats: **NO BACKUPS**, SSDs are node-local, $HOME is shared but on slow NFS.

WCWC - usage policy

Guiding philosophy: **make it easy to try new things**.

- **Not intended for production processing nor operational use.**
- Rough separation of usage:

  `wcgpu0` is for Prompt-Processing work
  `wcgpu1` is for "everything else".
- We manage the shared resources through ad-hoc coordination.
  - Talk to each other to work out resource conflicts.
  - Use `nvidia-smi` to determine if and which GPUs are already busy.
  - Check `top` to see if you have orphan VSCode/node or claude.
    - ★ These use a LOT of RAM. Sometimes 100% CPU zombies.

Avoid long running GPU usage if/when it impacts other users.

WCWC - some future plans

Hardware additions

I am pursuing some hand-me-down EOL hardware from SDCC:

- $\mathcal{O}(1)$ PB HDD storage server.
- 24 GB SSD storage server.
- May need to purchase/find 10+ Gbps switch to fully benefit.

OS upgrades

- Need to upgrade from Debian 12 to 13 at some point.
- Will fix bad partitioning choices (fix tiny `/tmp` problem).
- May re-do the authentication to separate from Phy dept servers.
- Will try to combine hardware and OS upgrades at same time.

WCWC $\rightarrow$ SDCC

SDCC interactive and batch access

Need SDCC account in the “dune” group to allow:

```
$ ssh -A user@ssh.sdcc.bnl.gov  # SSH keys
$ ssh    user@dunesub01         # SSH keys or Kerberos password
$ condor_submit myjob.sub
```

Aside: an SDCC login trick

Jump right to internal node

```
bv@external$ ssh dunesub01
bvlbne@dunesub01:~>
```

With `~/.ssh/config` bits like:

```
Host dunesub01 dunesub02
     Hostname %h.sdcc.bnl.gov
Match host *.sdcc.bnl.gov
      User USERNAME           # <-- replace with your SDCC user
Match host ssh.sdcc.bnl.gov
      ProxyJump none
Match host ssh*.sdcc.bnl.gov
      ProxyJump none
Match host *.sdcc.bnl.gov
      ProxyJump ssh.sdcc.bnl.gov
      PubkeyAcceptedKeyTypes +ssh-rsa
      PubkeyAcceptedAlgorithms +ssh-rsa
```

Can do similar for campus hosts using Physics gateway or SDCC's cssh.

SDCC GPUs

```
$ bvlbne@dunesub01:~> condor_status -constraint 'TotalGpus > 0' \
  -af Machine TotalGpus GPUs_DeviceName GPUs_Capability GPUs_GlobalMemoryMb

sgpu0001.usatlas.bnl.gov 4 undefined undefined undefined
sgpu0002.usatlas.bnl.gov 4 NVIDIA A100 80GB PCIe MIG 2g.20gb undefined 19894
sgpu0003.sdcc.bnl.gov 8 NVIDIA L40S 8.9 45458
sgpu0004.sdcc.bnl.gov 8 NVIDIA L40S 8.9 45458
```

- Prompt-processing R&D proposal payed for the `sgpu0004` node.
- $1\times$ **L40S** GPU $\approx 2\times$ RTX 4090 (by VRAM and CUDA core count).
- Only 8 of 10 GPUs shown, 2 held for JupyterHub access. This allocation may change.
- Not sure if the USATLAS **A100** node(s) are accessible to "dune".

SDCC GPU node access

Prompt Processing group's policy (for now):

Any "dune" user can land jobs on the GPU node.

If/when this puts a crunch on Prompt Processing R&D we may put up some technical guardrails.

Not yet clear if there is already a "special list" for GPU node access.

- PP group members have confirmed their access.
- When you try to submit to the GPU nodes, let me know how it goes.
- Also not yet clear if we can land jobs on the other visible GPU nodes.

SDCC batch GPU jobs - minimal example HTCondor file

```
$ cat myjob.sub
num_gpus = 1

executable     = /usr/bin/nvidia-smi  # A quick test
transfer_executable = False  # False if executable not on the submit host

universe       = vanilla
output         = nvidia_smi.$(Cluster).$(Process).out
error          = nvidia_smi.$(Cluster).$(Process).err
log            = nvidia_smi.log

request_gpus   = $(num_gpus)
Requirements = (GPUs_DeviceName == "NVIDIA L40S") && (GPUs_Capability == 8.9)

should_transfer_files = YES
when_to_transfer_output = ON_EXIT
queue 1

$ condor_submit myjob.sub
```

SDCC batch GPU - allocating GPUs

Set number of GPUs you need in HTCondor job script:

```
request_gpus = 2
```

In the job's environment:

```
echo $CUDA_VISIBLE_DEVICES
4,7
```

If your job does not honor CUDA_VISIBLE_DEVICES you must translate that into some way to set the GPU IDs to use.

Never manually set CUDA_VISIBLE_DEVICES in your code/script.

SDCC storage caveats

DUNE storage is gifted to us by SDCC

Until we can pay real money, we must beg and thank you for anything we get.

Storage for DUNE on SDCC

- Shared 100 GB global for $HOME.
- Shared 10 TB global at `/gpfs/mnt/gpfs01/lbne`.
- Node-local SSD tmp during job life (potentially up to 10 TB).
- **No long term storage, NO BACKUPS**

Suggestions / comments

- FR calculations are easy for GPU batch, minimal data files.
- FM training with sub-TB of data can put files into GPFS.
  - GPFS is network so not “SSD-fast”, GPFS → local disk caching may be useful.

JupyterHub

Web UI for Jupyter Notebooks and a web-based SSH terminal.

https://dune-jupyter.sdcc.bnl.gov/

- Access CPU-only and CPU+GPU nodes.
  - Select how many CPU cores, CPU RAM and number of GPU devices.
  - Your session will only "see" what was requested.
- Can use/supply containers to provide the environment.
- This service is still in development so fasten seat belts.
  - Doug Benjamin can provide help.
  - Moving to federated login and we will need to enroll existing SDCC account in "COmanage"
  - Requires "SDCC MFA" TOTP code + SDCC Kerberos password.

https://www.sdcc.bnl.gov/information/comanage-setup-new-accounts

$\mathcal{FIN}$