OPT-175 Logbook

Goal: Get a 175B dense model up and running by any means necessary.
Purpose of this document:
To provide a source of truth of what we did, when, and why, and any context that was important to those decisions. To provide each other with a clear place to find information about what is happening without having to ping.

Instructions

- Add a dated entry for each log, in reverse chronological order.
- Entries do not have to correspond to launches, but may include notes.
- HIGHLIGHT IN RED ANYTHING THE NEXT ONCALL SHOULD ABSOLUTELY NOTICE
- For all launches, include:
  - Date
  - Remember to update the pointers at the top of this document
  - **Context of why changes were necessary (Analysis of previous run)**
    - Include tensorboard screenshots of spikes or divergences if applicable
  - Launch steps:
    - Checkpoint/log folder
    - Relevant commits
    - PR of a change to sweep script if relevant
- Oncall responsibilities are at the bottom of this doc

Spare Node Tracker

NOTE: TRY TO KEEP KNOWN GREEN NODES IN IDLE AND DRAIN ALL BAD NODES.

NodeList	# nodes	State	Notes

175B Log

NOTE: You don't need to specify the log file when launching the monitor script, it'll find it automatically based on the job id.

2022-01-06 15:47 ET [Everyone]

```
2022-01-06 15:46:44 | INFO | fairseq_cli.train | done training in 89691.5 seconds
```

2022-01-05 14:10 [Stephen + Mikel]

Timeline

1. Stephen's devfair was hosting an ssh session running Tensorboard.
   a. The devfair was taken offline for yearly maintenance, causing ssh
2. An unknown person was running monitor.py, and that was also taken offline for an unknown reason.
3. Current oncall took action to relaunch monitor.py, which immediately detected a file-not-modified issue and attempted to restart the job.
   a. Due to user error, the wrong train.log file was specified, causing a false positive on the file not modified detector.
   b. Job was put on hold

    c. fixmycloud was run
    d. We reached the part about NCCL tests and silently failed on fixmycloud, and crashing monitor.py. It did attempt to send an email (into the void

Mitigations taken:

1. Noticed 10 nodes were in drain state, mostly from a “kill task failed”
    a. We don’t always know whether this indicates a bad node or not.
2. Manually undrained and launched fixmycloud.
    a. This initially hung because pg0-8 was unresponsive but scheduled a nccl test anyway.
    b. Manually drained pg0-8 and relaunched fixmycloud
    c. Two nodes were drained for failing tests.
3. Manually ran touch train.log on the latest run to ensure time modified would be new
4. Manually removed the nodelist via sudo scontrol update job=6848 'NodeList='
5. Manually ran sudo scontrol release 6848 to resume the job
6. Current oncall ran monitor.py with updated train.log argument and verified stability

Future actions required:

1. We need to add the nccl binaries directly to the repo, and remove the get_nccl_scripts.sh file/check.
2. monitor.py should log who is running it.
3. monitor.py should use scontrol show jobid to identify the logfile automatically, rather than require manual specification
4. Emails need to be fixed or another alerting system needs to be found.
5. monitor.py should probably run via some sort of nohup or as a daemon
6. tensorboard serve should probably run via some sort of nohup

2022-01-06 10:30ish [Susan + Mikel]

Timestamps in the logs were in ET time because the last oncall had the TZ environment variable set when launching the jobs. This made it look like the job was stuck for 5 hours at first glance.

Mitigations taken: patch to ensure logs are always in UTC.

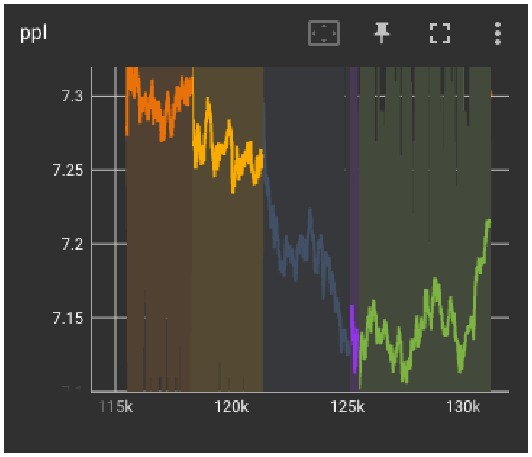
2022-01-04 15:22 ET [Sam]

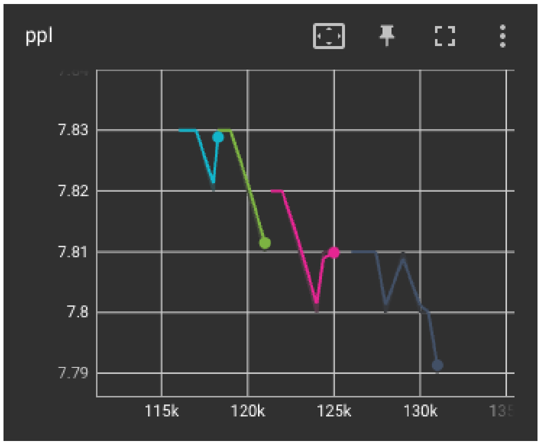
Auto-recovery failed with no email. Restarting from 2021-12-30 09:30 ET [Stephen] - Job recovery failed.

2022-01-03 05:10 ET [Daniel]

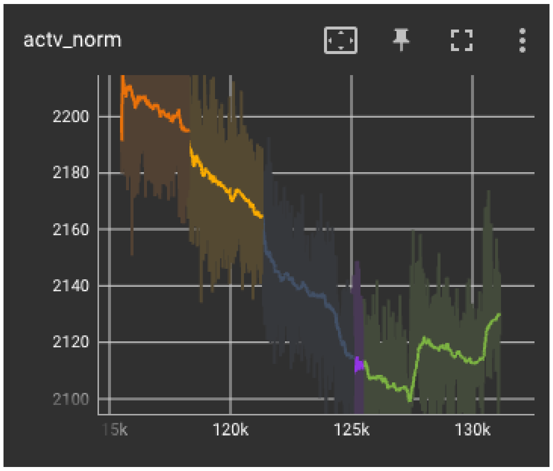
Update 12:47 ET [Stephen]: Trend is continuing. Agree with holding steady.

Train and validation ppl still show opposite trends (in the unusual way). Very confused, but based on a previous comment from Stephen in the chat I’ll leave the training untouched and wait for someone to help interpret this in the morning.



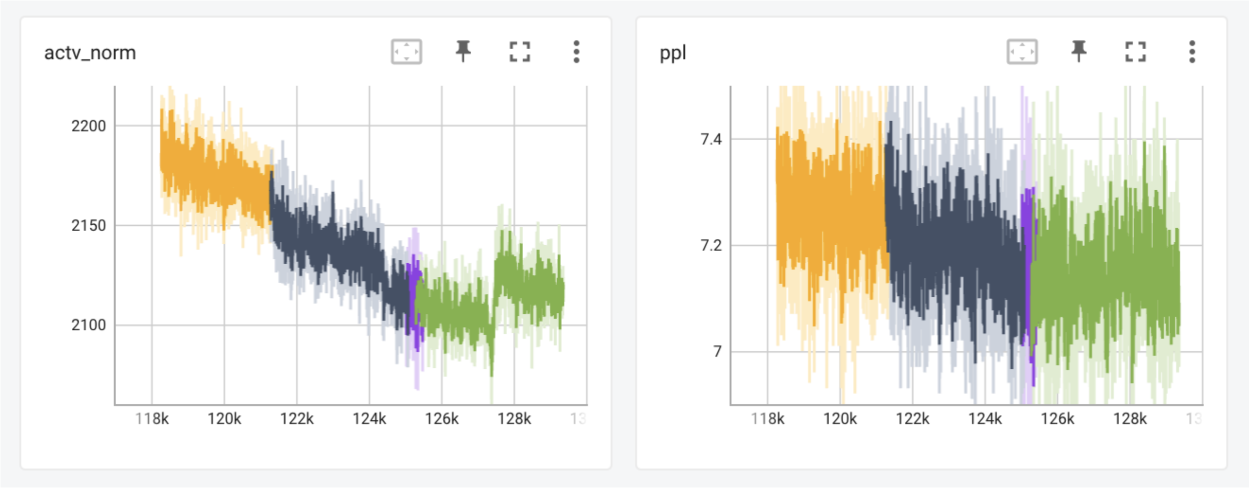


Activation norm seems to repeat an upward step pattern:

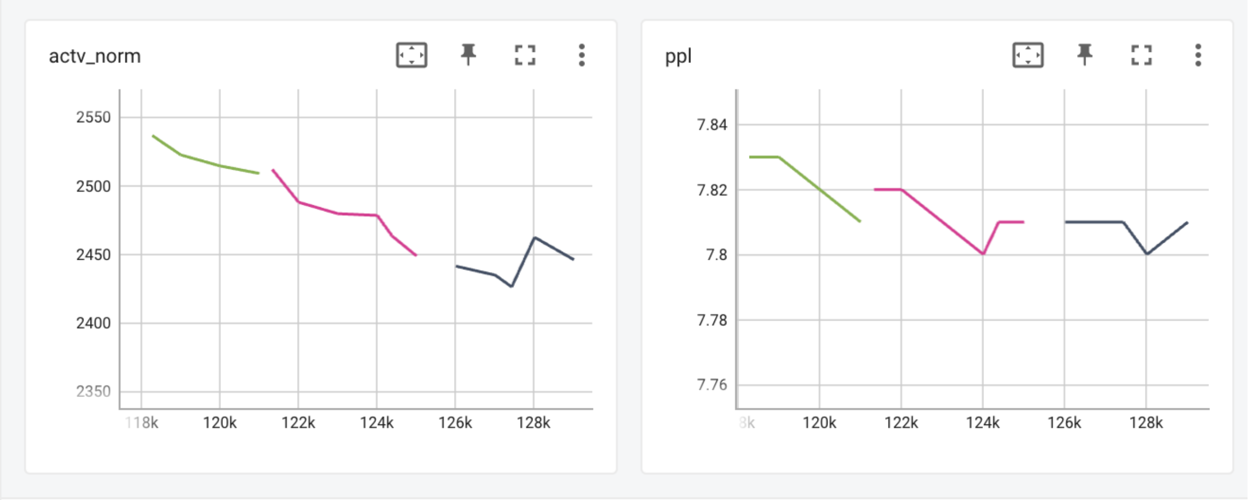


2022-01-02 17:26 ET [Stephen]

Noting that we have seen a spike in activation norms and train PPL is trending up



Validation not really affected:



2021-12-31 02:53 ET [Punit]

Started the training monitor script

```
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>@fb.com /shared/home/namangoyal/checkpoints/175B/175B_run12.57*/train.log
--modified-threshold 3600 --slurm-jobid 6214
```

2021-12-31 12:00 ET [Moya]

- Monitor script detects train.log not getting updated; tries to autorecover (6 am ET ish)
- Autorecover is successful.
- …however I had two monitor scripts running going to two separate emails (<scrubbed>)
  - Which meant recovery took twice as long. (Up at 9 am ish rather than sooner)
  - …and the emails didn't even send anyway :(
- node-[5,13] seem to be new drained nodes; undrain them and run ./fixmycloud on them to see what's up… Seems like ssh issue
  - pdsh@ip-0A1E0404: node-5: ssh exited with exit code 15
  - pdsh@ip-0A1E0404: node-13: ssh exited with exit code 15
- SSH into the two nodes run `nvidia-smi`
  - Get `Unable to determine the device handle for GPU 000B:00:00.0: GPU is lost.  Reboot the system to recover this GPU` for both
- Put node-[5,13] back into drain mode
- …and while I happened to be updating the log for this, 150 enters into a "connection timed out mode", same as drained* node.

```
hpc*      up   infinite      1 drain* node-82
hpc*      up   infinite      1  idle* node-150
hpc*      up   infinite      3  drain node-[5,13,148]
hpc*      up   infinite      1    mix node-11
hpc*      up   infinite    124  alloc node-[1-4,6-10,12,14-32,34-81,83-92,94,96-98,100-106,108-116,118-134]
hpc*      up   infinite     11   idle node-[137-146,149]
```

2021-12-30 17:00 ET [Moya] - nvidia_smi.py bug fix; machine check

- Some spares were caught under `fixmycloud` when they shouldn't have been.

- Undrained the relevant nodes (node-[5,18,47,118,120-121,149]) then reran fixmycloud after fixing the relevant bug in nvidia_smi
- Noticed infoROM corruption in one of the nodes and we've got enough spares to not need yellows, so marking as drain
    - 2021-12-30 22:58:27 WARNING nvidia_smi | node-148: infoROM is corrupted at gpu 0001:00:00.0
    - 2021-12-30 22:58:27 WARNING nvidia_smi | node-148: infoROM is corrupted at gpu 000E:00:00.0

Cluster stats after the above:

```
hpc*      up   infinite      1 drain* node-82          – [2021-12-30 8PM EST] CSP sync - Has wrong IP address. CSP looking into, UI issue
hpc*      up   infinite      1  drain node-148
hpc*      up   infinite      1    mix node-11
hpc*      up   infinite    124  alloc node-[1-4,6-10,12-17,19-32,34-46,48-81,83-92,94,96-98,100-106,108-116,119,122-134,137-140]
hpc*      up   infinite     14   idle node-[5,18,47,118,120-121,141-146,149-150]
```

2021-12-30 09:30 ET [Stephen] - Job recovery failed

- We crashed with some sort of hardware failure
- Job actually got dequeued from slurm before the auto-recovery could hold it
- As a result, the monitor script crashed and auto-recovery didn't execute
- No emails got sent.
- Manually resuming. Choosing to bump run ID bc our tensorboards are getting crowded.

Also note that I noticed the monitor script fails to auto-lock due to the original run directory being 0775. I've manually changed run directories to be 0777 for now.

Cluster status after fixmycloud:

```
hpc*      up   infinite      3 drain* node-[33,47,95]
hpc*      up   infinite      5  drain node-[82,118,120-121,149]
hpc*      up   infinite      1    mix node-11
hpc*      up   infinite    132   idle
node-[1-4,6-10,12-17,19-32,34-46,48-81,83-92,94,96-98,100-106,108-116,119,122-134,137-146,148,150]
```

Several of those drains might be false positives (ECC errors now are over aggressive, catching aggregate ones rather than recent ones)

```
# LAUNCH OF 12.57

# use updated fairscale
cd ~/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/Megatron-LM
git checkout tags/v2.6

cd ~/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp
```

```
# use previous blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RUN_ID=175B_run12.57

# Use hosts verified good from fixmycloud from above, but don't manually specify
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"

./scripts/cloud/monitor_train_log.py --mailto <scrubbed> /shared/home/namangoyal/checkpoints/175B/175B_run12.57*/train.log
--modified-threshold 3600 --slurm-jobid 6214
```

2021-12-29 13:40 ET [Stephen] - Cluster maintenance and relaunch

- Observed WPS drop
- Manually paused to run fixmycloud. Found node-95 had terrible NCCL
- Drained and reported
- Job requeued

Later:

- Noticed a typo in nvidia_smi.py
- Found node-149 had high uncorrectable ECCs after fixing typo

Also checked active nodes:

```
2021-12-29 18:40:12 CRITICAL nvidia_smi | node-118: ecc high uncorrectables: DRAM Uncorrectable: 1335
s
2021-12-29 18:40:12 CRITICAL nvidia_smi | node-121: ecc high uncorrectables: DRAM Uncorrectable: 8
```

Took no action. The job is running fine…

2021-12-28 16:00 PT [Susan] - cluster maintenance

- Job restarted and auto-recovered
- 5 nodes in drain:

```
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/tensorboard$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
hpc*         up   infinite      1 drain* node-33
hpc*         up   infinite      5  drain node-[11,47,111,132-133]
hpc*         up   infinite    124  alloc
node-[1-4,6-10,12-17,19-32,34-46,48-92,94-98,100-106,108-110,112-116,118-131,134,137-138]
hpc*         up   infinite     11   idle node-[139-146,148-150]
```

```
(base) susanz@ip-0A1E0465:~/fairseq-py$ python scripts/cloud/slurm.py summary
node-11: Kill task failed [root@2021-12-29T03:08:14]
node-47: Kill task failed [root@2021-12-29T03:08:13]
node-111: Kill task failed [root@2021-12-29T03:08:13]
node-132: Kill task failed [root@2021-12-29T03:08:14]
```

node-133: Kill task failed [root@2021-12-29T03:08:13]

- Ssh-ing to each:
  - 11 infoROM corrupted
  - 47 lost GPU
  - 111 infoROM corrupted
  - 132-133 not sure what's wrong here, undraining and running fixmycloud

```
(base) susanz@ip-0A1E0465:~/fairseq-py$ python scripts/cloud/slurm.py undrain node-132
2021-12-29 04:10:09 WARNING  slurm      | Undraining node-132 because "No reason given"
(base) susanz@ip-0A1E0465:~/fairseq-py$ python scripts/cloud/slurm.py undrain node-133
2021-12-29 04:10:12 WARNING  slurm      | Undraining node-133 because "No reason given"
```

- Fixmycloud shows 132 and 133 as fine. Leaving as idle.
- Reported 47 to CSP.

2021-12-28 9:10 ET [Myle/Stephen] - manually recovered job

- Autorecovery failed due to a permission error on the lock file :/
  - Fixed in PR #2845
- NOT relaunched monitoring script, will let someone else do it

Stephen:

- I'm the captain now
- Running fixmycloud & relaunching the monitor script
- Reported 136, 117, 19, and 18 to CSP
- Later reported 5 too

They gave us back node-[19,55,56,91,92].

2021-12-27 9:25 ET [Myle] - postmortem on autorecovery issue

- In the past, validation takes ~14 minutes for all subsets
  - Validation prints several lines to train.log, so it shouldn't have triggered the 15 minute timeout
- Based on train.log and monitor.log, it seems like the job genuinely hung in validation, but hanging 3 times in a row during validation seems very suspicious.

Hypotheses:

1. Something is buffering writes to train.log for the whole validation step, which tripped the 15 minute modified threshold.
   a. Moving the threshold to 1 hour (--modified-threshold=3600) seems like a good solution in this case.
2. Anecdotally we seem to crash during validation a lot. It'd be good to quantify, but perhaps something in our code or dataloader is causing hangs during validation?
   a. We should quantify frequency of TERM during validation vs. training
3. It's possible we did have three bad nodes in a short time period, but what are the odds that they all failed in validation? Even if we did have bad nodes, it's possible there's something in validation that makes validation more likely to break nodes.

Separately it'd be great to add timestamps to train.stderr somehow, to make it easier to cross-reference train.log and train.stderr

2021-12-27 13:19 CET [Mikel] - restarting autorecovery script

- The autorecovery script has kicked in 3 times in the last few hours, all 3 times during validation
- I suspect it could be because train.log is less verbose during validation and the previous --modified-threshold could be too agressive
- Killed the previous autorecovery job and relaunched it with --modified-threshold 3600:
```
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>@fb.com
/shared/home/namangoyal/checkpoints/175B/175B_run12.56*/train.log --modified-threshold 3600
--slurm-jobid 4136
```

2021-12-25 04:18 ET [Myle] - starting improved autorecovery script

- Autorecovery has been made more robust in #2842
- Relaunched with (note that --modified-threshold was originally 900 but has been adjusted in the command below to 3600): `./scripts/cloud/monitor_train_log.py --mailto <scrubbed>@fb.com /shared/home/namangoyal/checkpoints/175B/175B_run12.56*/train.log --modified-threshold 3600 --slurm-jobid 4136`
- **Oncall can tail with: `tail -f /data/users/common/monitor.log`**

2021-12-25 09:48 ET [Myle] - RCA on autorecovery failure

Timeline (all times UTC):

- *2021-12-25 09:56*: job hangs
  - Root cause is **node-19**
    - Stderr message: `mlx5: ip-0A1E0444: got completion with error`
    - scripts/cloud/find_host.py maps ip-0A1E0444 to node-19
- *2021-12-25 10:15*: monitoring script detects hung job and starts autorecovery
- *2021-12-25 10:17*: fixmycloud requeues job with new nodelist
- *2021-12-25 10:18*: requeued job begins
- *2021-12-25 10:22*: monitoring script thinks auto-recovery was successful, sends email
- *2021-12-25 10:23*: done with model init
- *2021-12-25 10:27*: done with blob download, begin fast forwarding dataloader
- *2021-12-25 10:42*: monitoring script attempts auto-recovery once again
  - Note: there was no log message about why it's launching autorecovery, but it's because the train.log had not been modified in the previous 15 min
- *2021-12-25 10:45*: job is resumed and the cycle repeats

2021-12-25 08:49 ET [Stephen]

```
$ python scripts/cloud/slurm.py summary
node-11: infoROM_corrupted [susanz@2021-12-25T11:52:58]
node-19: Kill task failed [root@2021-12-25T10:16:13]
node-111: infoROM_corrupted [susanz@2021-12-25T11:53:04]
node-148: infoROM_corrupted [susanz@2021-12-25T11:53:09]
```

```
$ ssh node-19
$ nvidia-smi
Unable to determine the device handle for GPU 0001:00:00.0: GPU is lost.  Reboot the system to recover this GPU
$ sudo reboot
```

Context: CSP had asked us if we had tried rebooting to fix this error recently. Giving it a shot.
This node seems to NOT BE COMING BACK.
Please HOLD ON TO IT FOR CSP.

Updated node list.

2021-12-25 06:25 ET: [Daniel/Susan]

Auto-recovery script in action. monitor_train_log.py managed to send warning email about progress but then my instance crashed with :

```
PermissionError: [Errno 13] Permission denied: '/shared/home/namangoyal/checkpoints/175B/175B_run12.56.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr3e-05.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log.autorecover_lock'
```

Monitoring the restart: the training seems to have gotten terminated by something. My first guess is that the monitor script is misbehaving, I’m killing all instances

```
ps aux | grep monitor
sudo kill 43600
```

[Susan butting in]

- Put job in requeue / held state
- Ran fixmycloud idle
- Put the infoROM corrupted nodes in drain: 11, 111, 148
- Updated nodelist:
  sudo scontrol update job=4136 NodeList=node-[1-10,12-18,20-32,34-54,56-90,94-98,100-106,108-110,112-131,136-138]
- Released the job

2021-12-24 12:40 ET: Auto-recovery script

- After merging #2837 it is now possible to auto-recover from node failures or hung jobs
  - In the case of auto-recovery, the oncall will get a sequence of emails:
  - Email #1: File not modified in 900 seconds
  - Email #2: Detected hang, auto-recovery in progress
  - Email #3: Auto-recovery was apparently successful
    - This last email should contain the last few train.log lines
- Single pass of updating oncall docs with auto-recovery information.

- Note: it is fine for multiple people to run this command, since there is a locking mechanism to prevent multiple scripts from auto-recovering the same job
- Invocation:

```
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>@fb.com
/shared/home/namangoyal/checkpoints/175B/175B_run12.56*/train.log --modified-threshold 900
--slurm-jobid 4136
```

2021-12-24 10:00 ET: Kurt

- Checking status of idle nodes via fixmycloud
    - Still getting infoROM warnings on 148
    - Everything else passes
- Updated the node tracker table above.

2021-12-23 6:00 PM ET: Kurt

- Checking status of idle nodes via fixmycloud:
    - `2021-12-23 23:06:23 WARNING  nvidia_smi | node-148: infoROM is corrupted at gpu 0001:00:00.0`
    - `2021-12-23 23:06:23 WARNING  nvidia_smi | node-148: infoROM is corrupted at gpu 000E:00:00.0`
    - Everything else seems OK

2021-12-23 8:30 ET: Myle

- Followups for CSP:
    - **node-107** died in the night
    - Why was **node-55** added to the cluster in an unhealthy state (slow NCCL)?
- Job died ~30 min ago:
    - `srun: error: Node failure on node-107`
    - Job actually died, so will need to manually relaunch
- Seeing 133 idle hosts, running fixmycloud.py to confirm they are healthy
    - Found 3 hosts with bad NCCL:
        - node-55: max bandwidth 147.82 below threshold 180
            - This was “Off” last night, so CSP must have added it to the cluster overnight (and it’s bad)
        - node-91: max bandwidth 49.03 below threshold 180
            - From notes, this was bad last night too
        - node-92: max bandwidth 134.91 below threshold 180
            - From notes, this was bad last night too
    - Down to 130 healthy nodes
        - node-[33,145-150] are idle/healthy after relaunching job

```
# LAUNCH OF 12.56

# use updated fairscale
cd ~/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/Megatron-LM
git checkout tags/v2.6

cd ~/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use previous blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

# Use `checkpoint_33_98000` which was the last successful checkpoint
RESTORE_FILE="${BLOB_PREFIX}/checkpoint_33_98000.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.56

# Use hosts verified good from fixmycloud from above
INCLUDED_HOSTS=node-[1-6,8-32,34-54,56-68,70-73,76,78-79,81-84,88-90,96,98,100-106,108-114,116-144] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"

./scripts/cloud/monitor_train_log.py --mailto <scrubbed> /shared/home/namangoyal/checkpoints/175B/175B_run12.56*/train.log
--modified-threshold 900
```

2021-12-23 7:00 ET: Stephen

Manually drained/undrained nodes to mark the correspondence to what CSP reports. Ran update_hosts. We still see some issues with slurm not forgetting the IP address of nodes that went down.

2021-12-22 3:30 pm ET: [Myle] new oncall

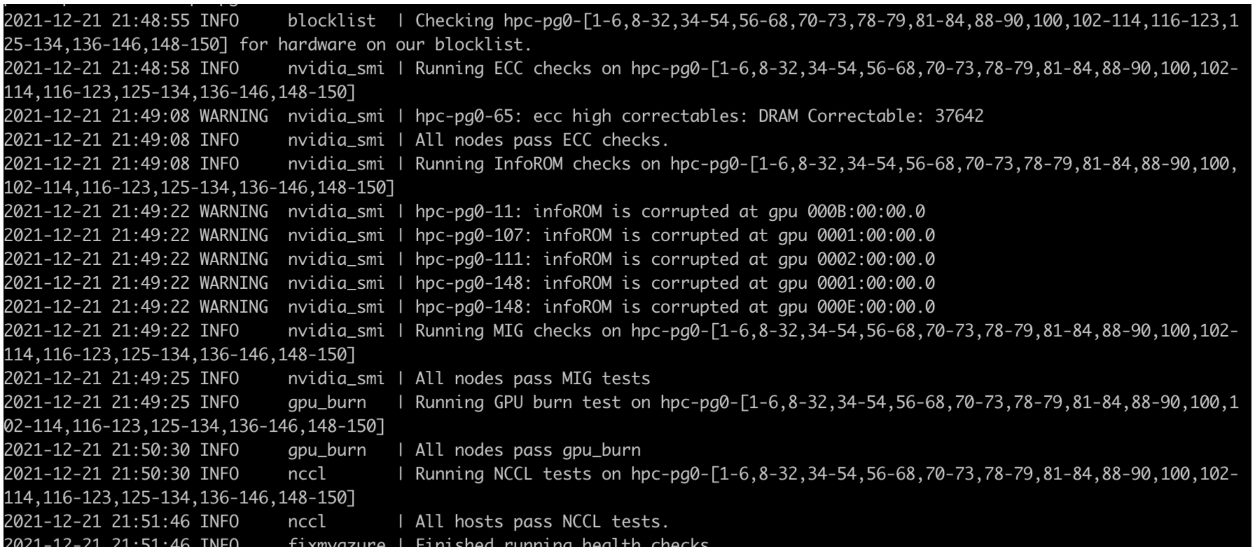
- Current status:
    - Job seems healthy
    - 3 healthy spare nodes (**node-[33,96,101]**)
    - 6 nodes in drain
        - **node-[55,124]**
            - “Off” according to cloud UI
        - **node-[91,92,135,147]**
            - “Ready” according to cloud UI
            - drained with reason Bad_infiniband
    - Target:
        - Increase to 7 nodes tonight
        - Tomorrow morning (coordinate on chat)
            - Release 5 nodes back to CSP
            - They will take a few hours to make this healthy
        - By end of day tomorrow 12 healthy nodes
    - RCA
        - Not sure about root cause
        - Could be “PCIe training”
        - Rebooting will cycle some systems, but not everything; can try it, but unlikely for it to work
        - CSP pre-flight tests didn’t cover multi-node tests previously
            - CSP has now recently added these to their preflight tests
            - + CSP has been able to repro the poor NCCL test results
- Note from Stephen (shared with CSP)

    - Being very explicit to sync both sides.
      Okay I confirmed we have the new 4 nodes, and all 4 still fail our NCCL tests, but are on standby as emergency replacements (at a 20% slowdown, but that’s better than 100%)
      33 and 101 are yellow for very high correctable ECC failures, as discussed earlier in chat. **They are our current main backups.**
      124 and 55 and still showing some slurm weirdness and are absolute no gos. Those nodes are down and I don’t know why slurm is confused.
      Thanks everyone.

2021-12-21 4:30 pm ET: [Moya] Kick off train 12.55

Ran fixmycloud on
node-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,102-114,116-123,125-134,136-146,148-150]
Since per thread those were the ones Stephen last used

- Todo - make error message on NCCL print and not just say “exit code 1”
- `please run /shared/home/mpchen/fairseq-py/scripts/cloud/nccl_tests/get_nccl_tests.sh first` was the error it ate



```
2021-12-21 21:48:55 INFO     blocklist  | Checking hpc-pg0-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,102-114,116-123,1
25-134,136-146,148-150] for hardware on our blocklist.
2021-12-21 21:48:58 INFO     nvidia_smi | Running ECC checks on hpc-pg0-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,102-
114,116-123,125-134,136-146,148-150]
2021-12-21 21:49:08 WARNING  nvidia_smi | hpc-pg0-65: ecc high correctables: DRAM Correctable: 37642
2021-12-21 21:49:08 INFO     nvidia_smi | All nodes pass ECC checks.
2021-12-21 21:49:08 INFO     nvidia_smi | Running InfoROM checks on hpc-pg0-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,
102-114,116-123,125-134,136-146,148-150]
2021-12-21 21:49:22 WARNING  nvidia_smi | hpc-pg0-11: infoROM is corrupted at gpu 000B:00:00.0
2021-12-21 21:49:22 WARNING  nvidia_smi | hpc-pg0-107: infoROM is corrupted at gpu 0001:00:00.0
2021-12-21 21:49:22 WARNING  nvidia_smi | hpc-pg0-111: infoROM is corrupted at gpu 0002:00:00.0
2021-12-21 21:49:22 WARNING  nvidia_smi | hpc-pg0-148: infoROM is corrupted at gpu 0001:00:00.0
2021-12-21 21:49:22 WARNING  nvidia_smi | hpc-pg0-148: infoROM is corrupted at gpu 000E:00:00.0
2021-12-21 21:49:22 INFO     nvidia_smi | Running MIG checks on hpc-pg0-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,102-
114,116-123,125-134,136-146,148-150]
2021-12-21 21:49:25 INFO     nvidia_smi | All nodes pass MIG tests
2021-12-21 21:49:25 INFO     gpu_burn   | Running GPU burn test on hpc-pg0-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,1
02-114,116-123,125-134,136-146,148-150]
2021-12-21 21:50:30 INFO     gpu_burn   | All nodes pass gpu_burn
2021-12-21 21:50:30 INFO     nccl       | Running NCCL tests on hpc-pg0-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,102-
114,116-123,125-134,136-146,148-150]
2021-12-21 21:51:46 INFO     nccl       | All hosts pass NCCL tests.
2021-12-21 21:51:46 INFO     fixmyazure | Finished running health checks.
```

InfoROM + ECC are yellows; have to run with anyway (cause this is 126 hosts, per conversing offline with Stephen)

```
# LAUNCH OF 12.55

# use updated fairscale
cd ~/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/Megatron-LM
git checkout tags/v2.6

cd ~/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
OLD_BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
NEW_BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>""
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

# Use `checkpoint_31_92000` per comment in thread of that being how far things got to
```

```
RESTORE_FILE="${OLD_BLOB_PREFIX}/checkpoint_31_92000.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.55

# Use hosts verified good from fixmycloud from above
INCLUDED_HOSTS=node-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,102-114,116-123,125-134,136-146,148-150] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${NEW_BLOB_PREFIX}/?${BLOB_AUTH}"
```

2021-12-21 (morning until 3 pm ish) [Stephen] Omicron Sev

CSP fat fingered and deleted our entire cluster when trying to replenish our buffer nodes.

At 14:25 ET they asked us to initiate our preflight checks

Upon reallocation we had:

```
2021-12-21 19:00:04 ERROR updatehost | Nodes node-42 and node-91 have same IP (10.30.4.95)
2021-12-21 19:00:04 ERROR updatehost | Nodes node-90 and node-92 have same IP (10.30.4.248)
2021-12-21 19:00:04 ERROR updatehost | Nodes node-9 and node-115 have same IP (10.30.4.23)
2021-12-21 19:00:04 ERROR updatehost | Nodes node-96 and node-135 have same IP (10.30.4.45)
2021-12-21 19:00:04 ERROR updatehost | Nodes node-99 and node-147 have same IP (10.30.4.60)
```

This was corrected by CSP. According to CSP, this is lag until slurm notices, but I repeated this procedure several times over at least 10 minutes.

92 and 115 also seemed unreachable but slurm wasn't having them fail their heartbeat. **Manually marked as drained**

Ran healthchecks:

```
2021-12-21 19:36:00 WARNING nvidia_smi | node-33: ecc high correctables: DRAM Correctable: 170435
2021-12-21 19:36:00 WARNING nvidia_smi | node-65: ecc high correctables: DRAM Correctable: 37642
2021-12-21 19:36:00 WARNING nvidia_smi | node-101: ecc high correctables: DRAM Correctable:
1700812021-12-21 19:36:21 WARNING nvidia_smi | node-11: infoROM is corrupted at gpu 000B:00:00.0
2021-12-21 19:36:21 WARNING nvidia_smi | node-93: infoROM is corrupted at gpu 0004:00:00.0
2021-12-21 19:36:21 WARNING nvidia_smi | node-93: infoROM is corrupted at gpu 000E:00:00.0
2021-12-21 19:36:21 WARNING nvidia_smi | node-107: infoROM is corrupted at gpu 0001:00:00.0
2021-12-21 19:36:21 WARNING nvidia_smi | node-111: infoROM is corrupted at gpu 0002:00:00.0
2021-12-21 19:36:21 WARNING nvidia_smi | node-148: infoROM is corrupted at gpu 0001:00:00.0
2021-12-21 19:36:21 WARNING nvidia_smi | node-148: infoROM is corrupted at gpu 000E:00:00.0
```

```
2021-12-21 19:39:23 ERROR nccl | node-7: max bandwidth 150.11 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-69: max bandwidth 143.8 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-74: max bandwidth 144.8 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-75: max bandwidth 145.35 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-76: max bandwidth 145.35 below threshold 180
```

```
2021-12-21 19:39:23 ERROR nccl | node-77: max bandwidth 145.15 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-80: max bandwidth 145.67 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-85: max bandwidth 149.82 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-86: max bandwidth 149.54 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-87: max bandwidth 144.02 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-93: max bandwidth 144.57 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-94: max bandwidth 140.92 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-95: max bandwidth 146.5 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-97: max bandwidth 146.5 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-98: max bandwidth 143.02 below threshold 180
2021-12-21 19:39:23 ERROR nccl | node-124: max bandwidth 148.34 below threshold 180
```

Replicated the list of bad nccl nodes with multiple tries.
Manually drained pg0-33 and pg0-105 as they had highest correctables.

Then I requeued susan’s job (which had automatically been put on hold when nodes went down) with

```
sudo scontrol hold job=3129
sudo scontrol update job=3129
'NodeList=node-[1-6,8-32,34-54,56-68,70-73,78-79,81-84,88-90,100,102-114,116-123,125-134,136-146,148-150]'
sudo scontrol release 3129
```

2021-12-21 5:30am ET: [Susan] Node down, restart from 91,250 with lower LR - Run 12.53, 12.54

- Running healthcheck: `python scripts/cloud/fixmycloud.py idle`
    - Checked node-81:
        - Unable to determine the device handle for GPU 0001:00:00.0: GPU is lost. Reboot the system to recover this GPU

- PR to lower LR: #2833
    - 12.53 crashed with tokenization error, reverted the cache tokenization change to resume and debug later
    - Luckily 91250 is early in a shard-epoch. Took only ~5 min for data loading to finish.

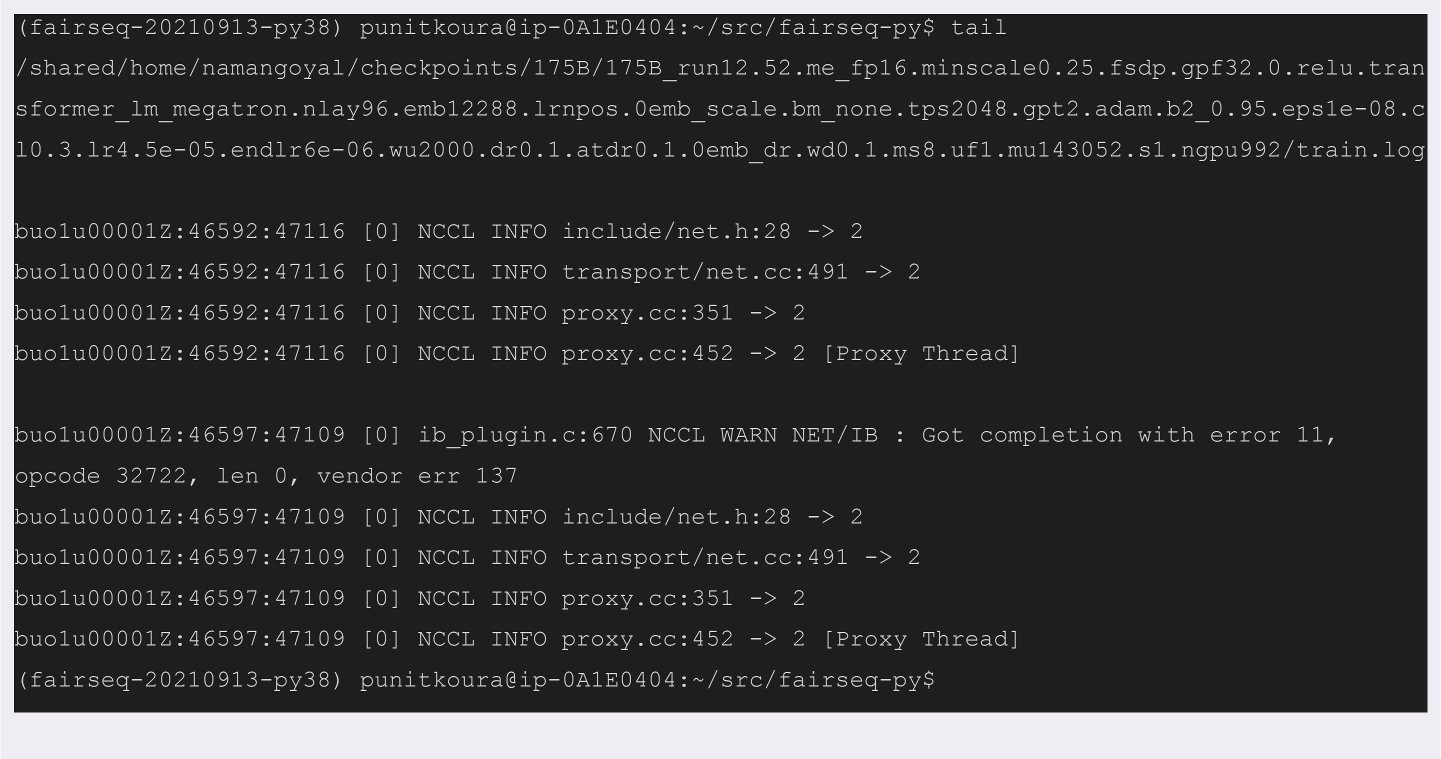
```
# LAUNCH OF 12.54

# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/Megatron-LM
git checkout tags/v2.6

cd ~/src/fairseq-py
git fetch origin susan/run12.53
git checkout susan/run12.53

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
```

```
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

# confirm epoch of checkpoint from previous train.log - we checkpoint every 250 steps
RESTORE_FILE="${BLOB_PREFIX}/checkpoint_31_91250.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.54

INCLUDED_HOSTS=node-[1-2,4-22,24-25,27-52,54-67,69-80,82-84,86-87,89-96,98-104,106-107,109-112,115-117,119-
135,147-148,150] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

2021-12-19 12pm ET: Crossing the epoch boundary

- Surprisingly uneventful, no large drop in training ppl
- Ran this command to make a backup of the epoch 1 checkpoint:
  `cp --recursive --include-pattern "checkpoint_last*.pt" <<<SCRUBBED FOR RELEASE>>>`
- Uploaded a version of the epoch 1 checkpoint without optimizer state here:
  `/opt/backups/175B/checkpoint1_eval/`
  - It's also located here on Cloud: `/data/175B_checkpoints/checkpoint1_eval`

2021-12-20 12:12 AM PT: [Punit] Node down - requeue for 12.52a



```
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$ tail
/shared/home/namangoyal/checkpoints/175B/175B_run12.52.me_fp16.minscale0.25.fsdp.gpf32.0.relu.tran
sformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.c
l0.3.lr4.5e-05.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

buo1u00001Z:46592:47116 [0] NCCL INFO include/net.h:28 -> 2
buo1u00001Z:46592:47116 [0] NCCL INFO transport/net.cc:491 -> 2
buo1u00001Z:46592:47116 [0] NCCL INFO proxy.cc:351 -> 2
buo1u00001Z:46592:47116 [0] NCCL INFO proxy.cc:452 -> 2 [Proxy Thread]

buo1u00001Z:46597:47109 [0] ib_plugin.c:670 NCCL WARN NET/IB : Got completion with error 11,
opcode 32722, len 0, vendor err 137
buo1u00001Z:46597:47109 [0] NCCL INFO include/net.h:28 -> 2
buo1u00001Z:46597:47109 [0] NCCL INFO transport/net.cc:491 -> 2
buo1u00001Z:46597:47109 [0] NCCL INFO proxy.cc:351 -> 2
buo1u00001Z:46597:47109 [0] NCCL INFO proxy.cc:452 -> 2 [Proxy Thread]
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$
```

Node failure for buo1u00001Z

```
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$ python scripts/cloud/find_host.py
buo1u00001Z
node-68
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$
```

node-68 is the problematic node.

```
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$ squeue
<scrubbed>
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$
```

The current job (2606) has
`node-[1-2,4-25,27-36,38-52,54-72,74-87,89-96,98-104,106-107,109-113,115-131,147-148,150]`
As the node list.
We need to swap out 68.

Checking current idle nodes

```
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
hpc*         up   infinite      1 drain~ node-146
hpc*         up   infinite      7 drain* node-[26,53,88,97,105,114,149]
hpc*         up   infinite      1  down* node-108
hpc*         up   infinite      2    mix node-[3,37]
hpc*         up   infinite    132  alloc
node-[1-2,4-25,27-36,38-52,54-72,74-87,89-96,98-104,106-107,109-113,115-135,139-142,147-148,150]
hpc*         up   infinite      7   idle node-[73,136-138,143-145]
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$
```

Potential new host list (adding 143)
`node-[1-2,4-25,27-36,38-52,54-67,69-72,74-87,89-96,98-104,106-107,109-113,115-131,143,147-148,150]`

Requeue + hold to pause the ongoing training job.

```
sudo scontrol requeue job=2606
sudo scontrol hold job=2606
```

Note that the job must be paused before fixmycloud, since NCCL tests require that. See **Performing health checks** section.

Running fixmycloud to check if the proposed new node list is healthy

```
(fairseq-20210913-py38) punitkoura@ip-0A1E0404:~/src/fairseq-py$ python scripts/cloud/fixmycloud.py --hosts
node-[1-2,4-25,27-36,38-52,54-67,69-72,74-87,89-96,98-104,106-107,109-113,115-131,143,147-148,150]
```

Looks like ECC, GPU burn tests etc passed. NCCL tests didn’t complete properly.

NCCL failures investigation

Re-ran the failing NCCL command, turns out there was another command which is to be run before the tests can go through.

```
missing libnccl and all_reduce_perf; please run /shared/home/punitkoura/src/fairseq-py/scripts/cloud/nccl_tests/get_nccl_tests.sh first!
```

After running the above command, fixmycloud worked fine. No errors found in the host list.

Resuming job with new host list

With confirmation from fixmycloud, the next step is to resume the job with a new host list.

```
sudo scontrol update job=2606 Nodelist=node-[1-2,4-25,27-36,38-52,54-67,69-72,74-87,89-96,98-104,106-107,109-113,115-131,143,147-148,150]
sudo scontrol release job=2606
```

Train.log seems to be working again.

Enable tensorboard

```
cd /shared/home/namangoyal/checkpoints/175B/tensorboard

sudo ln -s /shared/home/namangoyal/checkpoints/175B/175B_run12.52.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr4.5e-05.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/tbB run12.52b
```

Verified that 12.52b shows up on the tensorboard.

2021-12-17 15:34 ET: [Daniel] Node down - requeue for 12.52a

mlx5: buo1u00002X: got completion with error:
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000010 00000000 00000000 00000000
00000000 00008914 10001d2a f7d20ad3

python scripts/cloud/find_host.py buo1u00002X → node-105

sudo scontrol requeue job=2606
sudo scontrol hold job=2606

# Try swapping in 25:

```
sudo scontrol update job=2606 NodeList=node-[1-2,4-25,27-36,38-52,54-72,74-104,106, 107,109-113,115-131,147]
sudo scontrol release job=2606
```

# Turns out 88 and 97 are drained

Check that the proposed nodelist is healthy:
python scripts/cloud/fixmycloud.py --hosts node-[1-2,4-25,27-36,38-52,54-72,74-87,89-96,98-104,106,107,109-113,115-131,147-148,150]

```
# these two commands together pause the job
sudo scontrol requeue job=2606
sudo scontrol hold job=2606

sudo scontrol update job=2606
NodeList=node-[1-2,4-25,27-36,38-52,54-72,74-87,89-96,98-104,106,107,109-113,115-131,147-148,150]
sudo scontrol release job=2606
```

Diagnostics on original bad node:

ssh node-105
(base) danielsimig@buo1u00002X:~/fairseq-py$ python scripts/cloud/gather_diagnostics.py
Diagnostics uploaded to: <<<SCRUBBED FOR RELEASE>>>

Re-enable tensorboard

```
cd /shared/home/namangoyal/checkpoints/175B/tensorboard
```

sudo ln -s
/shared/home/namangoyal/checkpoints/175B/175B_run12.52.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr4.5e-05.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/tbA
175B_run12.52a

2021-12-16 12:15 ET: increased job time limit from 3 days to unlimited:

```
sudo scontrol update job=2606 TimeLimit=UNLIMITED
```

2021-12-14 18:30 ET: [Kurt] Drain a few nodes

Ran `python scripts/cloud/fixmycloud.py idle`, and found that two nodes broke on NCCL errors (53, 108); see entry for (2021-12-09 16:00 PT) where they were marked to be drained

Drained the nodes:

```
python scripts/cloud/slurm.py drain node-53 --reason="failing nccl tests"
2021-12-14 23:34:15 WARNING  slurm      | Draining node-53 because "failing nccl tests"
python scripts/cloud/slurm.py drain node-108 --reason="failing nccl tests"
2021-12-14 23:34:26 WARNING  slurm      | Draining node-108 because "failing nccl tests"
```

2021-12-14 13:30 ET: [Moya] Scancel + Resubmit

As decided in conversation, we restart at 72750 steps in order to lower the LR such that we're far away from a new shard. (Though, maybe could've done it an hour earlier as per where `loading train data` showed up in the logs.)

While running ran into a

```
    buo1u000088:16674:16674 [1] init.cc:988 NCCL WARN Cuda failure 'uncorrectable
NVLink error detected during the execution'
```

In the stdlog with "Please install the megatron submodule" in the stderr (despite having megatron installed)

Which was fixed by
`git submodule update --init fairseq/model_parallel/megatron`

Also took way too long to realize that it's completely kosher just outright copy/pasting BLOB_PREFIX and BLOB_AUTH from the previous runs… but I blame the peanut gallery. :P

```
squeue # Get id of the 175b run; also for the "INCLUDED_HOSTS" command below, to use recently determined cleaned hosts
scancel 2589 # The id of the 175 b run

## BEFORE: Change sweep_opt_en_lm_175b to 4.5 (+ commit diff as such)
#######################

# LAUNCH OF 12.52
# use updated fairscale
cd ~/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple


cd ~/Megatron-LM
git fetch --tags && git checkout v2.6 # more verbose in case you haven't already fetched the tags

cd ~/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RUN_ID=175B_run12.52 # New id to run

# Uses included hosts from currently existing run (ie, copy/pasted from squeue prior to cancelling before)
INCLUDED_HOSTS=node-[1-24,27-36,38-52,54-72,74-87,89-107,109-113,115-131,147] \
         python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

Also don't forget to run the command to update the tensorboard after all of this! (As an aside, I think someone else might've run this after I tried to since I forgot to do `sudo` when I ran it, but it's something a la)

```
cd /shared/home/namangoyal/checkpoints/175B/tensorboard
sudo ln -s <dir of new run in /shared/home/namangoyal/checkpoints/175B/$RUNID...> $RUNID
```

2021-12-14 03:30 ET: [Susan] Requeue

- Previous run crashed with CUDA launch failure again:
    - RuntimeError: CUDA error: unspecified launch failure
- Ran ecc error check:
    - WCOLL=~myleott/hosts PDSH_RCMD_TYPE=ssh pdsh nvidia-smi -q -d "ECC" > ecc_error.log

Saw:

    pdsh@ip-0A1E0404: node-88: ssh exited with exit code 15

SSH'ed over to node-88:
(base) susanz@buo1u00004D:~$ nvidia-smi

Unable to determine the device handle for GPU 000C:00:00.0: GPU is lost. Reboot the system to recover this GPU

- Swap in 3 in for 88

```
sudo scontrol requeue job=2589
sudo scontrol hold job=2589
sudo scontrol update job=2589 NodeList=node-[1-24,27-36,38-52,54-72,74-87,89-107,109-113,115-131,147]
sudo scontrol release job=2589
```

NOTE: “2021-12-13: Preemptive Plan” for changing LR still has not occurred yet

Initiated replace_node on node-88. Sent to Cormac.

2021-12-13 14:00 ET: Reverse Shadow and Oncall onboardings

Reverse Shadows
Monday 2pm: Myle
Tuesday 2pm: Stephen
Wednesday 2pm: Susan
Thursday 2pm: Sam
Friday 2pm: Naman

Main oncalls
Monday 2pm: Moya
Tuesday 2pm: Kurt
Wednesday 2pm: Punit
Thursday 2pm: Mikel
Friday 2pm: Daniel

2021-12-13: Preemptive Plan

On the next crash, we plan to lower the LR from 6.0 -> 4.5. We will observe that for a bit and lower it again to 3.0 subject to signal.

THIS SHOULD CAUSE THE RUN_ID AND THE BLOB FOLDER TO BOTH BE BUMPED.

2021-12-13 11:49 ET: [Stephen] Bumping timelimit

After we joked about the world record of hitting >2D, realized needed to update the timelimit of the job. Command executed:

```
sudo scontrol update job=2589 TimeLimit=UNLIMITED
```

2021-12-11 07:53 ET: [Stephen] Cluster Maintenance

5 down nodes. Time for some reprovisioning.
node-25: Kill task failed [root@2021-12-11T10:57:40]

```
node-26: Bad_infiniband [roller@2021-12-10T11:30:21]
node-53: No_reason_given [susanz@2021-12-10T00:12:45]
node-108: No_reason_given [susanz@2021-12-10T00:12:33]
node-114: Failed_GPU_burn [roller@2021-12-10T11:15:37]
```

Initiated replace_node on 3 bad nodes: 25, 26, 53

2021-12-11 02:52 PT: [Susan] Noticed IB issues/lost GPU.

Future readers: this is now simplified with #2785

Hotswap 25 <> 145

Previous run hung with:

```
        mlx5: buo1u00000S: got completion with error:
        00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000
        0000000d 00000000 00000000 00000000
        00000000 01005104 08001c17 10a7b3d3
```

```
# ssh'ing onto node shows GPU lost:
(base) susanz@buo1u00000S:~$ nvidia-smi
Unable to determine the device handle for GPU 000C:00:00.0: GPU is lost.  Reboot the system to recover this GPU

# Finding IP address for buo1u00000S
(base) susanz@buo1u00000S:~$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.30.4.29  netmask 255.255.252.0  broadcast 10.30.7.255
      inet6 fe80::222:48ff:fe25:4f2d  prefixlen 64  scopeid 0x20<link>
      ether 00:22:48:25:4f:2d  txqueuelen 1000  (Ethernet)
      RX packets 1223684089  bytes 1350989781677 (1.3 TB)
      RX errors 0  dropped 0  overruns 0  frame 0
      TX packets 163407144  bytes 3319218659584 (3.3 TB)
      TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

# Get hostname from IP address
(base) susanz@buo1u00000S:~$ grep 10.30.4.29 /etc/hosts

# Swap in 147 in for 25
sudo scontrol requeue job=2589
sudo scontrol hold job=2589
sudo scontrol update job=2589 NodeList=node-[1-2,4-24,27-36,38-52,54-72,74-107,109-113,115-131,147]
sudo scontrol release job=2589
```

Notes:

- Node 25 went into drain after job was updated with new hostlist
- We need to replace all the nodes in drain (?)

```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
hpc*        up   infinite      1 drain~ node-146
hpc*        up   infinite      5  drain node-[25-26,53,108,114]
hpc*        up   infinite      2    mix node-[3,37]
hpc*        up   infinite    136  alloc node-[1-2,4-24,27-36,38-52,54-107,109-113,115-142,147]
hpc*        up   infinite      3   idle node-[143-145]
```

2021-12-10 23:14: [Stephen] 12.51 Resuming

- Remembered I forgot to raise the learning rate
- Also took at the logs and saw it was fastforwarding the dataloader a lot. We should be right at an epoch boundary though! I think I didn't wait long enough for checkpoints to upload? -- looks like we only have 976/992 shards.
    - Observed via `ls "${BLOB_PREFIX}/?${BLOB_AUTH}" | grep 61000 | wc -l`

Wrote a fresh upload script and launched on nodes with:

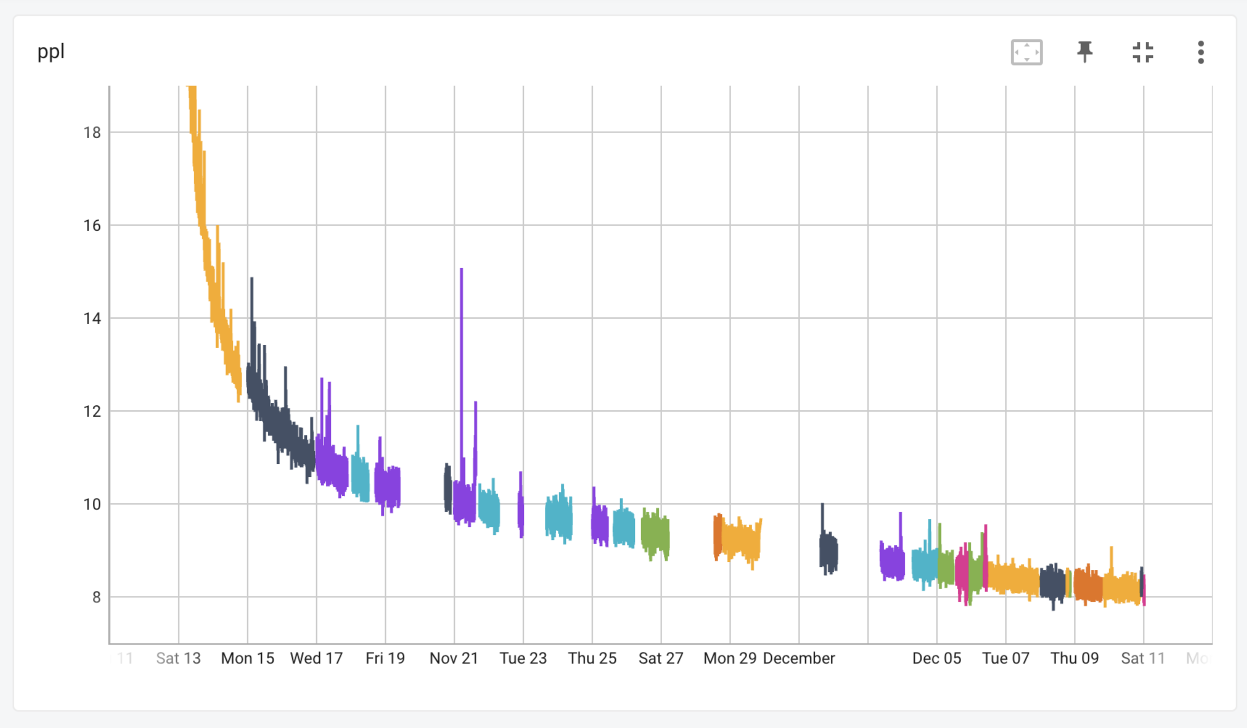
```
pdsh -R ssh -w 'node-[1-2,4-25,27-36,38-52,54-72,74-107,109-113,115-135,139-145,147]' bash ~/restore_61000.sh
```

Gave a lot of logspam bc of nodes that didn't participate in the previous job but that's okay. It was clearly copying on the ones left. Confirmed all 992 were uploaded at the end. Created a PR to increase the learning rate back to 6e-5 (GPT-3's value) and relaunched.

```
# LAUNCH OF 12.51

# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/Megatron-LM
git checkout tags/v2.6

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp_1251
git checkout gshard_combine_megatron_fsdp_1251

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RUN_ID=175B_run12.51 # big money no whammies

# Doing something radical. All idle nodes should now be safe nodes.
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"

# For the record, this ended up launching with
# node-[1-2,4-25,27-36,38-52,54-72,74-107,109-113,115-131]
```

Confirmed number of fast forward batches looks low (236). As of 00:23 ET looks healthy and signing off. Also left two baselines running for fun.

Just another last note: we seem to have improved massively in utilization. Here’s PPL with true wall clock:



Gaps are much better since the week of hell.

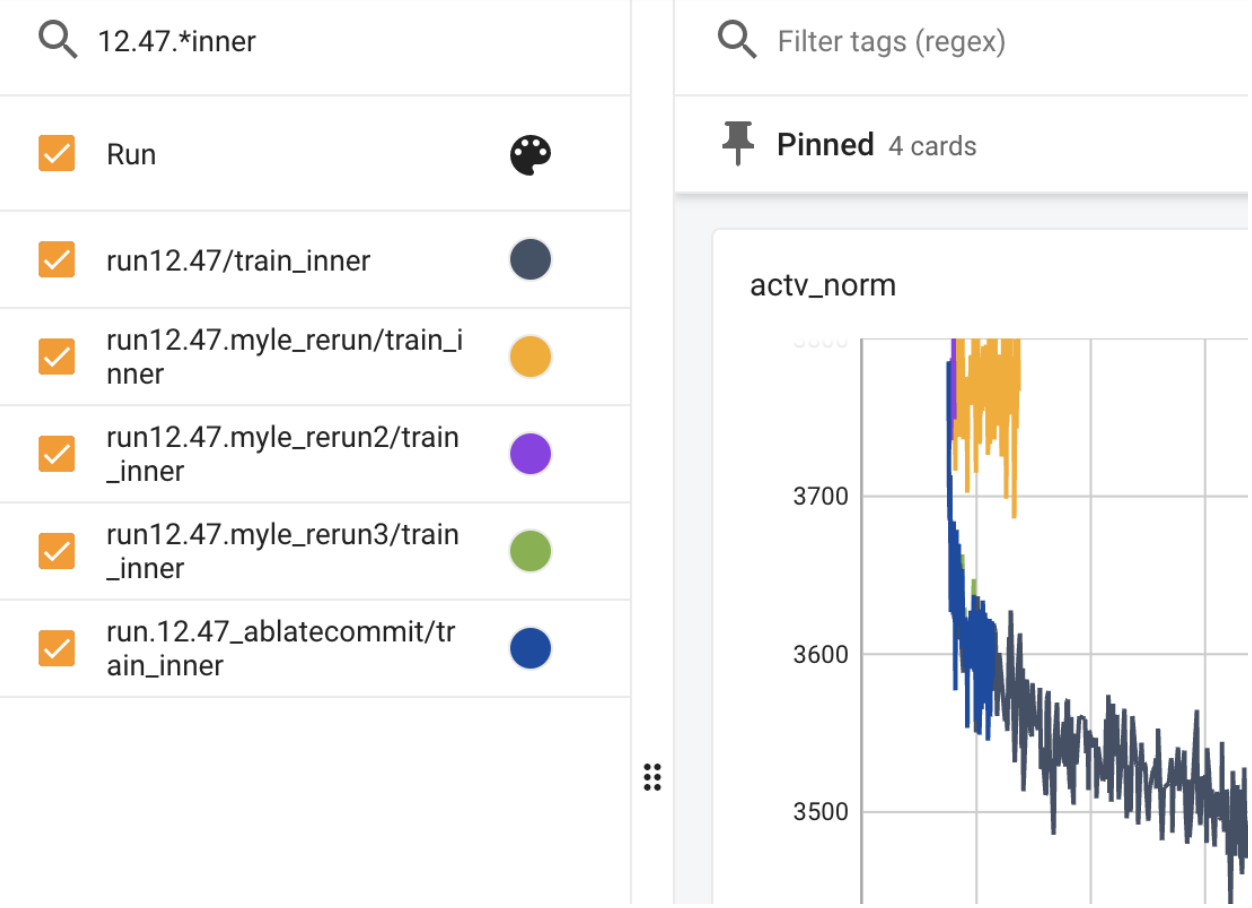
2021-12-10 22:42: [Stephen] 12.50 Resuming

- Ablation is done. Time to resume.
- Moved Megatron LM back to 2.6.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/Megatron-LM
git checkout tags/v2.6

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RUN_ID=175B_run12.50 # big money no whammies

# Doing something radical. All idle nodes should now be safe nodes.
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

2021-12-10 20:58 ET: [Stephen] The ablation

Update: Bad news y'all. That ain't the commit. New launch is clearly on the track as Naman's. Messing up my own environment can't explain it, as this run was when I first upgraded to 2.6.



☑ Run

☑ run12.47/train_inner ●

☑ run12.47.myle_rerun/train_inner ●

☑ run12.47.myle_rerun2/train_inner ●

☑ run12.47.myle_rerun3/train_inner ●

☑ run.12.47_ablatecommit/train_inner ●

Before launch:

- Lame. Hit shard boundary at 60765 updates. Waiting until 61k since I know we have had issues restoring from epoch boundaries before. Beginning ablation at 21:51.
- Reverting the bad commit in Megatron-LM with "git revert -m 1 0be405"
    - Confirmed files looked like the right ones touched.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp
```

```
# use new blob URL
OLD_BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"
RESTORE_FILE="${OLD_BLOB_PREFIX}/checkpoint_18_51750.pt?${BLOB_AUTH}"

# change blob prefix so that we don’t clobber Naman’s checkpoints
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>”

RUN_ID=175B_run12.47.byebad

# Doing something radical. All idle nodes should now be safe nodes.
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

2021-12-10 05:52 ET: [Stephen] Cluster maintenance

- Just generally checking health of all our fresh idle nodes, making sure fixmycloud is up to the task.
- Found update hosts to not quite be working
- After running update hosts, observed that pg0-114 and pg0-3 **have the same IP address.**
    - In “scontrol show nodes” pg0-114 has NodeHostName of 10.30.4.11
    - In “scontrol show nodes” pg0-3 has NodeHostName of 10.30.5.6
    - So why isn’t our script working?
- Also found hundreds of IP addresses that seem to only be listed once, don’t have slurm names. Ex 10.30.7.95. I assume those are nodes we used to have
- Also noticed pg0-146 seems unreachable
    - Its slurm NodeAddr is given as a name to itself, not an ip address like the others.
    - This exception was already carved out into the host updating script, so I guess it’s intentional
- Noticed bugs in update_hosts:
    - Clean known hosts was called before the hosts were updated lol
    - It could easily skip unreachable nodes like -146
    - After fixing these, the incongruities in the hosts file disappeared. I think we just hadn’t been able to run update hosts successfully.
- Ran fixmycloud with lots of fixes and improvements
    - #2776
    - Drained one node bc of bad infiniband, one node bc of gpu burn
    - **Started replacing pg0-146**
- Current status: 124 active; 18 good spares; 4 drained + 1 being replaced

2021-12-09 16:00 PT: [Susan] Run 12.49, restart due to NCCL errors

- Exactly the same garbage as “2021-12-05 12:15pm ET: Requeueing 12.44 and 12.45” run.
- Excluding 108 and 53 and putting in drain.
- Leaves us with:

(fairseq-20210913) susanz@ip-0A1E0404:~/fairseq-py$ sinfo

```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
hpc*         up   infinite      4 drain* node-[3,37,73,124]
hpc*         up   infinite      4  drain node-[26,53,108,114]
hpc*         up   infinite    124   idle node-[1-2,4-25,27-36,38-52,54-72,74-107,109-113,115-123,125-132]
```

- Restarting on the only 124 nodes we have that are functional:

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"
# confirm epoch of checkpoint from previous train.log - we checkpoint every 250 steps
RESTORE_FILE="${BLOB_PREFIX}/checkpoint_19_57000.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.49

INCLUDED_HOSTS=node-[1-2,4-25,27-36,38-52,54-72,74-107,109-113,115-123,125-132] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"

# monitor train.log - launch in a tmux session
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>@fb.com
/shared/home/namangoyal/checkpoints/175B/175B_run12.49.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_l
m_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr4.5e-05.end
lr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log --modified-threshold 900

# Add new entry for TB - TB dir takes a while to come up since training takes a while to restart (data
loader has to fast forward, goes through ~75 batches / min)
cd /shared/home/namangoyal/checkpoints/175B/tensorboard
sudo ln -s
../175B_run12.49.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb
_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr4.5e-05.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.w
d0.1.ms8.uf1.mu143052.s1.ngpu992/tb/ run12.49
```

- [16:22 PT] Launched
- [16:47 PT] Still waiting for data loader to fast forward
  - 2021-12-10 00:32:58 | WARNING | fairseq.data.iterators | Fast-forwarding dataloader by 2334 batches...
- [16:57 PT] Omg still waiting for data loader to fast forward ;_;
  - Our fast forwarding speed, based on notes from run 12.48, seems to be 75 batches / min.
- [17:04 PT] Oh hallelujah

    - 2021-12-10 01:04:14 | WARNING | fairseq.data.iterators | done fast-forwarding dataloader in 1939.7 seconds

2021-12-09 10:45am PT: Provisioning error in Cloud for node 124

- Reprovisioning, failed again.
- Yelled at CSP to give us more machines - will be getting 15 more at some point.
- Also need them to bump up ingress/egress limits on blob store

2021-12-09: Megatron v2.6 Debrief + CSP Sync

Discussion about what happened

- Discussion about how it was discovered, reviewing Susan's observations.
- Reviewed how we started going through to find what differed between Naman and Myle's env.
- Review that we have two versions megatron
    - Submodule, which lets us do the model parallel MHA
    - Imported, which lets us get fused_softmax

Future mitigations & Lessons learned:

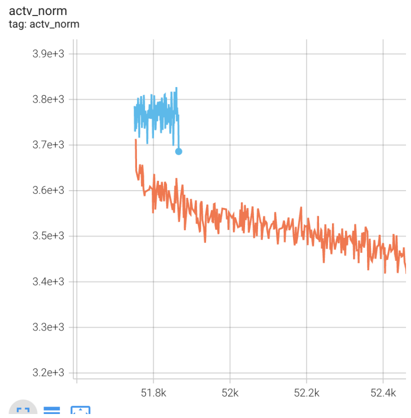
- Use single environment for all launches
    - Containers?
    - Add assert to sweep.py to check version numbers?
- We should test upgrading dependencies periodically, in case there are bug fixes
- Why aren't relaunches deterministic?
    - We assumed it was just loss scale history, but it seems things are different even when I relaunch the same thing multiple times
- What was the bug that Nvidia fixed?
    - Megatron v2.6 vs v2.4
        - Includes https://github.com/NVIDIA/Megatron-LM/pull/133
- What is the impact of this bug on the first 37% of training?
- Do we want to make any changes after fixing the bug?
    - Increase learning rate?
    - Increase clipping?

2021-12-08 8:55pm ET: run12.48: relaunch checkpoint_18_54250

- Run with Megatron v2.6 based on previous entries
- Resume from 54250, but has to fast-forward through 2630 batches, which takes 35 minutes with CPUs all at 100% :(

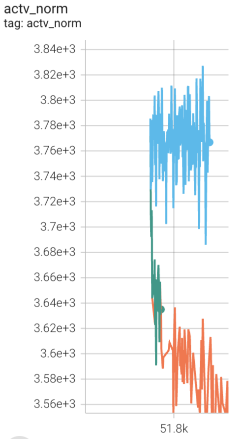
```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RESTORE_FILE="${BLOB_PREFIX}/checkpoint_18_54250.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.48

INCLUDED_HOSTS=node-[1-2,4-25,27-36,38-72,74-112,115-117,119-123,125-132] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

2021-12-08 TBD: Analysis from 12.47.myle

- The results look different between my run and Naman's original 12.47 run
- 

- Current hypothesis is something in `pip list` is different between the two
    - Make myself into Naman: sudo -i namangoyal
    - Naman and Myle seem to have differences in our pip-installed version of megatron-lm!
        - ~myleott/src/Megatron-LM
        - ~/namangoyal/src/Megatron-LM
        - Myle is on v2.4 (42c1cf4279acea5a554500dcb552211f44cbec45)
        - Naman is on v2.6 (3860e995269df61d234ed910d4756e104e1ab844)
- Going to relaunch a few more times:
    - run12.47.myle: my initial relaunch of 12.47 with my env
    - run12.47.myle2: a second relaunch of above to test determinism
        - Looks like it's not deterministic?!

    - run12.47.myle3: a third relaunch with Megatron v2.6
        - Looks like Megatron v2.6 was making things better:



2021-12-08 05:05pm ET: GPU failure; launch debug run with Myle’s env

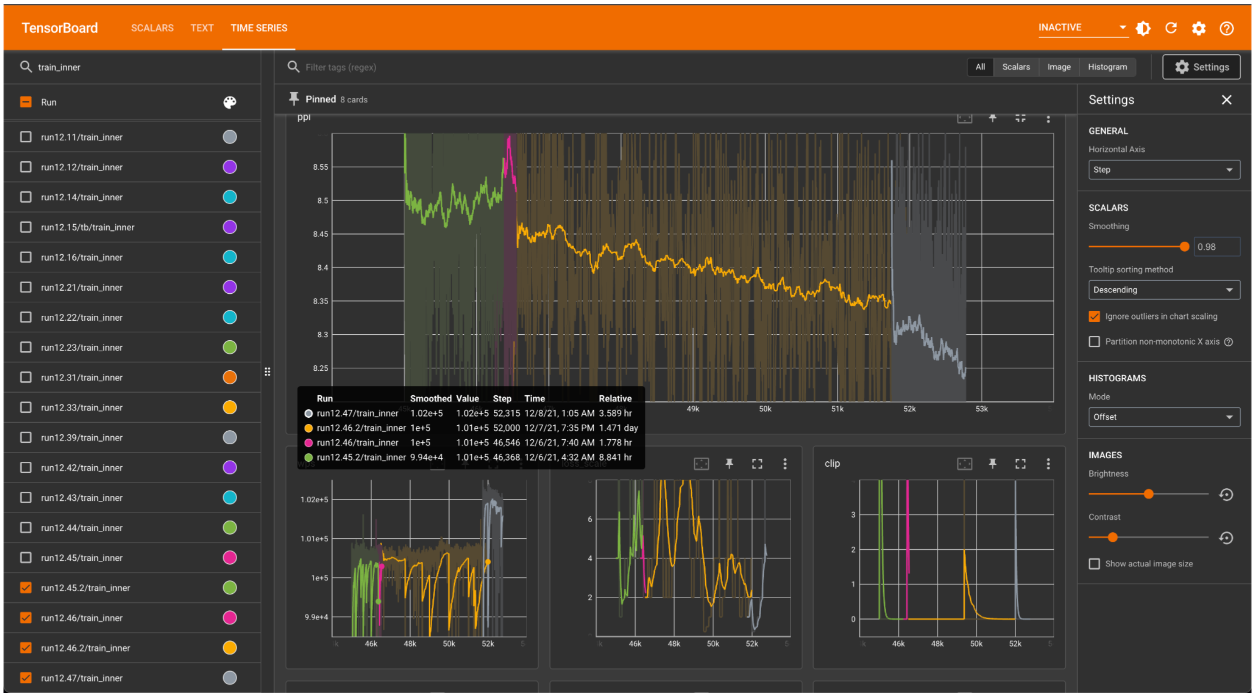
- There was a GPU failure, which caused 12.47 to hang
    - Latest checkpoint is **checkpoint_18_54250**
- Based on the discussion below, I will relaunch from **checkpoint_18_51750** with my env, to see if it matches Naman’s results. This run should be identical to Naman’s original 12.47 run, except launched from my environment
    - If it matches, then it seems we just got lucky!
    - If it doesn’t match, then we need to dig more and understand if there’s something unique about Naman’s environment, or if resuming from checkpoints is nondeterministic somehow
    - In either case, I will cancel and resume from **checkpoint_18_54250** thereafter

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RESTORE_FILE="${BLOB_PREFIX}/checkpoint_18_51750.pt?${BLOB_AUTH}"

# change blob prefix so that we don’t clobber Naman’s checkpoints
BLOB_PREFIX="/myleott/2021-12-08/175B_run12.47.myle"

RUN_ID=175B_run12.47.myle2

INCLUDED_HOSTS=node-[1-2,4-25,27-36,38-72,74-113,115-123,125-130] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

2021-12-08 04:00am PT: Checking in

[From Susan: How did Naman's run just get a free 2% bump in wps? Truly the golden touch. Also seems to be a significant drop in actv_norm as well, which generally helps us avoid overflow issues. Getting lucky here?]

- Comparison of config namespaces shows no significant differences:
  ```
  python scripts/compare_namespaces.py \
      175B_run12.47*/train.log \
      175B_run12.46*/train.log
  ```
- Perhaps some difference in environment? Cc Naman Goyal to confirm:
  - PyTorch version: '1.9.0+cu111'
    - Confirmed this matches Myle's env for 12.46
  - Fairscale version:
    - Branch: prefetch_fsdp_params_simple
    - Commit: 8820049331331c773077c257667aa81baf4cc9f9
      - Confirmed this matches Myle's env for 12.46
  - Megatron submodule version:
    - Branch: fairseq_v2 (16623c2dce9068f3f9574348b5b3c35c0c5a85c6)
      - Confirmed this matches Myle's env for 12.46 and Susan's env
  - Fairseq commit:
    - Commit: fc24ce0ae48626a6d18dbb45b486600c3732c14f
    - Branch: gshard_combine_megatron_fsdp
    - Myle's env for 12.46 was 6c973d4f92d0c9813f439a4920b23c7f93429511. There's only two commits separating this from Naman's and they are unrelated to training
  - Code snapshots, for reference:
    - **12.47 (Naman):** /shared/home/namangoyal/src/fairseq_gshard/fairseq-py/slurm_snapshot_code/2021-12-08T05_07_39.022577
    - **12.46 (Myle):** /shared/home/myleott/src/fairseq2/slurm_snapshot_code/2021-12-06T13_33_13.859846
    - Confirmed no meaningful differences with:
      ```
      diff -bur --exclude __pycache__
      /shared/home/namangoyal/src/fairseq_gshard/fairseq-py/slurm_snapshot_code/2021-12-08T05_07_39.022577
      /shared/home/myleott/src/fairseq2/slurm_snapshot_code/2021-12-06T13_33_13.859846
      ```
- For future reference, here are the machines used for each run:
  - **run 12.46.2**: node-[1-2,4-25,27-50,52-64,66-128]
  - **run 12.47**: node-[1-2,4-25,27-72,74-96,98-112,114-117,119-123,125-131]
  - Diff:
    - Remove node-[73,97,113,118,124]
    - Add node-[51,65,129,130,131]



☐ Run
☐ run12.11/train_inner
☐ run12.12/train_inner
☐ run12.14/train_inner
☐ run12.15/tb/train_inner
☐ run12.16/train_inner
☐ run12.21/train_inner
☐ run12.22/train_inner
☐ run12.23/train_inner
☐ run12.31/train_inner
☐ run12.33/train_inner
☐ run12.39/train_inner
☐ run12.42/train_inner
☐ run12.43/train_inner
☐ run12.44/train_inner
☐ run12.45/train_inner
☑ run12.45.2/train_inner
☑ run12.46/train_inner
☑ run12.46.2/train_inner
☑ run12.47/train_inner

Run	Smoothed	Value	Step	Time	Relative
● run12.47/train_inner	1.02e+5	1.02e+5	52,315	12/8/21, 1:05 AM	3.589 hr
● run12.46.2/train_inner	1e+5	1.01e+5	52,000	12/7/21, 7:35 PM	1.471 day
● run12.46/train_inner	1e+5	1.01e+5	46,546	12/6/21, 7:40 AM	1.778 hr
● run12.45.2/train_inner	9.94e+4	1.01e+5	46,368	12/6/21, 4:32 AM	8.841 hr

Settings

GENERAL

Horizontal Axis: Step

SCALARS

Smoothing: 0.98

Tooltip sorting method: Descending

☑ Ignore outliers in chart scaling

☐ Partition non-monotonic X axis

HISTOGRAMS

Mode: Offset

IMAGES

Brightness

Contrast

☐ Show actual image size

2021-12-07 10:59pm ET: RuntimeError: CUDA error: CUBLAS_STATUS_EXECUTION_FAILED when calling

NOTES FOR FUTURE:

1) 113 and 118 has info ram issue, look into if its a real issue or not, if it is then recycle

- Error stack trace:

```
data.storage().resize_(size.numel())
RuntimeError: CUDA error: unspecified launch failure
CUDA kernel errors might be asynchronously reported at some other API call,so the stacktrace below might be
incorrect.
For debugging consider passing CUDA_LAUNCH_BLOCKING=1.
```

- Ran ECC error check:
  - WCOLL=~/hosts PDSH_RCMD_TYPE=ssh pdsh nvidia-smi -q -d "ECC" > ecc_error.log
  - hcp-pg0-97 and node-124 hang and are completely unreachable
  - node-132 has “Could not resolve hostname node-132”
- Ran nvidia-smi check on all but above 3 nodes:

```
python scripts/cloud/nvidia_smi.py node-[1-2,4-25,27-72,74-96,98-123,125-131]

2021-12-08 04:33:21 WARNING  __main__   | node-23: ecc high correctables: DRAM Correctable: 170081
2021-12-08 04:33:21 WARNING  __main__   | node-88: ecc high correctables: DRAM Correctable: 506807
2021-12-08 04:33:21 WARNING  __main__   | node-85: ecc high correctables: DRAM Correctable: 16758
2021-12-08 04:33:21 WARNING  __main__   | node-85: ecc high correctables: DRAM Correctable: 16758
2021-12-08 04:33:21 INFO     __main__   | All nodes pass ECC checks.
2021-12-08 04:33:21 INFO     __main__   | Running MIG checks on node-[1-2,4-25,27-72,74-96,98-123,125-131]
2021-12-08 04:33:24 INFO     __main__   | All nodes pass MIG tests
2021-12-08 04:33:24 INFO     __main__   | Running InfoROM checks on
node-[1-2,4-25,27-72,74-96,98-123,125-131]
```

```
2021-12-08 04:33:38 WARNING  __main__   | node-113: infoROM is corrupted at gpu 0002:00:00.0
2021-12-08 04:33:38 WARNING  __main__   | node-118: infoROM is corrupted at gpu 000B:00:00.0
```

- Given above have to choose whether to exclude nodes with high correctable ecc error or inforam issues.
    - Taking a call to exclude “inforom is corrupted” hosts and choosing to go with following hosts:

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RESTORE_FILE="${BLOB_PREFIX}/checkpoint_18_51750.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.47

INCLUDED_HOSTS=node-[1-2,4-25,27-72,74-96,98-112,114-117,119-123,125-131] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"

# monitor train.log
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>
/shared/home/namangoyal/checkpoints/175B/175B_run12.46*/train.log --modified-threshold 900

# Add new entry for TB
cd /shared/home/namangoyal/checkpoints/175B/tensorboard
sudo ln -s
../175B_run12.47.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb
_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr4.5e-05.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.w
d0.1.ms8.uf1.mu143052.s1.ngpu992/tb/ run12.47
```

2021-12-07 10:48am ET: ECC error, requeueing 12.46

- Blame to **node-26**
    - Based on first line in stderr: “srun: error: node-26: task 196: Aborted (core dumped)”
    - ssh to node and nvidia-smi:
        - Unable to determine the device handle for GPU 000B:00:00.0: GPU is lost.
- Undraining **node-108**, which had “port error” previously
    - sudo scontrol update node=node-108 state=resume

- Requeue with:
    - sudo scontrol requeue job=2487
    - sudo scontrol hold job=2487
    - sudo scontrol update job=2487 NodeList=node-[1-2,4-25,27-50,52-64,66-128]
    - sudo scontrol release job=2487

2021-12-06 8:30am ET: Lowering LR and launching 12.46

- Lowering LR to 4e-5

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RESTORE_FILE="${BLOB_PREFIX}/checkpoint_16_46250.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.46

INCLUDED_HOSTS=node-[1-2,4-50,52-64,66-107,109-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"

# monitor train.log
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>
/shared/home/namangoyal/checkpoints/175B/175B_run12.46*/train.log --modified-threshold 900
```

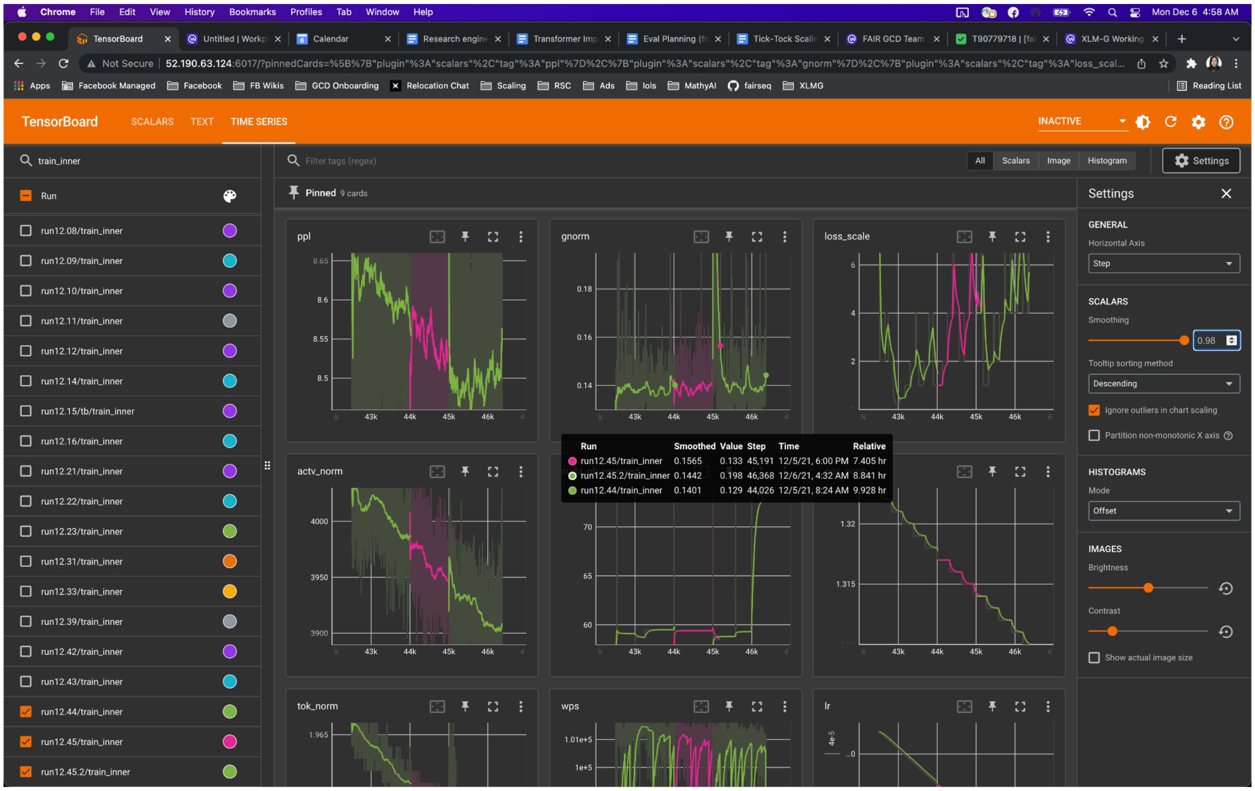
2021-12-06 05:13 PT: Job Hanging - 3 Machines Down

- 5 nodes are partly or fully compromised.
    - 2x: Unable to determine the device handle for GPU 0002:00:00.0: GPU is lost.
        - node-51
        - node-3
    - 2x: NCCL error (Got async event : port error)
        - [note: unclear if this means the nodes can’t be used; we’ve been using one of them successfully for the last 12 hours]
        - node-108
        - node-53
    - 1x: nvidia-smi hangs
        - node-65
- Next steps:

- ○ The "port error" nodes seem to be fine -- we were actually using one of them last night without a problem.
- ○ Will reprovision **node-51** and **node-3** now, since they will not get reallocated to us, since Cloud's built-in health checks will reject them

2021-12-06 05:00 PT: Grad norm spiking, ppl trending up

[From Susan: reading more tea leaves here, but seems like we've had a couple of grad norm spikes and our ppl is now slowly diverging. Recommending restarting with half the LR]



Not Secure | 52.190.63.124:6017/?pinnedCards=%5B%7B"plugin"%3A"scalars"%2C"tag"%3A"ppl"%7D%2C%7B"plugin"%3A"scalars"%2C"tag"%3A"gnorm"%7D%2C%7B"plugin"%3A"scalars"%2C"tag"%3A"loss_scal...

train_inner

Run

- □ run12.08/train_inner
- □ run12.09/train_inner
- □ run12.10/train_inner
- □ run12.11/train_inner
- □ run12.12/train_inner
- □ run12.14/train_inner
- □ run12.15/tb/train_inner
- □ run12.16/train_inner
- □ run12.21/train_inner
- □ run12.22/train_inner
- □ run12.23/train_inner
- □ run12.31/train_inner
- □ run12.33/train_inner
- □ run12.39/train_inner
- □ run12.42/train_inner
- □ run12.43/train_inner
- ☑ run12.44/train_inner
- ☑ run12.45/train_inner
- ☑ run12.45.2/train_inner

Pinned 9 cards

Run	Smoothed	Value	Step	Time	Relative
● run12.45/train_inner	0.1565	0.133	45,191	12/5/21, 6:00 PM	7.405 hr
○ run12.45.2/train_inner	0.1442	0.198	46,368	12/6/21, 4:32 AM	8.841 hr
● run12.44/train_inner	0.1401	0.129	44,026	12/5/21, 8:24 AM	9.928 hr

Settings

GENERAL
Horizontal Axis: Step

SCALARS
Smoothing: 0.98
Tooltip sorting method: Descending
☑ Ignore outliers in chart scaling
□ Partition non-monotonic X axis

HISTOGRAMS
Mode: Offset

IMAGES
Brightness
Contrast
□ Show actual image size

2021-12-05 9pm ET: Requeue after GPU error

- RuntimeError: CUDA error: CUBLAS_STATUS_EXECUTION_FAILED when calling `cublasGemmEx( handle, opa, opb, m, n, k, &falpha, a, CUDA_R_16F, lda, b, CUDA_R_16F, ldb, &fbeta, c, CUDA_R_16F, ldc, CUDA_R_32F, CUBLAS_GEMM_DFALT_TENSOR_OP)`
- Blame is **node-51**
  - ○ Ran nvidia-smi and **node-51** has "Unable to determine the device handle for GPU 0002:00:00.0: GPU is lost. Reboot the system to recover this GPU"
- Requeue while replacing **node-51** with **node-88**
  - ○ `sudo scontrol requeue job=2486`
  - ○ `sudo scontrol hold job=2486`
  - ○ `sudo scontrol update job=2486 NodeList=node-[1,3-50,52-64,66-107,109-128]`
    - ■ This replaces **node-51** with **node-88**
  - ○ `sudo scontrol release job=2486`
- Note: It took 50 minutes to resume training from checkpoint_15_45000!
  - ○ ~30 minutes just fast-forwarding the dataloader!
  - ○ Also added more logging: PR #2748

2021-12-05 18:30 ET: Poking through dmesg to see if we can find where we hung

- Noticed “nvidia-nvswitch: Version Mismatch”. Happens on both pg0-1 and pg0-53
- Dmesg actually reports when the job is officially hung: “task python:35662 blocked for more than 120 seconds” (Though I think this might be the dataloader workers, not the main proc?)
- When controlling for looking at timestamps, i don’t see a lot in dmesg :(

2021-12-05 12:15pm ET: Requeueing 12.44 and 12.45

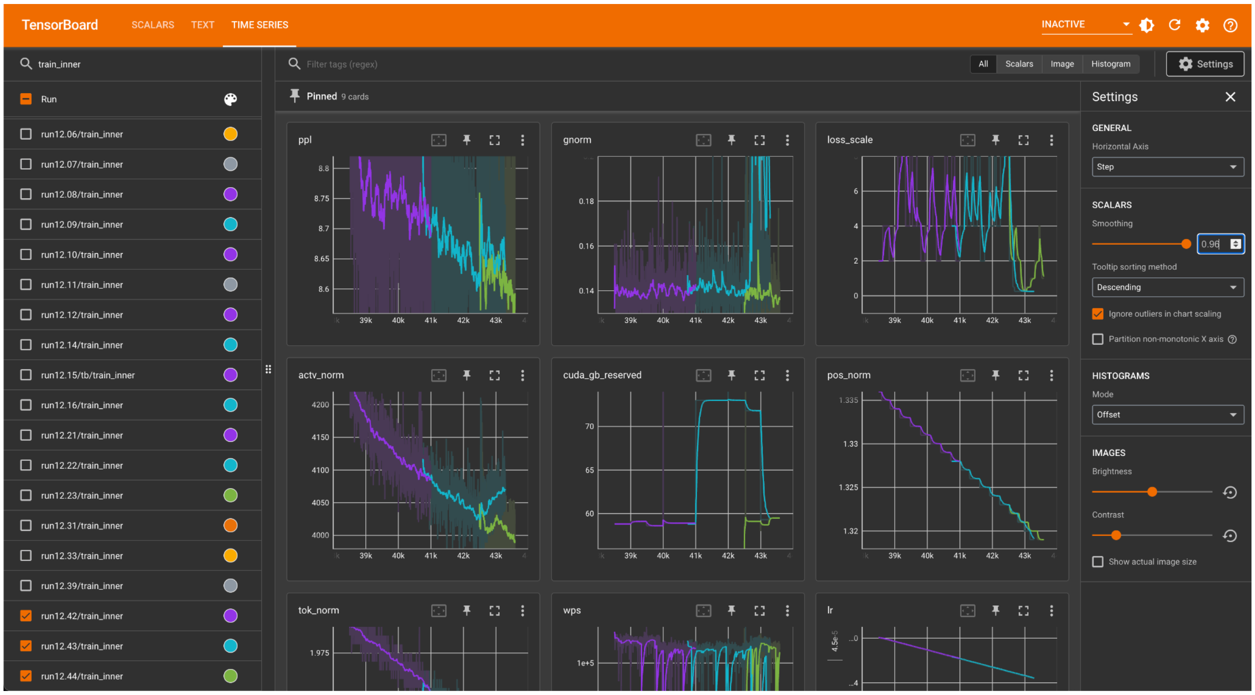
- There was a NCCL error (Got async event : port error) on **node-108** and **node-53**
    - It’s not clear which node is bad…
    - How to find the bad nodes:
        - There are two hosts that have port errors in train.log: buo1u000030 and buo1u00001K
        - To translate these into node-XX hostnames:
            - ssh to each host (e.g., buo1u000030)
            - get IP address (e.g., `ifconfig eth0` yields 10.30.4.109)
            - use /etc/hosts to map IP address (e.g., `grep 10.30.4.109 /etc/hosts` yields **node-108**)
- Action item: update monitor_train_log.py script to alert to port errors
    - “NCCL WARN NET/IB : Got async event :”
- Going to try requeueing and swapping the node out via scontrol
    - `sudo scontrol requeue job=2483`
    - `sudo scontrol hold job=2483`
    - `sudo scontrol update job=2483 NodeList=node-[1,3-64,66-87,89-107,109-128]`
        - This replaces **node-108** with **node-113**
    - `sudo scontrol release job=2483`
    - Didn’t work, seems to be trying to download into a blob URL :/
- Fall back to manual launch:

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RESTORE_FILE="${BLOB_PREFIX}/checkpoint_15_44000.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.45

INCLUDED_HOSTS=node-[1,3-64,66-87,89-107,109-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

```
# monitor train.log
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>
/shared/home/namangoyal/checkpoints/175B/175B_run12.44*/train.log
```

2021-12-05 05:35 PT: Checking on 12.44

[From Susan: looks like lowering LR helps keep us on track wrt ppl. Loss scale that stays below 1 for too long could be a leading indicator of instability going forward. Set smoothing to ~0.95 to see these trends.]



train_inner

■ Run

□ run12.06/train_inner
□ run12.07/train_inner
□ run12.08/train_inner
□ run12.09/train_inner
□ run12.10/train_inner
□ run12.11/train_inner
□ run12.12/train_inner
□ run12.14/train_inner
□ run12.15/tb/train_inner
□ run12.16/train_inner
□ run12.21/train_inner
□ run12.22/train_inner
□ run12.23/train_inner
□ run12.31/train_inner
□ run12.33/train_inner
□ run12.39/train_inner
☑ run12.42/train_inner
☑ run12.43/train_inner
☑ run12.44/train_inner

Settings

GENERAL
Horizontal Axis: Step

SCALARS
Smoothing: 0.96
Tooltip sorting method: Descending
☑ Ignore outliers in chart scaling
□ Partition non-monotonic X axis

HISTOGRAMS
Mode: Offset

IMAGES
Brightness
Contrast
□ Show actual image size

2021-12-05 02:24 ET: Side experiment

Note that I (Stephen) launched a 768M equivalent model to train for a bit to help unblock Anjali. If you need spare nodes, please just kill the job, as it is less important.

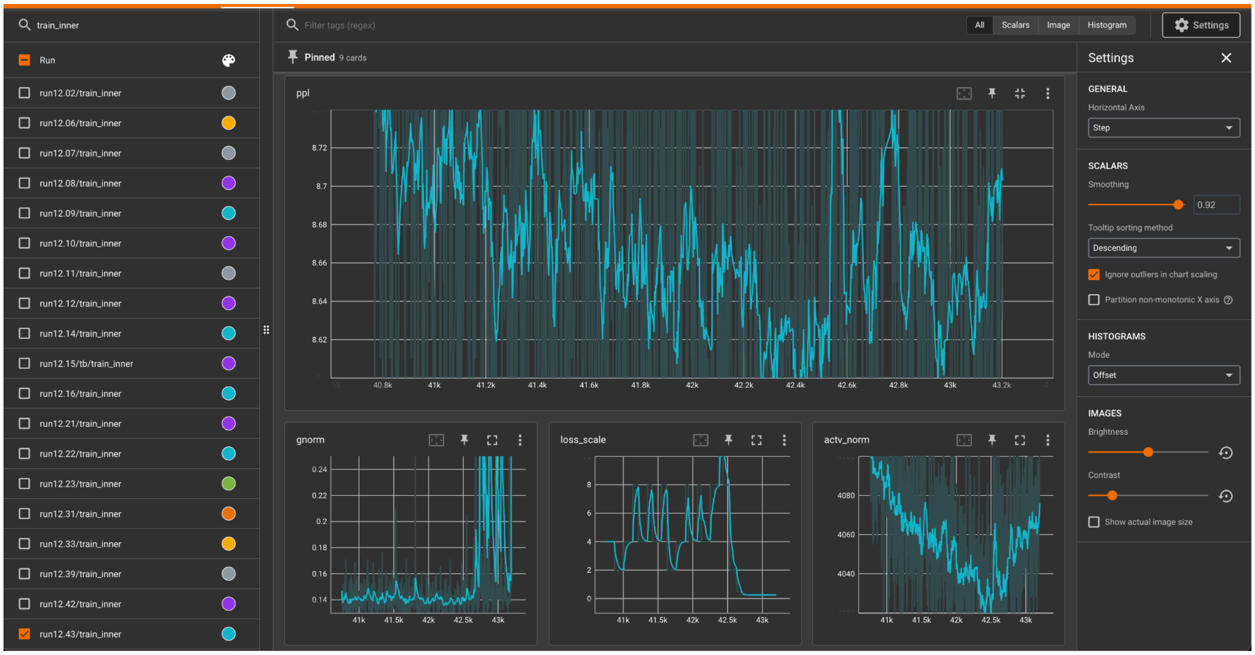
02:56 ET both runs look stable. Signing off.

2021-12-05- 00:00 ET: Loss scale exploding 2 - 12.44

Note that if we decide to relaunch this, there's an epoch boundary VERY SHORTLY after 42500 that might be preferable.

- Susan pointed out how gnorms seem unusually high. Individual updates are fine, but the frequency of spikes has definitely increased (from 12.42+12.43, 58/73 of the spikes >0.2 have been since 42500 updates)
- Proposed mitigation: Roll back to 42000 and resubmit.
- Alternative mitigation: Lower learning rate further
  - Opted to lower it by a factor of 0.9
  - And restore from 42500

- And **bumped the blob folder**



train_inner

☐ Run

☐ run12.02/train_inner
☐ run12.06/train_inner
☐ run12.07/train_inner
☐ run12.08/train_inner
☐ run12.09/train_inner
☐ run12.10/train_inner
☐ run12.11/train_inner
☐ run12.12/train_inner
☐ run12.14/train_inner
☐ run12.15/tb/train_inner
☐ run12.16/train_inner
☐ run12.21/train_inner
☐ run12.22/train_inner
☐ run12.23/train_inner
☐ run12.31/train_inner
☐ run12.33/train_inner
☐ run12.39/train_inner
☐ run12.42/train_inner
☑ run12.43/train_inner

Settings

GENERAL
Horizontal Axis: Step

SCALARS
Smoothing: 0.92
Tooltip sorting method: Descending
☑ Ignore outliers in chart scaling
☐ Partition non-monotonic X axis

HISTOGRAMS
Mode: Offset

IMAGES
Brightness
Contrast
☐ Show actual image size

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RESTORE_FILE="${OLD_BLOB_PREFIX}/checkpoint_14_42500.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.44

INCLUDED_HOSTS=node-[1,3-64,66-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"

# monitor train.log
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>
/shared/home/namangoyal/checkpoints/175B/175B_run12.44*/train.log
```

2021-12-04 20:24 ET: Loss scale exploding

- Observed loss scale exploding via alerting.
    - Waiting some short time to requeue
    - It's teetering. Checking back in 30.

- Note: we did manage to checkpoint during this (u 42750). We probably want to roll back to 42500 just because initializing from a bad loss scale is bad. Loss scale was 8! At that moment in time
- Still confirmed no uploads. Looks like to restore from checkpoint i will need to copy from all the nodes to a safe directory.
- Forced uploading of all local checkpoints via:
    - pdsh -R ssh -w “$(squeue | tail -n 1 | awk ‘{print $8}’)” ~roller/upload_checkpoints.sh | tee scary_upload_log
- By the time the upload had finished, we had hit 43000 updates and **out of alert territory**. However, loss scale was still only 0.25.
- Decided to again, let it live on. **NO ACTUAL ACTION WAS TAKEN WRT THE JOB**

2021-12-04 5:35am ET: Launch of 12.43: Fix blob upload

- Analysis of 12.42:
    - There was some CUDA error around 1am that caused training to hang
        - `RuntimeError: CUDA error: unspecified launch failure`
        - `CUDA kernel errors might be asynchronously reported at some other API call,so the stacktrace below might be incorrect.`
    - Looks like **node-65** is to blame (nvidia-smi is slow, has hung process), but I suggest leaving this node in drain for a couple days to report to CSP
- Launch of 12.43
    - Using this as an opportunity to replace the blob URL with a new one that works
        - Using a fresh blob URL that points to Susan’s blob container: `/susanz/2021-12-04/175B_run12.42`
        - Since checkpoints were sitting on local disk on each node, I manually uploaded all of them to the new path with this hacky script.
    - Replaced **node-65** with **node-71**

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

# use new blob URL
BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"

RESTORE_FILE="${BLOB_PREFIX}/checkpoint_14_40750.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.43

INCLUDED_HOSTS=node-[1,3-64,66-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}/?${BLOB_AUTH}"
```

```
# monitor train.log
./scripts/cloud/monitor_train_log.py --mailto <scrubbed>
/shared/home/namangoyal/checkpoints/175B/175B_run12.43*/train.log
```

2021-12-03, 10:35pm ET: DO NOT REBOOT OR REPROVISION ANY NODES UNTIL FURTHER NOTICE

It seems the upload-to-blob is broken due to a limit on the Cloud side, so the latest checkpoints are sitting on the local disks on each node and are not being uploaded to blob. I manually copied 40500 to /data for now, but please do not reprovision any nodes until this is resolved, since we will lose the latest checkpoint data in that case.

The error seems to be “409 The uncommitted block count cannot exceed the maximum limit of 100,000 blocks.”

- Some Google’ing suggests this is due to having too many incomplete/failed uploads to the given path.
- Manually uploading any file to the blob path consistently fails with the same error.
- I tried deleting some of the stored checkpoints to see if that resolved anything, but no luck.
    - Note: I’ve now deleted most of the historical checkpoints at *250 and *750 steps, so between ~25k and ~40k steps we now only have checkpoints at *500 and *000 steps
- This CSP help article suggests the “Wait 7 days for the uncommitted block list to garbage collect.”
- Seems the easiest fix for now is to switch to a new blob container.
- **Mitigation:** Switch to a new blob container: `susanz/2021-12-04/175B_run12.42`

2021-12-03

- ~~Myle to finish the SGD code~~ -- done
- Stephen to launch a 4 node tiny model for Anjali

2021-12-03 7:20am ET: Launch of 12.42: Switch back to Adam

- 12.41 seemed to make no progress in terms of pnorms or loss
- We also implemented SGD instead of SGDW (i.e., weight decay was wrong)
- Proposal: roll back to 12.39 with AdamW and further lower learning rate from 9e-5 to 6e-5 to match GPT-3
    - #2736

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gshard_combine_megatron_fsdp
git checkout gshard_combine_megatron_fsdp

BLOB_PREFIX="<<<SCRUBBED FOR RELEASE>>>"
BLOB_AUTH="<<<SCRUBBED FOR RELEASE>>>"
```

```
RESTORE_FILE="${BLOB_PREFIX}/checkpoint_13_38500.pt?${BLOB_AUTH}"

RUN_ID=175B_run12.42

INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/$(date +%Y-%m-%d).$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path "${BLOB_PREFIX}?${BLOB_AUTH}"
```

2021-12-02 17:16 ET: Fake SGD debacle: Debrief discussion

Summary of events and mitigations

Current hypothesis is that all our beta1=0 runs didn't work.

- Reason for this bug was because adam state dicts load the betas
- 12.35 (cloud checkpoints caused to never launch) and 12.36 (beta1=0) and 12.37 (simple requeue of 36)
- Note 12.38 and 12.39 were always meant to be true adam

Decision made to hard switch to true vanilla SGD (with launch of 12.41). Note, this required implementing fp16-friendly SGD.

NOTE FOR FUTURE: BECAUSE OF THE BETA1 BUG [WARNING: See 2021-12-02 17:16 ET: Debrief] therefore any ablations with 12.36/12.37 are no longer valid.

Next paths

- Prediction: 12.41 [true sgd] is probably going to drop rapidly due to rapidly annealing learning rate.
- Some debate
- Action Item: We should review the megatron code and see if they have anything about the switching. We don't believe they did this trick.

2021-12-02 16:08 ET: Launch of 12.41: Switching to true Vanilla SGD

DO NOT REQUEUE THIS RUN, IT CONTAINS RESTORE FILE LOGIC THAT WILL RESET BACK TO 37K EVERY TIME!

DO NOT REQUEUE THIS RUN, IT CONTAINS --RESET-OPTIMIZER LOGIC!

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gcmf-1241
git checkout gcmf-1241

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>”

CKPT_DIR=/data/users/susanz/checkpoints

# these two lines skipped because susan already downloaded them, but leaving them as future reference
#### cd $CKPT_DIR
#### cp --recursive --include-pattern "checkpoint_13_37000*.pt" "$BLOB_URL" checkpoint_13_37000

RESTORE_FILE=$CKPT_DIR/checkpoint_13_37000/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_13_37000.pt

RUN_ID=175B_run12.41

INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-12-02.$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path $BLOB_URL
```

2021-12-02 10:25am ET: Launch of 12.40: Intended to be fake SGD with lower learning rate [WARNING: See 2021-12-02 17:16 ET: Debrief on why that may not be]

Debate ensued. Decided to launch Fake SGD with:

- Lower learning rate
- Lower clip

Reasoning: The ablations give signal on which direction SGD learning should go. And both changing hyperparameters effectively lower LR so they won’t conflict. Signal is directionally sane.

NOTE FOR FUTURE: BECAUSE OF THE BETA1 BUG [WARNING: See 2021-12-02 17:16 ET: Debrief] therefore any ablations with 12.36/12.37 are no longer valid.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin gcmf-1240
git checkout gcmf-1240

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

CKPT_DIR=/data/users/susanz/checkpoints

# these two lines skipped because susan already downloaded them, but leaving them as future reference
#### cd $CKPT_DIR
#### cp --recursive --include-pattern "checkpoint_13_37000*.pt" "$BLOB_URL" checkpoint_13_37000

RESTORE_FILE=$CKPT_DIR/checkpoint_13_37000/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_13_37000.pt

RUN_ID=175B_run12.40

INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-12-02.$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path $BLOB_URL
```

2021-12-01 1:30pm ET: Launch of 12.39

Analysis of 12.38

Exploded 10 steps early compared to 12.33! Woot!
Spent a surprising amount of time riding the line of 0.0625 loss pretty happily

Launch of 12.39

DO NOT REQUEUE THIS RUN, IT CONTAINS RESTORE FILE LOGIC THAT WILL RESET BACK TO 37K EVERY TIME! Additionally, instead of requeueing, we should initiate the launch of 12.40!

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin back2adam12.39_gshard_combine_megatron_fsdp
git checkout back2adam12.39_gshard_combine_megatron_fsdp

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

CKPT_DIR=/data/users/susanz/checkpoints

# these two lines skipped because susan already downloaded them, but leaving them as future reference
#### cd $CKPT_DIR
#### cp --recursive --include-pattern "checkpoint_13_37000*.pt" "$BLOB_URL" checkpoint_13_37000

RESTORE_FILE=$CKPT_DIR/checkpoint_13_37000/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_13_37000.pt

RUN_ID=175B_run12.39

INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-12-01.$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path $BLOB_URL
```

Discussion

Decision: Why not both?

- Ablate and optimize for knowledge
- Both runs beginning at 37k
- Both runs will keep the loss scale changes. We handled lower rates for a healthy amount of time, and the end result was the same.
- Launch 12.39:
    - ADAM with max LR adjusted to 9e-5 (bringing us down to 6.8e-5 at current step)
    - Keep clip at 0.3
    - Launched first bc it’s more likely to die quickly, therefore make a happier oncall.
- Launch 12.40:
    - Fake SGD with max LR left the same (1.2e-4) and clip lowered to 0.3

Analysis

Observations:

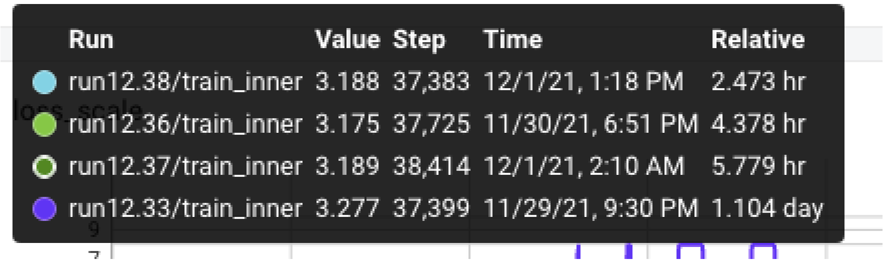
- Fake SGD made it further than ADAM
    - It made it a bit further from a requeue (1400 updates from CP, vs 400)
    - Without requeue it slightly later (725 updates from CP vs 400)
    - Regardless of ADAM/SGD decision, we *probably* should change something else.
- The loss scale changes didn’t change help ADAM:

    - It dies in the exact same places as before (within 10 updates)
- Gnorm is spiking in the same places for both (data!) but magn differ by method greatly.
- No signal on validation performance.

Key:

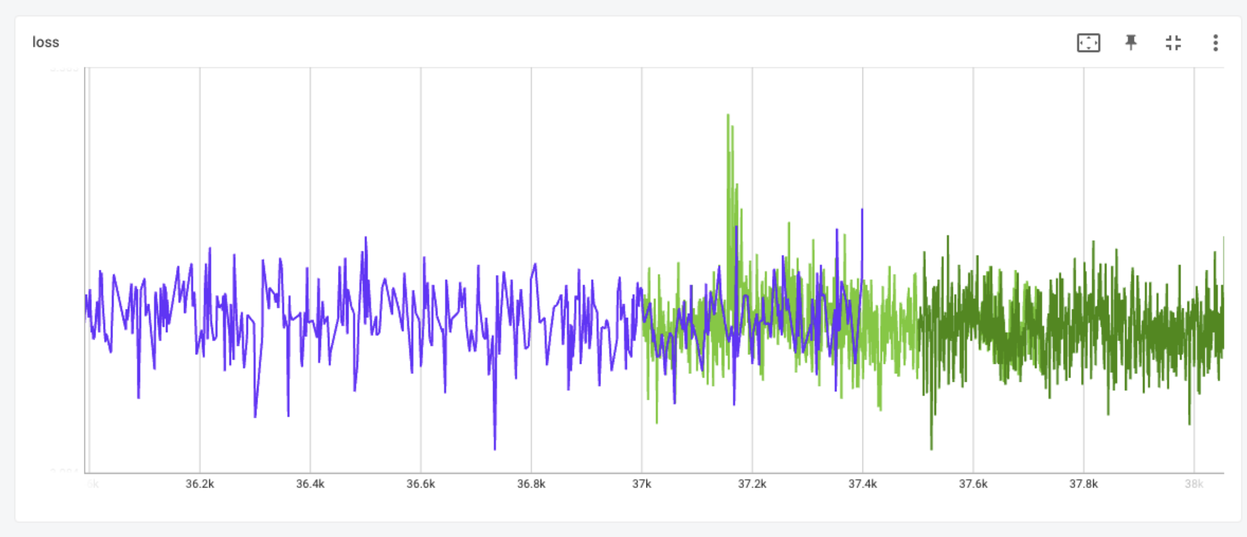
- Purple = Old ADAM run (12.33)
- Light green = Fake SGD Run (12.36)
- Dark green = Manual requeue of Light green after it hit 0.25 bottom out (12.37)
- Blue = New ADAM run (12.38)

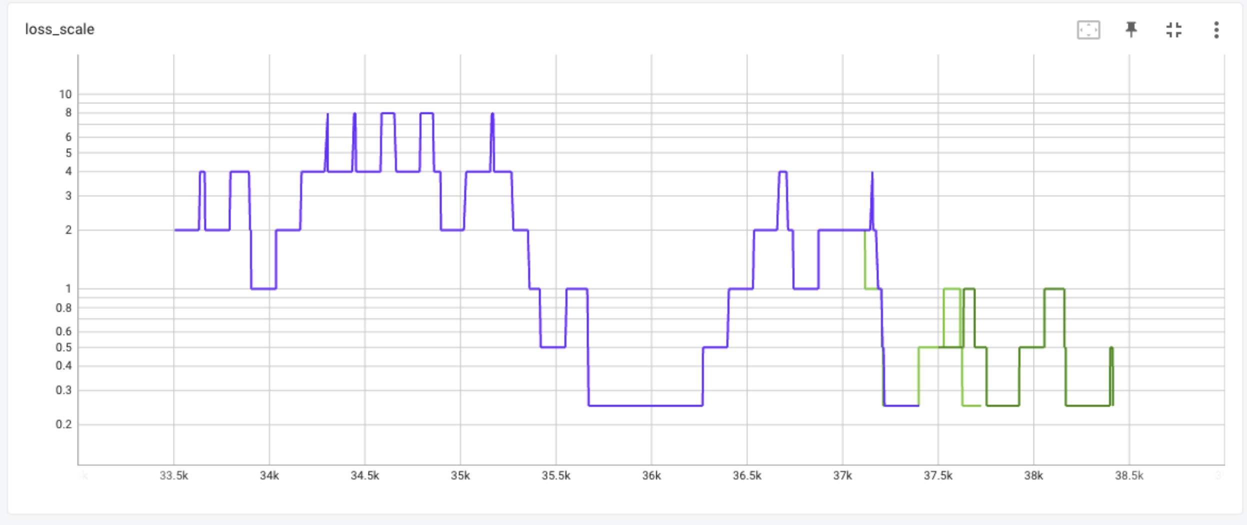
Max distance each got:

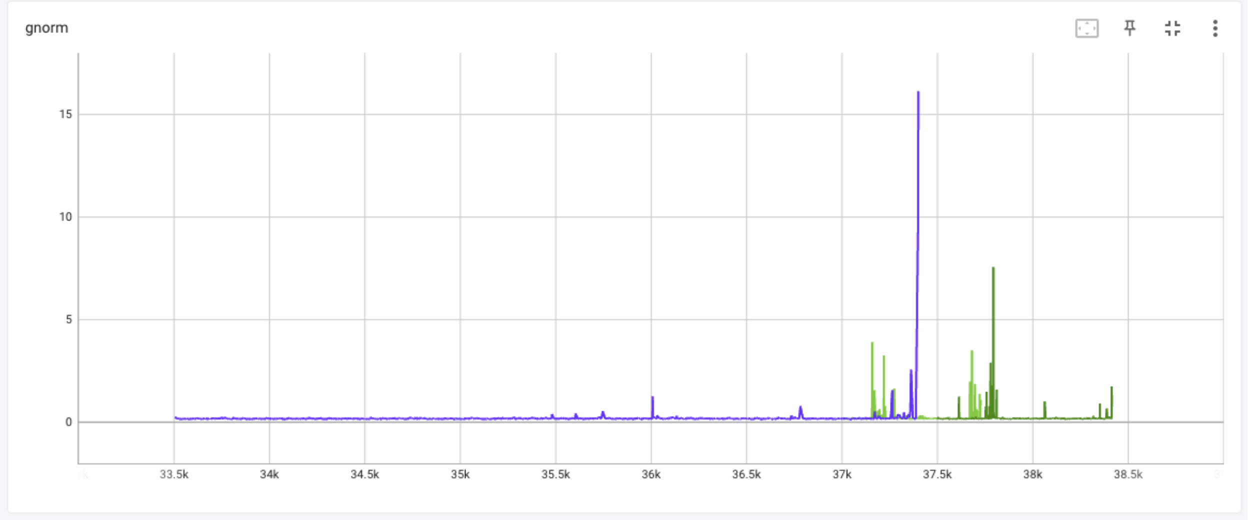


Run	Value	Step	Time	Relative
● run12.38/train_inner	3.188	37,383	12/1/21, 1:18 PM	2.473 hr
● run12.36/train_inner	3.175	37,725	11/30/21, 6:51 PM	4.378 hr
○ run12.37/train_inner	3.189	38,414	12/1/21, 2:10 AM	5.779 hr
● run12.33/train_inner	3.277	37,399	11/29/21, 9:30 PM	1.104 day

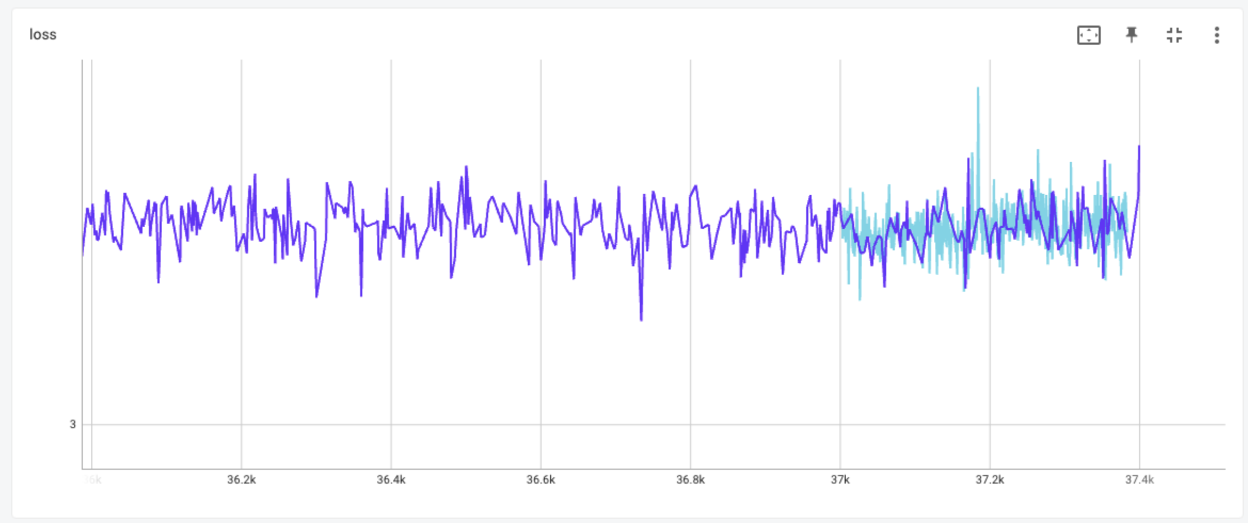
First just Fake SGD runs:

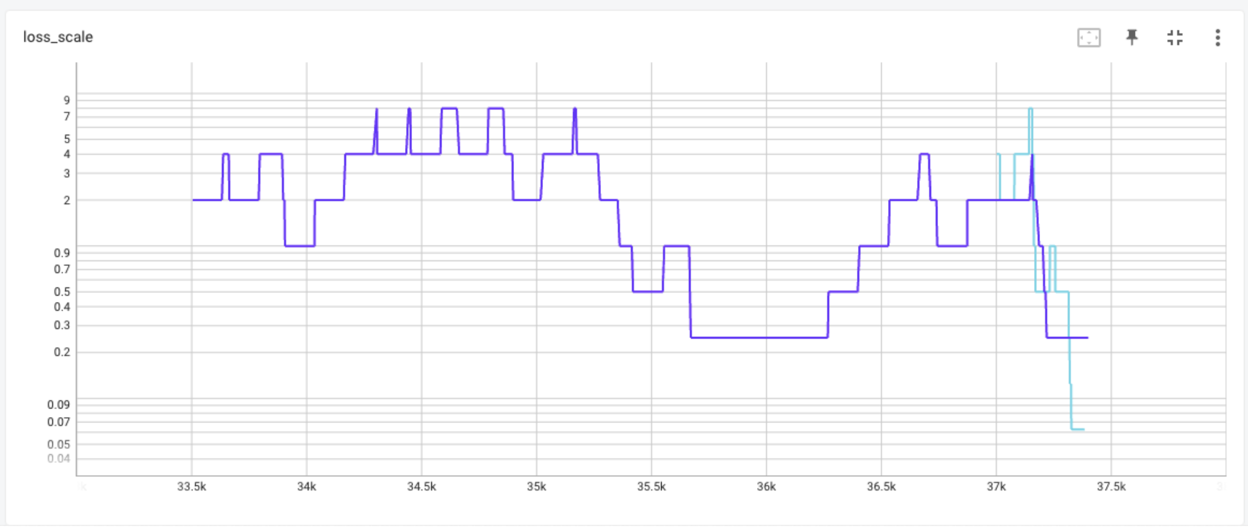


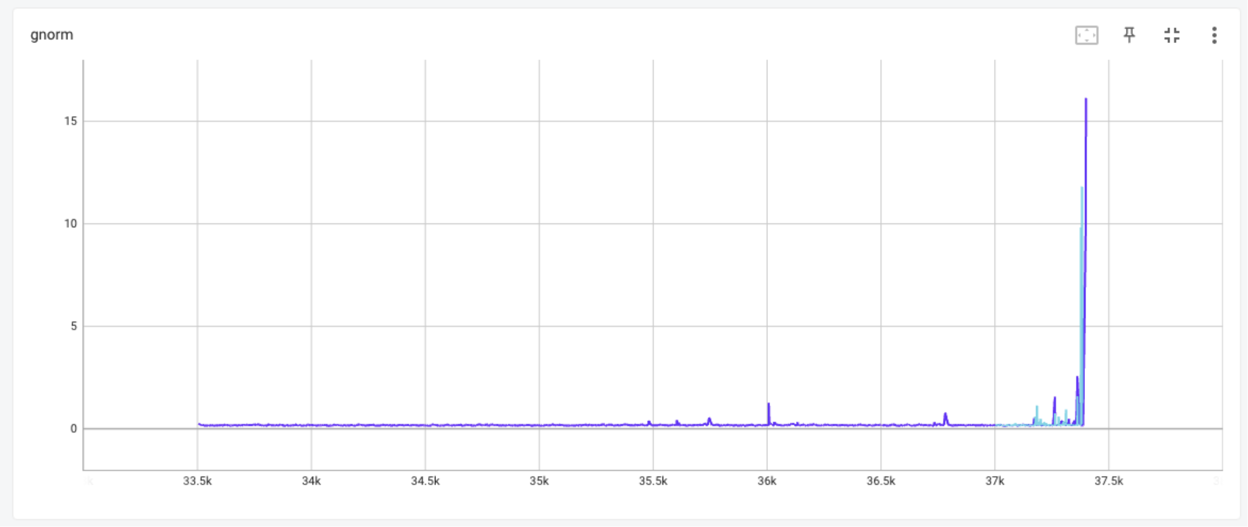




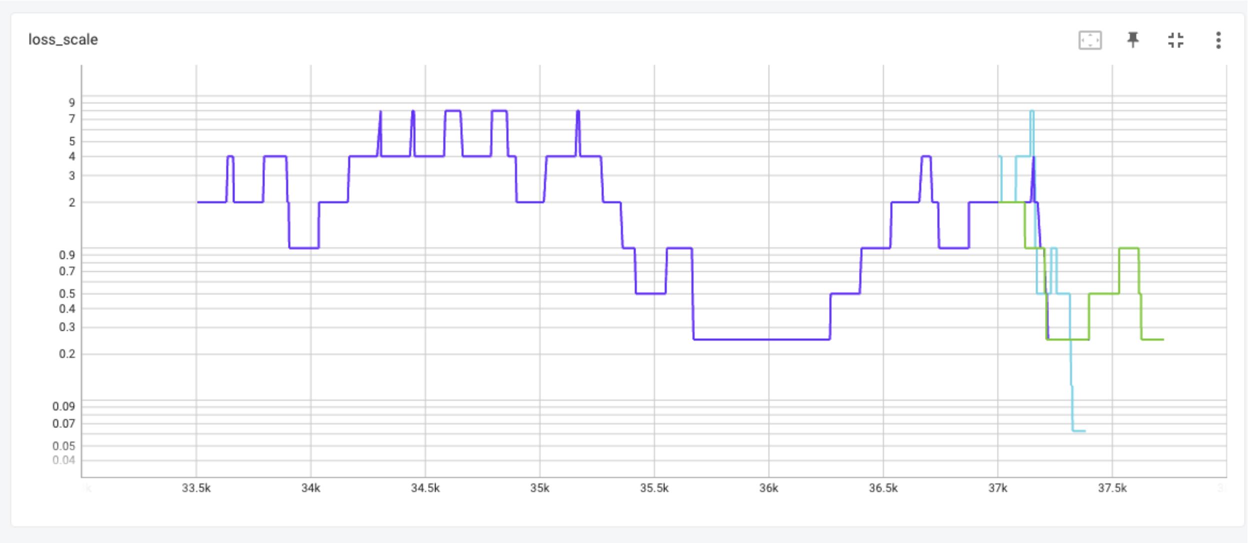
Next just the adam runs (Too difficult to distinguish on the same graph)







And here is both Fake SGD (with the requeue) and ADAM:



2021-12-01 8:39am ET: 12.38 True Adam with Lower LR

DO NOT REQUEUE THIS RUN, IT CONTAINS RESTORE FILE LOGIC THAT WILL RESET BACK TO 37K EVERY TIME!

And we broke at the exact same spot. (38414 updates)

Summary of discussions for next steps:

- Proposals:
    - Switch to true SGD
    - Futz with loss scale windows / logic
    - Change clipping back to 0.3
- Decision:
    - Revert to change 37k (last pure adam ckpt), back to pure adam / clip 0.3
    - Change the loss scale logic to halve the raise window 132 => 64

    - Change the loss scale logic to never load from checkpoint
    - Changes in #2714
- Rationale
    - We get a nice post-hoc comparison of fake-sgd vs adam
    - We're seeing this loss scale issue come up for both fake-sgd and adam
    - Requeues are regularly buying us second lives, which are essentially only messing with loss scale windows

Launching 12.38:

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin back2adam12.38_gshard_combine_megatron_fsdp
git checkout back2adam12.38_gshard_combine_megatron_fsdp

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

CKPT_DIR=/data/users/susanz/checkpoints

# these two lines skipped because susan already downloaded them, but leaving them as future reference
#### cd $CKPT_DIR
#### cp --recursive --include-pattern "checkpoint_13_37000*.pt" "$BLOB_URL" checkpoint_13_37000

RESTORE_FILE=$CKPT_DIR/checkpoint_13_37000/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_13_37000.pt

RUN_ID=175B_run12.38

INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-12-30.$RUN_ID \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path $BLOB_URL
```

2021-12-01 2:21am ET: [Stephen oncall] Run 12.37 [WARNING: See 2021-12-02 17:16 ET: Debrief on why it may not be SGD]

We got much further this time (38414 updates) but did start hitting 0.25's. Requeued.

2021-11-30 7:24pm ET: [Stephen oncall] Run 12.37 Manual requeue of 12.36. [WARNING: See 2021-12-02 17:16 ET: Debrief on why it may not be SGD]

Hit 37725 updates and started getting 0.25's. Couldn't requeue bc previous run ignored downloading checkpoints. Reverted that diff and relaunched unchanged.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
# changes for restoring without downloading, and also changes to loss scale logic
git fetch origin sgd_withdownload_gshard_combine_megatron_fsdp
git checkout sgd_withdownload_gshard_combine_megatron_fsdp

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

RUN_ID=175B_run12.37

INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-29.$RUN_ID \
    --full-cloud-upload-path $BLOB_URL
```

2021-11-20 7:24pm ET: [Stephen oncall]

Sam called out the loss scalar hitting minimum. Took action to requeue the job

A few minutes later, Susan pointed out that wouldn’t work because we reverted the download-from-cloud change.

See next entry for mitigation.

2021-11-30 10:10am PT: 12.36 restart from 37k, SGD mimicking [WARNING: See 2021-12-02 17:16 ET: Debrief on why it may not be SGD]

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

cd ~/src/fairseq-py
git fetch origin sgd_gshard_combine_megatron_fsdp
git checkout sgd_gshard_combine_megatron_fsdp

CKPT_DIR=/data/users/susanz/checkpoints

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_13_37000*.pt" "$BLOB_URL" checkpoint_13_37000

RESTORE_FILE=$CKPT_DIR/checkpoint_13_37000/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_13_37000.pt

RUN_ID=175B_run12.36
```

```
INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE \
    --full-cloud-upload-path $BLOB_URL
```

- Was just about to launch and saw all nodes to launch are were in drain:

```
(fairseq-20210913) susanz@ip-0A1E0404:~/fairseq-py$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
hpc*         up   infinite      1 drain~ node-129
hpc*         up   infinite      2  down~ node-[130-131]
hpc*         up   infinite      1  idle~ node-132
hpc*         up   infinite    124  drain node-[1,3-70,72-87,89-112,114-128]
hpc*         up   infinite      4   idle node-[2,71,88,113]
```

- sudo scontrol update node=node-[1,3-70,72-87,89-112,114-128] state=idle

2021-11-30 9:00am PT: 12.35 restart from 37k, SGD mimicking [WARNING: See 2021-12-02 17:16 ET: Debrief on why it may not be SGD]

- Downloaded checkpoints for 37k.
- Checkpoints got borked / clobbered by “always download cloud checkpoints” logic. Redownloading and reverting that change.
- Resized /data up to 85TB (was almost full at 70TB).

Note: This never actually ran

2021-11-30 9:00am ET: 12.34 requeue

[Stephen] Looks like it’s got an enormous number of 0.25 loss scales, basically all night long. Requeueing.

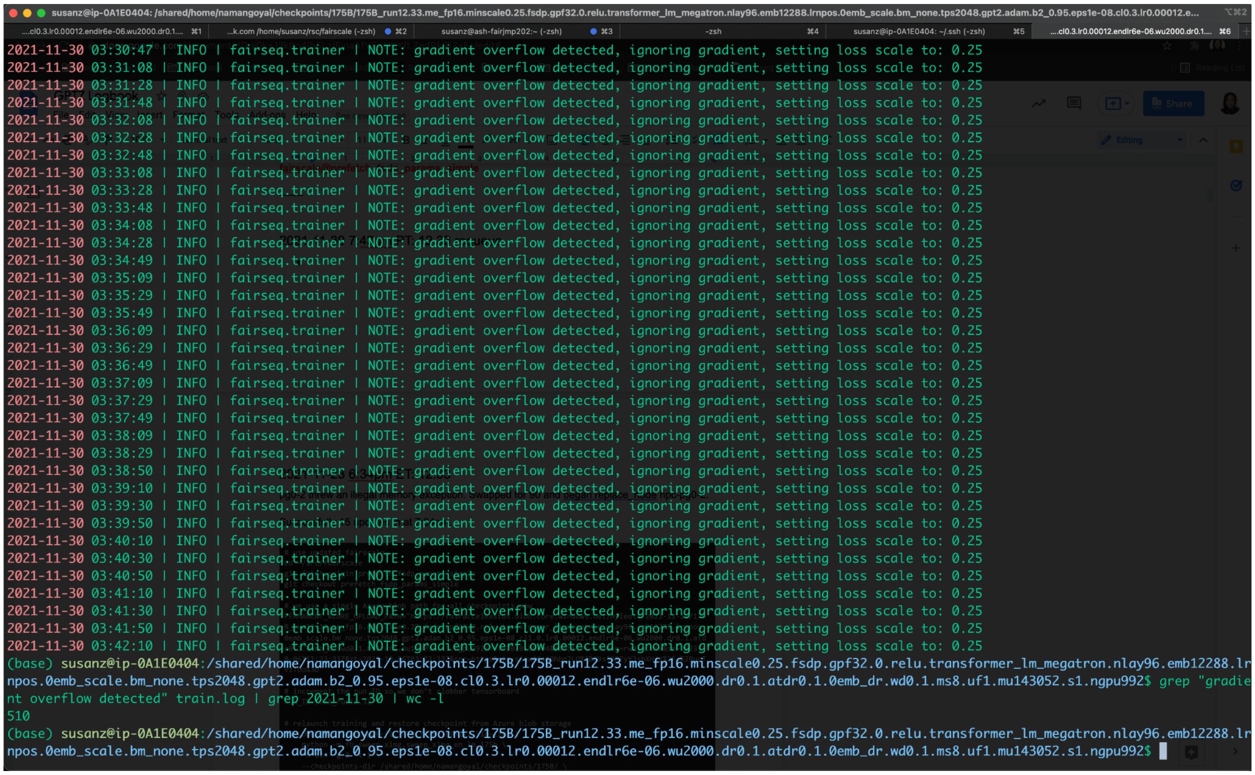
2021-11-29 7:43pm PT [Susan]: 12.34 restart

- Restarting with 6e4124680c960a8bc584f2fb0d4404a232745e8b pulled in

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
RUN_ID=175B_run12.34

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-29.$RUN_ID \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

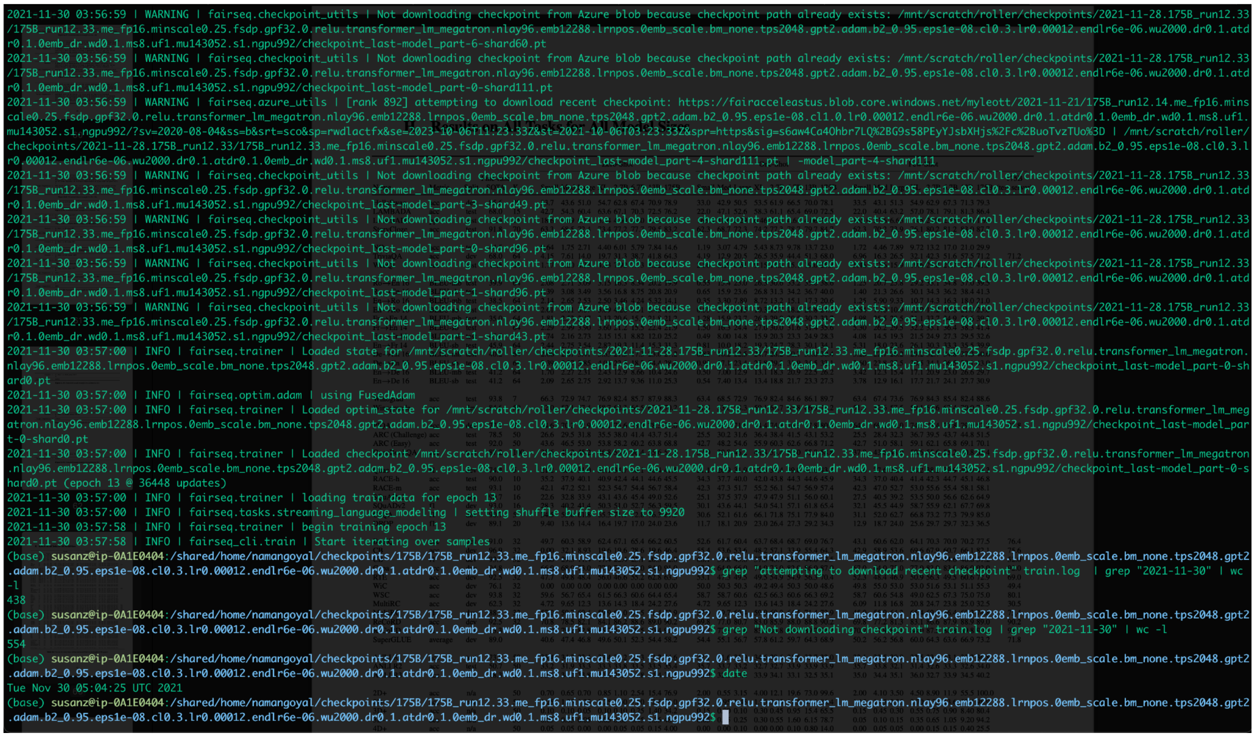
2021-11-30 7:43pm PT [Susan]: 12.33 requeue

- Got stuck for 510 iterations with gradient overflow / loss scale at 0.25


susanz@ip-0A1E0404: /shared/home/namangoyal/checkpoints/175B/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.e...
⌥⌘2
....cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.... ⌘1
...k.com /home/susanz/rsc/fairscale (-zsh) ⌘2
susanz@ash-fairjmp202:~ (-zsh) ⌘3
-zsh ⌘4
susanz@ip-0A1E0404: ~/.ssh (-zsh) ⌘5
....cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.... ⌘6

```
2021-11-30 03:30:47 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:31:08 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:31:28 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:31:48 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:32:08 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:32:28 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:32:48 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:33:08 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:33:28 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:33:48 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:34:08 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:34:28 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:34:49 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:35:09 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:35:29 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:35:49 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:36:09 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:36:29 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:36:49 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:37:09 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:37:29 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:37:49 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:38:09 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:38:29 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:38:50 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:39:10 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:39:30 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:39:50 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:40:10 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:40:30 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:40:50 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:41:10 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:41:30 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:41:50 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-30 03:42:10 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lr
npos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ grep "gradie
nt overflow detected" train.log | grep 2021-11-30 | wc -l
510
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lr
npos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$
```

- sudo scontrol requeue job=2229
  - Takes about 5 minutes to get nodelist out of (BeginTime) state
- This doesn’t look right:



```
2021-11-30 03:56:59 | WARNING | fairseq.checkpoint_utils | Not downloading checkpoint from Azure blob because checkpoint path already exists: /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33
/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atd
r0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-6-shard60.pt
2021-11-30 03:56:59 | WARNING | fairseq.checkpoint_utils | Not downloading checkpoint from Azure blob because checkpoint path already exists: /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33
/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atd
r0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-0-shard111.pt
2021-11-30 03:56:59 | WARNING | fairseq.azure_utils | [rank 892] attempting to download recent checkpoint: https://fairacceleastus.blob.core.windows.net/myleott/2021-11-21/175B_run12.14.me_fp16.mins
cale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.
mu143052.s1.ngpu992/?sv=2020-08-04&ss=b&srt=sco&sp=rwdlactfx&se=2023-10-06T11:23:33Z&st=2021-10-06T03:23:33Z&spr=https&sig=s6aw4Ca4Ohbr7LQ%2BG9s58PEyYJsbXHjs%2Fc%2BuoTvzTUo%3D | /mnt/scratch/roller/
checkpoints/2021-11-28.175B_run12.33/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.l
r0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-4-shard111.pt | -model_part-4-shard111
2021-11-30 03:56:59 | WARNING | fairseq.checkpoint_utils | Not downloading checkpoint from Azure blob because checkpoint path already exists: /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33
/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atd
r0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-3-shard49.pt
2021-11-30 03:56:59 | WARNING | fairseq.checkpoint_utils | Not downloading checkpoint from Azure blob because checkpoint path already exists: /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33
/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atd
r0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-0-shard96.pt
2021-11-30 03:56:59 | WARNING | fairseq.checkpoint_utils | Not downloading checkpoint from Azure blob because checkpoint path already exists: /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33
/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atd
r0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-1-shard96.pt
2021-11-30 03:56:59 | WARNING | fairseq.checkpoint_utils | Not downloading checkpoint from Azure blob because checkpoint path already exists: /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33
/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atd
r0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-1-shard43.pt
2021-11-30 03:57:00 | INFO | fairseq.trainer | Loaded state for /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.
nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-0-sh
ard0.pt
2021-11-30 03:57:00 | INFO | fairseq.optim.adam | using FusedAdam
2021-11-30 03:57:00 | INFO | fairseq.trainer | Loaded optim_state for /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_meg
atron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_par
t-0-shard0.pt
2021-11-30 03:57:00 | INFO | fairseq.trainer | Loaded checkpoint /mnt/scratch/roller/checkpoints/2021-11-28.175B_run12.33/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron
.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-0-s
hard0.pt (epoch 13 @ 36448 updates)
2021-11-30 03:57:00 | INFO | fairseq.trainer | loading train data for epoch 13
2021-11-30 03:57:00 | INFO | fairseq.tasks.streaming_language_modeling | setting shuffle buffer size to 9920
2021-11-30 03:57:58 | INFO | fairseq.trainer | begin training epoch 13
2021-11-30 03:57:58 | INFO | fairseq_cli.train | Start iterating over samples
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2
.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ grep "attempting to download recent checkpoint" train.log  | grep "2021-11-30" | wc
-l
438
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2
.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ grep "Not downloading checkpoint" train.log | grep "2021-11-30" | wc -l
554
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2
.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ date
Tue Nov 30 05:04:25 UTC 2021
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.33.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2
.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ 
```

- Seems like 438 checkpoints downloaded while 554 didn’t. Unclear if this is intended behavior. Since the job still hasn’t started after over an hour, scanceling and starting from a new checkpoint dir with a new run id.

2021-11-28 6:34pm ET [Stephen]: 12.33

pg0-2 threw an illegal memory exception. Swapped for 90 and began to replace node node-2.

Successfully 15 updates in at 7:04pm.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
RUN_ID=175B_run12.32

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-70,72-87,89-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-28.$RUN_ID \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-28 5:52pm ET [Stephen]: 12.32

- Looks like we've hit the zero-message hang. It's been almost 20 minutes without updates. We're at 33658, considerably further than we got before.
- Ran `pdsh -w 'node-[1,3-87,89,91-112,114-128]' -R ssh  nvidia-smi | vim -`
    - Observed that pg0-71 gave an exit code 15 and pg0-118 gave an exit code 14.
        - From https://developer.download.nvidia.com/compute/DCGM/docs/nvidia-smi-367.38.pdf
        - Return code 14 - infoROM is corrupted
        - Return code 15 - The GPU has fallen off the bus or has otherwise become inaccessible
    - All other hosts returned the expected 100% utilization from being blocked. We knew about pg0-118 and infoROM. Investigating pg0-71
- Lspci shows the right number of devices
- Investigating pg0-71
    - [putting off running nvidia-smi in case it hangs]
    - Htop: Shows 24 threads pegged. (By comparison, pg0-70 shows 24 threads pegged)
    - straced 2 child threads pegged and found them in sched_yield. Straced a parent process was pegged in futex_wait.
    - Gdb backtrace of a parent thread actually showed a rich backtrace (compared to the simple "stuck in cudart.so" I saw previously)
        - First non-OS Bt call I see:
        - 0x00007f2efbed1d6b in torch::autograd::ReadyQueue::pop() () from /shared/home/roller/miniconda3/envs/fairseq-20210913/lib/python3.8/site-packages/torch/lib/libtorch_cpu.so
        - Pg0-70 shows itself stuck in the same place
    - Rerunning pdsh nvidia-smi repeats pg0-71 as the problem node
    - Finally running nvidia-smi on pg0-71 gives "Unable to determine the device handle for GPU 000E:00:00.0: GPU is lost. Reboot the system to recover this GPU"
- Mitigations taken:
    - Launching 12.32, replacing pg0-71 with pg0-2
        - Thought about doing a scontrol hotswap, but because i have the --restore-file in my arguments I decided against it, falling back to cloud checkpoints this time.
    - [in progress] Rebooting and then maybe replacing pg0-71

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.32

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1-70,72-87,89,91-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-28.$RUN_ID \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-28 12:28pm ET [Stephen]: 12.31

See previous entry (below run block) for context.

- 12:42pm ET - looks like we're tokenizing. ETA based on 12.30 is 1:17pm.
- 1:15pm ET - Pg0-90 doesn't look like ssh is recovering from a reboot. Successfully replaced and marked as approved for usage
- 1:21pm ET - job is making updates
- 1:38pm ET - at 50 updates!!! I'm going for lunch.
- 14:52 ET - got past 250 updates. New checkpoint saved. Life is good.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

CKPT_DIR=/data/users/roller/175B_run12.27_restore/checkpoint_11_33416/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf
32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl
1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992
RESTORE_FILE=$CKPT_DIR/checkpoint_11_33000.pt

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.31

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-87,89,91-112,114-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-28.$RUN_ID \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    --restore-file $RESTORE_FILE \
    -p $RUN_ID
```

2021-11-28 10:09am ET [Stephen]: 12.30

- 12.29 failed with the same `filename storages not found`
    - Since the exception said pg0-55, i ssh'd into it and tried manually loading its checkpoints. All 6 parts got the same storages exception!
    - I could replicate this with the storages I had manually downloaded
    - Conclusion: 33250 checkpoints are corrupt. Maybe from R12.26 and R12.25 aggressively overwriting the checkpoints.
- Remedies taken (status: monitoring for success)
    - Add a quick patch to avoid the constant-rewrite bug witnessed. Testing in prod.
        - Note I now have local backups of 33000 and 33416.x
    - Since I don't like that 33250 is corrupted (I would rather us not have any corrupt checkpoints), I am rolling back to 33000.
    - This is frustrating since we have 33416 which is the epoch boundary, but let's just bite the bullet

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

CKPT_DIR=/data/users/roller/175B_run12.27_restore/checkpoint_11_33416/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992
RESTORE_FILE=$CKPT_DIR/checkpoint_11_33000.pt

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.30

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-87,89-112,114-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-28.$RUN_ID \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    --restore-file $RESTORE_FILE \
    -p $RUN_ID
```

Launched at 11:00am ET

- 11:23am - nodes just finished loading checkpoints
- 11:57am - still tokenizing, I think. We need to add a log line for after a checkpoint was successfully loaded and when the iterator is successfully fast forwarded
- 12:03am - looks like gpus are finally burning electricity, and we have our first step
    - Looks like the hot patch for checkpointing didn't get triggered here: loss scale didn't need to be lowered on the first step
    - WPS looks healthy for now. Monitoring for hangs
- 12:08pm Started replacing pg0-2
- 12:18pm Looks like we've hung again. No error message. Made it 26 updates.

While it was still running, tried pdsh nvidia-smi. Found nvidia-smi was hanging in node-90 when I eventually ctrl-c'd nvidia-smi. SSH'ing in and running nvidia-smi seemed to also hang.

Swapping for pg0-118, which has the inforam message but that was only a guess. Also attempting to reboot pg0-90

2021-11-28 9:41am ET [Stephen]: 12.29

- Wow this time we didn't even really get past init. On pg0-2: RuntimeError: CUDA error: an illegal memory access was encountered
- Note this is the same machine that gave me issues before
- Nvidia-smi shows no issue

Since 40 managed to successfully re-init overnight, swapping 2 for 40.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
# note we have the SECOND tensorboard clobbering fix, so we can requeue after this
RUN_ID=175B_run12.29

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-87,89-112,114-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-28.0 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-28 3:20am ET [Stephen]: 12.28

Last ditch resort. Doing the exact same as 27 except bumping the storage directory version.

Checkpoint_last is still downloading. See 12.27 entry for more info

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
# note we have the SECOND tensorboard clobbering fix, so we can requeue after this
RUN_ID=175B_run12.28

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1-39,41-87,89-112,114-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-27.3 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-28 1:50am ET [Stephen]: 12.27

Looks like 26 tried to immediately upload a checkpoint and failed its cp commands! Then it took another step, lowered its scalar, and tried uploading again! And again! The humanity! We’re already at loss scale 0.25.

After it did this 3 times, it looks like it’s hanging again. Some CPUs look pegged but using strace on the processes displays only sched_yield, so it’s probably just an idle loop.

Given this is our third hang in a row, somewhere must have something wrong. Leaving the job up to check on it.
Ssh'd into pg0-39 and ran "gdb -p <pid>" on one of the fairseq processes. Confirmed we were hanging in
`0x00007ff8d76eecb1 in ?? () from /usr/lib/x86_64-linux-gnu/libcuda.so.1`

Either we have a bug in our code causing workers to be out of sync, or there's a bad node somewhere. Initiated a cluster wide nccl test, it came back a bit slow(161 but using all nodes) but it finished.

Global run of nvidia-smi didn't find any uncorrectable errors, but pg-88 did show 506k correctable ones; and pg-23 had 170k correctable errors. Other nodes showed at most a couple thousand. Decided to launch again replacing 88 with 7.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
# note we have the SECOND tensorboard clobbering fix, so we can requeue after this
RUN_ID=175B_run12.27

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1-39,41-87,89-112,114-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-27.2 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

This one failed pretty fast with "KeyError: "filename 'storages' not found"" inside tarfile.py, suggesting that something got corrupted.

- Hypothesis: some node is hanging on uploading checkpoints?
- That doesn't make sense with Susan's hang some 8 steps in.

Observed 2 nodes (7 and 17) had checkpoint downloads that were the wrong file size.

In parallel, downloaded all our latest checkpoints (including checkpoint_last) and putting them in
`/data/users/roller/175B_run12.27_restore/checkpoint_11_33416/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992`

A good next move might be to roll back to 33000 (which is downloaded) or to checkpoint_last (33461).

2021-11-27 11:39 ET [Stephen]: Run 12.26

- Looks like things are hanging, debugging
- Stephen: fixing the TB clobber overgenerate thing
- nvidia-smi on:
    - **node-40**: WARNING: infoROM is corrupted at gpu 000B:00:00.0

        - 9:07pm PT: rebooting; 9:11pm alive again
        - **inforam message still appears!**
        - Currently replacing
    - **node-113**: WARNING: infoROM is corrupted at gpu 0002:00:00.0
        - 9:08pm PT: rebooting; 9:13pm alive again
        - **inforam message still appears!**
    - https://forums.developer.nvidia.com/t/inforom-is-corrupted-at-gpu/74277
        - "There is no publicly available utility to fix this. The card is damaged. Unless it is under warranty, there isn't anything you can do to repair it."
- Re: possible replacement nodes
    - **node-118** seems to have MIG enabled; Myle has noticed this in the past and we should add checks for this to our automation scripts. The fix is to do `sudo nvidia-smi -mig 0` and then restart.
        - 9pm PT: ran `sudo nvidia-smi -mig 0` and rebooted node-118
        - 9:07pm PT: node-118 came back without MIG
        - 9:10pm PT: ran NCCL tests on node-[3,118], but got only 144GB/s instead of 189GB/s!
        - **Conclusion: node-118 has bad IB**
            - Currently replacing
    - **node-65** seems to be in a weird state. NCCL tests on node-[12,65] showed 65 with an error: `Test CUDA failure common.cu:762 'all CUDA-capable devices are busy or unavailable'`
        - 9:06pm PT: rebooted node-65 (and also node-12 just because)
        - 9:15pm PT: NCCL tests on node-[3,65] came back good (189GB/s)
        - **Conclusion: node-65 is healthy**
    - **node-12**: rebooted as part of debug process above
        - 9:13pm PT: NCCL tests on node-[3,12] came back good (189GB/s)
        - **Conclusion: node-12 is healthy**
- Whoa, while fixing the tboard bug, ran test case on node-2 and got:
    - RuntimeError: CUDA error: misaligned address
    - O_o -- resolution: pray this is nothing

Other weird thing noticed in the logs: It seems like we immediately dumped a checkpoint before moving any iterations… 12.24 didn't do this, neither did 12.23!

Note: 12.26 seems to be doing this too, we may wish to roll back to 33000. Also note the last epoch was at 33416! We're already rolling back!

```
2021-11-28 02:35:19 | WARNING | fairseq.cloud_utils | [rank 830] done with cloud download in 788.0 Seconds
2021-11-28 03:00:18 | INFO | fairseq.trainer | begin training epoch 11
2021-11-28 03:00:18 | INFO | fairseq_cli.train | Start iterating over samples
2021-11-28 03:12:26 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 2.0
2021-11-28 03:12:26 | INFO | fairseq.checkpoint_utils | Preparing to save checkpoint for epoch 11 @ 33250 updates
2021-11-28 03:12:26 | INFO | fairseq.trainer | Saving checkpoint to
/mnt/scratch/susanz/checkpoints/2021-11-27/175B_run12.25.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.l
rnpos.0emb_scal
2021-11-28 03:12:27 | INFO | fairseq.trainer | Finished saving checkpoint to
/mnt/scratch/susanz/checkpoints/2021-11-27/175B_run12.25.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.l
rnpos.
2021-11-28 03:12:27 | INFO | fairseq.checkpoint_utils | Saved checkpoint
/mnt/scratch/susanz/checkpoints/2021-11-27/175B_run12.25.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.l
rnpos.0emb
2021-11-28 03:12:31 | INFO | fairseq_cli.train | preparing to copy
/mnt/scratch/susanz/checkpoints/2021-11-27/175B_run12.25.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.l
rnpos.0emb_sca
2021-11-28 03:12:50 | INFO | train_inner | {"epoch": 11, "gnorm__fsdp_wrapped…[truncated]
```

New Launch

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
# note we have the SECOND tensorboard clobbering fix, so we can requeue after this
RUN_ID=175B_run12.26

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1-6,8-39,41-112,114-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-27 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-27 6:10pm PT: Run 12.25 [Susan restart]

- Noticed job stuck for ~4 hours.
- Trying sudo scontrol requeue job=2194 to see if things can restart smoothly.
    - scanceling and relaunching with new id instead. We're out of tb directories.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.25

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1-6,8-11,13-64,66-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-27 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

- Stuck again - after 8 updates.

2021-11-27 10:59am PT: Run 12.24 [Myle rerunning job, but AFK rest of day]

- ...tbZ already exists. Ran out of possible suffixes.
- Relaunching with new RUN_ID

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
RUN_ID=175B_run12.24

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1-6,8-11,13-64,66-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-2 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-26 9:47am ET [Stephen managing cluster]

- Managed to get new nodes for pg0-7, pg0-85
    - Both are showing clean on the ECC uncorrectables
    - Both have been marked idle and safe to use
    - I didn’t touch pg0-88 but it also looks safe to use (maybe Susan did this one)
- Successfully updated hosts mapping and initialized the scratch directories
    - NOTE: Added these two lines to my ~/.ssh/config
    - Host node-*
    -    StrictHostKeyChecking no
- pg0-118 is looking a bit funny.
    - Some of its ECC’s logs say
        - SRAM Uncorrectable   : N/A
        - DRAM Uncorrectable   : N/A
    - Others only show 0’s, not N/A’s
    - Remedy: Draining it, rebooting it
    - After several minutes, host did not seem to recover from reboot
        - Started to re-alloc
        - And then it came right back! Omg. while it was in the process of terminating
        - Maybe I should’ve been slightly more patient :(. Maybe I needed to set it to “drain” instead of “drain*”
        - Fortunately the release and reclaim was fast
- Currently running reallocs:
    - Pg0-129 (Stuck on acquiring)
    - Pg0-118 (Current “creating vm” 10:18am)
- Now getting on a plane. Will pray for success.

Some observations:

- It seems like there might be a near 1:1 mapping between our hosts and what csp gives us. For example, if we already *have* a node and we terminate it, then restart it, then we seem to get an allocation immediately. However, nodes like 129--132 are simply never allocated (even though 118 was made available while that was queuing!)

2021-11-25 8:53am ET [Susan]: Run 12.23

Run 22.22 got stuck after IB issues cropped up and everyone came to rescue it at the exact same time:

```
mlx5: buo1u00003A: got completion with error:
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
0000000d 00000000 00000000 00000000
00000000 02005104 08001219 38dcf5d2
```

Lost **node-118** GPU (automatically put on drain after job was scanceled):

```
Unable to determine the device handle for GPU 0001:00:00.0: GPU is lost.  Reboot the system to recover this GPU
```

DRAM Uncorrectable on **node-7**:

```
GPU 0000000D:00:00.0
    Ecc Mode
        Current                         : Enabled
        Pending                         : Enabled
    ECC Errors
        Volatile
            SRAM Correctable            : 0
            SRAM Uncorrectable          : 0
            DRAM Correctable            : 0
            DRAM Uncorrectable          : 0
        Aggregate
            SRAM Correctable            : 0
            SRAM Uncorrectable          : 0
            DRAM Correctable            : 10
            DRAM Uncorrectable          : 101
```

- sudo scontrol update node=node-7 state=drain reason=dram

Current list of node states:

```
$ sinfo
    PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
    hpc*         up   infinite      2  down~ node-[129-130]
    hpc*         up   infinite      2  idle~ node-[131-132]
    hpc*         up   infinite      3  drain node-[7,85,118]
    hpc*         up   infinite    125   idle node-[1-6,8-84,86-117,119-128]
```

High amount of DRAM correctable errors on node-88 (506807), excluding to be safe.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.23

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1-6,8-84,86-87,89-117,119-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-25 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

- New log dir: /shared/home/namangoyal/checkpoints/175B/175B_run12.23.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

2021-11-25 11:35am ET [Myle]: Run 12.22

- Nvm, realized this won't resume properly due to cache, and will clobber tensorboard; will increment run ID instead
- Add 3rd party timeout to handle blob retries
    - #2686
    - timeout-decorator is a new dependency
        - pip install timeout-decorator
- Re: spare nodes:
    - Ran ./scripts/nccl_tests/cloud/run_nccl_allreduce.sh to validate NCCL perf
    - Confirmed node-[1,7,47] are good
    - Drained node-85, which had bad NCCL perf (only 140GB/s instead of 180)

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.22

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-42,44-57,59-84,86-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/$USER/checkpoints/2021-11-25 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-25 11:20am ET: Run 12.21 (requeue)

- Training got stuck in loss scale 0.25 loop, after step 28628
- Did scontrol requeue job=2167 to see if it magically fixes after restarting from checkpoint

2021-11-24 11:18pm ET [Susan]: Run 12.21

- Testing out #2681
- Remove node 58, replacing with 7.
    - Node 58 came up with GPUs! But, there's this:

      DRAM Correctable : 18446744073709551615

    - sudo scontrol update node=node-58 state=drain reason=dramtoast

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.21

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-42,44-57,59-84,86-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/susanz/checkpoints/2021-11-24.3 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-24 10:40pm ET [Susan]: Run 12.20

- Testing out #2681

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
RUN_ID=175B_run12.20

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-6,8-42,44-84,86-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/susanz/checkpoints/2021-11-24.3 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

- Lost GPU on node-58:

  ```
  (base) susanz@buo1u00001P:~$ nvidia-smi
  Unable to determine the device handle for GPU 0001:00:00.0: GPU is lost.  Reboot the system to recover this GPU
  ```

- sudo scontrol update node=node-58 state=drain reason=lostgpu

2021-11-24 3:30pm ET [Susan]: Run 12.19

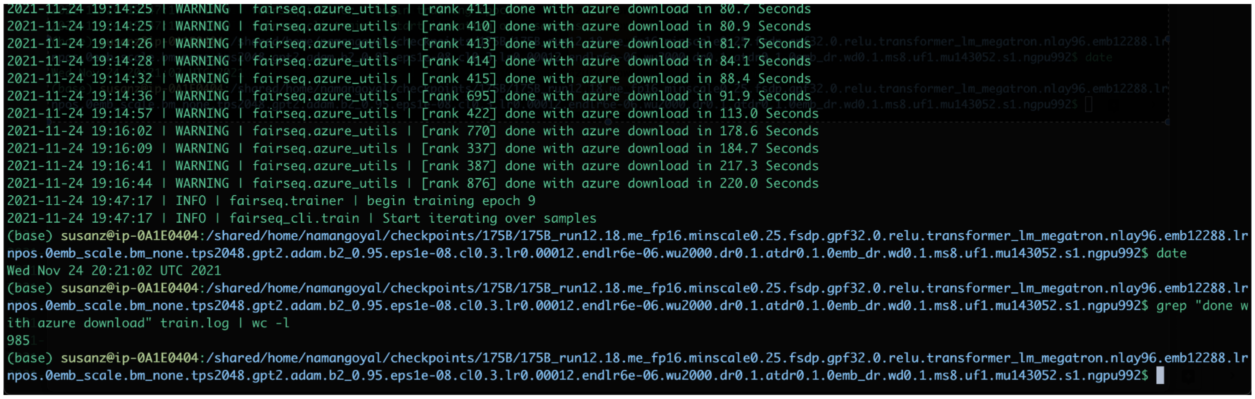
- Launching with node 85. Bringing in 47 instead.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
RUN_ID=175B_run12.19

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-6,8-42,44-84,86-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/susanz/checkpoints/2021-11-24.2 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

- Hung for 5 hours - missing 4 shards.

2021-11-24 2:10pm ET [Susan]: Run 12.18

- Relaunched exactly the same as 12.17 to see if the hanging was just data loading fast forwarding.
- It's actually something else?



```
2021-11-24 19:14:25 | WARNING | fairseq.azure_utils | [rank 411] done with azure download in 80.7 Seconds
2021-11-24 19:14:25 | WARNING | fairseq.azure_utils | [rank 410] done with azure download in 80.9 Seconds
2021-11-24 19:14:26 | WARNING | fairseq.azure_utils | [rank 413] done with azure download in 81.7 Seconds
2021-11-24 19:14:28 | WARNING | fairseq.azure_utils | [rank 414] done with azure download in 84.1 Seconds
2021-11-24 19:14:32 | WARNING | fairseq.azure_utils | [rank 415] done with azure download in 88.4 Seconds
2021-11-24 19:14:36 | WARNING | fairseq.azure_utils | [rank 695] done with azure download in 91.9 Seconds
2021-11-24 19:14:57 | WARNING | fairseq.azure_utils | [rank 422] done with azure download in 113.0 Seconds
2021-11-24 19:16:02 | WARNING | fairseq.azure_utils | [rank 770] done with azure download in 178.6 Seconds
2021-11-24 19:16:09 | WARNING | fairseq.azure_utils | [rank 337] done with azure download in 184.7 Seconds
2021-11-24 19:16:41 | WARNING | fairseq.azure_utils | [rank 387] done with azure download in 217.3 Seconds
2021-11-24 19:16:44 | WARNING | fairseq.azure_utils | [rank 876] done with azure download in 220.0 Seconds
2021-11-24 19:47:17 | INFO | fairseq.trainer | begin training epoch 9
2021-11-24 19:47:17 | INFO | fairseq_cli.train | Start iterating over samples
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.18.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lr
npos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ date
Wed Nov 24 20:21:02 UTC 2021
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.18.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lr
npos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ grep "done w
ith azure download" train.log | wc -l
985
(base) susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.18.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lr
npos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ 
```

  - Something is taking half an hour before "begin training epoch 9"
  - We don't seem to print all ranks for cloud download.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.18

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-6,8-42,44-46,48-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/susanz/checkpoints/2021-11-24.1 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-24 1:00pm ET [Susan]: Run 12.17

- Run 12.16 got stuck at step 27247 after hitting ECC error.
- `WCOLL=~myleott/hosts PDSH_RCMD_TYPE=ssh pdsh nvidia-smi -q -d "ECC" > ecc_error.log`
- Shows 2 and 43 with ECC errors. Restarting in cloud UI.
  - Taking a long time to provision for some reason.
  - 2 failed to provision with:
    "*Error: ProvisioningState/failed/OSProvisioningTimedOut OS Provisioning for VM nv6ormrlczgkx_245' did not finish in the allotted time. The VM may still finish provisioning successfully. Please check provisioning state later.*"

    Shutting down and restarting got:
    "*Reimaging virtual machine due to error on creation*"
- Also restarting 85 since VM failed to start for some reason

- 85 came back. Keep 7,47 removed in case they’re the bad IB nodes. Taking a gamble on 1.
- 43 came back as well.

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
RUN_ID=175B_run12.17

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[1,3-6,8-42,44-46,48-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/susanz/checkpoints/2021-11-24.1 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

- Notes:
  - Update local-checkpoints-dir! Permissions issue if using any other username there but your own.

  - Confirmed right checkpoint loaded (should see this by ~6 minutes in):

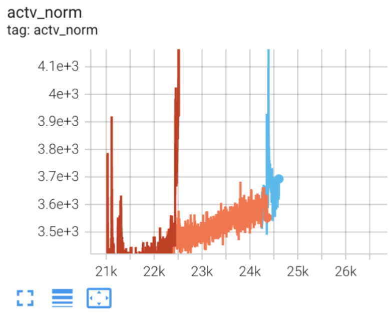
    2021-11-24 18:19:14 | INFO | fairseq.trainer | Loaded checkpoint /mnt/scratch/susanz/checkpoints/2021-11-24.1/175B_run12.17.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-0-shard0.pt (epoch 9 @ 27000 updates)

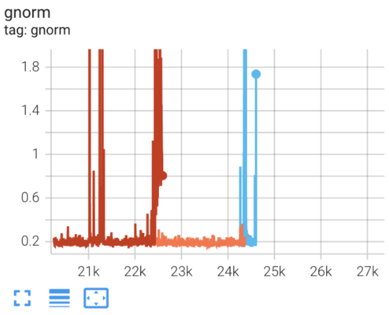
  - Got stuck at 988 log lines of cloud downloads, even though 992 shards exist on machines.

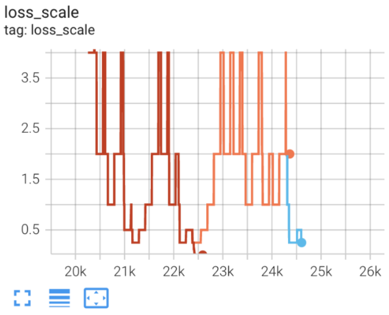
  -
    ```
    WCOLL=~myleott/hosts PDSH_RCMD_TYPE=ssh pdsh ls -la /mnt/scratch/susanz/checkpoints/2021-11-24.1/175B_run12.17.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992 > copied_files.log
    ```

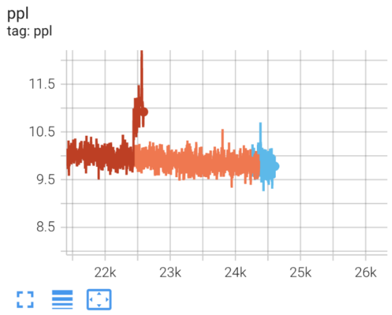
  -
    ```
    (fairseq-20210913) susanz@ip-0A1E0404:~/fairseq-py$ grep shard copied_files.log | wc -l
    992
    ```

2021-11-23 10:50am [Myle]: Run 12.16









- Decided to reduce clipping to 0.3 and relaunch from 24.5K checkpoint
  - If this fails, we will try resetting adam stats and do a fresh warmup.
- Uncovered a hang when restoring from Cloud blob; fix in PR #2673
- New train log: /shared/home/namangoyal/checkpoints/175B/175B_run12.16.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl0.3.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don’t clobber tensorboard
RUN_ID=175B_run12.16

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[2-6,8-46,48-84,86-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/myleott/checkpoints/2021-11-23.9 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID
```

2021-11-22 7:45am [Myle]: Run 12.15

- 12.14 failed with CUDA error:

    - THCudaCheck FAIL file=/pytorch/aten/src/THC/THCCachingHostAllocator.cpp line=278 error=999 : unknown error
    - RuntimeError: Caught RuntimeError in pin memory thread for device 4.
- To find bad node:
    - Run nvidia-smi on all hosts and save output in nvidia_smi_logs directory
        - `mkdir nvidia_smi_logs`
        - `scontrol show hostnames node-[2-6,8-46,48-94,96-128] | PDSH_RCMD_TYPE=ssh pdsh -w - nvidia-smi -f nvidia_smi_logs/%h`
    - **Conclusion:** node-85: Unable to determine the device handle for GPU 000B:00:00.0: GPU is lost. Reboot the system to recover this GPU
- Reboot the bad node
    - ssh node-85
    - sudo reboot
- Bad node never came back!
    - Try manually restarting from UI
- Current status:
    - we technically have quota up to 132 nodes
    - we currently have 125 nodes running, but only 123 are good
    - two nodes have IB issues that make them slow when part of any distributed jobs
    - we are not able to grow beyond 125 nodes – this makes me think the cluster does not have any spare nodes at this point

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.15

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[2-6,8-46,48-72,74-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/myleott/checkpoints/2021-11-22.3 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID

After launch:
sudo scontrol update job=2028 TimeLimit=UNLIMITED
sudo scontrol update job=2028 MailUser=<scrubbed> MailType=ALL
./scripts/poll_file.py /shared/home/namangoyal/checkpoints/175B/175B_run12.15*/train.log --mailto <scrubbed>
```

2021-11-21 4:15pm [Myle]: Run 12.14

- Run 12.13
    - Successfully tested the automatic-resume-from-latest-blob-checkpoint functionality

    - But loss exploded even faster than 12.12, since the loss scale state is not properly reloaded from checkpoint, causing the loss scale to stay low since there isn't enough history built up to increase it
- **Next step:** roll back to a slightly older checkpoint (22,500 => 22,250) and set --threshold-loss-scale=0.25
- New train log: /shared/home/namangoyal/checkpoints/175B/175B_run12.14.me_fp16.minscale0.25.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

CKPT_DIR=/data/users/myleott/175B_run12.12.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_8_22000*.pt" "$BLOB_URL" checkpoint_8_22000
cp --recursive --include-pattern "checkpoint_8_22250*.pt" "$BLOB_URL" checkpoint_8_22250

RESTORE_FILE=$CKPT_DIR/checkpoint_8_22250/checkpoint_8_22250.pt
RUN_ID=175B_run12.14

INCLUDED_HOSTS=node-[2-6,8-46,48-94,96-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=2021 TimeLimit=UNLIMITED
sudo scontrol update job=2021 MailUser=<scrubbed> MailType=ALL
```

2021-11-21 3pm [Myle]: Run 12.13

- Manually killed and relaunched with blob requeueing fixes from #2666
- New train log: /shared/home/namangoyal/checkpoints/175B/175B_run12.13.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# we use a single Cloud blob path for all checkpoints now
PERMANENT_CLOUD_UPLOAD_PATH="<<<SCRUBBED FOR RELEASE>>>"

# increment the run ID so we don't clobber tensorboard
RUN_ID=175B_run12.13

# relaunch training and restore checkpoint from Cloud blob storage
INCLUDED_HOSTS=node-[2-6,8-46,48-94,96-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --local-checkpoints-dir /mnt/scratch/myleott/checkpoints/2021-11-21 \
    --full-cloud-upload-path $PERMANENT_CLOUD_UPLOAD_PATH \
    -p $RUN_ID

After launch:
sudo scontrol update job=2019 TimeLimit=UNLIMITED
sudo scontrol update job=2019 MailUser=<scrubbed> MailType=ALL
```

2021-11-21 Analysis of 12.X series

Eta of completion:

- At observed rate (including downtime, total average WPS, etc)
    - Goal: 144k updates; Currently at: 22k
    - Started Nov 11 22:00; Currently Nov 21, 13:00; duration 231 hours.
        - = 38 seconds per update with downtime
        - = 53 more days at total average pace
        - = Jan 3
- At optimal rate (including WPS improvements from Naman, no downtime):
    - 122k updates to go
    - 19.3 seconds per update with no downtime
    - = 27.2523148 days
    - = Dec 18

Related idle thoughts [Stephen] about a potential v2:

- Tuning WD or LR might be most beneficial
- I still think fresh BPE might be nice
- I'd really like to add in multilingual/non-English data as the next big chunk of data. (Would definitely necessitate a new BPE)

2021-11-20 9:30pm [Myle]: Run 12.12

- Had previously tried requeuing job, but now realized it loaded the previous checkpoint at 17750 (likely due to the presence of --restore-file)
- Wasn't able to get Sam's blob reload logic to work, reverted to manual download
- New log: /shared/home/namangoyal/checkpoints/175B/175B_run12.12.me_fp16.fsdp.gpf32.0.relu.transformer_l

m_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

CKPT_DIR=/data/users/myleott/175B_run12.11.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_7_20250*.pt" "$BLOB_URL" checkpoint_7_20250

RESTORE_FILE=$CKPT_DIR/checkpoint_7_20250/checkpoint_7_20250.pt
RUN_ID=175B_run12.12

INCLUDED_HOSTS=node-[2-6,8-46,48-94,96-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=2001 TimeLimit=UNLIMITED
sudo scontrol update job=2001 MailUser=<scrubbed> MailType=ALL
```

2021-11-19 9:30am [Myle]: Run 12.11

- Sam Shleifer traced down IB issues with node-[7,43,90,95].
- Susan Zhang found ECC errors on node-1
- **Updated fairscale with Naman's FSDP speedup: fairscale@8820049331331c773077c257667aa81baf4cc9f9**

```
CKPT_DIR=/data/users/myleott/175B_run12.10.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_6_17750*.pt" "$BLOB_URL" checkpoint_6_17750

export RESTORE_FILE=$CKPT_DIR/checkpoint_6_17750/checkpoint_6_17750.pt

export RUN_ID=175B_run12.11

# use updated fairscale
cd ~/src/fairscale
git fetch origin prefetch_fsdp_params_simple
git checkout prefetch_fsdp_params_simple

# initial test run to validate nodes
```

```
INCLUDED_HOSTS=node-[2-6,8-89,91-94,96-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p test4_${RUN_ID} \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/

# resume from previous checkpoint
INCLUDED_HOSTS=node-[2-6,8-89,91-94,96-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1984 TimeLimit=UNLIMITED
sudo scontrol update job=1984 MailUser=<scrubbed> MailType=ALL
```

2021-11-18 List of issues for Cloud Complaints

Unable to train continuously for more than 1-2 days on a cluster of 128 nodes. Many failures require manual detection and remediation, wasting compute resources and researcher time:

- GPU reliability issues (e.g., ECC errors) leading to frequent job restarts and manual reprovisioning of problematic nodes/GPUs
- IB issues lead to degraded training speed (-20% throughput), requiring manual bisection of problematic nodes and manual reprovisioning
- Unexpected job hangs, likely due to IB/NCCL issues, requiring manual detection

Concrete failures seen:

- GPUs randomly disconnecting
    - “Unable to determine the device handle for GPU 000B:00:00.0: GPU is lost. Reboot the system to recover this GPU”
- Nodes with bad IB
    - “p2p_plugin.c:141 NCCL WARN NET/IB : Got async event : port error”
- Frequently see GPUs with high rates of uncorrectable ECC errors
    - Occurs every 1-2 days on a cluster of 128 nodes
    - Reprovisioning nodes often returns the same node with ECC errors

Asks:

- On CSP side:
    - GPUs should be validated to be sufficiently burned-in and ECC checked.
    - CSP to check for IB issues before allocating to us.
    - Need mechanism to retire problematic nodes so that they are not reassigned to us before they have been fixed/validated.
- On our side:
    - Automated health checks to be run at the start and end of each job.
    - Automatic detection and requeuing of jobs that hang due to IB/NCCL issues.

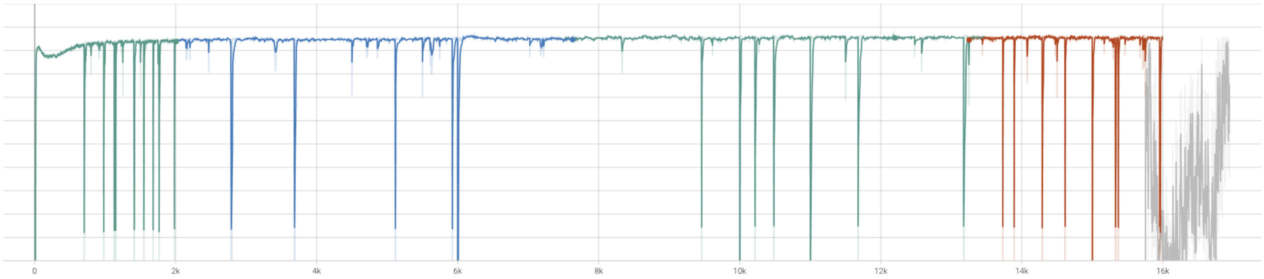
2021-11-18 5:30pm [Stephen]: Notes from 12.10

- Observed slow down is persistent

- Observe that the model will reach the next epoch boundary (and minimize wasted tokenization) in about 12-15 hours.
- Decision:
  - We will let it continue to run at lower WPS overnight
  - In the morning we will kill it, bisect and find the node with bad IB
  - We will simultaneously launch with Naman's improvements to WPS.
  - Once we identify the bad node, we need to document these lists of things to complain to CSP and get our money back
- Replaced node-[1,39] in cloud UI
  - node-1 came back up with ECC errors

2021-11-18 3pm [Stephen]: Run 12.10

Relaunching for observed slowness.



```
CKPT_DIR=/data/users/roller/175B_run12.09.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

CHKPT=checkpoint_6_16750

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

mkdir $CKPT_DIR
cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_6_16750*.pt" "$BLOB_URL" checkpoint_6_16750

export RESTORE_FILE=$CKPT_DIR/checkpoint_6_16750/checkpoint_6_16750.pt

export RUN_ID=175B_run12.10

cd ~/working/fairseq
INCLUDED_HOSTS=node-[2-38,40-89,91-119,121-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

# After launch:
sudo scontrol update job=1477 TimeLimit=UNLIMITED
sudo scontrol update job=1477 MailUser=<scrubbed> MailType=ALL

# update tensorboard
```

- Observed exceptions during workflow:

    - Had to change my SAS url to a new value (was given the new value in the exception)
    - Had to `pip install cloud-storage-blob`
    - Some nodes were left stuck in a drain state (1, 39)
    - Ssh'd into 43 after receiving guidance it was probably okay, and ran `nvidia-smi -q -d "ECC"` to check for ECC errors. All reported 0 so switched 1 to that node.
    - Needed to cd back to my fairseq directory before launch
    - Had to relaunch due to a typo in the checkpoint filenames (updated instructions)
- Launched at 16:04 ET
- New Log file:
    - /shared/home/namangoyal/checkpoints/175B/175B_run12.10.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log
- Potential problematic machines (nov 18):
    - Diagnosis: ssh'ing in and running nvidia-smi reports "Unable to determine the device handle for GPU 000B:00:00.0: GPU is lost. Reboot the system to recover this GPU"
    - node-1
    - node-39
    - Suggested remedy is both of them need to be rebooted manually by ssh + sudo reboot
- Potential Problematic Machines [updated Nov 17]:
    - **node-90** (N/A uncorrectable errors instead of 0)
        - SRAM Correctable   : N/A
        - SRAM Uncorrectable   : N/A
        - DRAM Correctable   : N/A
        - DRAM Uncorrectable   : N/A

2021-11-17 11pm [Myle]: Run 12.09

- Previous run failed with mysterious error: "RuntimeError: CUDA error: unknown error"
    - Happened as validation was starting on run 12.08:
      2021-11-18 02:24:01 | INFO | fairseq_cli.train | Begin looping over validation "valid/Gutenberg_PG-19" subset with length "0"
- New log file:
  /shared/home/namangoyal/checkpoints/175B/175B_run12.09.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
CKPT_DIR=/data/users/myleott/175B_run12.08.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_6_15750*.pt" "$BLOB_URL" checkpoint_6_15750

export RESTORE_FILE=$CKPT_DIR/checkpoint_6_15750/checkpoint_6_15750.pt

export RUN_ID=175B_run12.09

INCLUDED_HOSTS=node-[1-38,40-42,44-89,91-119,121-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1403 TimeLimit=UNLIMITED
sudo scontrol update job=1403 MailUser=<scrubbed> MailType=ALL
```

2021-11-16 11pm [Myle]: Run 12.08

- Previous run failed with mysterious error: “p2p_plugin.c:141 NCCL WARN NET/IB : Got async event : port error”
- New log file: /shared/home/namangoyal/checkpoints/175B/175B_run12.08.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
CKPT_DIR=/data/users/myleott/175B_run12.07.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_5_13250*.pt" "$BLOB_URL" checkpoint_5_13250

export RESTORE_FILE=$CKPT_DIR/checkpoint_5_13250/checkpoint_5_13250.pt

export RUN_ID=175B_run12.08

INCLUDED_HOSTS=node-[1-38,40-89,91-94,96-119,121-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1394 TimeLimit=UNLIMITED
sudo scontrol update job=1394 MailUser=<scrubbed> MailType=ALL
```

2021-11-14 9:45pm [Myle]: Run 12.07

- Previous run silently hung
- New log file: /shared/home/namangoyal/checkpoints/175B/175B_run12.07.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
CKPT_DIR=/data/users/myleott/175B_run12.06.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cd $CKPT_DIR
cp --recursive --include-pattern "checkpoint_3_7500*.pt" "$BLOB_URL/*" checkpoint_3_7500

export RESTORE_FILE=$CKPT_DIR/checkpoint_3_7500/checkpoint_3_7500.pt

export RUN_ID=175B_run12.07

INCLUDED_HOSTS=node-[1-38,40-89,91-94,96-119,121-128] \
    python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1390 TimeLimit=UNLIMITED
sudo scontrol update job=1390 MailUser=<scrubbed> MailType=ALL
```

2021-11-12 10:30pm [Myle]: Run 12.06

- Previous restore failed because checkpoint_1_2000 is missing shards in Cloud blob!
    - Run 12.02 must have been interrupted before all the checkpoints were uploaded.
    - Fortunately we can get the shards from local storage on the nodes

```
# copy missing checkpoints from local storage on each node
MNT_DIR=/mnt/scratch/susanz/checkpoints/2021-11-12/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

WCOLL=~/hosts PDSH_RCMD_TYPE=ssh pdsh ls $MNT_DIR | grep "checkpoint_1_2000.*pt$"
(...)

cd $RESTORE_DIR
scp node-30:$MNT_DIR/checkpoint_1_2000-model_part-5-shard29.pt .
scp node-19:$MNT_DIR/checkpoint_1_2000-model_part-4-shard18.pt .
scp node-37:$MNT_DIR/checkpoint_1_2000-model_part-0-shard36.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-0-shard54.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-1-shard54.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-2-shard54.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-3-shard54.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-4-shard54.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-5-shard54.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-6-shard54.pt .
scp node-56:$MNT_DIR/checkpoint_1_2000-model_part-7-shard54.pt .
scp node-68:$MNT_DIR/checkpoint_1_2000-model_part-0-shard66.pt .
scp node-103:$MNT_DIR/checkpoint_1_2000-model_part-0-shard99.pt .

export RESTORE_FILE=/data/users/susanz/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2000/checkpoint_1_2000.pt

export RUN_ID=175B_run12.06
```

```
INCLUDED_HOSTS=node-[1-38,41-94,96-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1387 TimeLimit=UNLIMITED
sudo scontrol update job=1387 MailUser=<scrubbed> MailType=ALL
```

- Misc
    - Re-enable tensor init on GPU
    - Also run `sudo reboot` on node-[39,40]

```
git checkout 08cb44d9dc3dcbe90605dd03b4f2156996ea2bac

export
RESTORE_FILE=/data/users/susanz/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.l
rnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_
dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2000/checkpoint_1_2000.pt

export RUN_ID=175B_run12.06

INCLUDED_HOSTS=node-[1-38,41-94,96-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1386 TimeLimit=UNLIMITED
sudo scontrol update job=1386 MailUser=<scrubbed> MailType=ALL
```

2021-11-12 7pm [Susan]: Run 12.05

- Removed tensor init on gpu.

```
git checkout ddbb690ed49d49653a1c12374386de1a2102d3a2

export
RESTORE_FILE=/data/users/susanz/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.l
rnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_
dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2000/checkpoint_1_2000.pt

export RUN_ID=175B_run12.05

INCLUDED_HOSTS=node-[1-38,40-94,96,98-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1385 TimeLimit=UNLIMITED
sudo scontrol update job=1385 MailUser=<scrubbed> MailType=ALL
```

2021-11-12 6pm [Susan]: Run 12.04

- Host 95 and 120 are both full of ECC errors. Put both in drain. Replacing 95 seems to give the same machine, so will wait to restart later.

```
git checkout 38dab5485ef7d5c1e29187185680b6e4f314e7b9

export
RESTORE_FILE=/data/users/susanz/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2000/checkpoint_1_2000.pt

export RUN_ID=175B_run12.04

INCLUDED_HOSTS=node-[1-38,40-94,96,98-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1384 TimeLimit=UNLIMITED
sudo scontrol update job=1384 MailUser=<scrubbed> MailType=ALL
```

2021-11-12 3pm [Susan]: Run 12.03

- Restarting requires following the same steps as 11.2 to download checkpoints (takes ~25 minutes).
- Downloading into /data/users/susanz/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992
- Loading checkpoint when run comes up takes ~15 minutes.
- Loading data when run comes up takes ~30 minutes (to fast forward to data point within epoch).

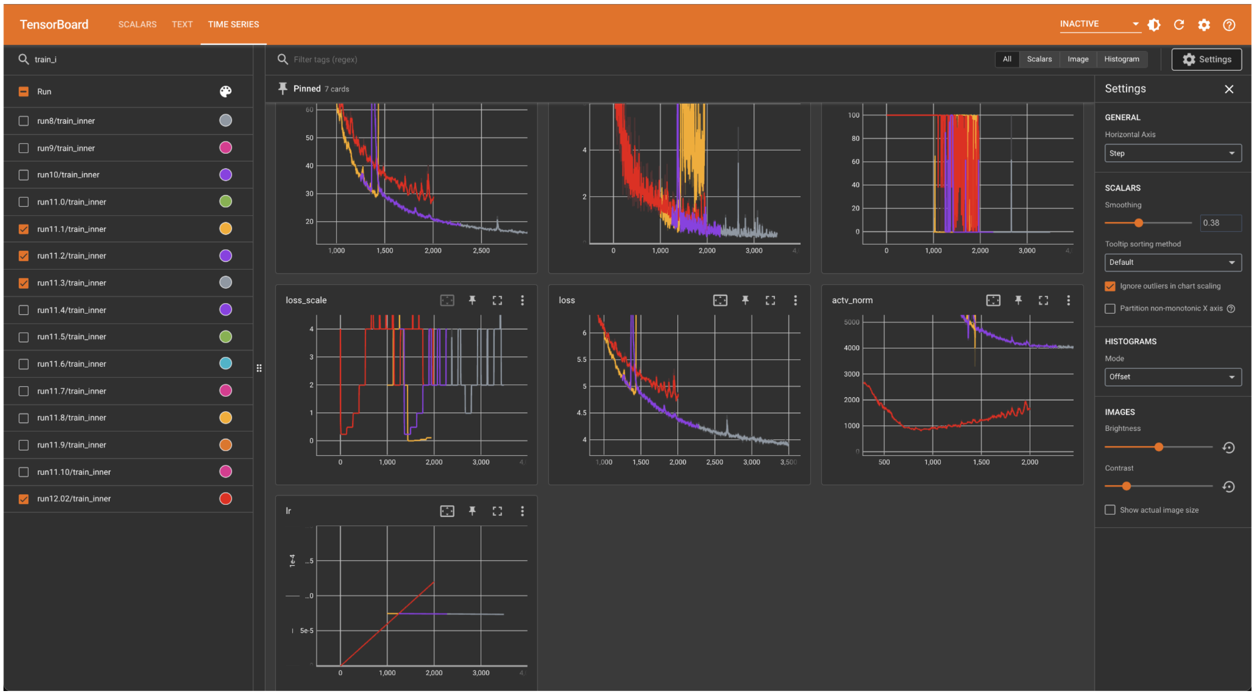
```
# Find cloud path of where the checkpoints went - grab last one

(fairseq-20210913)
susanz@ip-0A1E0404:/shared/home/namangoyal/checkpoints/175B/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992$ grep "<<< SCRUBBED FOR RELEASE >>>" train.log | tail -n 1

2021-11-12 19:34:30 | INFO | fairseq_cli.train | preparing to copy
/mnt/scratch/susanz/checkpoints/2021-11-12/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2000-model_part-0-shard0.pt to <<<SCRUBBED FOR RELEASE>>>


export
RUN_ID="175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992"
```

```
export CHECKPOINT="checkpoint_1_2000"

mkdir /data/users/susanz/$RUN_ID
cd /data/users/susanz/$RUN_ID

mkdir $CHECKPOINT

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cp --recursive --include-pattern "$CHECKPOINT-*.pt" "$BLOB_URL/*" $CHECKPOINT/
```

Launch commands:

```
git checkout 38dab5485ef7d5c1e29187185680b6e4f314e7b9

export
RESTORE_FILE=/data/users/susanz/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.l
rnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_
dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2000/checkpoint_1_2000.pt

export RUN_ID=175B_run12.03

# DO DRY RUN!! Node configuration may have been changed !!!
INCLUDED_HOSTS=node-[1-38,40-87,89-96,98-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE \
    --dry-run

INCLUDED_HOSTS=node-[1-38,40-87,89-94,96,98-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE

After launch:
sudo scontrol update job=1382 TimeLimit=UNLIMITED
sudo scontrol update job=1382 MailUser=<scrubbed> MailType=ALL
```

Analysis of Run 12.02 [Susan]



Settings

GENERAL

Horizontal Axis: Step

SCALARS

Smoothing: 0.38

Tooltip sorting method: Default

☑ Ignore outliers in chart scaling

□ Partition non-monotonic X axis

HISTOGRAMS

Mode: Offset

IMAGES

Brightness

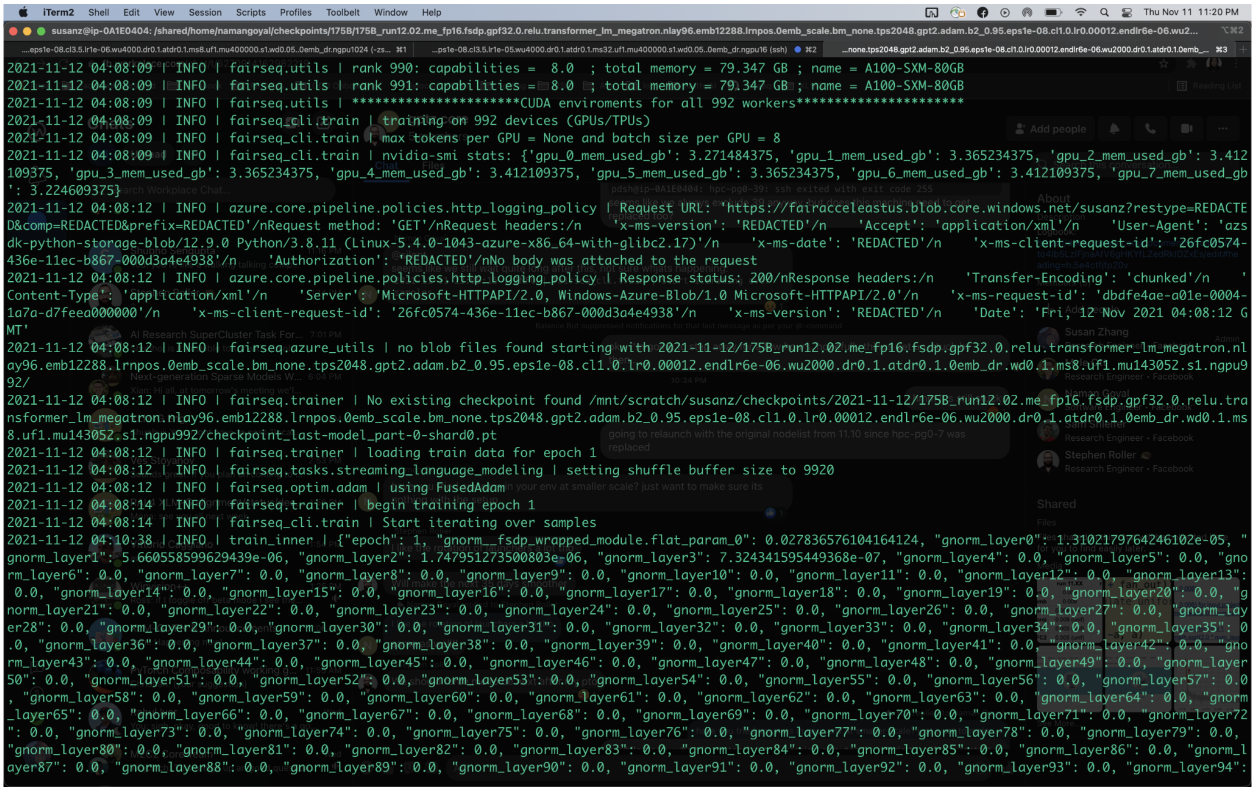
Contrast

□ Show actual image size

- Loss of ppl starting to oscillate more heavily, potentially indicating that LR is too high.
- CUDA error crashed the run after 2008 updates.

2021-11-11 11pm [Susan]: Run 12.02

- Relaunched with node 7 put back, same nodelist as 11.10.
- This worked! Expect roughly 2 minutes between “Start iterating over samples” and the first log line:


iTerm2
Shell
Edit
View
Session
Scripts
Profiles
Toolbelt
Window
Help
Thu Nov 11 11:20 PM
susanz@ip-0A1E0404: /shared/home/namangoyal/checkpoints/175B/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2...
...eps1e-08.cl3.5.lr1e-06.wu4000.dr0.1.atdr0.1.ms8.uf1.mu400000.s1.wd0.05..0emb_dr.ngpu1024 (-zs... ⌘1
...ps1e-08.cl3.5.lr1e-05.wu4000.dr0.1.atdr0.1.ms32.uf1.mu400000.s1.wd0.05..0emb_dr.ngpu16 (ssh) ⌘2
...none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_... ⌘3

```
2021-11-12 04:08:09 | INFO | fairseq.utils | rank 990: capabilities =  8.0  ; total memory = 79.347 GB ; name = A100-SXM-80GB
2021-11-12 04:08:09 | INFO | fairseq.utils | rank 991: capabilities =  8.0  ; total memory = 79.347 GB ; name = A100-SXM-80GB
2021-11-12 04:08:09 | INFO | fairseq.utils | ***********************CUDA environments for all 992 workers***********************
2021-11-12 04:08:09 | INFO | fairseq_cli.train | training on 992 devices (GPUs/TPUs)
2021-11-12 04:08:09 | INFO | fairseq_cli.train | max tokens per GPU = None and batch size per GPU = 8
2021-11-12 04:08:09 | INFO | fairseq_cli.train | nvidia-smi stats: {'gpu_0_mem_used_gb': 3.271484375, 'gpu_1_mem_used_gb': 3.365234375, 'gpu_2_mem_used_gb': 3.412
109375, 'gpu_3_mem_used_gb': 3.365234375, 'gpu_4_mem_used_gb': 3.412109375, 'gpu_5_mem_used_gb': 3.365234375, 'gpu_6_mem_used_gb': 3.412109375, 'gpu_7_mem_used_gb
': 3.224609375}
2021-11-12 04:08:12 | INFO | azure.core.pipeline.policies.http_logging_policy | Request URL: 'https://fairacceleastus.blob.core.windows.net/susanz?restype=REDACTE
D&comp=REDACTED&prefix=REDACTED'/nRequest method: 'GET'/nRequest headers:/n    'x-ms-version': 'REDACTED'/n    'Accept': 'application/xml'/n    'User-Agent': 'azs
dk-python-storage-blob/12.9.0 Python/3.8.11 (Linux-5.4.0-1043-azure-x86_64-with-glibc2.17)'/n    'x-ms-date': 'REDACTED'/n    'x-ms-client-request-id': '26fc0574-
436e-11ec-b867-000d3a4e4938'/n    'Authorization': 'REDACTED'/nNo body was attached to the request
2021-11-12 04:08:12 | INFO | azure.core.pipeline.policies.http_logging_policy | Response status: 200/nResponse headers:/n    'Transfer-Encoding': 'chunked'/n    '
Content-Type': 'application/xml'/n    'Server': 'Microsoft-HTTPAPI/2.0, Windows-Azure-Blob/1.0 Microsoft-HTTPAPI/2.0'/n    'x-ms-request-id': 'dbdfe4ae-a01e-0004-
1a7a-d7feea000000'/n    'x-ms-client-request-id': '26fc0574-436e-11ec-b867-000d3a4e4938'/n    'x-ms-version': 'REDACTED'/n    'Date': 'Fri, 12 Nov 2021 04:08:12 G
MT'
2021-11-12 04:08:12 | INFO | fairseq.azure_utils | no blob files found starting with 2021-11-12/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nl
ay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu9
92/
2021-11-12 04:08:12 | INFO | fairseq.trainer | No existing checkpoint found /mnt/scratch/susanz/checkpoints/2021-11-12/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.tra
nsformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms
8.uf1.mu143052.s1.ngpu992/checkpoint_last-model_part-0-shard0.pt
2021-11-12 04:08:12 | INFO | fairseq.trainer | loading train data for epoch 1
2021-11-12 04:08:12 | INFO | fairseq.tasks.streaming_language_modeling | setting shuffle buffer size to 9920
2021-11-12 04:08:12 | INFO | fairseq.optim.adam | using FusedAdam
2021-11-12 04:08:14 | INFO | fairseq.trainer | begin training epoch 1
2021-11-12 04:08:14 | INFO | fairseq_cli.train | Start iterating over samples
2021-11-12 04:10:38 | INFO | train_inner | {"epoch": 1, "gnorm__fsdp_wrapped_module.flat_param_0": 0.027836576104164124, "gnorm_layer0": 1.3102179764246102e-05, "
gnorm_layer1": 5.660558599629439e-06, "gnorm_layer2": 1.747951273500803e-06, "gnorm_layer3": 7.324341595449368e-07, "gnorm_layer4": 0.0, "gnorm_layer5": 0.0, "gno
rm_layer6": 0.0, "gnorm_layer7": 0.0, "gnorm_layer8": 0.0, "gnorm_layer9": 0.0, "gnorm_layer10": 0.0, "gnorm_layer11": 0.0, "gnorm_layer12": 0.0, "gnorm_layer13":
 0.0, "gnorm_layer14": 0.0, "gnorm_layer15": 0.0, "gnorm_layer16": 0.0, "gnorm_layer17": 0.0, "gnorm_layer18": 0.0, "gnorm_layer19": 0.0, "gnorm_layer20": 0.0, "g
norm_layer21": 0.0, "gnorm_layer22": 0.0, "gnorm_layer23": 0.0, "gnorm_layer24": 0.0, "gnorm_layer25": 0.0, "gnorm_layer26": 0.0, "gnorm_layer27": 0.0, "gnorm_lay
er28": 0.0, "gnorm_layer29": 0.0, "gnorm_layer30": 0.0, "gnorm_layer31": 0.0, "gnorm_layer32": 0.0, "gnorm_layer33": 0.0, "gnorm_layer34": 0.0, "gnorm_layer35": 0
.0, "gnorm_layer36": 0.0, "gnorm_layer37": 0.0, "gnorm_layer38": 0.0, "gnorm_layer39": 0.0, "gnorm_layer40": 0.0, "gnorm_layer41": 0.0, "gnorm_layer42": 0.0, "gno
rm_layer43": 0.0, "gnorm_layer44": 0.0, "gnorm_layer45": 0.0, "gnorm_layer46": 0.0, "gnorm_layer47": 0.0, "gnorm_layer48": 0.0, "gnorm_layer49": 0.0, "gnorm_layer
50": 0.0, "gnorm_layer51": 0.0, "gnorm_layer52": 0.0, "gnorm_layer53": 0.0, "gnorm_layer54": 0.0, "gnorm_layer55": 0.0, "gnorm_layer56": 0.0, "gnorm_layer57": 0.0
, "gnorm_layer58": 0.0, "gnorm_layer59": 0.0, "gnorm_layer60": 0.0, "gnorm_layer61": 0.0, "gnorm_layer62": 0.0, "gnorm_layer63": 0.0, "gnorm_layer64": 0.0, "gnorm
_layer65": 0.0, "gnorm_layer66": 0.0, "gnorm_layer67": 0.0, "gnorm_layer68": 0.0, "gnorm_layer69": 0.0, "gnorm_layer70": 0.0, "gnorm_layer71": 0.0, "gnorm_layer72
": 0.0, "gnorm_layer73": 0.0, "gnorm_layer74": 0.0, "gnorm_layer75": 0.0, "gnorm_layer76": 0.0, "gnorm_layer77": 0.0, "gnorm_layer78": 0.0, "gnorm_layer79": 0.0,
"gnorm_layer80": 0.0, "gnorm_layer81": 0.0, "gnorm_layer82": 0.0, "gnorm_layer83": 0.0, "gnorm_layer84": 0.0, "gnorm_layer85": 0.0, "gnorm_layer86": 0.0, "gnorm_l
ayer87": 0.0, "gnorm_layer88": 0.0, "gnorm_layer89": 0.0, "gnorm_layer90": 0.0, "gnorm_layer91": 0.0, "gnorm_layer92": 0.0, "gnorm_layer93": 0.0, "gnorm_layer94":
```

Anything longer than that may be an issue…

- New train dir: /shared/home/namangoyal/checkpoints/175B/175B_run12.02.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

- Command ran:

```
git checkout 7b7ccd38f30a9db9c32df360922d803620268ce6

INCLUDED_HOSTS=node-[1-38,40-87,89-96,98-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p 175B_run12.01 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/

After launch:
sudo scontrol update job=1378 TimeLimit=UNLIMITED
sudo scontrol update job=1378 MailUser=<scrubbed> MailType=ALL
```

2021-11-11 5:40pm [Susan]: Run 12.01

- Same as 12.00 but with node 7 removed, and 97 swapped in.
- Still got stuck. Relaunching with node 7 put back.
- New train dir: /shared/home/namangoyal/checkpoints/175B/175B_run12.01.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992
- Command ran:

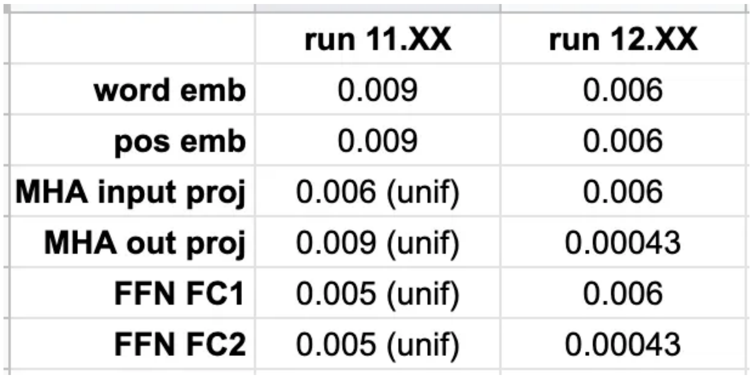
```
git checkout 7b7ccd38f30a9db9c32df360922d803620268ce6

INCLUDED_HOSTS=node-[1-6,8-38,40-87,89-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p 175B_run12.01 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/

After launch:
sudo scontrol update job=1377 TimeLimit=UNLIMITED
sudo scontrol update job=1377 MailUser=<scrubbed> MailType=ALL
```

- Rebooting node-7:
    - Drain first: `sudo scontrol update node=node-7 state=drain reason=lostgpu`

2021-11-11 5:30pm [Susan]: Run 12.00 - Lost a GPU on node-7, restarting.

- Discussion on gradient predivide: Quick primer on gradient predivide

- Summary of weight init decision:



	run 11.XX	run 12.XX
word emb	0.009	0.006
pos emb	0.009	0.006
MHA input proj	0.006 (unif)	0.006
MHA out proj	0.009 (unif)	0.00043
FFN FC1	0.005 (unif)	0.006
FFN FC2	0.005 (unif)	0.00043

- Weight Init from different codebases
    - Megatron-LM uses sigma / math.sqrt(2.0 * num_layers)
        - NOTE: The layer wise scaling is only being applied to the output layer i.e fc2 of ffn and out_proj of attn and not all the weight matrix. (this is mostly already implemented in gshard_combine_megatron_fsdp but with small differences.)
        - sigma = 0.006 for 175B
        - Word Embedding
            - normal(0, 0.006)
        - Position Embedding
            - normal(0, 0.006)
        - MHA
            - QKV input projection
                - normal(0, 0.006)
            - Output projection
                - sigma / math.sqrt(2.0 * num_layers)
                - Stddev: 0.00043
            - All biases: zero
        - FFN
            - FC1
                - normal(0, 0.006)
            - FC2
                - sigma / math.sqrt(2.0 * num_layers)
                    - Stddev: 0.00043
            - All biases: zero
        - Layer norms
            - Gamma: 1.0
            - Beta: 0.0
    - Fairseq gshard_combine_megatron_fsdp does:
        - Word Embedding
            - normal(0, 0.009)
        - Position Embedding:
            - normal(0, 0.009)
        - MHA
            - QKV input projection
                - Stddev: 0.006
                    - Model parallel fairseq is approximately normal(0, 0.006)

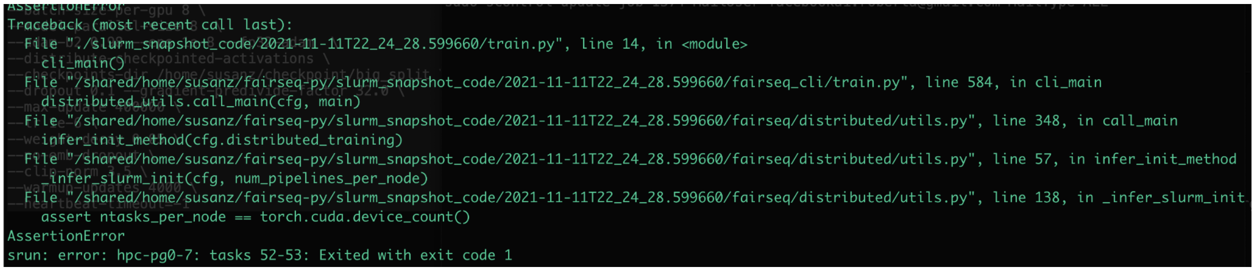
          - Note: this matches non-model parallel fairseq’s xaviar_uniform(..., gain=1 / math.sqrt(2)), which is similar to normal(0, 0.006)
      - Output projection
        - Stddev: 0.009
          - Model parallel fairseq is approximately normal(0, 0.009)
          - Note: this matches non-model parallel fairseq’s xavier_uniform, which is similar to normal(0, 0.009)
    - FFN
      - FC1
        - Stddev: 0.005
          - Uses kaiming_uniform(..., a=math.sqrt(5))
      - FC2
        - Stddev: 0.005
          - Uses kaiming_uniform(..., a=math.sqrt(5))
    - Layer norms
      - Gamma: 1.0
      - Beta: 0.0
  - DeepSpeed uses sigma / math.sqrt(2.0 * num_layers)
  - Mesh tensorflow doesn’t seem to scale init based on num layers
  - Lingvo doesn’t seem to scale init at all
  - Paddle Paddle uses pytorch defaults lmao
  - GPT NeoX uses same as deepspeed

- New train dir: /shared/home/namangoyal/checkpoints/175B/175B_run12.00.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnpos.0emb_scale.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr0.00012.endlr6e-06.wu2000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992

- Command ran:

```
git checkout 7b7ccd38f30a9db9c32df360922d803620268ce6

INCLUDED_HOSTS=node-[1-38,40-87,89-96,98-119,121-128] \
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p 175B_run12.00 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/

After launch:
sudo scontrol update job=1374 TimeLimit=UNLIMITED
sudo scontrol update job=1374 MailUser=<scrubbed> MailType=ALL
```

- Lost a GPU:



```
AssertionError
Traceback (most recent call last):
  File "./slurm_snapshot_code/2021-11-11T22_24_28.599660/train.py", line 14, in <module>
    cli_main()
  File "/shared/home/susanz/fairseq-py/slurm_snapshot_code/2021-11-11T22_24_28.599660/fairseq_cli/train.py", line 584, in cli_main
    distributed_utils.call_main(cfg, main)
  File "/shared/home/susanz/fairseq-py/slurm_snapshot_code/2021-11-11T22_24_28.599660/fairseq/distributed/utils.py", line 348, in call_main
    infer_init_method(cfg.distributed_training)
  File "/shared/home/susanz/fairseq-py/slurm_snapshot_code/2021-11-11T22_24_28.599660/fairseq/distributed/utils.py", line 57, in infer_init_method
    _infer_slurm_init(cfg, num_pipelines_per_node)
  File "/shared/home/susanz/fairseq-py/slurm_snapshot_code/2021-11-11T22_24_28.599660/fairseq/distributed/utils.py", line 138, in _infer_slurm_init
    assert ntasks_per_node == torch.cuda.device_count()
AssertionError
srun: error: hpc-pg0-7: tasks 52-53: Exited with exit code 1
```

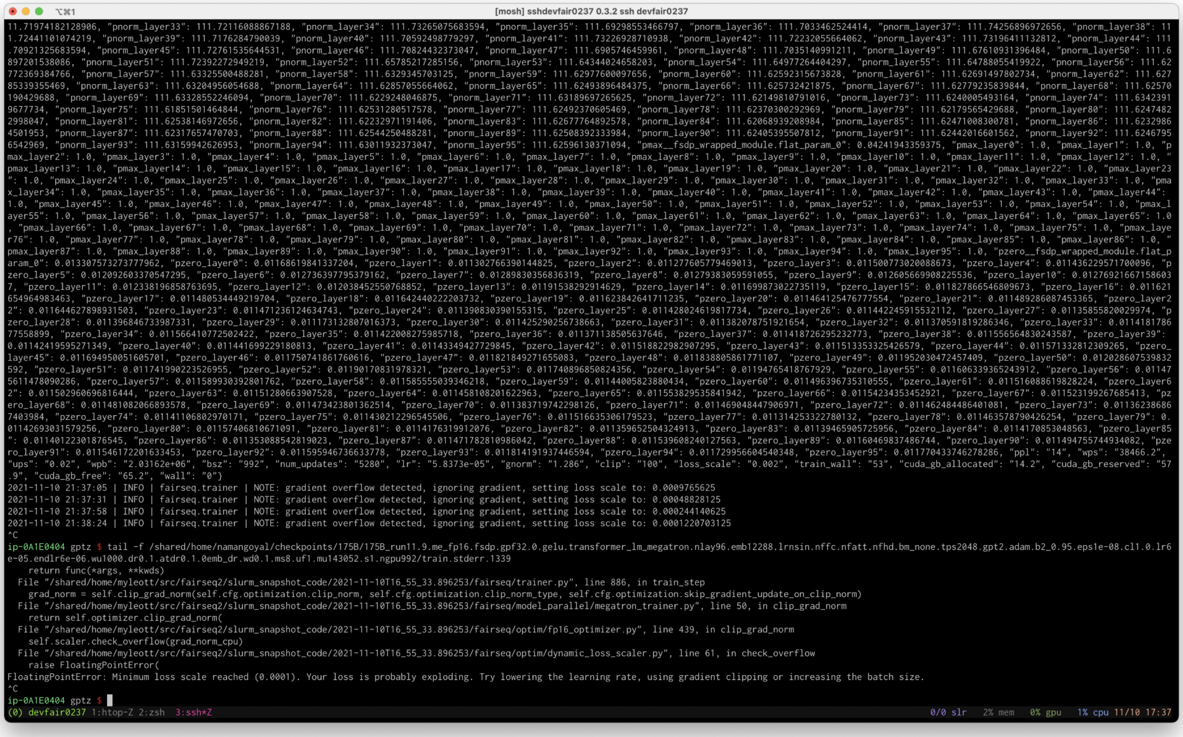
Restarting and rotating in 97 (rotating out 7).

2021-11-10 5pm: Run 11.10

NOTE: To resume Run 11, we must revert 7eacba2

Analysis of 11.9

Loss scale started dropping and hit min at 5280 updates



```
11.71974182128906, "pnorm_layer33": 111.72116088867188, "pnorm_layer34": 111.73265075683594, "pnorm_layer35": 111.69298553466797, "pnorm_layer36": 111.7033462524414, "pnorm_layer37": 111.74256896972656, "pnorm_layer38": 11
1.72441101074219, "pnorm_layer39": 111.7176284790039, "pnorm_layer40": 111.70592498779297, "pnorm_layer41": 111.73226928710938, "pnorm_layer42": 111.72232055664062, "pnorm_layer43": 111.73196411132812, "pnorm_layer44": 111
.70921325683594, "pnorm_layer45": 111.72761535644531, "pnorm_layer46": 111.70824432373047, "pnorm_layer47": 111.6905746459961, "pnorm_layer48": 111.7035140991211, "pnorm_layer49": 111.67610931396484, "pnorm_layer50": 111.6
897201538086, "pnorm_layer51": 111.72392272949219, "pnorm_layer52": 111.65785217285156, "pnorm_layer53": 111.64344024658203, "pnorm_layer54": 111.64977264404297, "pnorm_layer55": 111.64788055419922, "pnorm_layer56": 111.62
772369384766, "pnorm_layer57": 111.63325500488281, "pnorm_layer58": 111.6329345703125, "pnorm_layer59": 111.62977600097656, "pnorm_layer60": 111.62592315673828, "pnorm_layer61": 111.62691497802734, "pnorm_layer62": 111.627
85339355469, "pnorm_layer63": 111.63204956054688, "pnorm_layer64": 111.62857055664062, "pnorm_layer65": 111.62493896484375, "pnorm_layer66": 111.625732421875, "pnorm_layer67": 111.62779235839844, "pnorm_layer68": 111.62570
190429688, "pnorm_layer69": 111.63328552246094, "pnorm_layer70": 111.6229248046875, "pnorm_layer71": 111.63189697265625, "pnorm_layer72": 111.62149810791016, "pnorm_layer73": 111.6240005493164, "pnorm_layer74": 111.6342391
9677734, "pnorm_layer75": 111.61851501464844, "pnorm_layer76": 111.62531280517578, "pnorm_layer77": 111.62492370605469, "pnorm_layer78": 111.62370300292969, "pnorm_layer79": 111.62179565429688, "pnorm_layer80": 111.6247482
2998047, "pnorm_layer81": 111.62538146972656, "pnorm_layer82": 111.62232971191406, "pnorm_layer83": 111.62677764892578, "pnorm_layer84": 111.62068939208984, "pnorm_layer85": 111.62471008300781, "pnorm_layer86": 111.6232986
4501953, "pnorm_layer87": 111.62317657470703, "pnorm_layer88": 111.62544250488281, "pnorm_layer89": 111.62508392333984, "pnorm_layer90": 111.62405395507812, "pnorm_layer91": 111.62442016601562, "pnorm_layer92": 111.6246795
6542969, "pnorm_layer93": 111.63159942626953, "pnorm_layer94": 111.63011932373047, "pnorm_layer95": 111.62596130371094, "pmax__fsdp_wrapped_module.flat_param_0": 0.04241943359375, "pmax_layer0": 1.0, "pmax_layer1": 1.0, "p
max_layer2": 1.0, "pmax_layer3": 1.0, "pmax_layer4": 1.0, "pmax_layer5": 1.0, "pmax_layer6": 1.0, "pmax_layer7": 1.0, "pmax_layer8": 1.0, "pmax_layer9": 1.0, "pmax_layer10": 1.0, "pmax_layer11": 1.0, "pmax_layer12": 1.0, "
pmax_layer13": 1.0, "pmax_layer14": 1.0, "pmax_layer15": 1.0, "pmax_layer16": 1.0, "pmax_layer17": 1.0, "pmax_layer18": 1.0, "pmax_layer19": 1.0, "pmax_layer20": 1.0, "pmax_layer21": 1.0, "pmax_layer22": 1.0, "pmax_layer23
": 1.0, "pmax_layer24": 1.0, "pmax_layer25": 1.0, "pmax_layer26": 1.0, "pmax_layer27": 1.0, "pmax_layer28": 1.0, "pmax_layer29": 1.0, "pmax_layer30": 1.0, "pmax_layer31": 1.0, "pmax_layer32": 1.0, "pmax_layer33": 1.0, "pma
x_layer34": 1.0, "pmax_layer35": 1.0, "pmax_layer36": 1.0, "pmax_layer37": 1.0, "pmax_layer38": 1.0, "pmax_layer39": 1.0, "pmax_layer40": 1.0, "pmax_layer41": 1.0, "pmax_layer42": 1.0, "pmax_layer43": 1.0, "pmax_layer44":
1.0, "pmax_layer45": 1.0, "pmax_layer46": 1.0, "pmax_layer47": 1.0, "pmax_layer48": 1.0, "pmax_layer49": 1.0, "pmax_layer50": 1.0, "pmax_layer51": 1.0, "pmax_layer52": 1.0, "pmax_layer53": 1.0, "pmax_layer54": 1.0, "pmax_l
ayer55": 1.0, "pmax_layer56": 1.0, "pmax_layer57": 1.0, "pmax_layer58": 1.0, "pmax_layer59": 1.0, "pmax_layer60": 1.0, "pmax_layer61": 1.0, "pmax_layer62": 1.0, "pmax_layer63": 1.0, "pmax_layer64": 1.0, "pmax_layer65": 1.0
, "pmax_layer66": 1.0, "pmax_layer67": 1.0, "pmax_layer68": 1.0, "pmax_layer69": 1.0, "pmax_layer70": 1.0, "pmax_layer71": 1.0, "pmax_layer72": 1.0, "pmax_layer73": 1.0, "pmax_layer74": 1.0, "pmax_layer75": 1.0, "pmax_laye
r76": 1.0, "pmax_layer77": 1.0, "pmax_layer78": 1.0, "pmax_layer79": 1.0, "pmax_layer80": 1.0, "pmax_layer81": 1.0, "pmax_layer82": 1.0, "pmax_layer83": 1.0, "pmax_layer84": 1.0, "pmax_layer85": 1.0, "pmax_layer86": 1.0, "
pmax_layer87": 1.0, "pmax_layer88": 1.0, "pmax_layer89": 1.0, "pmax_layer90": 1.0, "pmax_layer91": 1.0, "pmax_layer92": 1.0, "pmax_layer93": 1.0, "pmax_layer94": 1.0, "pmax_layer95": 1.0, "pzero__fsdp_wrapped_module.flat_p
aram_0": 0.013307573273777962, "pzero_layer0": 0.01168619841337204, "pzero_layer1": 0.011302766390144825, "pzero_layer2": 0.011277605779469013, "pzero_layer3": 0.011500773020088673, "pzero_layer4": 0.011436229571700096, "p
zero_layer5": 0.012092603370547295, "pzero_layer6": 0.012736397795379162, "pzero_layer7": 0.01289830356836319, "pzero_layer8": 0.01279383059591055, "pzero_layer9": 0.012605669908225536, "pzero_layer10": 0.01276921667158603
7, "pzero_layer11": 0.012338196858763695, "pzero_layer12": 0.012038452550768852, "pzero_layer13": 0.01191538292914629, "pzero_layer14": 0.011699873022735119, "pzero_layer15": 0.011827866546809673, "pzero_layer16": 0.011621
654964983463, "pzero_layer17": 0.011480534449219704, "pzero_layer18": 0.011642440222203732, "pzero_layer19": 0.011623842641711235, "pzero_layer20": 0.011464125476777554, "pzero_layer21": 0.011489286087453365, "pzero_layer2
2": 0.011644627898931503, "pzero_layer23": 0.011471236124634743, "pzero_layer24": 0.011390830390155315, "pzero_layer25": 0.011428024619817734, "pzero_layer26": 0.011442245915532112, "pzero_layer27": 0.01135855820029974, "p
zero_layer28": 0.011396846733987331, "pzero_layer29": 0.011173132807016373, "pzero_layer30": 0.011425290256738663, "pzero_layer31": 0.011382078751921654, "pzero_layer32": 0.011370591819286346, "pzero_layer33": 0.0114181786
77558899, "pzero_layer34": 0.011566410772502422, "pzero_layer35": 0.011422008275985718, "pzero_layer36": 0.011371138505637646, "pzero_layer37": 0.011418726295232773, "pzero_layer38": 0.011556564830243587, "pzero_layer39":
0.01142419595271349, "pzero_layer40": 0.011441699229180813, "pzero_layer41": 0.011433494277729845, "pzero_layer42": 0.011518822982907295, "pzero_layer43": 0.011513353325426579, "pzero_layer44": 0.011571332812309265, "pzero_
layer45": 0.011694950051605701, "pzero_layer46": 0.011750741861760616, "pzero_layer47": 0.011821849271655083, "pzero_layer48": 0.011838805861771107, "pzero_layer49": 0.011952030472457409, "pzero_layer50": 0.01202860753983
2592, "pzero_layer51": 0.011741990223526955, "pzero_layer52": 0.01190170831978321, "pzero_layer53": 0.011740896850824356, "pzero_layer54": 0.01194765418767929, "pzero_layer55": 0.011606339365243912, "pzero_layer56": 0.01147
5611478090286, "pzero_layer57": 0.011589930392801762, "pzero_layer58": 0.011585555039346218, "pzero_layer59": 0.01144005823880434, "pzero_layer60": 0.011496396735310555, "pzero_layer61": 0.011516088619828224, "pzero_layer6
2": 0.011502960696816444, "pzero_layer63": 0.01151280663907528, "pzero_layer64": 0.011458108201622963, "pzero_layer65": 0.011553829535841942, "pzero_layer66": 0.01154234353452921, "pzero_layer67": 0.011523199267685413, "pz
ero_layer68": 0.011481082066893578, "pzero_layer69": 0.011473423801362514, "pzero_layer70": 0.011383719742298126, "pzero_layer71": 0.011469048447906971, "pzero_layer72": 0.011462484486401081, "pzero_layer73": 0.01136238686
7403984, "pzero_layer74": 0.01141106802970171, "pzero_layer75": 0.011430212296545506, "pzero_layer76": 0.011516635306179523, "pzero_layer77": 0.011314253322780132, "pzero_layer78": 0.011463578790426254, "pzero_layer79": 0.
01142693031579256, "pzero_layer80": 0.01157406810671091, "pzero_layer81": 0.0114176319912076, "pzero_layer82": 0.011359652504324913, "pzero_layer83": 0.01139465905725956, "pzero_layer84": 0.0114170853048563, "pzero_layer85
": 0.01140122301876545, "pzero_layer86": 0.011353088542819023, "pzero_layer87": 0.011471782810986042, "pzero_layer88": 0.011539608240127563, "pzero_layer89": 0.01160469837486744, "pzero_layer90": 0.011494755744934082, "pze
ro_layer91": 0.011546172201633453, "pzero_layer92": 0.011595946736633778, "pzero_layer93": 0.011814191937446594, "pzero_layer94": 0.011729956604540348, "pzero_layer95": 0.011770433746278286, "ppl": "14", "wps": "38466.2",
"ups": "0.02", "wpb": "2.03162e+06", "bsz": "992", "num_updates": "5280", "lr": "5.8373e-05", "gnorm": "1.286", "clip": "100", "loss_scale": "0.002", "train_wall": "53", "cuda_gb_allocated": "14.2", "cuda_gb_reserved": "57
.9", "cuda_gb_free": "65.2", "wall": "0"}
2021-11-10 21:37:05 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.0009765625
2021-11-10 21:37:31 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.00048828125
2021-11-10 21:37:58 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.000244140625
2021-11-10 21:38:24 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.0001220703125
^C
ip-0A1E0404 gptz $ tail -f /shared/home/namangoyal/checkpoints/175B/175B_run11.9.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr6
e-05.endlr6e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.stderr.1339
    return func(*args, **kwds)
  File "/shared/home/myleott/src/fairseq2/slurm_snapshot_code/2021-11-10T16_55_33.896253/fairseq/trainer.py", line 886, in train_step
    grad_norm = self.clip_grad_norm(self.cfg.optimization.clip_norm, self.cfg.optimization.clip_norm_type, self.cfg.optimization.skip_gradient_update_on_clip_norm)
  File "/shared/home/myleott/src/fairseq2/slurm_snapshot_code/2021-11-10T16_55_33.896253/fairseq/model_parallel/megatron_trainer.py", line 50, in clip_grad_norm
    return self.optimizer.clip_grad_norm(
  File "/shared/home/myleott/src/fairseq2/slurm_snapshot_code/2021-11-10T16_55_33.896253/fairseq/optim/fp16_optimizer.py", line 439, in clip_grad_norm
    self.scaler.check_overflow(grad_norm_cpu)
  File "/shared/home/myleott/src/fairseq2/slurm_snapshot_code/2021-11-10T16_55_33.896253/fairseq/optim/dynamic_loss_scaler.py", line 61, in check_overflow
    raise FloatingPointError(
FloatingPointError: Minimum loss scale reached (0.0001). Your loss is probably exploding. Try lowering the learning rate, using gradient clipping or increasing the batch size.
^C
ip-0A1E0404 gptz $ █
(0) devfair0237 1:htop-Z 2:zsh  3:ssh*Z                                                                        0/0 slr   2% mem   0% gpu   1% cpu 11/10 17:37
```

Decision

- Launch 11.10 where we:
    - Switch to RELU
    - Switch to Stable MHA
    - We assume this will die
- Presumably Thursday AM we will launch **12.00** with:
    - Kill normformer
    - Lower LR
    - Batch Skipping

        - Actual LPE, no scale emb
        - And possibly some changes to init (layerwise init possibly)
    - In parallel, we will use the RSC to ablate different layerwise init options:
        - Default (as is now)
        - Layerwise init scaled by a global constant (as a function of total layers)
        - Layerwise init scaled by per-layer-index

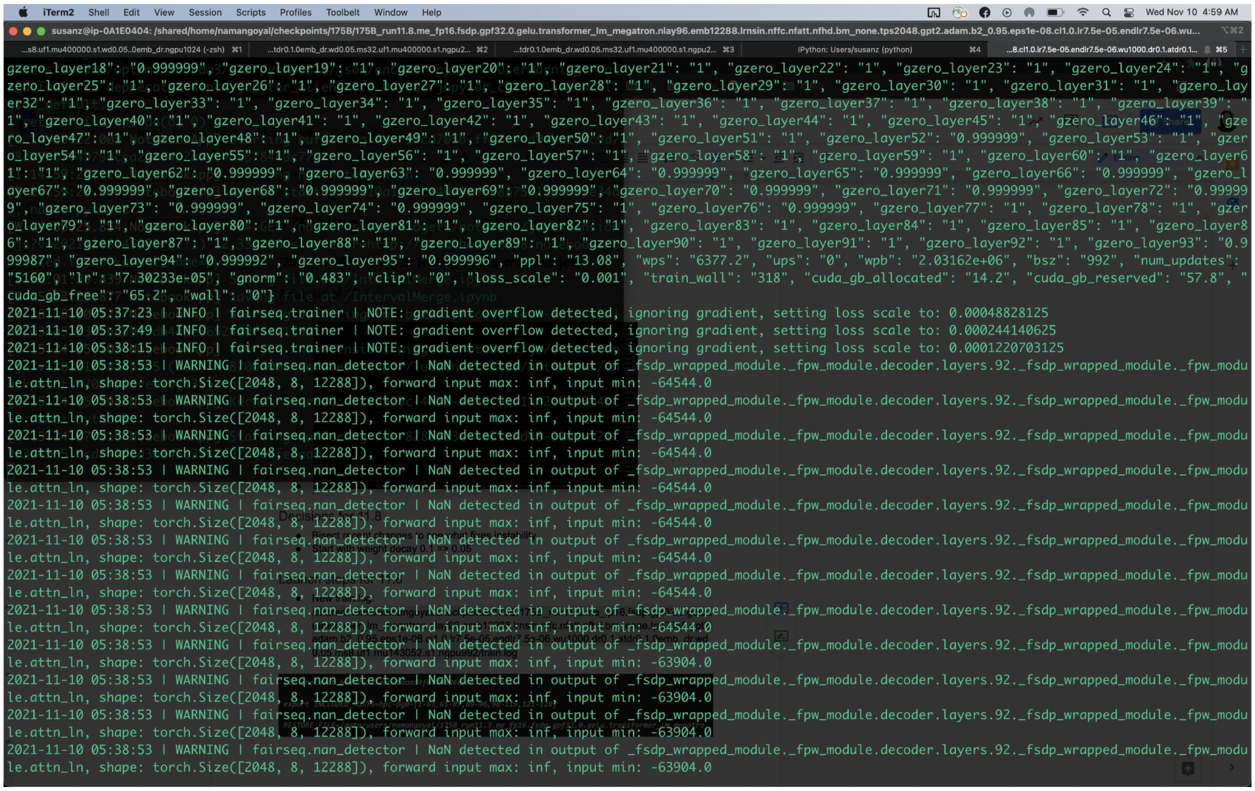
Launch steps for Run 11.10

- New train log: /shared/home/namangoyal/checkpoints/175B/175B_run11.10.me_fp16.fsdp.gpf32.0.relu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr6e-05.endlr6e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
git checkout 075f54e1b8da88ab90d6a0717d6e73eba33b98a0

export INCLUDED_HOSTS=node-[1-38,40-87,89-96,98-119,121-128]

RESTORE_FILE=/data/users/namangoyal/175B_run11.3.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/checkpoint_2_4750/checkpoint_2_4750.pt

RUN_ID=175B_run11.10

# note that this command has --reset-dataloader, which is not desired in general

python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE \
    --reset-dataloader

After launch:
sudo scontrol update job=1365 TimeLimit=UNLIMITED
sudo scontrol update job=1365 MailUser=<scrubbed> MailType=ALL
```

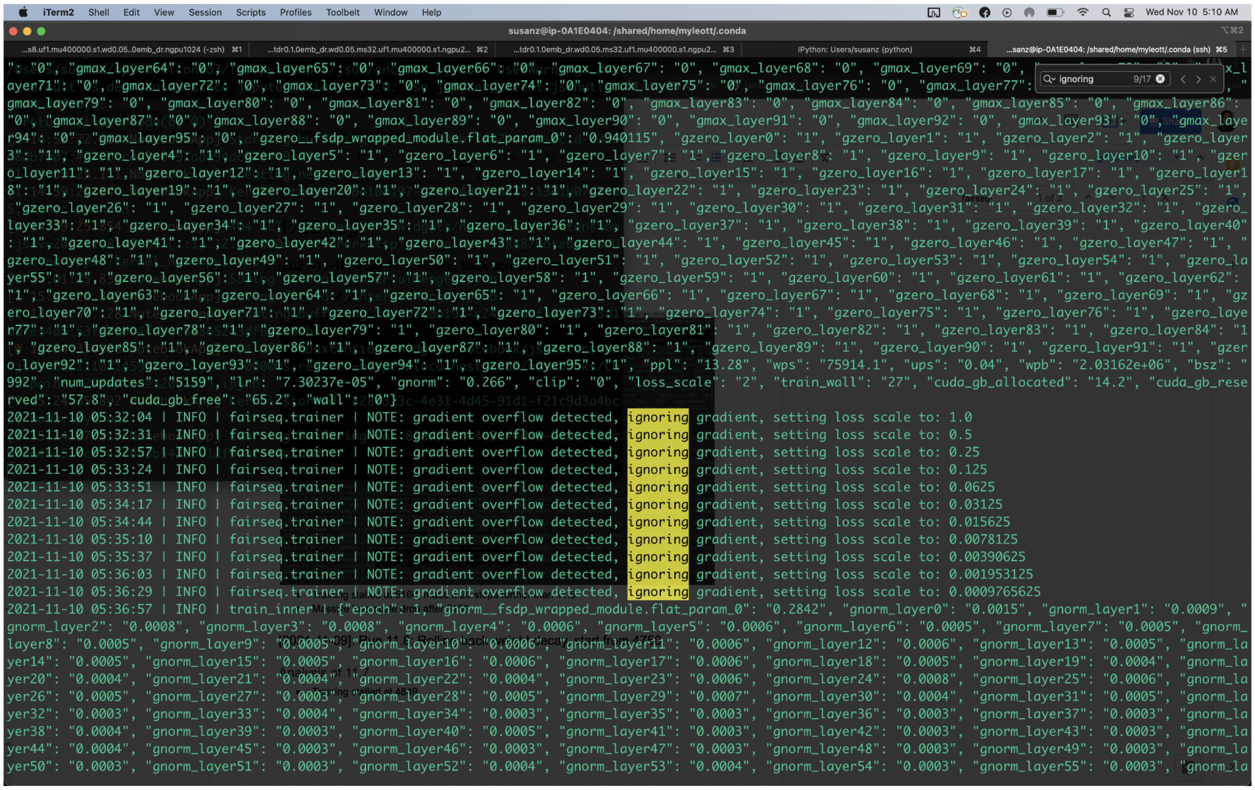
[2021-11-10]: Run 11.9: Lowered LR to 6e-5, match exp 11.6 otherwise

Analysis of 11.8


iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
Wed Nov 10 4:59 AM
susanz@ip-0A1E0404: /shared/home/namangoyal/checkpoints/175B/175B_run11.8.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr7.5e-05.endlr7.5e-06.wu...
...s8.uf1.mu400000.s1.wd0.05..0emb_dr.ngpu1024 (-zsh) ⌘1
...tdr0.1.0emb_dr.wd0.05.ms32.uf1.mu400000.s1.ngpu2... ⌘2
...tdr0.1.0emb_dr.wd0.05.ms32.uf1.mu400000.s1.ngpu2... ⌘3
IPython: Users/susanz (python) ⌘4
...8.cl1.0.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1... ⌘5

```
gzero_layer18": "0.999999", "gzero_layer19": "1", "gzero_layer20": "1", "gzero_layer21": "1", "gzero_layer22": "1", "gzero_layer23": "1", "gzero_layer24": "1", "g
zero_layer25": "1", "gzero_layer26": "1", "gzero_layer27": "1", "gzero_layer28": "1", "gzero_layer29": "1", "gzero_layer30": "1", "gzero_layer31": "1", "gzero_lay
er32": "1", "gzero_layer33": "1", "gzero_layer34": "1", "gzero_layer35": "1", "gzero_layer36": "1", "gzero_layer37": "1", "gzero_layer38": "1", "gzero_layer39": "
1", "gzero_layer40": "1", "gzero_layer41": "1", "gzero_layer42": "1", "gzero_layer43": "1", "gzero_layer44": "1", "gzero_layer45": "1", "gzero_layer46": "1", "gze
ro_layer47": "1", "gzero_layer48": "1", "gzero_layer49": "1", "gzero_layer50": "1", "gzero_layer51": "1", "gzero_layer52": "0.999999", "gzero_layer53": "1", "gzer
o_layer54": "1", "gzero_layer55": "1", "gzero_layer56": "1", "gzero_layer57": "1", "gzero_layer58": "1", "gzero_layer59": "1", "gzero_layer60": "1", "gzero_layer6
1": "1", "gzero_layer62": "0.999999", "gzero_layer63": "0.999999", "gzero_layer64": "0.999999", "gzero_layer65": "0.999999", "gzero_layer66": "0.999999", "gzero_l
ayer67": "0.999999", "gzero_layer68": "0.999999", "gzero_layer69": "0.999999", "gzero_layer70": "0.999999", "gzero_layer71": "0.999999", "gzero_layer72": "0.99999
9", "gzero_layer73": "0.999999", "gzero_layer74": "0.999999", "gzero_layer75": "1", "gzero_layer76": "0.999999", "gzero_layer77": "1", "gzero_layer78": "1", "gzer
o_layer79": "1", "gzero_layer80": "1", "gzero_layer81": "1", "gzero_layer82": "1", "gzero_layer83": "1", "gzero_layer84": "1", "gzero_layer85": "1", "gzero_layer8
6": "1", "gzero_layer87": "1", "gzero_layer88": "1", "gzero_layer89": "1", "gzero_layer90": "1", "gzero_layer91": "1", "gzero_layer92": "1", "gzero_layer93": "0.9
99987", "gzero_layer94": "0.999992", "gzero_layer95": "0.999996", "ppl": "13.08", "wps": "6377.2", "ups": "0", "wpb": "2.03162e+06", "bsz": "992", "num_updates":
"5160", "lr": "7.30233e-05", "gnorm": "0.483", "clip": "0", "loss_scale": "0.001", "train_wall": "318", "cuda_gb_allocated": "14.2", "cuda_gb_reserved": "57.8", "
cuda_gb_free": "65.2", "wall": "0"}
2021-11-10 05:37:23 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.00048828125
2021-11-10 05:37:49 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.000244140625
2021-11-10 05:38:15 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.0001220703125
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -64544.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -63904.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -63904.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -63904.0
2021-11-10 05:38:53 | WARNING | fairseq.nan_detector | NaN detected in output of _fsdp_wrapped_module._fpw_module.decoder.layers.92._fsdp_wrapped_module._fpw_modu
le.attn_ln, shape: torch.Size([2048, 8, 12288]), forward input max: inf, input min: -63904.0
```

- Training stalled at **5160**, made it 41 steps further than 11.6 (when weight decay was also 0.05)
  - 11.6 got to num_updates 5119 before naning out
- Massive loss scale drop after **5159**:


iTerm2 Shell Edit View Session Scripts Profiles Toolbelt Window Help
Wed Nov 10 5:10 AM
susanz@ip-0A1E0404: /shared/home/myleott/.conda
...s8.uf1.mu400000.s1.wd0.05..0emb_dr.ngpu1024 (-zsh) ⌘1
...tdr0.1.0emb_dr.wd0.05.ms32.uf1.mu400000.s1.ngpu2... ⌘2
...tdr0.1.0emb_dr.wd0.05.ms32.uf1.mu400000.s1.ngpu2... ⌘3
IPython: Users/susanz (python) ⌘4
...sanz@ip-0A1E0404: /shared/home/myleott/.conda (ssh) ⌘5
ignoring
9/17

```
": "0", "gmax_layer64": "0", "gmax_layer65": "0", "gmax_layer66": "0", "gmax_layer67": "0", "gmax_layer68": "0", "gmax_layer69": "0",                        x_l
ayer71": "0", "gmax_layer72": "0", "gmax_layer73": "0", "gmax_layer74": "0", "gmax_layer75": "0", "gmax_layer76": "0", "gmax_layer77":                        "
gmax_layer79": "0", "gmax_layer80": "0", "gmax_layer81": "0", "gmax_layer82": "0", "gmax_layer83": "0", "gmax_layer84": "0", "gmax_layer85": "0", "gmax_layer86":
"0", "gmax_layer87": "0", "gmax_layer88": "0", "gmax_layer89": "0", "gmax_layer90": "0", "gmax_layer91": "0", "gmax_layer92": "0", "gmax_layer93": "0", "gmax_laye
r94": "0", "gmax_layer95": "0", "gzero__fsdp_wrapped_module.flat_param_0": "0.940115", "gzero_layer0": "1", "gzero_layer1": "1", "gzero_layer2": "1", "gzero_layer
3": "1", "gzero_layer4": "1", "gzero_layer5": "1", "gzero_layer6": "1", "gzero_layer7": "1", "gzero_layer8": "1", "gzero_layer9": "1", "gzero_layer10": "1", "gzer
o_layer11": "1", "gzero_layer12": "1", "gzero_layer13": "1", "gzero_layer14": "1", "gzero_layer15": "1", "gzero_layer16": "1", "gzero_layer17": "1", "gzero_layer1
8": "1", "gzero_layer19": "1", "gzero_layer20": "1", "gzero_layer21": "1", "gzero_layer22": "1", "gzero_layer23": "1", "gzero_layer24": "1", "gzero_layer25": "1",
"gzero_layer26": "1", "gzero_layer27": "1", "gzero_layer28": "1", "gzero_layer29": "1", "gzero_layer30": "1", "gzero_layer31": "1", "gzero_layer32": "1", "gzero_
layer33": "1", "gzero_layer34": "1", "gzero_layer35": "1", "gzero_layer36": "1", "gzero_layer37": "1", "gzero_layer38": "1", "gzero_layer39": "1", "gzero_layer40"
: "1", "gzero_layer41": "1", "gzero_layer42": "1", "gzero_layer43": "1", "gzero_layer44": "1", "gzero_layer45": "1", "gzero_layer46": "1", "gzero_layer47": "1", "
gzero_layer48": "1", "gzero_layer49": "1", "gzero_layer50": "1", "gzero_layer51": "1", "gzero_layer52": "1", "gzero_layer53": "1", "gzero_layer54": "1", "gzero_la
yer55": "1", "gzero_layer56": "1", "gzero_layer57": "1", "gzero_layer58": "1", "gzero_layer59": "1", "gzero_layer60": "1", "gzero_layer61": "1", "gzero_layer62":
"1", "gzero_layer63": "1", "gzero_layer64": "1", "gzero_layer65": "1", "gzero_layer66": "1", "gzero_layer67": "1", "gzero_layer68": "1", "gzero_layer69": "1", "gz
ero_layer70": "1", "gzero_layer71": "1", "gzero_layer72": "1", "gzero_layer73": "1", "gzero_layer74": "1", "gzero_layer75": "1", "gzero_layer76": "1", "gzero_laye
r77": "1", "gzero_layer78": "1", "gzero_layer79": "1", "gzero_layer80": "1", "gzero_layer81": "1", "gzero_layer82": "1", "gzero_layer83": "1", "gzero_layer84": "1
", "gzero_layer85": "1", "gzero_layer86": "1", "gzero_layer87": "1", "gzero_layer88": "1", "gzero_layer89": "1", "gzero_layer90": "1", "gzero_layer91": "1", "gzer
o_layer92": "1", "gzero_layer93": "1", "gzero_layer94": "1", "gzero_layer95": "1", "ppl": "13.28", "wps": "75914.1", "ups": "0.04", "wpb": "2.03162e+06", "bsz": "
992", "num_updates": "5159", "lr": "7.30237e-05", "gnorm": "0.266", "clip": "0", "loss_scale": "2", "train_wall": "27", "cuda_gb_allocated": "14.2", "cuda_gb_rese
rved": "57.8", "cuda_gb_free": "65.2", "wall": "0"}
2021-11-10 05:32:04 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 1.0
2021-11-10 05:32:31 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.5
2021-11-10 05:32:57 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.25
2021-11-10 05:33:24 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.125
2021-11-10 05:33:51 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.0625
2021-11-10 05:34:17 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.03125
2021-11-10 05:34:44 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.015625
2021-11-10 05:35:10 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.0078125
2021-11-10 05:35:37 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.00390625
2021-11-10 05:36:03 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.001953125
2021-11-10 05:36:29 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, setting loss scale to: 0.0009765625
2021-11-10 05:36:57 | INFO | train_inner | {"epoch": 1, "gnorm__fsdp_wrapped_module.flat_param_0": "0.2842", "gnorm_layer0": "0.0015", "gnorm_layer1": "0.0009", "
gnorm_layer2": "0.0008", "gnorm_layer3": "0.0008", "gnorm_layer4": "0.0006", "gnorm_layer5": "0.0006", "gnorm_layer6": "0.0005", "gnorm_layer7": "0.0005", "gnorm_
layer8": "0.0005", "gnorm_layer9": "0.0005", "gnorm_layer10": "0.0006", "gnorm_layer11": "0.0006", "gnorm_layer12": "0.0006", "gnorm_layer13": "0.0005", "gnorm_la
yer14": "0.0005", "gnorm_layer15": "0.0005", "gnorm_layer16": "0.0006", "gnorm_layer17": "0.0006", "gnorm_layer18": "0.0005", "gnorm_layer19": "0.0004", "gnorm_la
yer20": "0.0004", "gnorm_layer21": "0.0004", "gnorm_layer22": "0.0004", "gnorm_layer23": "0.0006", "gnorm_layer24": "0.0008", "gnorm_layer25": "0.0006", "gnorm_la
yer26": "0.0005", "gnorm_layer27": "0.0003", "gnorm_layer28": "0.0005", "gnorm_layer29": "0.0007", "gnorm_layer30": "0.0004", "gnorm_layer31": "0.0005", "gnorm_la
yer32": "0.0003", "gnorm_layer33": "0.0004", "gnorm_layer34": "0.0003", "gnorm_layer35": "0.0003", "gnorm_layer36": "0.0003", "gnorm_layer37": "0.0003", "gnorm_la
yer38": "0.0004", "gnorm_layer39": "0.0003", "gnorm_layer40": "0.0005", "gnorm_layer41": "0.0003", "gnorm_layer42": "0.0003", "gnorm_layer43": "0.0003", "gnorm_la
yer44": "0.0004", "gnorm_layer45": "0.0003", "gnorm_layer46": "0.0003", "gnorm_layer47": "0.0003", "gnorm_layer48": "0.0003", "gnorm_layer49": "0.0003", "gnorm_la
yer50": "0.0003", "gnorm_layer51": "0.0003", "gnorm_layer52": "0.0004", "gnorm_layer53": "0.0004", "gnorm_layer54": "0.0003", "gnorm_layer55": "0.0003", "gnorm_la
```

- Still layer 92 that infs
- Last log shows lr of 7.30233e-05

Decisions for 11.9

- Turn clipping back on (removing the “throw batch out” logic)
- Lower LR to 6e-5
- Otherwise match run 11.7:
    - Increase weight decay to 0.1
    - Lower beta2 to 0.95
    - Roll back to checkpoint @ 4750 steps
    - Reset dataloader
- Misc:
    - Add pnorm logging

Launch steps for 11.9

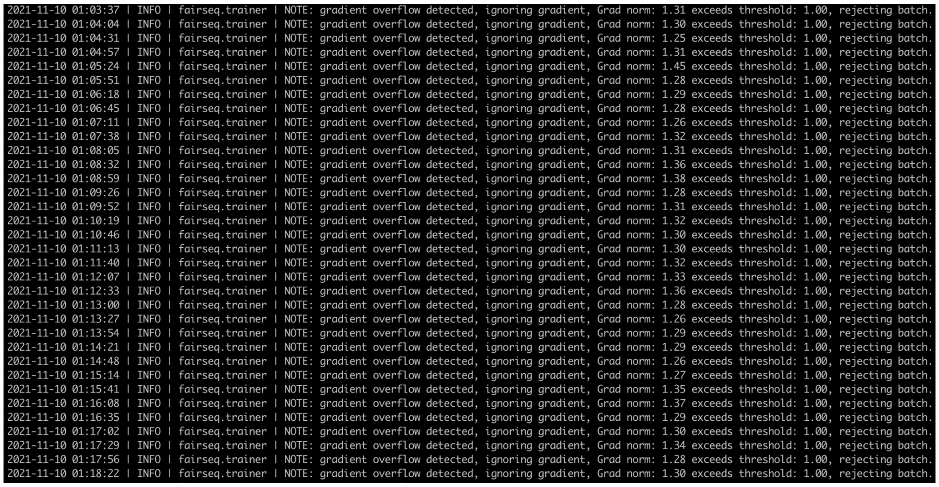
- New train log: /shared/home/namangoyal/checkpoints/175B/175B_run11.9.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr6e-05.endlr6e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

```
git checkout a69e5d30b4199f40c4651deac918c58ab594f018

export INCLUDED_HOSTS=node-[1-38,40-87,89-96,98-119,121-128]

RESTORE_FILE=/data/users/namangoyal/175B_run11.3.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb1228
8.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atd
r0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/checkpoint_2_4750/checkpoint_2_4750.pt

RUN_ID=175B_run11.9

# note that this command has --reset-dataloader, which is not desired in general

python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE \
    --reset-dataloader

After launch:
sudo scontrol update job=1336 TimeLimit=UNLIMITED
sudo scontrol update job=1336 MailUser=<scrubbed> MailType=ALL
```

[2021-11-09]: Run 11.8: Rolling back weight decay, start from 4750

Analysis of 11.7

- Training stalled at 4849

- 

```
2021-11-10 01:03:37 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.31 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:04:04 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.30 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:04:31 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.25 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:04:57 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.31 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:05:24 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.45 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:05:51 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.28 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:06:18 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.29 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:06:45 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.28 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:07:11 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.26 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:07:38 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.32 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:08:05 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.31 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:08:32 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.36 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:08:59 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.38 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:09:26 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.28 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:09:52 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.31 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:10:19 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.32 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:10:46 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.30 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:11:13 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.30 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:11:40 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.32 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:12:07 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.33 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:12:33 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.36 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:13:00 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.28 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:13:27 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.26 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:13:54 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.29 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:14:21 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.29 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:14:48 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.26 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:15:14 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.27 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:15:41 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.35 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:16:08 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.37 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:16:35 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.29 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:17:02 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.30 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:17:29 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.34 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:17:56 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.28 exceeds threshold: 1.00, rejecting batch.
2021-11-10 01:18:22 | INFO | fairseq.trainer | NOTE: gradient overflow detected, ignoring gradient, Grad norm: 1.30 exceeds threshold: 1.00, rejecting batch.
```

Decisions for 11.8

- Bisect recent changes to see what fixes instability
- Start with weight decay 0.1 => 0.05

Launch steps for 11.8

- New train.log: /shared/home/namangoyal/checkpoints/175B/175B_run11.8.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/train.log

```
git checkout b4842e181dababaaad4f0170ad6c90b782877b1c

export INCLUDED_HOSTS=node-[1-63,65-87,89-96,98-119,121-128]

RESTORE_FILE=/data/users/namangoyal/175B_run11.3.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb1228
8.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atd
r0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/checkpoint_2_4750/checkpoint_2_4750.pt

RUN_ID=175B_run11.8

# note that this command has --reset-dataloader, which is not desired in general

python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE \
    --reset-dataloader

After launch:
sudo scontrol update job=1336 TimeLimit=UNLIMITED
sudo scontrol update job=1336 MailUser=<scrubbed> MailType=ALL
```

[2021-11-09]: Run 11.7: Changing tons of stuff, start from 4750

Analysis

Potential concerns:

- [Preemptive] Data composition

Launch steps for 11.7

- Analysis to come
- Fairseq commit: d9c903ea0a2894071fb5bee96b0c9612f1e5a402
- new data: /data/opt/corpus_dedup_10_10_1_0.05_run11.7/
- Beta2: 0.95 , weight_decay: 0.1, reset dataloader true
- Log: /shared/home/namangoyal/checkpoints/175B/175B_run11.7.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.95.eps1e-08.cl1.0.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.1.ms8.uf1.mu143052.s1.ngpu992/train.log

[2021-11-09]: Run 11.6: Starting to skip batches

Emergency meeting notes

- Started by cleaning up tech debt of these notes
- Performed analysis of 11.6 (see below)
- Brainstormed intervention paths

Decided actions (becomes 11.7)

- Reset dataloader (see description in brainstorm)
- Resume from 4750
- Increase WD to 0.05 -> 0.1
- Lower beta2 to 0.98 -> 0.95
- (Note we need a new runid)
- [Myle] swap the shards
- [Naman] Launch the run

Backup option if things die in the middle of the night:

- [Sam] Check at 11pm
- Lower the LR to 6e-5

Brainstorm of interventions (With initializing from some checkpoint):

- Reset dataloader for faster startup
    - --reset-dataloader
    - Swap shard1 with shard29
    - Swap shard2 with shard30
    - Reasoning: we’re late in shard 2 and so we end up doing 45 minutes of tokenization before we can see updates
- To be done together:
    - Lower beta2 to 0.95
    - Increase Weight Decay to 0.1

        - How much would we need to roll back to address this?
- Lower LR to 6e-5
- Hotswap GELU => RELU
    - Unsure if the code path works due to fused -- need to test on 100M params
- Clamp activations
- Shuffle the data
- Adding the max to the layer norm
- Manually shrink gradient to embeddings
- Increase layernorm epsilon
- [bottom pri] Adam epsilon 1e-8 => 1e-6

Brainstorm of interventions that require full restart [Unanimously unpopular:

- Cold swap GELU => RELU (with restart)
- Remove normformer
- Different data
- Lower LR with cold restart

Analysis of 11.6

Hypothesis:

- Instead of clipping, start throwing stuff away
- Resume from 5000 steps (140 steps rewound)

Note 11.6 is NOT equivalent to 11.5 even with restart

- We reset the loss scale state (accidentally - this isn't checkpointed: num updates for which overflow didn't happen)
    - Losses match to 2nd decimal, gnorms start diverging
    - Loss scales start diverging slightly after gnorm
    - Interesting: slightly higher loss scale made it survive slightly more updates

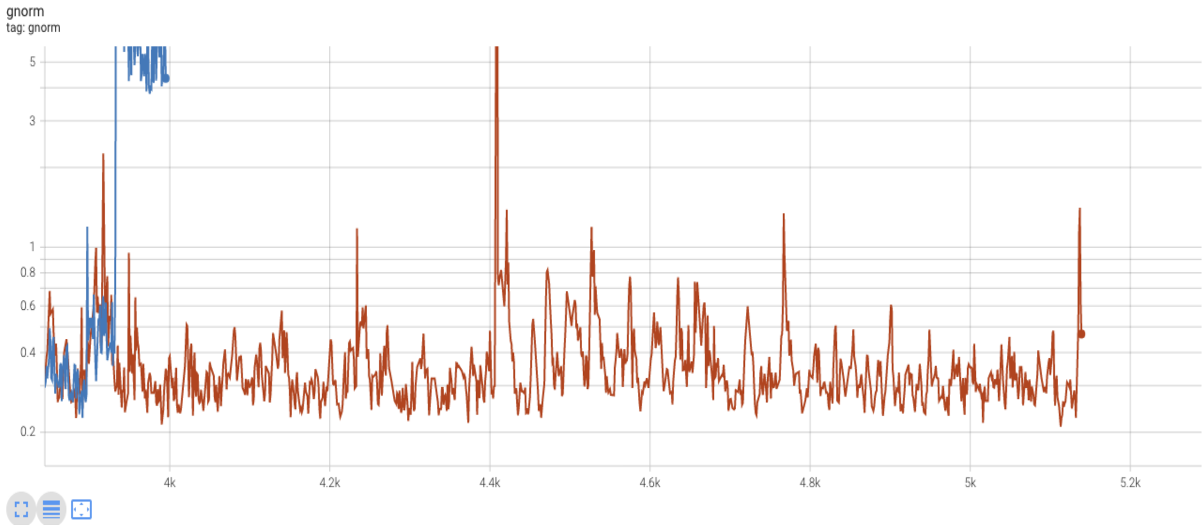
Cute observations:

- We are seeing a very the same thing: layer 92 attn goes in the range of [-inf, *almost* inf]
    - AND this is happening in the forward pass
- We only threw away one batch before the loss scale went crazy

[2021-11-08] Run 11.5: clip 1.5 -> 1.0

Analysis of 11.5

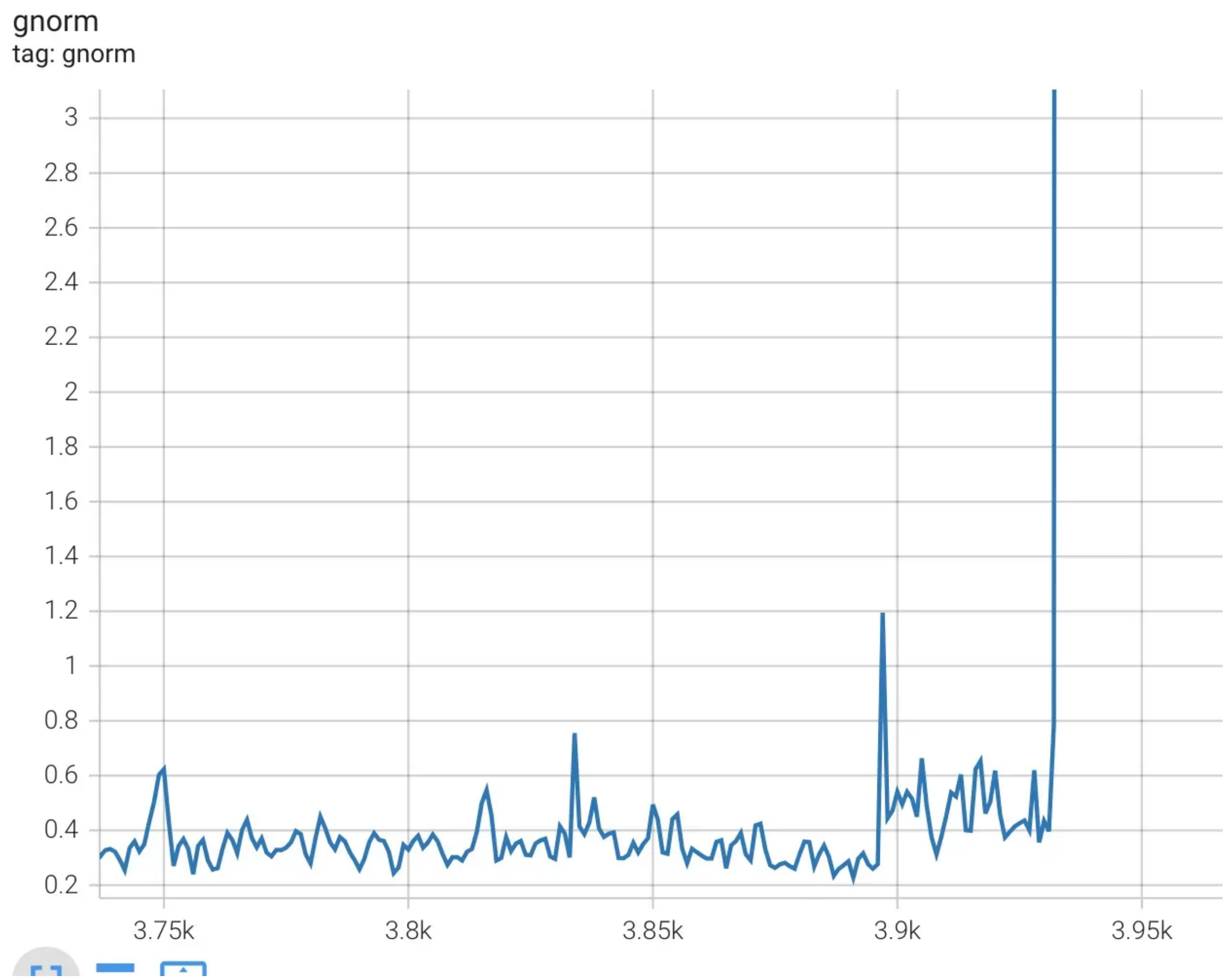
- Reason: 11.5 exploded at 5139 updates
- Diagnosis: gnorm spike at end: (we had survived bigger spikes but this one exploded the run)
- Note: loss scale is very low (.000122…)
- Layer 92 attention layer norm *inputs* seemed to be where the input to attn layer norm had -inf (and a maximum value pretty damn close to +inf)
    - We were already dropping loss scale really far back.
    - Large gnorm at step 5136 there was a large, clipped grad norm
        - Unsolved mystery: which layer really the problem?
    - We do a handful more updates and we -inf out



- Action:
  - We decided to skip gradient update when gnorm is higher than clip norm threshold: #2602
  - Other theories: attn is numerically unstable
- Restart logs:
  - /shared/home/namangoyal/checkpoints/175B/175B_run11.6.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.0.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/train.log

Analysis of 11.4

- 11.4 exploded at around ~3.94k.
- The gnorm looks like below
- Reduced clipping to 1.0 and restarted
- Logs:
- /shared/home/namangoyal/checkpoints/175B/175B_run11.5.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.0.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/train.log



[Undated] Run 11.4: Changed validation freq

Just trying to spend less time in validation/saving.

Note: validation sets changed, so valid_ppl is not comparable to previous runs

- Restarted with following changes (1959e415390560c7b2d680317ca3c07a5f24f8cc):
  - Reducing validation frequency to 1000 instead of 250
  - Model Initialization on gpu
  - Reduced validation set sizes
- Logs:
  - /shared/home/namangoyal/checkpoints/175B/175B_run11.4.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/train.log
- Added to tensorboard:
  - cd /shared/home/namangoyal/checkpoints/175B/tensorboard
  - sudo ln -s ../175B_run11.4.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/tb/ run11.4

[2021-11-07] Run 11.3: ECC Failure

No configuration changes.

- ECC error for node node-64 at 2285 updates
    - Action: Node put on drain
    - Downloaded 2250 checkpoint to:
        - /data/users/namangoyal/175B_run11.3.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2250

- **New train.log path:** /shared/home/namangoyal/checkpoints/175B/175B_run11.3.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/train.log

```
git checkout 9b0645d7779a247c2861742b46112a131bf0a67b

export INCLUDED_HOSTS=node-[1-63,65-87,89-96,98-119,121-128]
RESTORE_FILE=/data/users/namangoyal/175B_run11.3.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_2250/checkpoint_1_2250.pt
RUN_ID=175B_run11.3
python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE
```

[2021-11-06] Run 11.2: clip 2.5 -> 1.5

- Resumed from 1K steps with peak LR of 7.5e-5
- Node failure at ~1950 steps
    - The good news is that slurm did try to requeue the job, so once we build the automatic download-latest-checkpoint-from-blob-and-resume functionality, this kind of problem will recover automatically
    - We also had some ECC errors from yesterday, so all of our 4 buffer nodes are bad (node-[61,88,97,120]). Manually recycled them this morning following the instructions at the top of the Cloud cluster admin doc.
- Relaunching run on remaining 124 nodes
    - There was a big ppl jump ~1430 steps, and the model was just recovering between 1430-1950 steps (after which the node failure stopped the job). Looking at gnorms, decided to roll back to 1250 steps and lower clipping from 2.5 => 1.5.
    - Myle is AFK for the rest of the day. Will defer to Naman, Sam, Stephen and Susan to react to job failures if needed
- Known problems / TODOs

    - Blob storage URL seems to change across requeues, so we'll need to address this (probably in fb_sweep/sweep/slurm.py) before we can automatically download checkpoints from blob and resume training
    - Tensorboard seems to be bad at resuming training from a checkpoint. If we reuse the same tensorboard dir, then the graphs have a weird overlap. If we switch to a new save_dir (tensorboard dir), then we lose the previous training history :/
        - Solution for now is new tensorboard dir. It's easy enough to copy the tfevent files if we later decide to stitch things back together.
- New fairseq commit: 9b0645d7779a247c2861742b46112a131bf0a67b
- **New train.log path:**
  `/shared/home/namangoyal/checkpoints/175B/175B_run11.2.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl1.5.lr7.5e-05.endlr7.5e-06.wu1000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/train.log`

```
cd
/data/users/myleott/175B_run11.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfat
t.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl2.5.lr0.0003.endlr
1e-05.wu4000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992

# Note that the blob storage URL seems to change across requeues,
# need to get the latest one from train.log

BLOB_URL="<<<SCRUBBED FOR RELEASE>>>"

cp --recursive --include-pattern "checkpoint_1_1250-*.pt" "$BLOB_URL/*" checkpoint_1_1250/

cd /path/to/fairseq-py

RESTORE_FILE=/data/users/myleott/175B_run11.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrn
sin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl2.5.lr0.0003.endlr1e-05.wu4000.dr0.1.atdr0.1.0em
b_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_1250/checkpoint_1_1250.pt

# increment the minor version to get a new save dir / tensorboard dir
RUN_ID=175B_run11.2

INCLUDED_HOSTS=node-[1-60,62-87,89-96,98-119,121-128] python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p $RUN_ID \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE \
    --dry-run

After launch:
sudo scontrol update job=1286 TimeLimit=UNLIMITED
sudo scontrol update job=1286 MailUser=<scrubbed> MailType=ALL
```

[2021-11-05] Run 11.1: peak LR down to 7.5e-5

- Note: This was actually run in the same train.log file as 11.0
- Here we capped the warmup at 1000 steps, using a peak LR of 7.5e-5

[2021-11-05] Run 11.0: LETS GO

- 2M bsz
- FP32 Adam

- Tensor parallel (8x MP)
- New data - from experiment 29
- Learned positional embeddings with sinusoidal init
- Weight decay of 0.05
- LR of 3e-4, end LR of 1e-5
- No dropout on embeddings
- Normformer (impact on grad norm is making earlier layers be more similar with later layers)
- Gradient pre-divide factor: 32 (Naman has been running with this)
- Clip (l2 norm): 2.5
- Fairseq commit: 52ac2df400bd3f42301438217151826b0853c43c

Log path:

```
/shared/home/namangoyal/checkpoints/175B/175B_run11.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.
emb12288.lrnsin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl2.5.lr7.5e-05.endlr7.5e-06.wu100
0.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/train.log
```

```
INCLUDED_HOSTS=node-[1-5,7-34,36-87,89-127] python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p 175B_run11 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/
After launch (slurm job id 1264):
sudo scontrol update job=1264 TimeLimit=UNLIMITED
sudo scontrol update job=1264 MailUser=<scrubbed> MailType=ALL
```

Nov 6, 2021 Loss exploded between 1K and 1.25K steps. Decided to roll back to 1K steps and set peak LR to 7.5e-5: 76ae8349c0180daf90174c05d88ba7b6075cde51

```
cd
/data/users/myleott/175B_run11.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrnsin.nffc.nfat
t.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl2.5.lr0.0003.endlr
1e-05.wu4000.dr0.1.atdr0.1.0emb_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992

cp --recursive --include-pattern "checkpoint_1_1000-*.pt" <<<SCRUBBED FOR RELEASE>>>

RESTORE_FILE=/data/users/myleott/175B_run11.me_fp16.fsdp.gpf32.0.gelu.transformer_lm_megatron.nlay96.emb12288.lrn
sin.nffc.nfatt.nfhd.bm_none.tps2048.gpt2.adam.b2_0.98.eps1e-08.cl2.5.lr0.0003.endlr1e-05.wu4000.dr0.1.atdr0.1.0em
b_dr.wd0.05.ms8.uf1.mu143052.s1.ngpu992/checkpoint_1_1000/checkpoint_1_1000.pt

# Note that the new save_dir is different, because we adjusted the peak LR.
# I manually copied the contents of the old checkpoint dir (tb, train.log)
# to the new save_dir for continuity

INCLUDED_HOSTS=node-[1-60,62-87,89-119,121-127] python -m fb_sweep.opt.sweep_opt_en_lm_175b \
    -n 124 -g 8 -t 1 \
    -p 175B_run11 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --restore-file $RESTORE_FILE \
    --resume-failed --dry-run

After launch:
sudo scontrol update job=1284 TimeLimit=UNLIMITED
sudo scontrol update job=1284 MailUser=<scrubbed> MailType=ALL
```

Run 10

- Try to make run 9 identical to run 8, but with tensor parallelism
- 4M bsz
- FP16 Adam
- Tensor parallel (8x MP)
- --gradient-predivide-factor 11.1
- Fairseq commit: gshard-175b-run10

Log path:
/shared/home/namangoyal/checkpoints/175B/175B_run10.me_fp16.fsdp.gelu.transformer_lm_megatron.nlay96.emb12288.bm_none.tps2048.adam.fp16adam.b2_0.98.eps1e-08.cl1.0.lr6e-05.wu290.dr0.1.atdr0.1.ms16.uf1.0.mu73832.s1.wd0.1.ngpu992/train.log

```
INCLUDED_HOSTS=node-[1-7,9-88,90-108,110-127] python -m fb_sweep.opt.sweep_opt_en_lm \
    -n 124 -g 8 -t 1 \
    --weight-decay 0.1 --gradient-predivide-factor 11.1 \
    --model-parallel-size 8 --distribute-checkpointed-activations --batch-size 16 \
    -p 175B_run10 --model-size 175B_opt_h2_2021 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/
```

Outcome: OOM during validation after 300 steps

Run 9

- Similar to run 8, except:
- Tensor parallel (8x MP)
- 2M bsz
- FP32 Adam
- Fairseq commit: gshard-175b-run9

Log path:
/shared/home/namangoyal/checkpoints/175B/175B_run9.me_fp16.fsdp.gelu.transformer_lm_megatron.nlay96.emb12288.bm_none.tps2048.adam.b2_0.98.eps1e-08.cl1.0.lr6e-05.wu580.dr0.1.atdr0.1.ms8.uf1.0.mu147665.s1.wd0.1.ngpu992/train.log

```
INCLUDED_HOSTS=node-[1-7,9-74,76-78,80-127] python -m fb_sweep.opt.sweep_opt_en_lm \
    -n 124 -g 8 -t 1 \
    --weight-decay 0.1 --fp32-adam \
    --model-parallel-size 8 --distribute-checkpointed-activations --batch-size 8 \
    -p 175B_run9 --model-size 175B_opt_h2_2021 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/
```

Outcome: loss exploded

Run 8

- Revert to Run 5 config, but with use-sharded-state
- Fairseq commit: gshard-175b-run8

Log path:
/shared/home/namangoyal/checkpoints/175B/175B_run8.fsdp.me_fp16.transformer_lm_gpt.nlay96.emb12288.bm_none.tps2048.adam.fp16adam.b2_0.98.eps1e-08.cl1.0.lr6e-05.wu290.dr0.1.atdr0.1.wd0.1.ms2.uf1.mu73832.s1.ngpu992/train.log

```
INCLUDED_HOSTS=node-[1-7,9-74,76-78,80-127] python -m fb_sweep.opt.sweep_opt_en_lm \
    -n 124 -g 8 -t 1 \
    -p 175B_run8 --model-size 175B_opt_h2_2021 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/
```

Outcome: Good! See workplace post.

Run 7

- Revert to Run 5 config, but with tensor parallelism
    - One difference: run 5 had a 4M bsz, but here we are using 2M bsz
- Clipping 1.0
- Weight decay 0.1
- Longer warmup (290 steps)
- This also hardcodes skip_remainder_batch=True for Trainer.get_valid_iterator
- Fairseq commit: gshard-175b-run7
- Fairscale commit: 3584965cd4356c3c522e7d97aa13994cfa95ea5b

Log path:
/shared/home/namangoyal/checkpoints/175B/run7.me_fp16.fsdp.gelu.transformer_lm_megatron.nlay96.emb12288.bm_none.tps2048.adam.b2_0.98.eps1e-08.cl1.0.lr6e-05.wu290.dr0.1.atdr0.1.ms8.uf1.mu73832.s1.ngpu992/train.log

```
INCLUDED_HOSTS=node-[1-7,9-74,76-78,80-127] python -m fb_sweep.opt.sweep_opt_en_lm \
    -n 124 -g 8 -t 1 \
    -p run7 --model-size 175B_opt_h2_2021 --update-freq 1 \
    --batch-size-per-gpu 8 --model-parallel-size 8   --dropout 0.1 \
    --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --distribute-checkpointed-activations --fp32-adam \
    --save-interval 300 --validate-interval 300 --gradient-predivide-factor 11.1 --dry-run
```

Outcome: seems to get stuck at 10.7 loss again, seems tensor parallel isn’t fixed

2021-10-22: Run 6

3ad9e48cc started Oct 22 1:01 AM

- Tensor parallelism
- Clipping 1.0
- Weight decay 0.01
- 1x warmup
- Adam Beta2 0.95
- Adam Eps 1e-6

Log Path:

/shared/home/namangoyal/checkpoints/175B/run6.me_fp16.fsdp.gelu.transformer_lm_megatron.nlay96.emb12288.bm_none.tps2048.adam.b2_0.95.eps1e-06.cl1.0.lr6e-05.wu93.dr0.1.atdr0.1.ms8.uf1.mu147666.s1.wd0.01.ngpu992//train.log

Job 378

- Back to tensor parallel with cpu weight init (which seems to improve things, at least in my env).
- weight-decay .01 because all the evidence I have points against increasing.
- A little more save-interval, validate-interval to get more signal overnight.

```
INCLUDED_HOSTS=node-[1-7,9-74,76-78,80-127] python -m fb_sweep.opt.sweep_opt_en_lm  -n 124 -g 8 -t 1 \
    -p run6 --model-size 175B_opt_h2_2021 --update-freq 1 \
    --batch-size-per-gpu 8   --model-parallel-size 8   --dropout 0.1 \
    --max-update 147666 --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ \
    --distribute-checkpointed-activations --fp32-adam --save-interval 500  --validate-interval 500
    --gradient-predivide-factor 11.1 \
    --weight-decay .01
```

Outcome: seems to get stuck at 10.7 loss again!

Run 5

- Similar to Run4, but with some extra safety knobs
- Clipping 1.0
- Weight decay 0.1
- 3x longer warmup
- Fairseq commit: d70abfba80ded958cde92af62fc505bbb99ea170
- **New log path:** /shared/home/namangoyal/checkpoints/175B/175B_run5.fsdp.me_fp16.transformer_lm_gpt.nlay96.emb12288.bm_none.tps2048.adam.fp16adam.b2_0.98.eps1e-08.cl1.0.lr6e-05.wu290.dr0.1.atdr0.1.wd0.1.ms2.uf1.mu73832.s1.ngpu992/train.log
- Relaunched with: INCLUDED_HOSTS=node-[1-7,9-74,76-78,80-127] python -m fb_sweep.opt.sweep_opt_en_lm -n 124 -g 8 -t 1 -p 175B_run5 --model-size 175B_opt_h2_2021 --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/

Outcome:

Loss went down for 300 steps, validation ran.
Htop & nvidia-smi looked fine on a random node 2 hours after the last log.
Started stalling Oct 21, 2021 20:40 -> Oct 21, 2021 23:34 (killed by Sam)
Update by Myle, 10/22 @ 8am: it seems I forgot to add use-sharded-state, so it was trying to consolidate state on a rank 0! PR with fix pushed here: #2490

Run 4

- Revert to Run1 config on gshard stable with zero3
- Fairseq commit: 53d993880508f1ea3272b406e8ecc99298305c7b
- Fairscale commit: 8acbec718f3c70a6b9785470bb9e05cd84fc3f8e
- **New log path:** /shared/home/namangoyal/checkpoints/175B/175B_run4.fsdp.me_fp16.transformer_lm_gpt.nlay96.em

b12288.bm_none.tps2048.adam.fp16adam.b2_0.98.eps1e-08.cl0.0.lr6e-05.wu96.dr0.1.atdr0.1.wd0.01.ms2.uf1.mu73832.s1.ngpu992/train.log
- Relaunched with: INCLUDED_HOSTS=node-[1-7,9-74,76-78,80-127] python -m fb_sweep.opt.sweep_opt_en_lm -n 124 -g 8 -t 1 -p 175B_run4 --model-size 175B_opt_h2_2021 --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/

Outcome: Loss exploding again

Run 3

- Clipping 1.0
- Adam beta2 0.95
- Adam eps 1e-6
- ~~--fp32-reduce-scatter~~
- New fairseq commit hash: 6b629aec74a917f4fbaf3b208f261aa4fc375c84
- This was buggy (don’t use): ~~Also changed fairscale a08a523f6b2fb401f1e12522af9160673fe41e32~~
- **New log path:** /shared/home/namangoyal/checkpoints/175B/175B_third_attempt.2.me_fp16.fsdp.gelu.transformer_lm_megatron.nlay96.emb12288.fp32reduce.bm_none.tps2048.adam.b2_0.95.eps1e-06.cl1.0.lr6e-05.wu193.dr0.1.atdr0.1.wd0.1.ms8.uf1.mu147666.s1.ngpu992/train.log
- Relaunched with: INCLUDED_HOSTS=node-[1-7,9-74,76-78,80-127] python -m fb_sweep.opt.sweep_opt_en_lm   -n 124 -g 8 -t 1   -p 175B_third_attempt --model-size 175B_opt_h2_2021   --update-freq 1   --batch-size-per-gpu 8   --model-parallel-size 8   --dropout 0.1 --max-update 147666   --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/ --distribute-checkpointed-activations --fp32-adam --tensor-parallel-init-model-on-gpu --save-interval 300 --validate-interval 300 --gradient-predivide-factor 11.1 --warmup-updates 193

Outcome: Sam discovered that model parallel branch doesn’t converge even for 125M model

[Undated] Run 2

- Increase weight decay to 0.1
- Log path: /shared/home/namangoyal/checkpoints/175B/175B_first_attempt.me_fp16.fsdp.gelu.transformer_lm_megatron.nlay96.emb12288.bm_none.tps2048.adam.b2_0.98.eps1e-08.cl0.0.lr6e-05.wu193.dr0.1.atdr0.1.wd0.1.ms8.uf1.mu147666.s1.ngpu992/train.log

Outcome: loss plateaus, fails to go below 10.7

2021-10-20: Run 1

20th October

Fairseq: 468a050d4996a4f18c99519c0cfc2d9512f5ab6f
Fairscale: 3584965cd4356c3c522e7d97aa13994cfa95ea5b
Megatron submodule: b6a6ed16ae0a6ff5a57089f66f13a617e1390d1f

Conda env on cloud:

```
Cmd: python -m fb_sweep.opt.sweep_opt_en_lm     -n 124 -g 8 -t 1     -p 175B_first_attempt --model-size
175B_opt_h2_2021     --update-freq 1     --batch-size-per-gpu 8   --model-parallel-size 8   --dropout 0.1
--max-update 147666       --checkpoints-dir /shared/home/namangoyal/checkpoints/175B/
--distribute-checkpointed-activations --fp32-adam  --salloc  --tensor-parallel-init-model-on-gpu  --save-interval
300  --validate-interval 300 --gradient-predivide-factor 11.1 --warmup-updates 193
```

Nodes: node-[1-7,9-74,76-78,80-127]
log:
/shared/home/namangoyal/checkpoints/175B/175B_first_attempt.me_fp16.fsdp.gelu.transformer_lm_megatron.nlay96.emb12288.bm_none.tps2048.adam.b2_0.98.eps1e-08.cl0.0.lr6e-05.wu193.dr0.1.atdr0.1.wd0.01.ms8.uf1.mu147666.s1.ngpu992/train.log

Outcome: Loss kinda exploded. Trying weight decay next

Kitchen sink: Analysis of Exp 21--29

This keeps record of some of our kitchen sink findings.

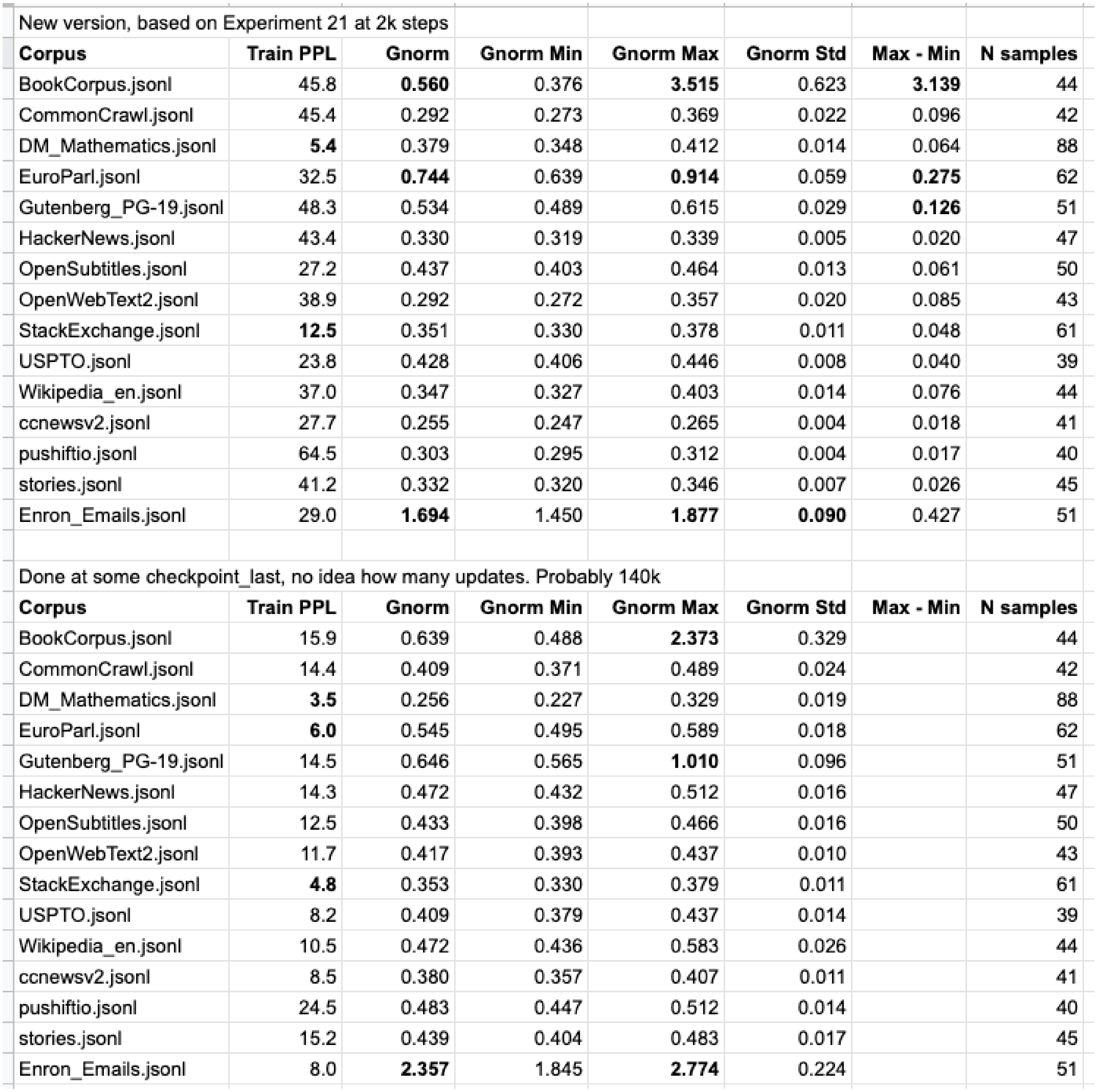
Description of experiments:

- 21: New data after fixing encoding issues and collapsing newlines
- 22: Old data + pushift.io (control)
- 23: Subset of new data closest to the “classic” roberta data. (BookCorpus + CC + OWT2 + wiki + ccnews2 + pushift.io + stories)
- 24: Ablate out BookCorpus from 23. So new versions of (CC + OWT2 + wiki + ccnews2 + pushift.io + stories)
- 25: Safer version of all new data (cc + pg19 + hn + OST + OWT2 + USPTO + wiki + ccnews + pushift.io + stories)
- 26: Add in the FAIR version of BookCorpus from 23. So new versions of (BookCorpusFair + CC + OWT2 + wiki + ccnews2 + pushift.io + stories)
- 27: Safer versions of all new data (BookCorpusFair + cc + pg19 + hn + OST + OWT2 + USPTO + wiki + ccnews + pushift.io + stories)
- 28: learned positional embeddings vs sinusoidal (with caveats)
- 29: Manually cleaned version of corpora via handcrafted regexes

Exp 21 -- 23

Wanted to see what was the “dangerous” corpora in 21. Performed this experiment and found BookCorpus, EuroParl, and Pg19, Enron, and EuroParl all look iffy for gnorms

DM and SX look kinda sketchy for PPL reasons



New version, based on Experiment 21 at 2k steps

Corpus	Train PPL	Gnorm	Gnorm Min	Gnorm Max	Gnorm Std	Max - Min	N samples
BookCorpus.jsonl	45.8	**0.560**	0.376	**3.515**	0.623	**3.139**	44
CommonCrawl.jsonl	45.4	0.292	0.273	0.369	0.022	0.096	42
DM_Mathematics.jsonl	**5.4**	0.379	0.348	0.412	0.014	0.064	88
EuroParl.jsonl	32.5	**0.744**	0.639	**0.914**	0.059	**0.275**	62
Gutenberg_PG-19.jsonl	48.3	0.534	0.489	0.615	0.029	**0.126**	51
HackerNews.jsonl	43.4	0.330	0.319	0.339	0.005	0.020	47
OpenSubtitles.jsonl	27.2	0.437	0.403	0.464	0.013	0.061	50
OpenWebText2.jsonl	38.9	0.292	0.272	0.357	0.020	0.085	43
StackExchange.jsonl	**12.5**	0.351	0.330	0.378	0.011	0.048	61
USPTO.jsonl	23.8	0.428	0.406	0.446	0.008	0.040	39
Wikipedia_en.jsonl	37.0	0.347	0.327	0.403	0.014	0.076	44
ccnewsv2.jsonl	27.7	0.255	0.247	0.265	0.004	0.018	41
pushiftio.jsonl	64.5	0.303	0.295	0.312	0.004	0.017	40
stories.jsonl	41.2	0.332	0.320	0.346	0.007	0.026	45
Enron_Emails.jsonl	29.0	**1.694**	1.450	**1.877**	**0.090**	0.427	51

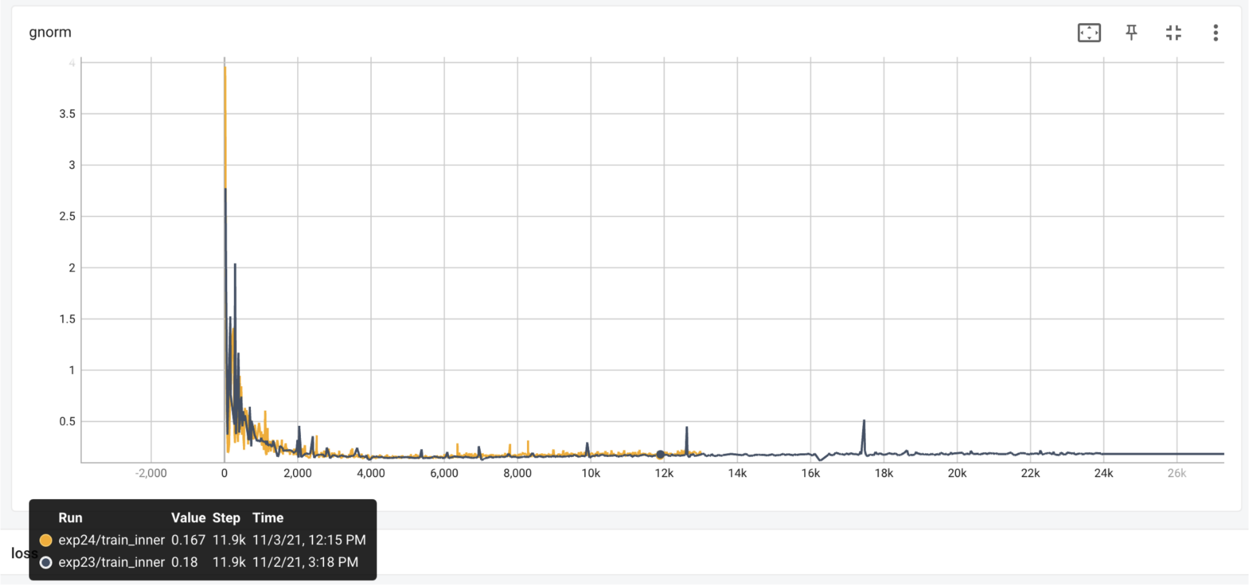
Done at some checkpoint_last, no idea how many updates. Probably 140k

Corpus	Train PPL	Gnorm	Gnorm Min	Gnorm Max	Gnorm Std	Max - Min	N samples
BookCorpus.jsonl	15.9	0.639	0.488	**2.373**	0.329		44
CommonCrawl.jsonl	14.4	0.409	0.371	0.489	0.024		42
DM_Mathematics.jsonl	**3.5**	0.256	0.227	0.329	0.019		88
EuroParl.jsonl	**6.0**	0.545	0.495	0.589	0.018		62
Gutenberg_PG-19.jsonl	14.5	0.646	0.565	**1.010**	0.096		51
HackerNews.jsonl	14.3	0.472	0.432	0.512	0.016		47
OpenSubtitles.jsonl	12.5	0.433	0.398	0.466	0.016		50
OpenWebText2.jsonl	11.7	0.417	0.393	0.437	0.010		43
StackExchange.jsonl	**4.8**	0.353	0.330	0.379	0.011		61
USPTO.jsonl	8.2	0.409	0.379	0.437	0.014		39
Wikipedia_en.jsonl	10.5	0.472	0.436	0.583	0.026		44
ccnewsv2.jsonl	8.5	0.380	0.357	0.407	0.011		41
pushiftio.jsonl	24.5	0.483	0.447	0.512	0.014		40
stories.jsonl	15.2	0.439	0.404	0.483	0.017		45
Enron_Emails.jsonl	8.0	**2.357**	1.845	**2.774**	0.224		51

Exp 23, 24 and 25 (Drop out BookCorpus)

Our predominant goal here is to consider whether the “safe” version of BookCorpus is really safer.

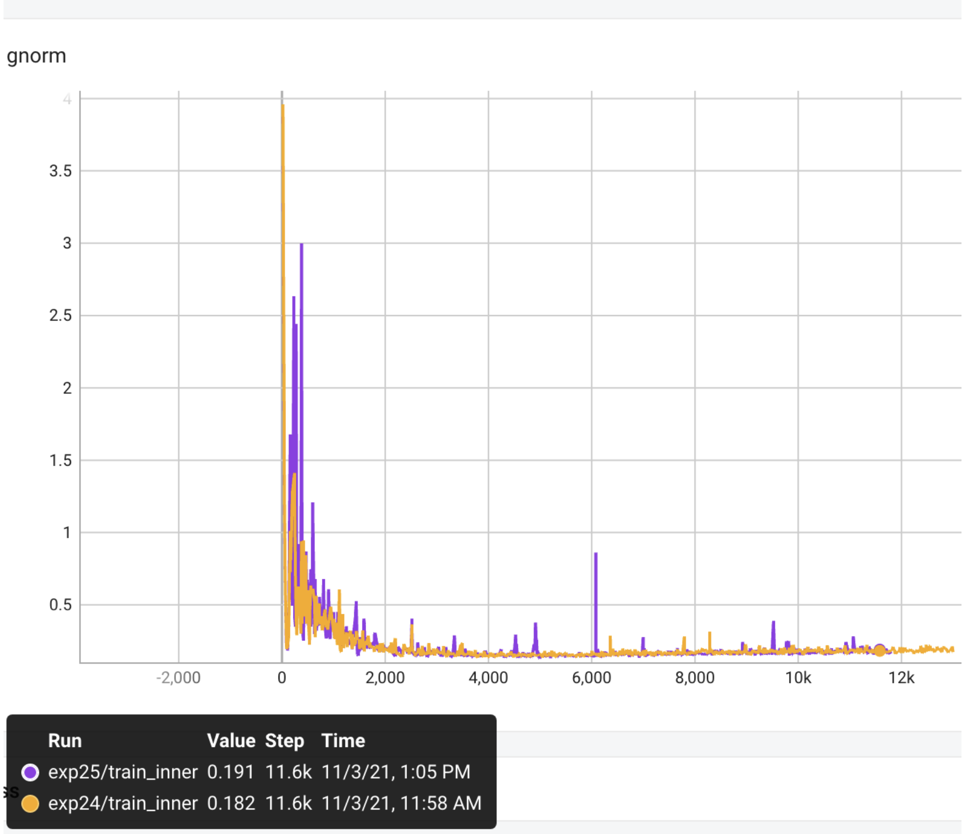
To that end, our main comparison is 23 vs 24.



Run	Value	Step	Time
● exp24/train_inner	0.167	11.9k	11/3/21, 12:15 PM
○ exp23/train_inner	0.18	11.9k	11/2/21, 3:18 PM

In our judgment, 24 was a little bit less spiky than 23, indicating that we feel comfortable that the main problem with our “new data” is the BookCorpus coming from the Pile.

We can also compare 24 vs 25 to see if we managed to exclude the dangerous corpora:



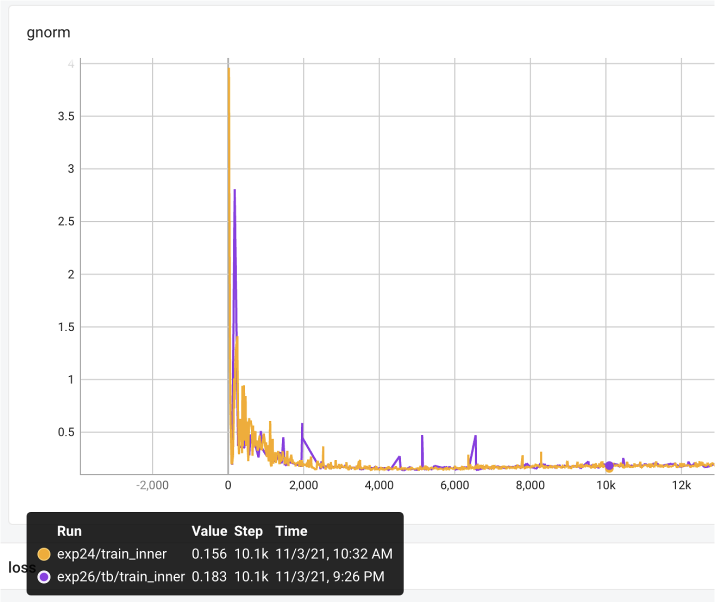
Run	Value	Step	Time
● exp25/train_inner	0.191	11.6k	11/3/21, 1:05 PM
● exp24/train_inner	0.182	11.6k	11/3/21, 11:58 AM

In general, 24 looks fairly smooth (after you consider this is log interval 1), and 25 still has some issues.

At this point, we killed them early and grabbed a copy of BookCorpus from the old cluster.

Exp 26 and 27: Adding in Book Corpus

On stability via only the new books corpus (24 vs 26)



Run	Value	Step	Time
exp24/train_inner	0.156	10.1k	11/3/21, 10:32 AM
exp26/tb/train_inner	0.183	10.1k	11/3/21, 9:26 PM

Overall they seem about the same if one were to average 26 at a smaller log interval.

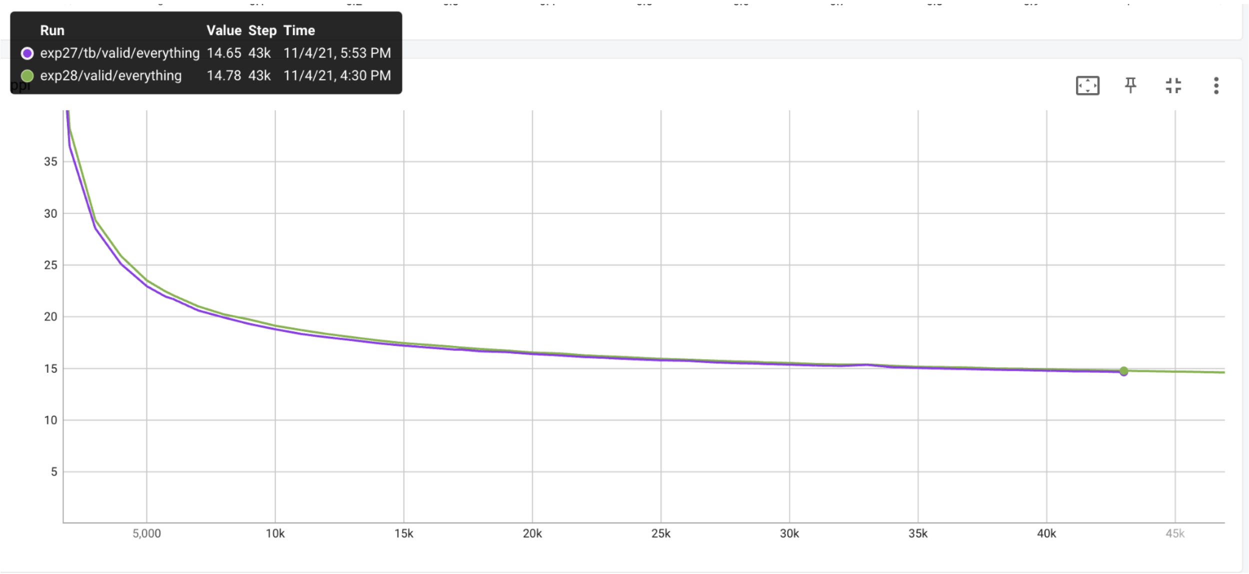
Exp 29: Manually cleaned up corpora

Learned Embeddings

Exp 27 vs 28

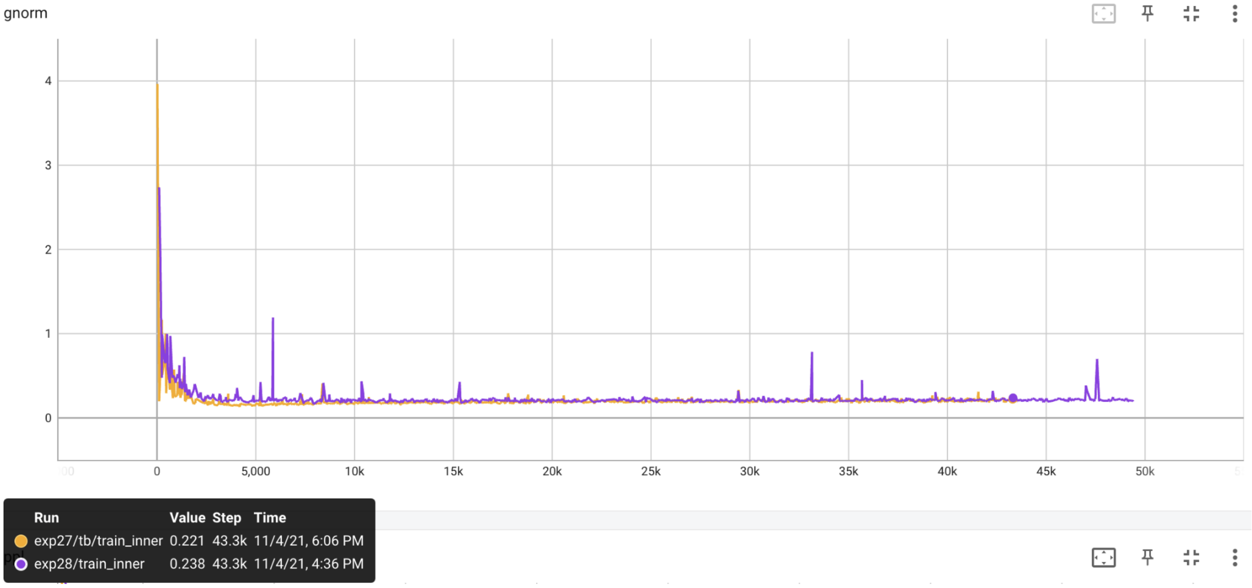
These two runs use the same dataset, and predominantly differ choices for positional embeddings. In particular, 27 uses sinusoidal embeddings, while 28 uses LPE + some tricks around initialization and scaling. See the PR for 28 for an exact description.

They are neck to neck, but 27 does come out on top after 45k updates.



Run	Value	Step	Time
exp27/tb/valid/everything	14.65	43k	11/4/21, 5:53 PM
exp28/valid/everything	14.78	43k	11/4/21, 4:30 PM

We also observe that the gnorms of 28 are a bit spikier:



Run	Value	Step	Time
exp27/tb/train_inner	0.221	43.3k	11/4/21, 6:06 PM
exp28/train_inner	0.238	43.3k	11/4/21, 4:36 PM

Oncall Debugging

AKA: Help! I'm oncall, it's 3am, and everything is on fire!

Philosophy

When you are on call, you are ON CALL. You are fully expected to fulfill the responsibilities within a reasonable time frame.

But when you are not on call, you are NOT on call. Try to leave monitoring and resolution to the oncall as much as you can. It both prevents burnout, and the oncall from growing lazy.

If you have conflicts or cannot fulfill your shift, that's okay! Just make sure to arrange a trade with someone.

Remember to document your actions. This helps people understand what you did without relying on pinging you.

Responsibilities

1. Regularly monitor jobs (by checking tensorboard and tailed logs) for aberrant behavior.
   a. Run `./scripts/cloud/monitor_train_log.py --mailto <scrubbed>@fb.com --slurm-jobid JOBID --modified-threshold 900` to get emails about aberrations (the email listed will send to the oncall) and enable auto-recovery.
   b. See links at the top of the document for the latest tensorboard
2. Perform cluster maintenance in the case of hardware failures
   a. Most of this is done via fixmycloud and the auto-recovery monitor now.
   b. Identify the broken node
   c. Relaunch the slurm job with it replaced using one of the spares
   d. Drain the bad node, note it in the Spare Node Tracker table.
   e. Potentially replace the node.
3. Keep nodes from being idle and ensure that *something* productive is always running.
   a. Preferably the main job.
   b. If it looks unrecoverable, execute any pre-documented Plan B choices
   c. Bias for action. Make discretionary decisions if other members are not available.

Onboarding & Gotchas

These are the instructions for getting onboarded into the cluster to fulfill your oncall duties.

- Getting onto Cloud and setting up environment:
  - < link to cloud cluster instructions doc >
  - Begin following this including the “getting on” and “setup environment”
  - You can have a tmux session on cloud, but Stephen usually just keeps tmux on his devfair and ssh’s in from the devfair.
  - Once you’re onboarded, make sure you have the ability to run “sudo”. You will need it, a lot.
  - `git clone git@github.com:facebookresearch/fairscale` and do install instructions
    - Make sure to be on the right checkpoints for `fairscale` and `fairseq-py` (Look at the most recent copy/paste of run+install instructions in the oncall log; scripts won’t exist otherwise)
  - Also make sure to once run “python3 scripts/cloud/ssh.py” Which will ensure your ssh is configured to disregard host keys, as our nodes are re-imaged and change keys frequently
- **Gotcha**: Avoid doing anything compute-heavy on the login node. It is a very light machine and a shared resource. Heavy operations on it can slow down EVERYONE.
- **Gotcha**: most of our oncall tooling expects to be run from the login node.
  - However, sometimes it’s useful to directly log into nodes when trying to debug a failure. Just “ssh node-X”

Questions

- Is it safe to assume the checkpoint last used in the log is the right one?
  - We are constantly writing checkpoints to blob storage. Checkpoints are not written to any sort of shared disk (like /shared/home)

  - (We actually write them to local SSD on every worker, and then initiate a separate upload on every worker, which makes it very fast).
  - When launched without `--restore-file`, our code will automatically fetch the latest checkpoint from blob and resume from that.
  - But in the case of things like loss exploding, or more radical changes, you may need to manually provide a `--restore-file` in order to resume from that checkpoint.
  - See entry “2021-12-06 8:30am ET: Lowering LR and launching 12.46” for the latest example of doing that.
- Where will I find log files that I need to tail?
  - /shared/home/namangoyal/checkpoints/175B/ contains many folders. Each folder has its “12.46” or “12.37” folder or what not.
- What is this BLOB_PREFIX and BLOB_AUTH stuff?
  - BLOB_AUTH contains authentication variables so that Cloud’s blob storage knows what account we are using. It generally will never change.
  - BLOB_PREFIX is roughly the bucket/folder that we are uploading to. Due to limitations on the Cloud side, sometimes we need to switch to new buckets.
    - The BLOB_PREFIX also contains a “run id” in it generally speaking. We only bump this when we do something that could potentially clobber checkpoints. **Therefore you will often find yourself using BLOB PREFIXES with RunIDs that are not 1:1 with the run you are launching.**

Run is stuck in loop of “lower loss scale”

Uh oh. The loss exploded. We don’t know why.

Remember to document your actions in the logbook.

Actions:

1. **Don’t panic.**
2. Ping the group chat to discuss potential options. If no one responds within 10 minutes, you will have to make a decision by yourself. Letting nodes idle costs $2500/hour so it is strongly discouraged.
   a. Bias for action, but also double check your commands to avoid catastrophic losses (clobbering checkpoints, etc). As expensive as it is, an extra idle hour is still cheaper than having to redo multiple days.
      i. It’s generally a good idea to bump your blob RUNID variables. Search this document for OLD_BLOB_PREFIX for the most recent examples of that.
3. Go read the logs and check the tensorboard.
   a. What happened to Gnorm in the updates preceding the explosion? Did we see a large gnorm spike? (~0.7 or above)?
   b. What about actv norm, pnorms, loss, etc? Did those spike too?
   c. What was the loss scale doing during the moments before? Did it drop very rapidly (20-50 updates)? Or was it a slow gradual fall out?
4. Go read the log book. Was there a plan B put in place already? If so, execute it. If not, you will have to make some decisions yourself.
5. Actions you might take:
   a. Let it keep running another hour or so. Sometimes we recover. This is unlikely if you are seeing 40+ loss scale flatlines.

    b. Requeue. This is a generally safe option, *unless the logbook tells you otherwise*. This will restart the job from the last checkpoint and just pray for the best. Frequently this just kicks the can down the road 2-6 more hours, but that can buy you some time to sleep and discuss.
    c. More extreme measures are then suitable. Preferably these should be group discussion, but that's not always possible.
    d. Taking extreme measures should always cause you to bump the run ID and the BLOB_URL.
    e. Prior actions we have taken as examples:
        i. Lowering the LR, usually by some factor (0.9x or 0.75x).
            1. This is our current default mode of action.
            2. As of Dec 7, this seems to buy us a couple days.
        ii. Implementing clamping of activations (Not actually tried yet, but a plan B)
        iii. Lowering the clip 1.0 -> 0.3.
        iv. Hot swap the optimizer from Adam to SGD. This has gone poorly and is not currently recommended.

WPS has dropped a lot

As of 2021-12-02, WPS is consistently around ~100k and does not fluctuate more than 1%, except during validation.

If you observe WPS dropping below ~90k or more for a sustained period (>20 minutes), we probably have an infiniband problem.

Actions:

1. Run health checks only on idle nodes. Identify a suitable replacement.
    a. If you cannot, then you should leave the job as is. Better to run slow than not at all.
2. Pause the job
3. Run global health checks. You are particularly looking out for NCCL issues this time.
4. Replace the node and resume the job.

Job is Hanging

Remember to document your actions in the logbook.

Possible Actions:

1. **Do nothing.** Auto recovery in monitor.py may handle this for you. **You should receive an email regarding success/failure.**
2. Read the train.log. Check the timestamp. Remember it's in UTC time (run "date" to see what time the computer thinks it is now)
3. Check the stderr. Is there a mention of a CUDA or NCCL Failure?
    a. A node has probably kicked the bucket. Hopefully the auto-recovery script has kicked in. Check its logs (which should be coming in via email)
    b. If not, pause the job (see below) and follow the Health Checks guidelines.
4. Is there really nothing in the stderr? Let's check on the CPUs, the GPUs, and the Disk to see if there are active bottlenecks we just need to be patient for.
    a. CPU: Nodes may be stuck doing some tokenization. ssh into one and run htop.
        i. If many CPUs are used or spiking, then just be patient. We may be tokenizing.

        ii. If you see ~16 processes pegged at 100% but the other 80 unused, you should continue down this checklist.
    b. GPU: run “watch -n0.5 nvidia-smi”. Watch for about a minute. Are all the GPUs at 100%? Is the temperature 40C or 72C? Is the power usage ~70W or or ~385W?
        i. Hotter and higher power mean the GPU is active. Let it be.
        ii. Cooler and low power means we are hanging on communication, and probably have a bad node.
    c. Disk: Run “sudo iotop” and watch.
        i. Is the top process sitting at just 0mb/s read/write? We are not disk bottlenecked.
        ii. Is it very high? We might be doing I/O or writing a checkpoint.

5. Full health checks are called for.
    a. Pause or cancel the job
    b. Run fixmycloud all
    c. Replace the bad node(s) with good nodes in the INCLUDED_HOSTS environmental variable.
    d. Relaunch the job

Performing health checks

Use fixmycloud to check quality across the entire cluster

```
# check only idle, unused nodes
python scripts/cloud/fixmycloud.py idle
# check all nodes. Should only be done when a job is not running
python scripts/cloud/fixmycloud.py all
```

It will output information about potential warnings and possibly suggested mitigations.

You may wish to run partial checks, which can be done by calling individual scripts rather than the fixmycloud script. The health checks include:

- **nvidia-smi checks**. These do not interfere with running nodes, and are safe to run on active nodes.
- **gpu-burn checks**. These DO interfere with running nodes, and should not be used
- **nccl checks**. These require the node be undrained and idle, and require you to check 3+ nodes at once.
- **blocklist checks**. These check if a node is a known bad host, and are safe to run on active nodes.

I need to reprovision/replace a node

We no longer replace nodes ourselves, and instead CSP does it for us. As of 2022-01-13, the fixmycloud tool should upload a diagnostics dump to a public Blob bucket and CSP is responsible for monitoring this bucket actively.

Before you reprovision a node it’s a good idea to get the metadata and hardware ID for it because you need to report it to CSP as a bad node. To get the metadata:

1. ssh to the node you plan to reprovision. Some notes about hostnames:
    ○ Each node has multiple hostnames, any of which can be used with ssh:
        ■ **node-XXX**: this is the alias used by Slurm
        ■ **ip-XXXXXX**: this is the alias used by Cloud
        ■ **10.30.4.XXX**: this is the internal IP address of the node

        - **buoXXXX**: this is another alias used by Cloud
    - You can resolve all of these to node-XXX format using the “scripts/cloud/find_host.py” script.
2. Run the scripts/cloud/gather_diagnostics.py script and save the blob URL that is printed and send to CSP.

Using the command line
Warning: You probably don’t want this unless you really know what you’re doing. CSP is supposed to be handling this for us.

```
# reprovision node node-2
# warning: you probably don't want to do this on idle or utilized nodes
python scripts/cloud/replace_node.py node-2
```

Make sure you update the “Spare Node Tracker” to indicate it is being replaced and update the timestamp.

I need to mark a node as unusable

Draining nodes is useful when you want to mark them as unusable by SLURM. You can then take other mitigation actions (like rebooting a node, or reprovisioning it). As of 2021-12-02, nodes with serious errors (marked as error/fatal/critical in fixmycloud) should be drained and reported to CSP via chat using the diagnostics script.

```
# drain node node-2, marking it unusable
python scripts/cloud/slurm.py drain node-2 --reason=”Why I am draining”
```

You may also **undrain** a node similarly with “slurm.py undrain [node]”

Rebooting a node

```
$ ssh node-X
$ sudo reboot
# [ you will be logged out ]
$ ping -O node-X
# [ wait until you start getting "no answer yet"]
# [ start monitoring on cloud UI]
# [ wait until pings return ]
# [ then wait an extra minute ]
$ ssh node-X # assert you can get back in. do a health check
```

If it takes longer than say, 20 minutes, you should probably switch to replacing the node (see above).

I want to hot swap a node in a running job without explicitly re-launching

Note: Monitor script should now do this automatically for you.

This is useful if you want to correct for a hardware failure without changing any hyperparameters, and just resume training.

Figure out the job ID using squeue. The main job is going to be the one with 124 nodes. Let's assume the job ID is 2589.

```
# these two commands together pause the job
sudo scontrol requeue job=2589
sudo scontrol hold job=2589
# find whatever node(s) caused the issue and drain it
python scripts/cloud/slurm.py drain node-25
# OPTIONAL: you can explicitly tell SLURM which nodes to use
# However, if you are only interested in using any idle node, you can skip
sudo scontrol update job=2589 NodeList=node-[1-2,4-24,27-36,38-52,54-72,74-107,109-113,115-131,147]
# allow it to resume
sudo scontrol release job=2589
# note the job will be in a "BeginTime" held state for 1 minute. This is done intentionally to allow for cleanups on shutdown.
```

Table of Contents/Index