CERN IT GPUs Update

Ricardo Rocha

Compute Accelerator Forum, Feb 10th 2021
https://indico.cern.ch/event/975007/

Reminder

https://clouddocs.web.cern.ch/gpu/README.html

GPU availability on virtual machines, batch, kubernetes clusters, ...

Working on integrating them with all relevant higher level services

Request for GPUs: *GPU Platform Consultancy Functional Element*

https://cern.service-now.com/service-portal?id=functional_element&name=gpu-platform

#GPU channel on IT-dep mattermost

Access to GPU resources

Fill this form for requesting **access to GPU resources**.
If you don't need access but you have another kind of request for the GPU Platform Consultancy, please use **this form** instead.

N.B. it will create a ticket directly into "GPU Platform Consultancy" 2nd level.

Usage pattern expected (spiky if <30% overall usage, full if >80%)

(•) Spiky ( ) Medium ( ) Full

Specific performance requirements for floating point precision

(•) Double ( ) Single ( ) None

Type of interface desired

(•) Notebook ( ) Batch ( ) Kubernetes ( ) VM ( ) Other

CUDA drivers and versions required (custom if you need specific drivers)

(•) Custom ( ) Any

ML framework being used (for machine learning workloads only)

(•) Tensorflow ( ) PyTorch ( ) scikit-learn ( ) Other

Distributed training possible or desired (for machine learning workloads only)

No

* Further details

What's New

Action items from the last update in this forum (Oct 2020)

https://indico.cern.ch/event/950196/

Availability of Virtual GPUs (vGPUs): done

GPUs on Gitlab CI: done (not yet as shared runners)

And a bonus with TPU and ARM availability, more later

GPUs on lxplus: ongoing

Virtual GPUs

https://clouddocs.web.cern.ch/gpu/README.html#virtual-gpus

Offers a fraction of a GPU (time sharing), no physical partitioning as with A100s

Alternative to PCI passthrough where a full GPU is assigned

Use Cases

- Spiky workloads with no strong performance requirements
- Development, testing, validation
- Notebooks, workshops, tutorials, trainings, ...

Virtual GPUs

https://clouddocs.web.cern.ch/gpu/README.html#virtual-gpus

Currently backed by Nvidia T4s (ratio of 4 vGPUs per physical GPU)

Requires a license setup on the node, we provide recipes

- See instructions above for virtual machines
- Finishing automation for Kubernetes clusters

GitLab CI / Public Cloud Runners

https://gitlab.cern.ch/rbritoda/gitlab-runner-public

Experimental setup with runners using public cloud resources

- Relying on resources from the Cloudbank EU (Broker Pilot) project
- Using both GCP and AWS
- Resources available until June 2021, follow up being worked on

Requires explicitly enabling runners in the project

- See docs in the link above

GitLab CI / Public Cloud Runners

https://gitlab.cern.ch/rbritoda/gitlab-runner-public

Cluster(s) auto scaling on demand, cost effective

- Slight delay (<2min) for job start when a fresh node is required

Large numbers and types of resources available

- GPUs: Nvidia A100, V100, P100, K80, P4, T4, M60
- TPUs: v2-8, v3-8, can add larger instances if needed
- ARM: AWS Graviton2 (m6g instances)

Familiar job environment, including CVMFS (immediate request by ATLAS)

```yaml
# Sample for a generic GPU job, which can land in any type of GPU
sample-gpu:
  image: docker.io/nvidia/cuda:11.1.1-base
  script:
    - nvidia-smi
  tags:
    - docker-gpu

# You can also pass a more specific GPU tag, asking for a specific vendor
sample-gpu-nvidia:
  image: docker.io/nvidia/cuda:11.1.1-base
  script:
    - nvidia-smi
  tags:
    - docker-gpu-nvidia

# You can also pass a more specific GPU tag, asking for a specific GPU type
sample-gpu-nvidia-a100:
  image: docker.io/nvidia/cuda:11.1.1-base
  script:
    - nvidia-smi
  tags:
    - docker-gpu-a100
```

Pipeline Needs **Jobs 13** Tests 0

Status	Job ID	Name		Coverage
Test				
passed	#12061346 docker-arm	sample-arm	00:00:05 2 hours ago	
passed	#12061347 docker-arm-graviton2	sample-arm-graviton2	00:00:08 2 hours ago	
passed	#12061338 docker-gpu	sample-gpu	00:00:05 2 hours ago	
passed	#12061339 docker-gpu-nvidia	sample-gpu-nvidia	00:00:15 2 hours ago	
passed	#12061340 docker-gpu-a100	sample-gpu-nvidia-a100	00:00:15 2 hours ago	
passed	#12061345 docker-gpu-k80	sample-gpu-nvidia-k80	00:00:06 2 hours ago	
passed	#12061342 docker-gpu-p100	sample-gpu-nvidia-p100	00:00:05 2 hours ago	
passed	#12061344 docker-gpu-p4	sample-gpu-nvidia-p4	00:00:12 2 hours ago	
passed	#12061343 docker-gpu-t4	sample-gpu-nvidia-t4	00:00:06 2 hours ago	
passed	#12061341 docker-gpu-v100	sample-gpu-nvidia-v100	00:00:12 2 hours ago	
passed	#12061348 docker-tpu	sample-tpu	00:03:03 2 hours ago	
passed	#12061349 docker-tpu-v2-8	sample-tpu-v2-8	00:03:01 2 hours ago	
passed	#12061350 docker-tpu-v3-8	sample-tpu-v3-8	00:02:58 2 hours ago	

```
Waiting for pod gitlab-runner/runner-ka9reaze-project-108952-concurrent-1dgllp to be running, status is Pending
       ContainersNotReady: "containers with unready status: [build helper]"
       ContainersNotReady: "containers with unready status: [build helper]"
Running on runner-ka9reaze-project-108952-concurrent-1dgllp via gitlab-runner-gpu-a100-gitlab-runner-854f7d54b7-z9t9f...
Getting source from Git repository                                                                    00:01
Fetching changes with git depth set to 50...
Initialized empty Git repository in /builds/rbritoda/gitlab-runner-public/.git/
Created fresh repository.
Checking out 828259f8 as load...
Skipping Git submodules setup
Executing "step_script" stage of the job script                                                       00:00
$ nvidia-smi
Tue Feb  9 20:55:13 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 450.51.06    Driver Version: 450.51.06    CUDA Version: 11.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  A100-SXM4-40GB      Off  | 00000000:00:04.0 Off |                    0 |
| N/A   33C    P0    42W / 400W |      0MiB / 40537MiB |      0%      Default |
|                               |                      |             Disabled |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
Cleaning up file based variables                                                                      00:00
Job succeeded
```

```
Waiting for pod gitlab-runner/runner-hdqec7c-project-108952-concurrent-1fbmnv to be running, status is Pending
       ContainersNotReady: "containers with unready status: [build helper]"
       ContainersNotReady: "containers with unready status: [build helper]"
Running on runner-hdqec7c-project-108952-concurrent-1fbmnv via gitlab-runner-tpu-v3-8-gitlab-runner-5d54df77b-wcgq8...
Getting source from Git repository                                                                    00:01
Fetching changes with git depth set to 50...
Initialized empty Git repository in /builds/rbritoda/gitlab-runner-public/.git/
Created fresh repository.
Checking out 828259f8 as load...
Skipping Git submodules setup
Executing "step_script" stage of the job script                                                       00:12
$ ./check_tpu.py
2021-02-09 21:01:12.670321: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-02-09 21:01:12.670363: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-02-09 21:01:12.670393: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (runner-hdqec7c-project-108952-concurrent-1fbmnv): /proc/driver/nvidia/version does not exist
2021-02-09 21:01:12.671344: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 AVX512F FMA
2021-02-09 21:01:12.679627: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2000170000 Hz
2021-02-09 21:01:12.680091: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f4538000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-02-09 21:01:12.680128: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
2021-02-09 21:01:12.687470: I tensorflow/core/distributed_runtime/rpc/grpc_channel.cc:301] Initialize GrpcChannelCache for job worker -> {0 -> 10.116.18.178:8470}
2021-02-09 21:01:12.687498: I tensorflow/core/distributed_runtime/rpc/grpc_channel.cc:301] Initialize GrpcChannelCache for job localhost -> {0 -> localhost:30018}
2021-02-09 21:01:12.701793: I tensorflow/core/distributed_runtime/rpc/grpc_channel.cc:301] Initialize GrpcChannelCache for job worker -> {0 -> 10.116.18.178:8470}
2021-02-09 21:01:12.701830: I tensorflow/core/distributed_runtime/rpc/grpc_channel.cc:301] Initialize GrpcChannelCache for job localhost -> {0 -> localhost:30018}
2021-02-09 21:01:12.702455: I tensorflow/core/distributed_runtime/rpc/grpc_server_lib.cc:390] Started server with target: grpc://localhost:30018
Connecting to TPU at grpc://10.116.18.178:8470
Running on TPU  ['10.116.18.178:8470']
  Timestamp: 21:01:22
  TPU type: TPU v3
  Utilization of TPU Matrix Units (higher is better): 0.000%
Cleaning up file based variables                                                                      00:00
Job succeeded
```

```
Waiting for pod gitlab-runner/runner-hdqec7c-project-108952-concurrent-1fbmnv to be running, status is Pending
       ContainersNotReady: "containers with unready status: [build helper]"
       ContainersNotReady: "containers with unready status: [build helper]"
Running on runner-hdqec7c-project-108952-concurrent-1fbmnv via gitlab-runner-tpu-v3-8-gitlab-runner-5d54df77b-wcgq8...
Getting source from Git repository                                                                    00:01
Fetching changes with git depth set to 50...
Initialized empty Git repository in /builds/rbritoda/gitlab-runner-public/.git/
Created fresh repository.
Checking out 828259f8 as load...
Skipping Git submodules setup
Executing "step_script" stage of the job script                                                       00:12
```

```python
#!/usr/bin/python3
import tensorflow as tf
import os
from tensorflow.python.profiler import profiler_client

endpoint = os.environ['KUBE_GOOGLE_CLOUD_TPU_ENDPOINTS']
print("Connecting to TPU at %s\n" % endpoint)

tpu = tf.distribute.cluster_resolver.TPUClusterResolver()
print('Running on TPU ', tpu.cluster_spec().as_dict()['worker'])

tf.config.experimental_connect_to_cluster(tpu)
tf.tpu.experimental.initialize_tpu_system(tpu)

print(profiler_client.monitor(endpoint.replace('8470', '8466'), 100, 2))
```

```
2021-02-09 21:01:12.702455: I tensorflow/core/distributed_runtime/rpc/grpc_server_lib.cc:390] Started server with target: grpc://localhost:30018
Connecting to TPU at grpc://10.116.18.178:8470
Running on TPU  ['10.116.18.178:8470']
  Timestamp: 21:01:22
  TPU type: TPU v3
  Utilization of TPU Matrix Units (higher is better): 0.000%
Cleaning up file based variables                                                                      00:00
Job succeeded
```

```
Running on runner-yawjs17o-project-108952-concurrent-18ffzd via gitlab-runner-arm-graviton2-gitlab-runner-575dd7ff87-tlpbc...
Getting source from Git repository                                                                    00:00
Fetching changes with git depth set to 50...
Initialized empty Git repository in /builds/rbritoda/gitlab-runner-public/.git/
Created fresh repository.
Checking out 828259f8 as load...
Skipping Git submodules setup
Executing "step_script" stage of the job script                                                       00:00
$ lscpu
Architecture:                    aarch64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
CPU(s):                          2
On-line CPU(s) list:             0,1
Thread(s) per core:              1
Core(s) per socket:              2
Socket(s):                       1
NUMA node(s):                    1
Vendor ID:                       ARM
Model:                           1
Model name:                      Neoverse-N1
Stepping:                        r3p1
BogoMIPS:                        243.75
L1d cache:                       128 KiB
L1i cache:                       128 KiB
L2 cache:                        2 MiB
L3 cache:                        32 MiB
NUMA node0 CPU(s):               0,1
Vulnerability Itlb multihit:     Not affected
Vulnerability L1tf:              Not affected
Vulnerability Mds:               Not affected
Vulnerability Meltdown:          Not affected
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl
Vulnerability Spectre v1:        Mitigation; __user pointer sanitization
Vulnerability Spectre v2:        Not affected
Vulnerability Srbds:             Not affected
Vulnerability Tsx async abort:   Not affected
Flags:                           fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid asimdrdm lrcpc dcpop asimddp ssbs
Cleaning up file based variables                                                                      00:00
Job succeeded
```

GitLab CI / Public Cloud Runners

GPUs

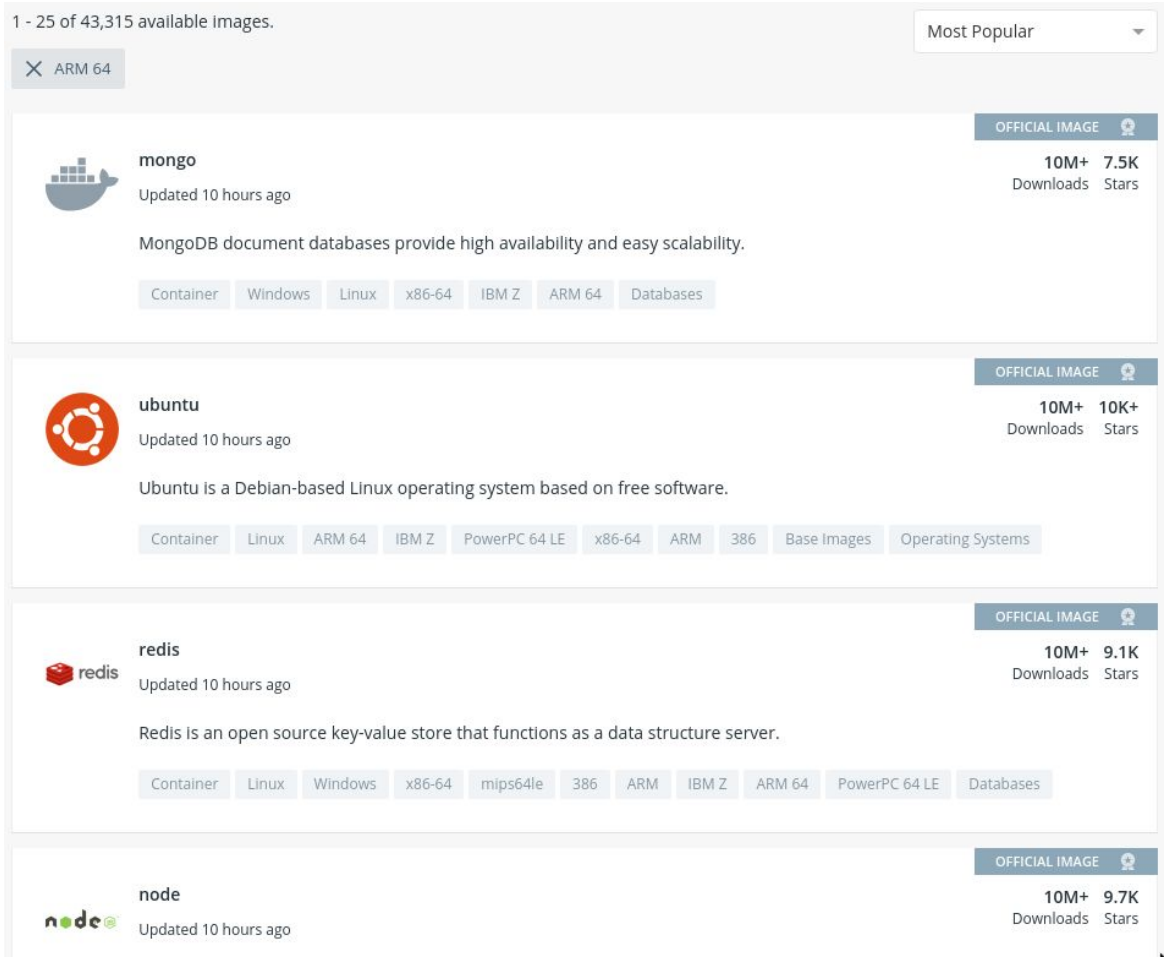
- Jobs get a dedicated GPU node
- Useful also to compare performance between cards

TPUs

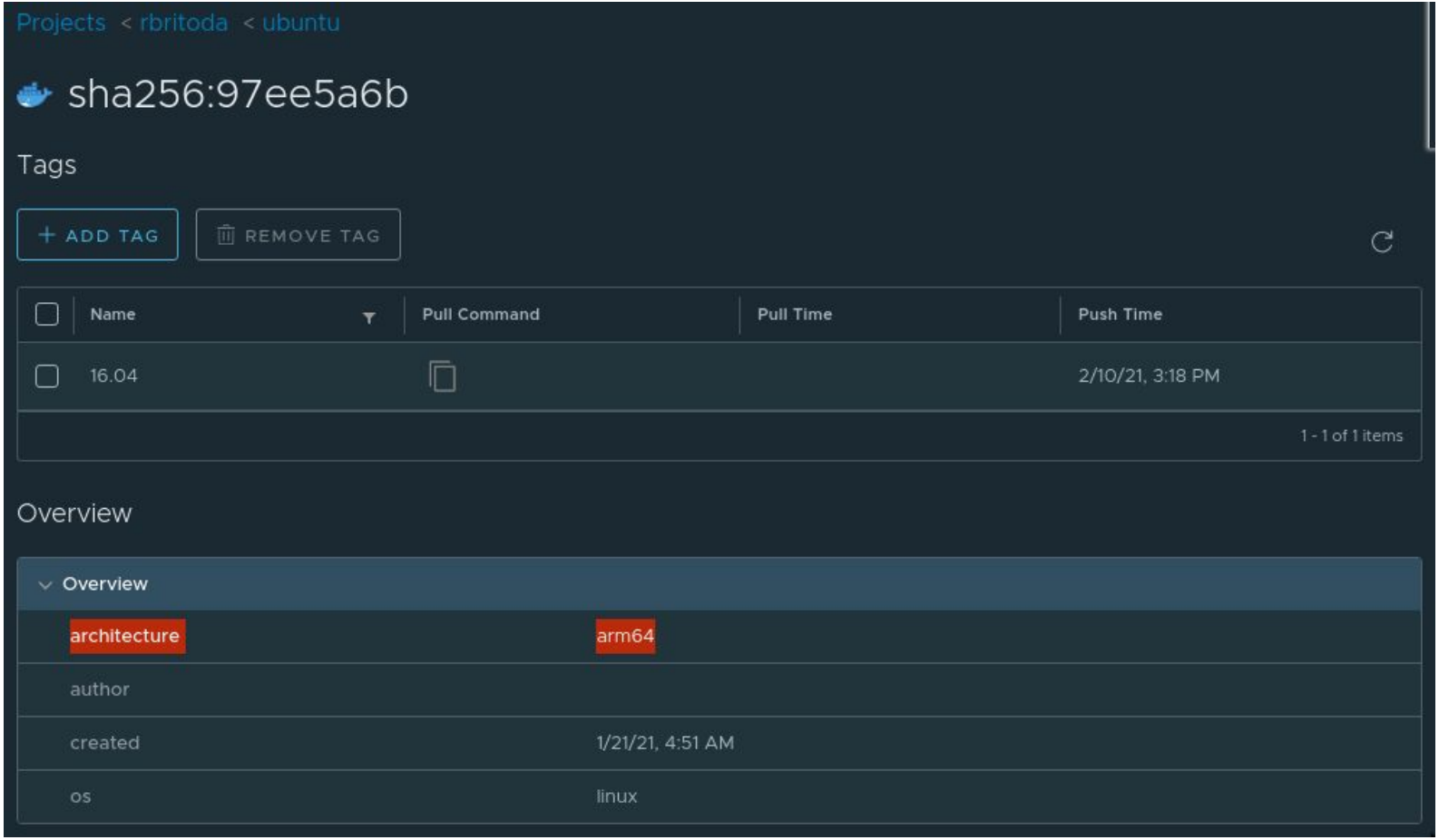
- Currently up to 8 cores TPUs available, can easily add bigger ones
- Useful to validate TensorFlow or JAX based code, performance

ARM

- Relying on m6g.large instances (2 vCPUs), request for 64 vCPUs and baremetal (soon)
- Job images must be ARM64 compatible, dockerhub and registry.cern.ch have multi-arch support
- Useful to build packages and images, benchmarking and performance validation



hub.docker.com



registry.cern.ch

Questions?