Using GPU On YARN

Prerequisites

- As of now, only Nvidia GPUs are supported by YARN
- YARN node managers have to be pre-installed with Nvidia drivers.
- (When Docker container is being used), install nvidia-docker (only 1.0 is supported by YARN). And run command to make sure it works.

Configs

GPU scheduling

In `resource-types.xml`

Add following properties

```
<configuration>
  <property>
     <name>yarn.resource-types</name>
     <value>yarn.io/gpu</value>
  </property>
</configuration>
```

In `yarn-site.xml`

`DominantResourceCalculator` MUST be configured to enable GPU scheduling/isolation.

For `Capacity Scheduler`, use following property to configure `DominantResourceCalculator` (In `capacity-scheduler.xml`):

Property	Default value

yarn.scheduler.capacity.resource-calculator	org.apache.hadoop.yarn.util.resource.DominantResourceCalculator

GPU Isolation

In `yarn-site.xml`

```
<property>
  <name>yarn.nodemanager.resource-plugins</name>
  <value>yarn.io/gpu</value>
</property>
```

This is to enable GPU isolation module on NodeManager side.

By default, YARN will automatically detect and config GPUs when above config is set. Following configs need to be set in `yarn-site.xml` only if admin has specialized requirements.

1) Allowed GPU Devices

Property	Default value
yarn.nodemanager.resource-plugins.gpu.allowed-gpu-devices	auto

Specify GPU devices which can be managed by YARN NodeManager, split by comma
Number of GPU devices will be reported to RM to make scheduling decisions.
Set to auto (default) let YARN automatically discover GPU resource from
system.

Manually specify GPU devices if auto detect GPU device failed or admin
only want subset of GPU devices managed by YARN. GPU device is identified
by their minor device number and index. A common approach to get minor
device number of GPUs is using `nvidia-smi -q` and search `Minor Number`
output.

When manual specify minor numbers, admin needs to include indice of GPUs
as well, format is `index:minor_number[,index:minor_number...]`. An example
of manual specification is `0:0,1:1,2:2,3:4"` to allow YARN NodeManager to

manage GPU devices with indices `0/1/2/3` and minor number `0/1/2/4`.
numbers .

2) Executable to discover GPUs

Property	value
yarn.nodemanager.resource-plugins.gpu.path-to-discovery-executables	/absolute/path/to/nvidia-smi

When `yarn.nodemanager.resource.gpu.allowed-gpu-devices=auto` specified,
YARN NodeManager needs to run GPU discovery binary (now only support `nvidia-smi`) to get GPU-related information.
When value is empty (default), YARN NodeManager will try to locate
discovery executable itself.
An example of the config value is: `/usr/local/bin/nvidia-smi`

3) Docker Plugin Related Configs

Following configs can be customized when user needs to run GPU applications inside Docker container. They're not required if admin follows default installation/configuration of `nvidia-docker`.

Property	Default value
yarn.nodemanager.resource-plugins.gpu.docker-plugin	nvidia-docker-v1

Specify docker command plugin for GPU. By default uses Nvidia docker V1.0.

Property	Default value
yarn.nodemanager.resource-plugins.gpu.docker-plugin.nvidia-docker-v1.endpoint	http://localhost:3476/v1.0/docker/cli

Specify end point of `nvidia-docker-plugin`. Please find documentation: https://github.com/NVIDIA/nvidia-docker/wiki For more details.

4) CGroups mount

GPU isolation uses CGroup devices controller to do per-GPU device isolation. Following configs

should be added to `yarn-site.xml` to automatically mount CGroup sub devices, otherwise admin has to manually create devices subfolder in order to use this feature.

Property	Default value
yarn.nodemanager.linux-container-executor.cgroups.mount	true

In `container-executor.cfg`

In general, following config needs to be added to `container-executor.cfg`

```
[gpu]
module.enabled=true
```

When user needs to run GPU applications under non-Docker environment:

```
[cgroups]
# This should be same as yarn.nodemanager.linux-container-executor.cgroups.mount-path inside yarn-site.xml
root=/sys/fs/cgroup
# This should be same as yarn.nodemanager.linux-container-executor.cgroups.hierarchy inside yarn-site.xml
yarn-hierarchy=yarn
```

When user needs to run GPU applications under Docker environment:

1) Add GPU related devices to docker section:

Values separated by comma, you can get this by running `ls /dev/nvidia*`

```
[docker]
docker.allowed.devices=/dev/nvidiactl,/dev/nvidia-uvm,/dev/nvidia-uvm-tools,/dev/nvidia1,/dev/nvidia0
```

2) Add `nvidia-docker` to volume-driver whitelist.

```
[docker]
...
```

```
docker.allowed.volume-drivers
```

3) Add `nvidia_driver_<version>` to readonly mounts whitelist.

```
[docker]
...
docker.allowed.ro-mounts=nvidia_driver_375.66
```

Use it

Distributed-shell + GPU

Distributed shell currently support specify additional resource types other than memory and vcores.

Distributed-shell + GPU without Docker

Run distributed shell without using docker container (Asks 2 tasks, each task has 3GB memory, 1 vcore, 2 GPU device resource):

```
yarn jar <path/to/hadoop-yarn-applications-distributedshell.jar> \
  -jar <path/to/hadoop-yarn-applications-distributedshell.jar> \
  -shell_command /usr/local/nvidia/bin/nvidia-smi \
  -container_resources memory-mb=3072,vcores=1,yarn.io/gpu=2 \
  -num_containers 2
```

You should be able to see output like

```
Tue Dec  5 22:21:47 2017
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.66                 Driver Version: 375.66                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
```

```
|   0  Tesla P100-PCIE...  Off  | 0000:04:00.0     Off |                    0 |
| N/A   30C    P0    24W / 250W |      0MiB / 12193MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla P100-PCIE...  Off  | 0000:82:00.0     Off |                    0 |
| N/A   34C    P0    25W / 250W |      0MiB / 12193MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

For launched container task.

Distributed-shell + GPU with Docker

You can also run distributed shell with Docker container.
`YARN_CONTAINER_RUNTIME_TYPE` / `YARN_CONTAINER_RUNTIME_DOCKER_IMAGE` must be specified to use docker container.

```
yarn jar <path/to/hadoop-yarn-applications-distributedshell.jar> \
       -jar <path/to/hadoop-yarn-applications-distributedshell.jar> \
       -shell_env YARN_CONTAINER_RUNTIME_TYPE=docker \
       -shell_env YARN_CONTAINER_RUNTIME_DOCKER_IMAGE=<docker-image-name> \
       -shell_command nvidia-smi \
       -container_resources memory-mb=3072,vcores=1,yarn.io/gpu=2 \
       -num_containers 2
```