GPU Monitoring with DCGM

Unit 5

Outline

- DCGM Overview
- DCGM CLI Usage and Examples

Objectives

By the end of this unit, you will be able to:

- Define the purpose and features of DCGM
- Apply the DCGM command-line interface (CLI) to perform common tasks

DCGM Overview

NVIDIA Data Center GPU Management (DCGM)

- The NVIDIA® Data Center GPU Manager (DCGM) is an intelligent, lightweight user space agent that provides monitoring capabilities of NVIDIA GPUs in the cluster.
- DCGM simplifies administration by performing a variety of functions related to the GPUs available on the system, such as:
  - GPU behavior monitoring
  - GPU configuration management
  - GPU policy oversight
  - GPU health and diagnostics
  - GPU accounting and process statistics
  - NVSwitch configuration

DCGM on a DGX System

Focus Areas:

- Manage GPUs as collections of related resources
- Configure NVSwitch
- Define and enforce GPU configuration state
- Automate GPU management policies
- Provide robust, online health and diagnostics
- Enable job-level statistics and accounting

DCGM Framework Includes:

- Programmatic access via C and Python
- Python interfaces for admin-centric scripting environments
- CLI-based tools to provide an interactive out-of-the-box

DCGM CLI Usage and Examples

DCGM CLI

- Standalone Mode is the NVIDIA preferred mode of operation
- DCGMI is the CLI interface for the standalone mode of DCGM
- Allows users to easily configure, manage and monitor GPUs and interconnections
- For more details on the DCGM usage refer to the user guide:
- https://docs.nvidia.com/datacenter/dcgm/latest/dcgm-user-guide/index.html

DCGM Help Page

- Use '**dcgmi**' to run DCGM CLI commands
- Run '**dcgmi --help**' command for details
- Subsystems are primary method used to interact with DCGM CLI

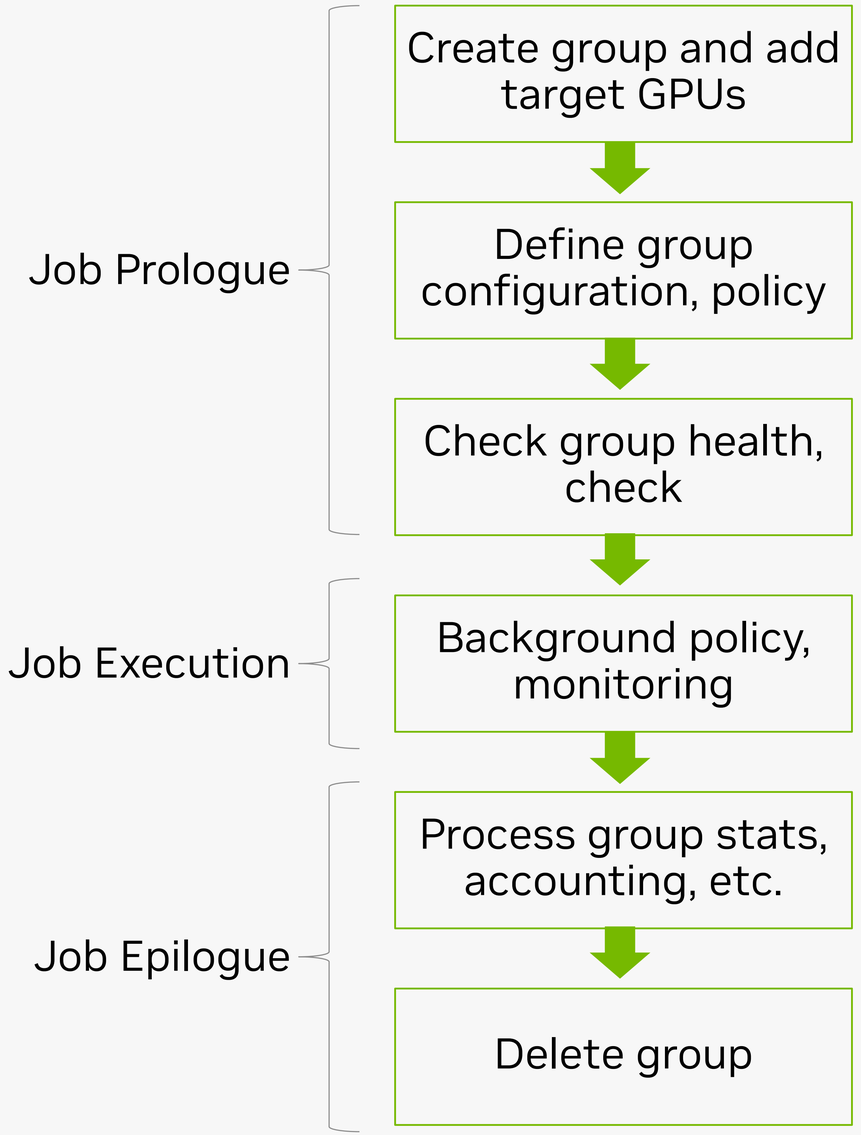
```
~ $ dcgmi --help
Usage: dcgmi
   dcgmi <subsystem>
   dcgmi -v

Flags:
  -v  vv             (OR required)  Get DCGMI version information
      subsystem      (OR required)  The desired subsystem to be accessed.
                      Subsystems Available:
…

Please email dcgm-support@nvidia.com with any questions, bug reports, etc.

NVIDIA Datacenter GPU Management Interface

...[output truncated]
```

DCGM CLI Subsystems

Subsystem	Description
set	Configure hostengine settings
profile	Control and list DCGM profiling metrics
modules	Control and list DCGM modules
dmon	Stats Monitoring of GPUs [dcgmi dmon –h for more info]
nvlink	Displays NvLink link statuses and error counts [dcgmi nvlink –h for more info]
introspect	Gather info about DCGM itself [dcgmi introspect –h for more info]
discovery	Discover GPUs on the system [dcgmi discovery –h for more info]
group	GPU Group Management
config	Configuration Management
health	Health Monitoring
policy	Policy Management
diag	System Validation/Diagnostic
stats	Process Statistics
topo	GPU Topology

DCGMI CLI Usage

- Almost all DCGM operations take place on groups
- Groups are intended to help the user to manage collections of GPUs as a single abstract resource
- Setting up a group of GPUs for a job would follow the flowchart



DCGMI Creating and Deleting a Group

- A group can be created using '**dcgmi group –c <Group_Name>**'
- To get a listing of existing groups, use '**dcgmi group –l**'
- To delete an existing group, use '**dcgmi group –d <Group ID>**'

```
~ $ dcgmi group -c GPU_Group

GPU_Group Successfully created group "GPU_Group" with a group ID of 1


~ $ dcgmi group -l
1 group found.
+-------------------------------------------------------------------------+
| GROUPS                                                                  |
+============+============================================================+
| Group ID   | 1                                                          |
| Group Name | GPU_Group                                                  |
| GPU ID(s)  | None                                                       |
+------------+------------------------------------------------------------+

$ dcgmi group -d 1
Successfully removed group 1
```

DCGMI Adding GPUs to a Group

- DCGMI discovery can be used to find a list of available GPUs
- Selected GPUs can be added to an existing group

```
~ $ dcgmi discovery -l
2 GPUs found.

+--------+----------------------------------------------------------------+
| GPU ID | Device Information                                             |
+========+================================================================+
| 0      | Name: Tesla K80                                                |
|        | PCI Bus ID: 0000:07:00.0                                       |
|        | Device UUID: GPU-00000000000000000000000000000000              |
+--------+----------------------------------------------------------------+
| 1      | Name: Tesla K80                                                |
|        | PCI Bus ID: 0000:08:00.0                                       |
|        | Device UUID: GPU-11111111111111111111111111111111              |
+--------+----------------------------------------------------------------+

~ $ dcgmi group -g 1 -a 0,1
Add to group operation successful.
```

DCGMI GPU Configuration & Policy

- Use the **config** subcommand to set the target configuration for the group
- The **policy** subcommand is used to set up automatic GPU behaviors (e.g., action, notification) based on various conditions.
- The **--set** and **--get** flags are used to set and read configuration or policy information
- The **--help** flag gives the details on the settings

```
~ $ dcgmi config –g 1 –-set –c 2
Configuration successfully set.


~ $ dcgmi policy –g 2 –-set 0,0 -p
Policy successfully set.
```

DCGMI Enable and Check Stats

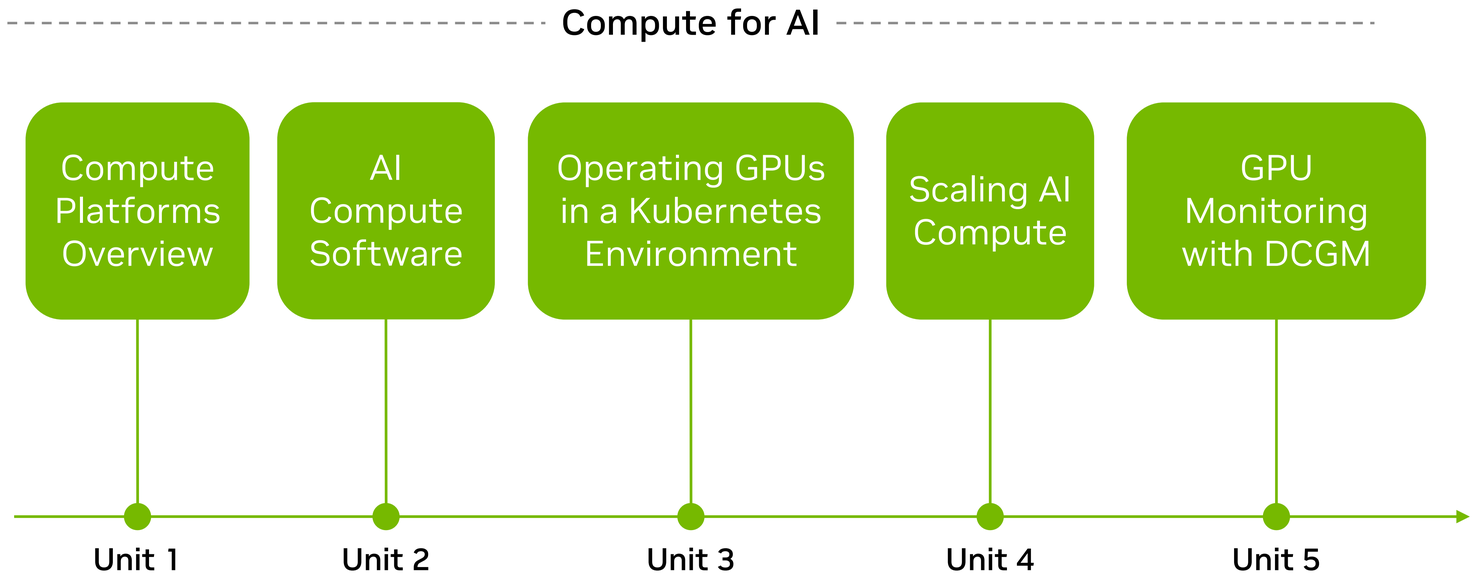
- DCGMI stats must be enabled to track GPUs status
- Status can be tracked across jobs and groups of GPUs

```
~ $ dcgmi stats -g 2 --enable
Successfully started process watches.

~ $ dcgmi stats --pid 1234 -v
+-------------------------------------------------------------------------+
| GPU ID: 0 |
+=================================+=======================================+
|------- Execution Stats ----------+---------------------------------------|
| Start Time *                    | Tue Nov 3 17:36:43 2015               |
| End Time *                      | Tue Nov 3 17:38:33 2015               |
| Total Execution Time (sec) *    | 110.33                                |
| No. of Conflicting Processes *  | 0                                     |
+------- Performance Stats --------+---------------------------------------+

[output truncated]

(*) Represents a process statistic. Otherwise device statistic during process lifetime listed.
```

Summary

Now that you have completed this unit, you should be able to:

- Define the purpose and features of DCGM
- Apply the DCGM command-line interface (CLI) to perform common tasks




NVIDIA