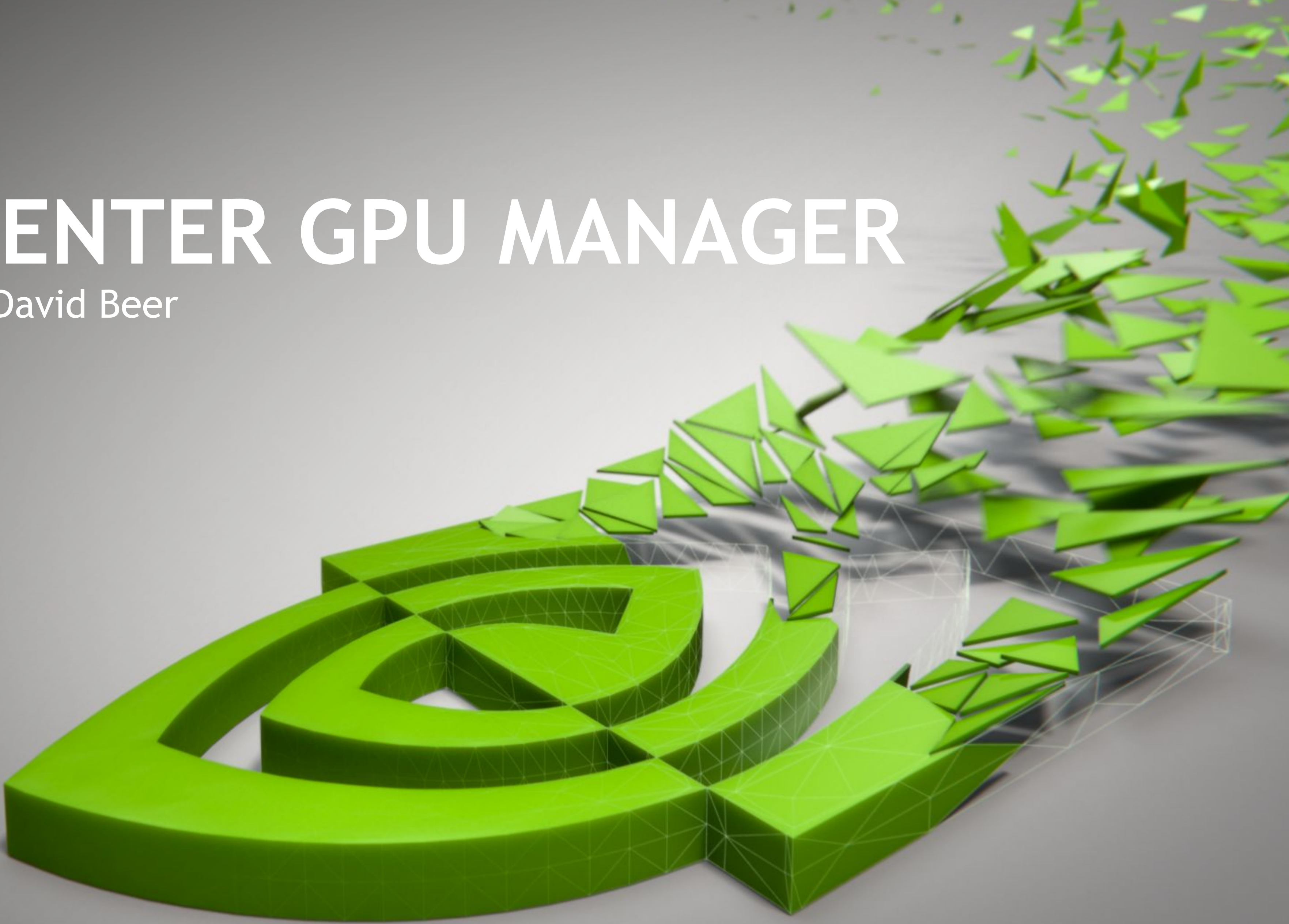
DATA CENTER GPU MANAGER

Brent Stolle and David Beer

March 2018



TOOLS FOR MANAGING GPUs

Out-of-Band

GPU Metrics and Monitoring via BMC (SMBPBI)

Provide metrics (thermals, power, etc.) without the NVIDIA driver

Typically used at public CSPs (i.e. multi-tenant environments)

In-Band

Tools use the NVIDIA driver to provide GPU and NVSwitch metrics

DCGM, NVML (smi) are in-band tools

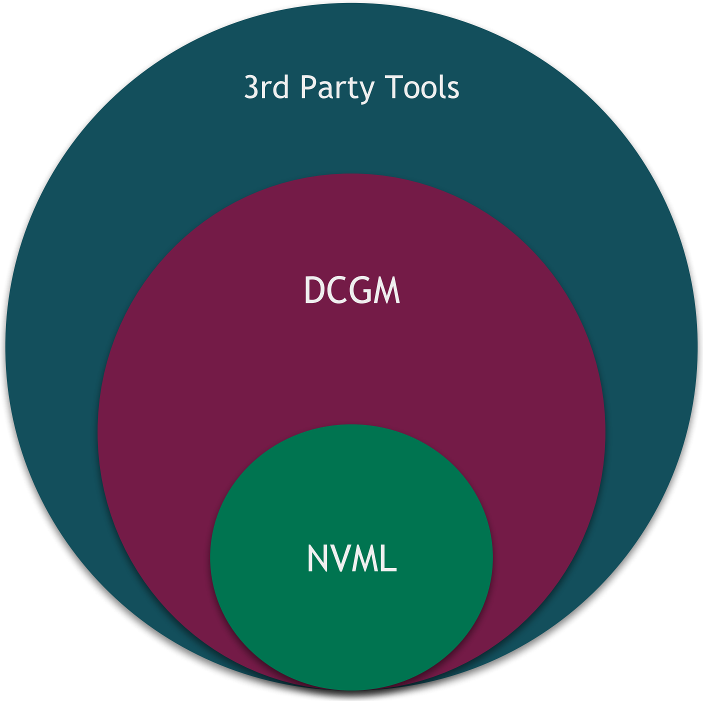
Typically used at single tenant environments

NVIDIA IN-BAND TOOLS ECOSYSTEM

- Cluster managers, Job schedulers, TSDBs, Visualization tools

- Customers integrating DCGM; CSPs for system validation

- Customers building their own GPU metrics/monitoring stack using NVML



Bright Computing
IBM PLATFORM COMPUTING
Grafana
UNIVA
Prometheus
Altair | PBS Works
Adaptive COMPUTING

HOW SHOULD I MANAGE MY GPUS?

NVML

Stateless queries. Can only query current data

Low overhead while running, high overhead to develop

Low-level control of GPUs

Management app must run on same box as GPUs

DCGM

Can query a few hours of metrics

Provides health checks and diagnostics

Can batch queries/operations to groups of GPUs

Can be remote or local

3RD PARTY TOOLS

Provide database, graphs, and a nice UI

Need management node(s)

Development already done. You just have to configure the tools.

DATA CENTER GPU MANAGER (DCGM)

ACTIVE HEALTH MONITORING

- Runtime Health Checks
- Prologue Checks
- Epilogue Checks

GPU DIAGNOSTICS

- Software Deployment Tests
- Stress Tests
- Hardware Issues and Interface Tests (PCIe, NVLink)

POLICY AND ALERTING

- Pre-configured Policies
- Job Level Statistics
- Stateful Configuration

CONFIGURATION MANAGEMENT

- Dynamic Power Capping
- Synchronous Clock Boost
- Fixed Clocks

DCGM OVERVIEW

GPU Management in the Accelerated Data Center

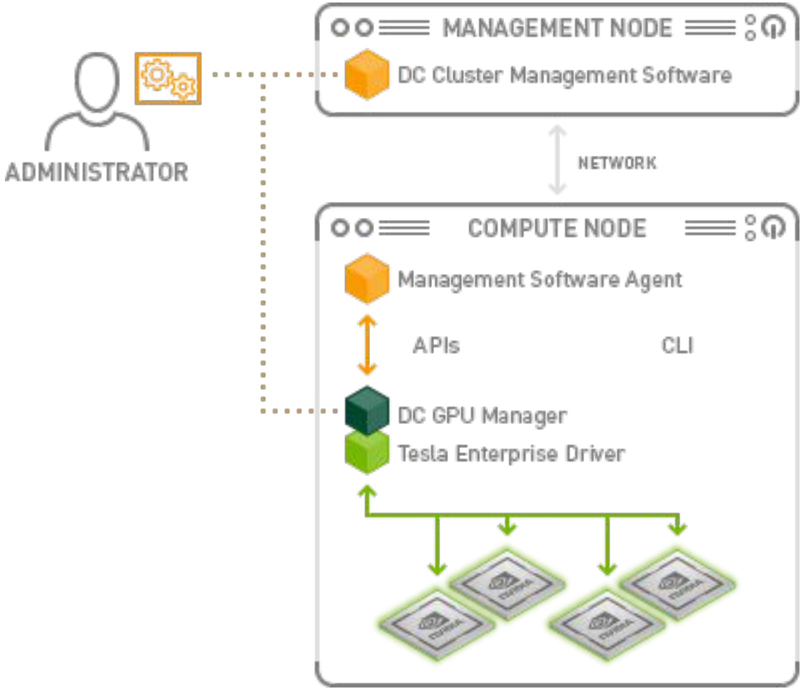
Supported NVIDIA Hardware

- Fully supported on Tesla GPUs (Kepler+)
- Supported on Quadro, GeForce, and Titan GPUs (Maxwell+)
- Supports NvSwitch and DGX-2
- Driver R384 or Later (Linux only)

SDK Installer Packages

- .deb and .rpm Packages
- Includes Binaries - CLI (`dcgmi`) and daemon (`nv-hostengine`)
- Libraries and Headers (includes NVML)
- C and Python Bindings and Code samples
- Documentation - User Guides and API docs

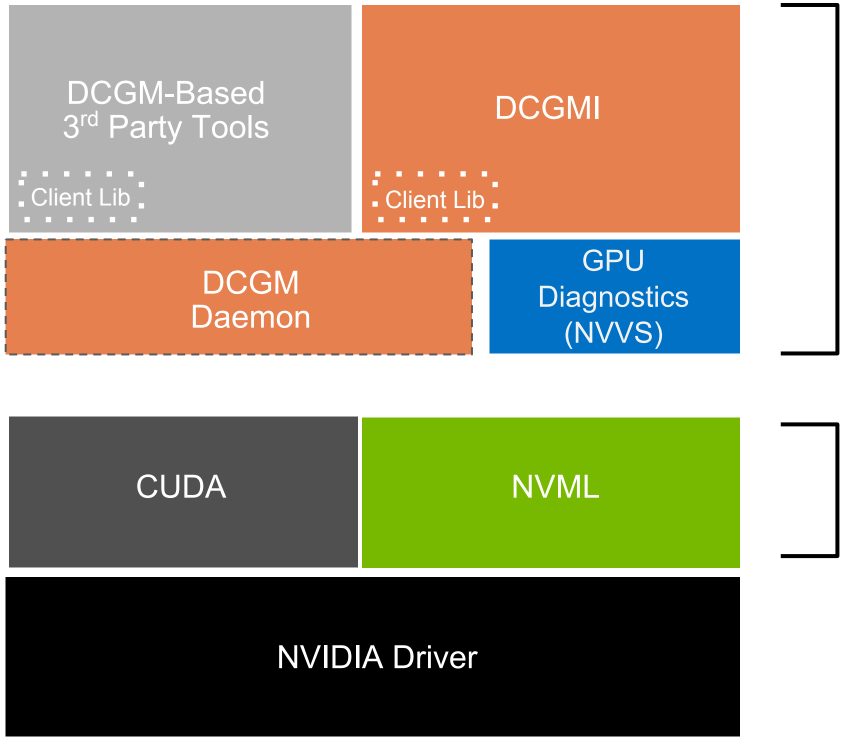
https://developer.nvidia.com/data-center-gpu-manager-dcgm



Latest Release: v1.3.3 (Jan 2018)

AVAILABLE NVIDIA MANAGEMENT TOOLS

Software Stack



Data Center GPU Manager (DCGM)

- Additional diagnostics (aka NVVS) and active health monitoring
- Policy management and more

NVIDIA Management Library (NVML)

- Low level control of GPUs
- Included as part of driver
- Header is part of CUDA Toolkit / DCGM

ACTIVE HEALTH MONITORING & ANALYSIS

NON INVASIVE CHECKS

Real-time monitoring & aggregated health indicator

Checks health of all GPUs and NVSwitch subsystems

- PCIe, ECC, Inforom, Power Thermal, NVLink

Run Health Check : Healthy System

```
dcgmi health --check -g 1

Health Monitor Report
+------------------+---------------------------------------------------------+
| Overall Health:   Healthy                                                  |
+==================+=========================================================+
```

Run Health Check : System with problems

```
dcgmi health -g 1 -c
Health Monitor Report
+----------------------------------------------------------------------------+
| Group 1          | Overall Health: Warning                                 |
+==================+=========================================================+
| GPU ID: 0        | Warning                                                 |
|                  | PCIe system: Warning - Detected more than 8 PCIe        |
|                  | replays per minute for GPU 0: 13                        |
+------------------+---------------------------------------------------------+
| GPU ID: 1        | Warning                                                 |
|                  | InfoROM system: Warning - A corrupt InfoROM has been    |
|                  | detected in GPU 1.                                      |
+------------------+---------------------------------------------------------+
```

Demo: Health Checks

GPU DIAGNOSTICS (NVVS) - COVERAGE AREAS

DEPLOYMENT AND SOFTWARE ISSUES

- NVML library access and versioning
- CUDA library access and versioning
- Software conflicts

HARDWARE ISSUES AND DIAGNOSTICS

- PCIe and NVLink interface checks
- Framebuffer and memory checks
- Compute engine checks

STRESS CHECKS

- Power and thermal stress
- Throughput stress
- Constant relative system performance
- Maximum relative system performance

INTEGRATION ISSUES

- PCIe and NVLink replay counter checks
- Topological limitations
- Permissions, driver and cgroups checks
- Basic power and thermal constraint checks

COMPREHENSIVE DIAGNOSTICS

ACTIVE HEALTH CHECKS

Identification, recovery & isolation of failed GPUs and NVSwitches.

Diagnostics to root cause failures, Pre & post job GPU health checks

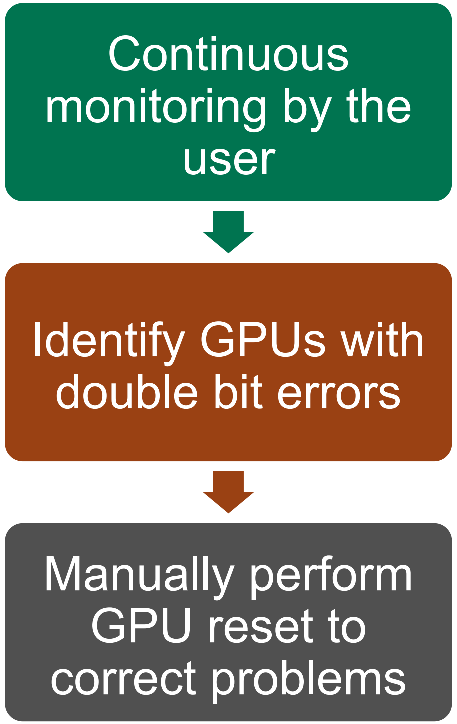
System sanity to stress performance, bandwidth, power and thermal characteristics

Multi-level diagnostic options from few seconds to minutes

```
dcgmi diag -r 3
+---------------------------+-------------+
| Diagnostic                | Result      |
+===========================+=============+
|-----  Deployment  --------+-------------|
| Blacklist                 | Pass        |
| NVML Library              | Pass        |
| CUDA Main Library         | Pass        |
| CUDA Toolkit Library      | Pass        |
| Permissions and OS Blocks | Pass        |
| Persistence Mode          | Pass        |
| Environment Variables     | Pass        |
| Page Retirement           | Pass        |
| Graphics Processes        | Pass        |
| Inforom                   | Pass        |
+-----  Hardware  ----------+-------------+
| GPU Memory                | Pass - All  |
| Diagnostic                | Pass - All  |
+-----  Integration  -------+-------------+
| PCIe                      | Pass - All  |
+-----  Stress  ------------+-------------+
| SM Stress                 | Pass - All  |
| Targeted Stress           | Pass - All  |
| Targeted Power            | Warn - All  |
| Memory Bandwidth          | Pass - All  |
+---------------------------+-------------+
```

FLEXIBLE GPU GOVERNANCE POLICIES

With Existing Tools

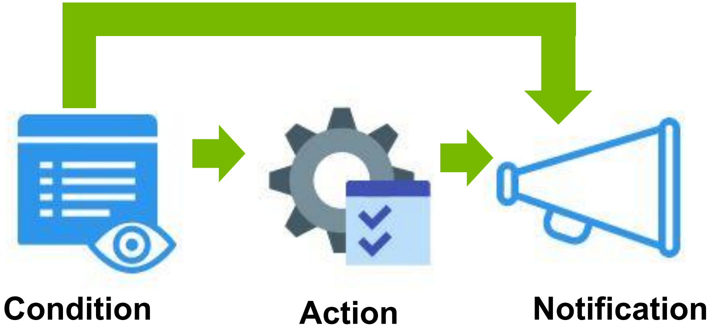


Continuous monitoring by the user

Identify GPUs with double bit errors

Manually perform GPU reset to correct problems

Using DCGM



Auto-detects double bit errors, performs page retirement, and notifies the user

Condition: Watch for DBE
Action: Page retirement
Notification: Callback

Demo: Policy Alerting

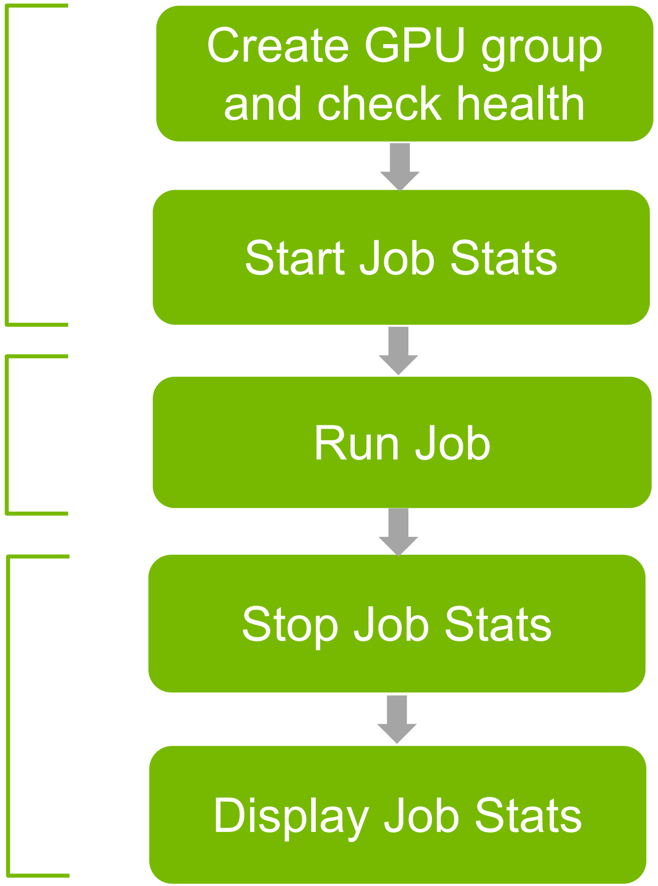
MANAGING JOB LIFECYCLE

Which GPUs did my job run on?

How much of the GPUs did my job use?

Any error or warning conditions during my job (ECC errors, clock throttling, etc)

Are the GPUs healthy and ready for the next job?



Create GPU group and check health

Start Job Stats

Run Job

Stop Job Stats

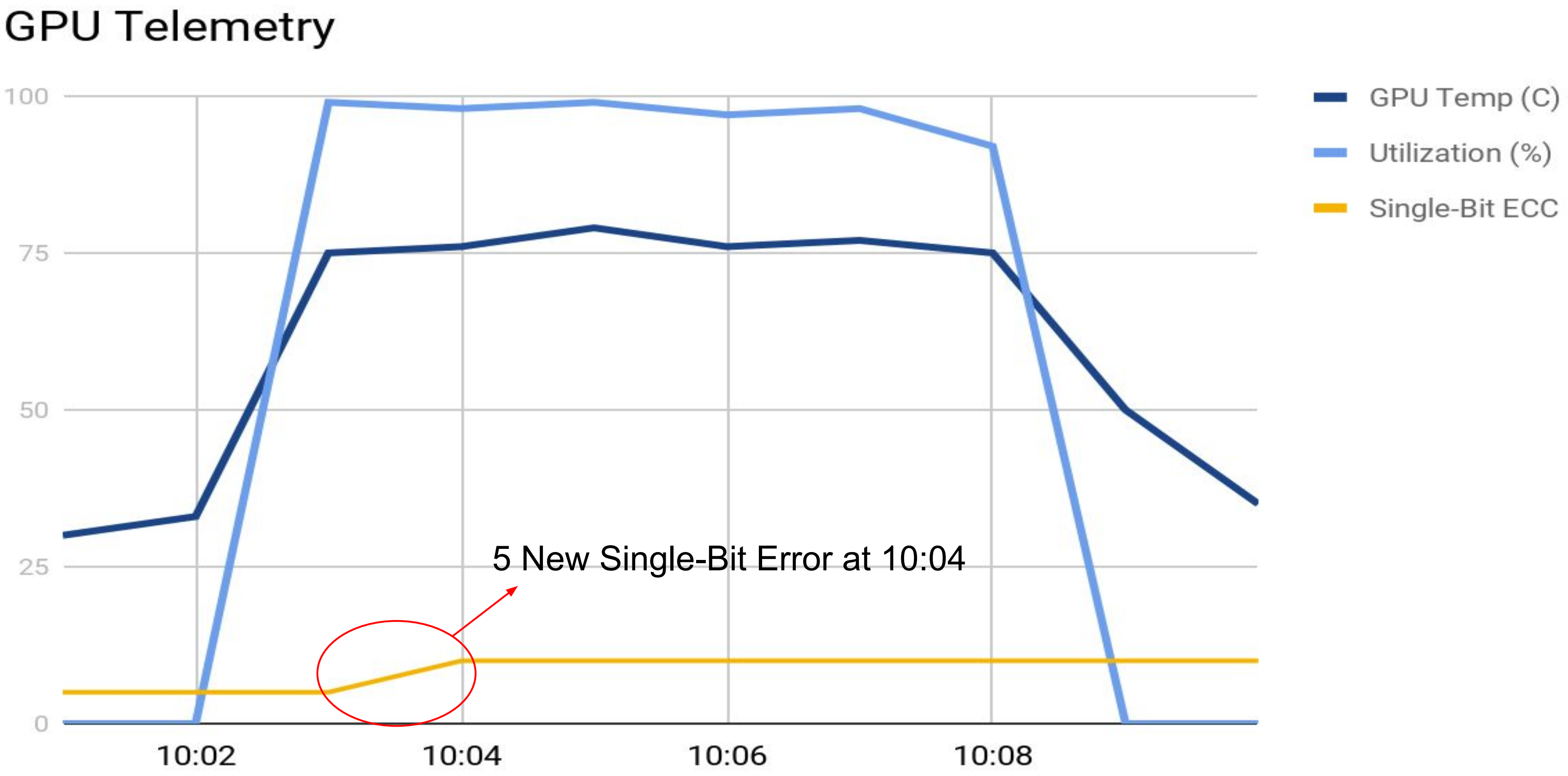
Display Job Stats

JOB STATISTICS

Detailed stats show utilization, performance and more…

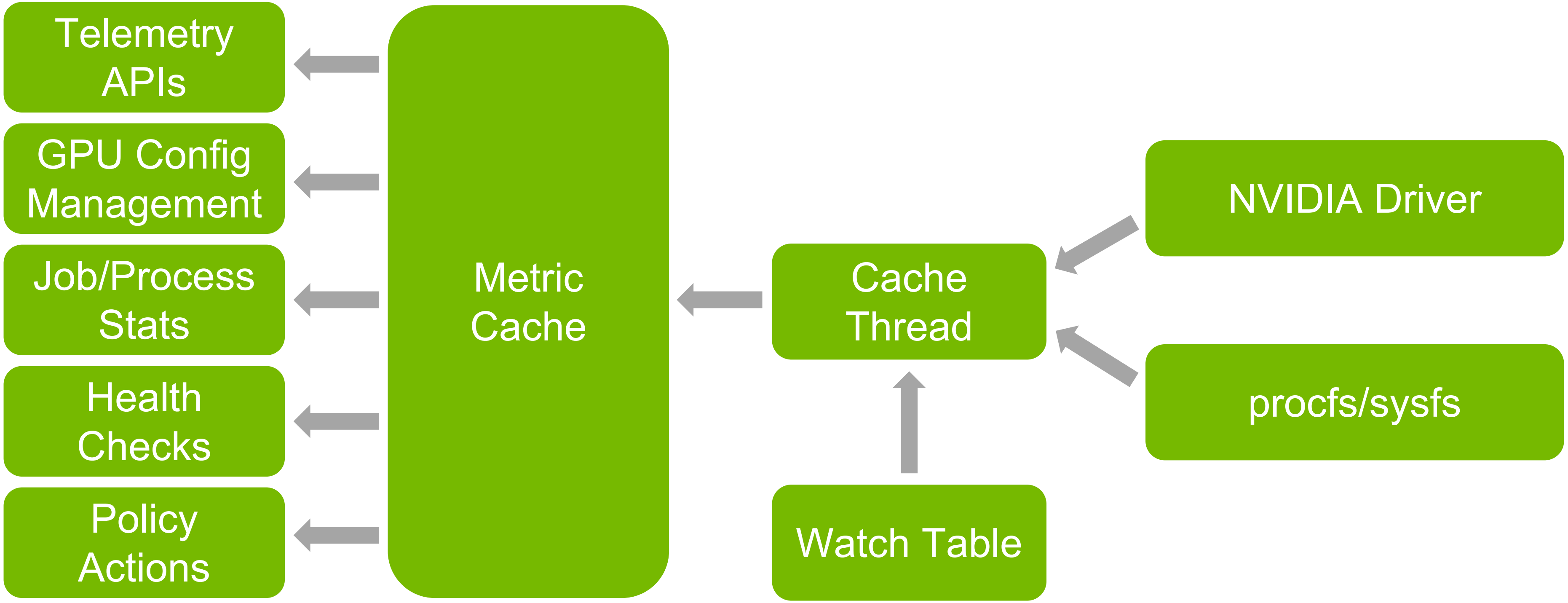
```
dcgmi stats --job demojob -v -g 2
Successfully retrieved statistics for job: demojob.
+----------------------------------------------------------------------------+
| GPU ID: 0                                                                  |
+==================================+=========================================+
|-----  Execution Stats  ----------+-----------------------------------------|
| Start Time                       | Wed Mar  7 10:02:34 2018                |
| End Time                         | Wed Mar  7 10:10:00 2018                |
| Total Execution Time (sec)       | 445.48                                  |
| No. of Processes                 | 1                                       |
| Compute PID                      | 23112                                   |
+-----  Performance Stats  --------+-----------------------------------------+
| Energy Consumed (Joules)         | 1437                                    |
| Max GPU Memory Used (bytes)      | 120324096                               |
| SM Clock (MHz)                   | Avg: 998, Max: 1177, Min: 405           |
| Memory Clock (MHz)               | Avg: 2068, Max: 2505, Min: 324          |
| SM Utilization (%)               | Avg: 76, Max: 100, Min: 0               |
| Memory Utilization (%)           | Avg: 0, Max: 1, Min: 0                  |
| PCIe Rx Bandwidth (megabytes)    | Avg: 0, Max: 0, Min: 0                  |
| PCIe Tx Bandwidth (megabytes)    | Avg: 0, Max: 0, Min: 0                  |
+-----  Event Stats  --------------+-----------------------------------------+
| Single Bit ECC Errors            | 5                                       |
| Double Bit ECC Errors            | 0                                       |
| PCIe Replay Warnings             | 0                                       |
| Critical XID Errors              | 0                                       |
+-----  Slowdown Stats  -----------+-----------------------------------------+
| Due to - Power (%)               | 0                                       |
|        - Thermal (%)             | Not Supported                           |
|        - Reliability (%)         | Not Supported                           |
|        - Board Limit (%)         | Not Supported                           |
|        - Low Utilization (%)     | Not Supported                           |
|        - Sync Boost (%)          | 0                                       |
+----------------------------------+-----------------------------------------+
```

WHY A DAEMON? STATEFULNESS



5 New Single-Bit Error at 10:04

DCGM DAEMON INTERNALS



GPU CONFIGURATION MANAGEMENT

MAINTAINS CONFIGURATION

Initialization: Configure all GPUs (global group)

Per-job basis: Individual partitioned group settings

Maintains settings across driver restarts, GPU resets or at job start

Supports SET, GET and ENFORCE

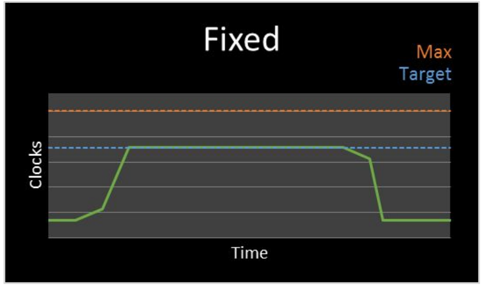
Disable ECC mode

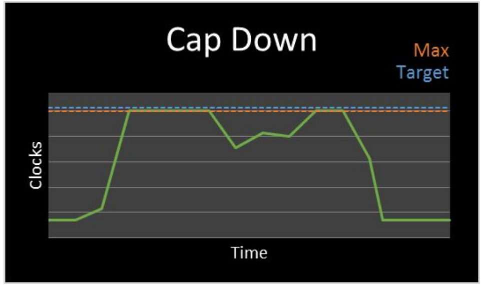
```
dcgmi config -g 1 --set -P 200
Configuration successfully set.
```

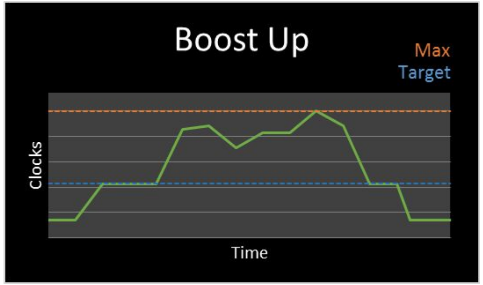
Get Group config [Note DCGM performed reset]

```
dcgmi config -g 1 --get
+---------------------------+------------------------+------------------------+
| all_gpu_group             |                        |                        |
| Group of 2 GPUs           | TARGET CONFIGURATION   | CURRENT CONFIGURATION  |
+===========================+========================+========================+
| Sync Boost                |  Not Specified         |  Disabled              |
| SM Application Clock      |  Not Specified         |  705                   |
| Memory Application Clock  |  Not Specified         |  2600                  |
| ECC Mode                  |  Disabled              |  Disabled              |
| Power Limit               |  200                   |  225                   |
| Compute Mode              |  Not Specified         |  E. Process            |
+---------------------------+------------------------+------------------------+
```

ENHANCED POWER & CLOCK MGMT.







- **Dynamic Power Capping**
  - Drive better power density through dynamic power capping
  - Apply power capping to a single or a group of GPUs
- **Fixed Clocks**
  - Target conservative clock rate for fixed performance
  - Useful for profiling
- **Synchronous Clock Boost**
  - Predictable performance through group GPU clock boost in lockstep
  - Dynamically modulate mutli-gpu clocks across multiple boards in unison based on target workload, power budgets or other criteria

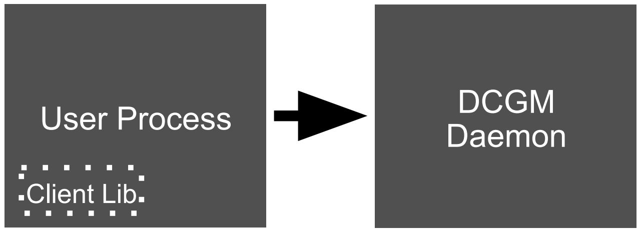
DCGM MODES OF OPERATION

STANDALONE

Runs as daemon

Client libraries connect via TCP/IP

1 DCGM for several clients



User Process

Client Lib

DCGM Daemon

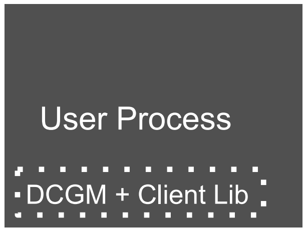
EMBEDDED

Runs within client process

Even within python

1 DCGM per client process

No TCP/IP necessary



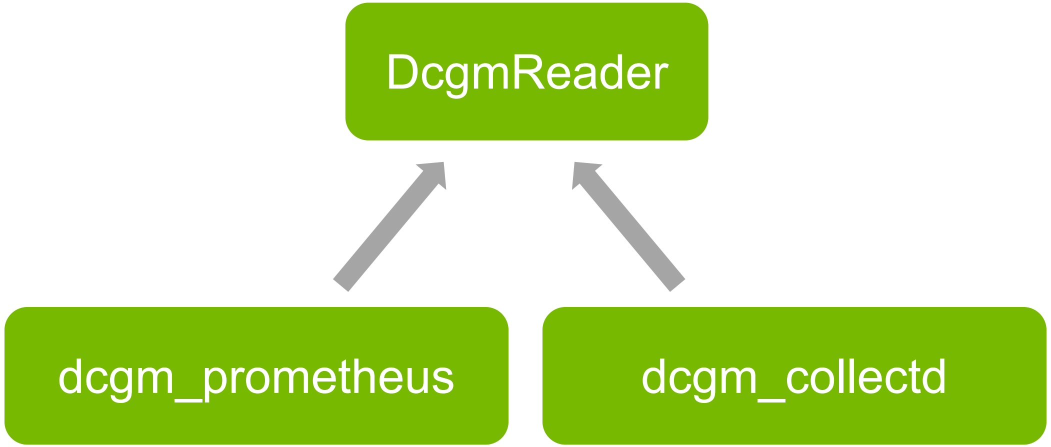
User Process

DCGM + Client Lib

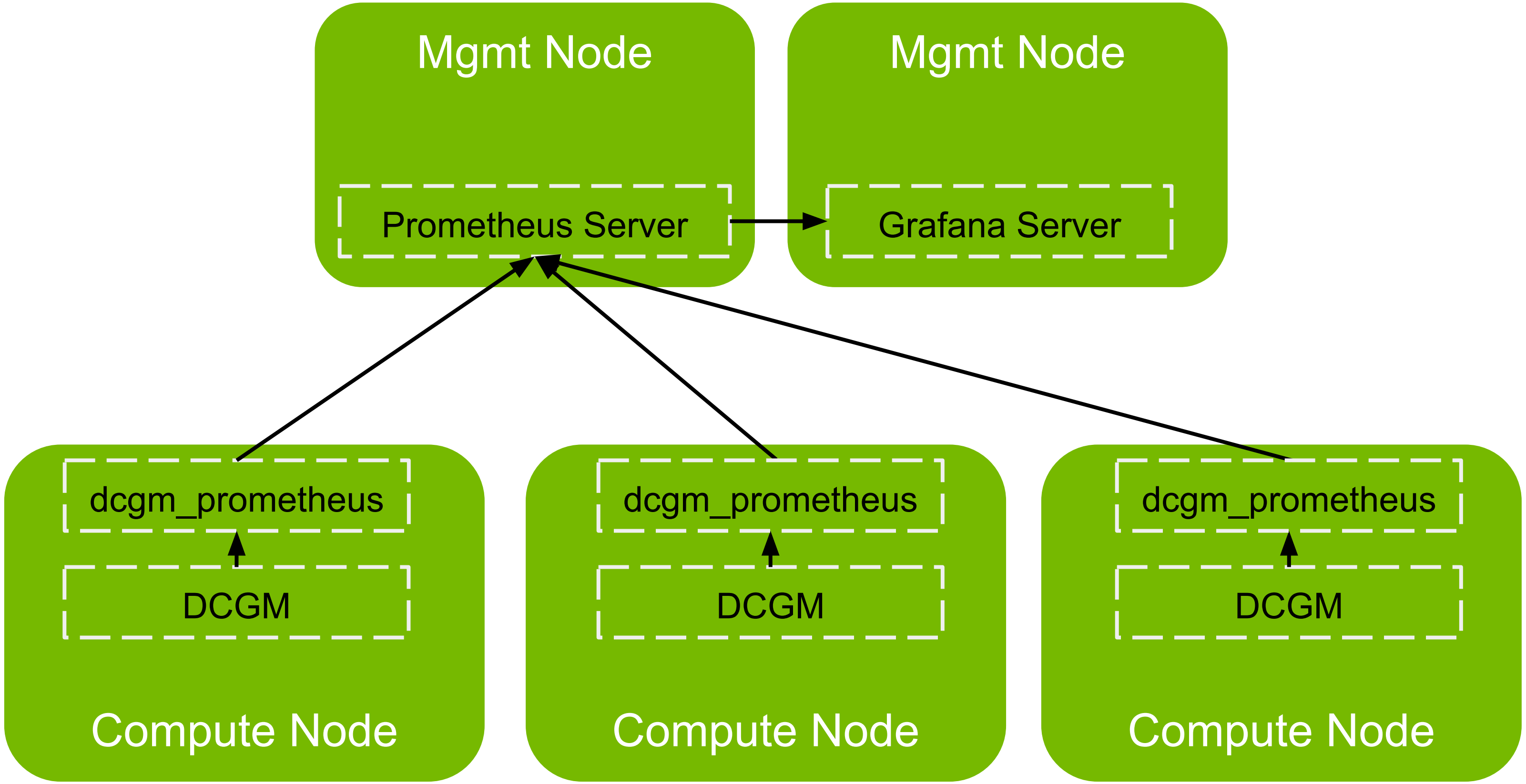
THIRD-PARTY INTEGRATIONS

Provide DcgmReader base python class for GPU / NvSwitch telemetry monitoring

Provide working examples for popular monitoring tools based on DcgmReader



EXAMPLE DEPLOYMENT: PROMETHEUS

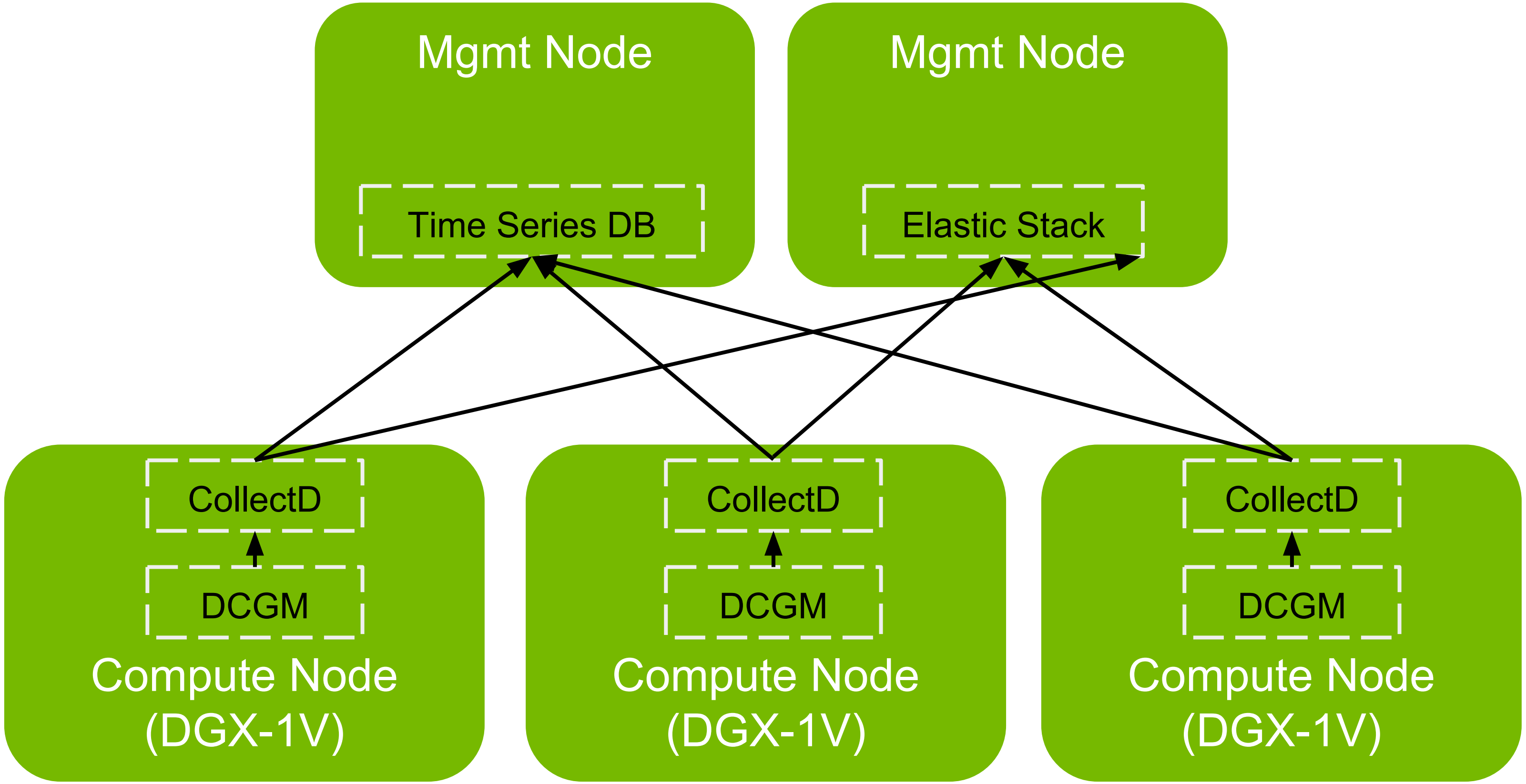


Demo: DCGM + Prometheus + Grafana

Example Deployments

NVIDIA SATURNV CLUSTER

660 Compute Nodes



DCGM ROADMAP*

v1.3.3

Container Ecosystem Enablement

- DCGM enablement for non-Tesla GPUs (Maxwell+)
- Interactive Device Monitoring with ‘dmon’
- New Diagnostics to stress GPUs
- Deprecation of standalone NVVS

Jan 2018

v1.4

Improved User Experience

- Integration with 3rd party monitoring/metrics stacks (Prometheus, Grafana)
- Container orchestration (Kubernetes) support (cAdvisor metrics, health checks)
- Go Bindings
- Job Scheduler Hints
- Packages on compute/cuda repo

Apr 2018

vNext

Next Generation Systems

- DGX-2 and NVSwitch monitoring and diagnostics
- Container orchestration continued

Summer 2018

* Roadmap Subject to Change

NVIDIA