

## ubuntu监控GPU

### 0. 基础通用设备信息查询

- `lspci -v | grep -i VGA`: 列出PCI设备信息, 通过grep过滤VGA(GPU)
- `lshw -C display`: 列出所有硬件设备信息
- `inxi -G`: 更为具体的显卡信息
  - 需要先下载 `apt install inxi`

### 1. 命令行简要监控

- NVIDIA
  - 通过官方驱动
    - `sudo apt install nvidia-driver-xxx` # 替换为你的驱动版本 (如 535)
    - `sudo apt install nvidia-utils-xxx` # 安装驱动版本所属工具包
    - `nvidia-smi`: 查看GPU详细信息
    - `watch -n 1 nvidia-smi`: 实现实时监控
  - `gpustat`
    - `pip install gpustat`
    - `gpustat`
  - `nvitop`
    - `pip install nvitop` 有auto\compact\full三种展示模式
    - `nvitop -m full`: 展示完整信息
- AMD
  - 安装AMD官方ROCm驱动
    - `sudo wget -qO - https://repo.radeon.com/rocm/rocm.asc | sudo apt-key add`  
- 官方GPG密钥
    - `sudo sh -c 'echo "deb [arch=amd64] https://repo.radeon.com/rocm/apt/debian/ xenial main" > /etc/apt/sources.list.d/rocm.list'` 官方软件源
      - 对于最新的 Ubuntu 版本, 可能需要替换 **xenial** 为相应的版本名称, 比如 bionic 或 focal
    - `sudo apt install rocm-smi`
    - `rocm-smi`
  - 使用 `radeontop` AMD开源工具
    - `sudo apt install radeontop`
    - `radeontop`
- Intel集成显卡
  - `sudo apt install intel-gpu-tools`
  - `sudo intel_gpu_top`

### 2. 系统级监控工具

- `glances`(支持Nvidia及Intel)
  - 提供全面的系统资源监控, 包括 CPU、内存、磁盘 I/O、网络流量、文件系统、进程负载、系统温度等
  - `sudo apt install glances`
  - `glances --enable-plugin gpu` # 启用 GPU 插件

### 3. Prometheus + Grafana

- 以[172.20.88.178]监控[172.20.88.52]GPU-**Tesla-T4**为例

#### 1. 安装dcgm-exporter/Node-exporter(GPU服务器)

- dcgm-exporter(**Nvidia专用**)
  - 需要安装Nvidia驱动, 并支持DCGM (官方dcgm-exporter的release版本只提供**source code**)
  - Nvidia驱动安装
    - `ubuntu-drivers devices` 查询设备支持的驱动版本
    - `apt-get install xxx`
    - `dpkg -l | grep nvidia-driver` 查询已安装的驱动版本
    - `lsmod | grep nouveau` 查询是否使用通用驱动程序nouveau, **没有无输出**
      - 若要启用Nvidia驱动, 则nouveau必须禁用
  - DCGM安装(以**ubuntu20.04**为例)
    - CUDA仓库添加

```
//安装依赖工具包
sudo apt-get install -y software-properties-common
//下载 CUDA 仓库的 PIN 文件
wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/cuda-ubuntu2004.pin
//移动 PIN 文件到 APT 配置目录, 设置APT包管理器优先级, 确保优先从Nvidia仓库下载
sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600
//添加GPG密钥
sudo apt-key adv --fetch-keys
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/3bf863cc.pub
//添加 CUDA 仓库
sudo add-apt-repository "deb
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86_64/ "
```

- `sudo apt-get update`
- `sudo apt-get install -y datacenter-gpu-manager` 安装
- `dcgmi -v` 检查版本

#### Docker-安装dcgm-exporter

- `docker login nvcr.io` 受限Registry, 需要Nvidia账户生成apikey
  - Username: \$oauthtoken
  - Password: cTVibWE1NXAzC21xcTVmZzVhamNiM2Jhc2o6NWFiMGJiYmMtZjAyYy00YmJjLWl0Zm

UtNmMyMGE4MTdhNzY4

- `docker pull nvcr.io/nvidia/k8s/dcgm-exporter:3.3.9-3.6.1-ubi9` 拉取镜像
  - [<https://github.com/NVIDIA/dcgm-exporter/releases>] 查找对应DCGM的版本
- `docker run --gpus all -d -p 9400:9400 nvcr.io/nvidia/k8s/dcgm-exporter:3.3.9-3.6.1-ubi9` 运行容器
  - `docker run --gpus all -d -p 9400:9400 --cap-add SYS_ADMIN nvcr.io/nvidia/dcgm-exporter:latest` 权限不足时使用

### 源码构建安装dcgm-exporter

- 需要go语言环境
  - `wget https://dl.google.com/go/go1.22.0.linux-amd64.tar.gz` 下载
  - `sudo tar -C /usr/local -xzf go1.22.0.linux-amd64.tar.gz` 解压
  - 配置 `/etc/profile`

```
//添加如下内容
export GOROOT=/usr/local/go
export GOPATH=$HOME/go
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin
```

- `source /etc/profile`
- `go version` 检查go版本
- `git clone https://github.com/NVIDIA/dcgm-exporter.git` 克隆
  - `cd dcgm-exporter` 进入仓库目录
  - `make binary` 编译
  - `cd cmd/dcgm-exporter`
  - `./dcgm-exporter` 运行

### 编译运行后会在9400端口暴露Prometheus 指标

- [<http://<主机IP>:9400/metrics>]查询是否启用
- Node-exporter(**Prometheus的通用嗅探器**)
  - [[https://github.com/prometheus/node\\_exporter/releases](https://github.com/prometheus/node_exporter/releases)]下载需求版本传输并解压
    - `wget https://github.com/prometheus/node_exporter/releases/download/v1.8.2/node_exporter-1.8.2.linux-amd64.tar.gz`
    - `tar -xvzf node_exporter-1.8.2.linux-amd64.tar.gz` 解压
    - `cd node_exporter-1.8.2.linux-amd64` 进入目录
    - `./node_exporter &` 运行
  - `apt-get install prometheus-node-exporter` 或直接使用包管理器下载 [/etc/...]
  - 编译运行后会在9100端口暴露指标
    - [<http://<主机IP>:9100/metrics>]查询是否启用
- 配置为系统服务
  - `vi /etc/systemd/system/node_exporter.service`

```
//配置以下内容
[Unit]
Description=Prometheus Node Exporter
Wants=network.target
After=network.target

[Service]
ExecStart=/path/to/node_exporter  /// 修改为实际node_exporter路径
Restart=on-failure
User=nobody
Group=nogroup

[Install]
WantedBy=multi-user.target

// 配置完成后的更新启动命令
sudo systemctl daemon-reload          //重新加载systemd配置
sudo systemctl enable node_exporter    // 启用
sudo systemctl start node_exporter     // 启动
sudo systemctl status node_exporter    // 检查服务状态

//dcgm-exporter配置方式一样
```

## 2. 安装prometheus (监控服务器)

- `apt-get install prometheus`
- 通过包管理器安装的地址为 `/etc/prometheus`

## 3. 配置 prometheus.yml

```
//dcgm-exporter, Nvidia专用
scrape_configs:
  - job_name: 'gpu'
    static_configs:
      - targets: ['<GPU服务器IP地址>:9400'] # 替换 GPU 服务器 IP 地址
```

```
// node-exporter, 通用
scrape_configs:
  # ... 其他 job 配置 ...
  - job_name: 'node-exporter'
    static_configs:
      - targets: ['<GPU服务器IP地址>:9100']
```

- 启动Prometheus
  - `systemctl start prometheus` 启动服务
  - `systemctl reload prometheus` 重启服务

#### 4. 安装Grafana(监控服务器)

- apt-get安装

```
// 添加grafana仓库GPG密钥及源
sudo apt-get install -y apt-transport-https //下载 HTTPS 软件包仓库中的软件包
sudo wget -q -O - https://apt.grafana.com/gpg.key | sudo apt-key add -
echo "deb https://apt.grafana.com stable main" | sudo tee
/etc/apt/sources.list.d/grafana.list

//更新软件源并安装
sudo apt-get update
sudo apt-get install -y grafana

sudo systemctl start grafana-server //开启
sudo systemctl enable grafana-server // 设置开机自启动
sudo systemctl status grafana-server // 检查服务状态
```

- deb包安装

- 官网下载地址 [<https://grafana.com/grafana/download?edition=oss&platform=linux>]

```
sudo apt-get install -y adduser libfontconfig1 musl
wget https://dl.grafana.com/oss/release/grafana_11.5.1_amd64.deb
sudo dpkg -i grafana_11.5.1_amd64.deb
```

#### 5. 配置Grafana数据源及创建仪表盘(Web界面)

- 访问 **Grafana Web**界面 [<http://<监控服务器IP地址>:3000>]
- 默认初始用户[admin], 初始密码[admin] -> 测试已更改为[www223344]
- Grafana 首页左侧导航栏 -> Connections -> Data sources -> Add data source -> Prometheus
- 配置**Prometheus**数据源
  - URL: [<http://<Prometheus服务器IP地址>:<Prometheus端口>>] 默认端口为9090
  - 其它按需设置或默认
- 创建仪表盘
  - 添加图标面板**Dashboards**
  - Query部分, 使用PromQL查询GPU指标
    - dcgm-exporter
      - `nvidia_gpu_utilization_percent` (GPU 利用率)
      - `nvidia_gpu_memory_used_bytes` (显存已用)
      - `nvidia_gpu_temperature_celsius` (GPU 温度)
      - 其余参考 **dcgm-exporter** 的文档
    - Node-exporter
      - `curl http://172.20.88.178:9100/metrics | grep gpu`
      - 在命令行查询GPU相关信息, 再复制到Grafana -> query
  - 配置图标标题及轴标签等信息