在 vSphere Bitfusion 中启动应用程序

3

您可以在整个 GPU 内存中运行应用程序，也可以仅在内存的专用分区中运行应用程序。vSphere Bitfusion 可以使用单个 CLI 命令分配 GPU、运行应用程序和解除分配 GPU，或者可以使用各个命令执行相同的任务。

本章讨论了以下主题：

通过 run 命令运行应用程序

vSphere Bitfusion 客户端可以在远程共享 GPU 上运行机器学习应用程序。通过使用 `run` 命令，可在 vSphere Bitfusion 中启动单个应用程序。

用于启动应用程序的 vSphere Bitfusion 命令是 `run` 和必需参数（GPU 数量）。为将 vSphere Bitfusion 参数与应用程序区分开，请使用双连字符分隔符或将应用程序放在引号内。可以通过将占位符值替换为实际值并运行以下命令之一，在 vSphere Bitfusion 中启动应用程序。

- `bitfusion run -n num_gpus other switches -- applications and arguments`
- `bitfusion run -n num_gpus other switches "applications and arguments"`

通过运行 `run` 命令，可以执行以下三个任务。

1 从共享池分配 GPU

2 在应用程序执行 CUDA 调用时可访问 GPU 的环境中启动应用程序

3 在应用程序关闭时解除分配 GPU

`run` 命令封装了 `request_gpus`、`client` 和 `release_gpus` 命令。您可以使用各个命令分配 GPU 并在同一 GPU 上运行多个应用程序。有关详细信息，请参见通过预留 GPU 启动应用程序。

使用 RUN 命令分配 GPU

可以通过运行 `run` 命令为单个应用程序分配 GPU。应用程序在 GPU 的整个内存资源中运行。

使用 `run` 命令请求的所有 GPU 必须从单个 vSphere Bitfusion 服务器进行分配，并且服务器必须将 GPU 列为具有不同 PCIe 地址的单独设备。

例如，AI 应用程序 `asimov_i.py` 采用两个参数：GPU 数量和批处理大小。

- 当应用程序需要 1 个 GPU 时，运行 `bitfusion run -n 1 -- python asimov_i.py --num_gpus=1 --batchsz=64`
- 当应用程序需要 2 个 GPU 时，运行 `bitfusion run -n 2 -- python asimov_i.py --num_gpus=2 --batchsz=64`

默认情况下，vSphere Bitfusion 等待 30 分钟，以便有足够的 GPU 可用。要修改默认间隔，请使用 `--timeout` *`value`*, `-t` *`value`* 参数。输入超时（以秒为单位），或者时间和单位，例如秒 (s)、分钟 (m) 和小时 (h)。

例如，可以为 *value* 参数定义以下值。

10	10 秒
10s	10 秒
10m	10 分钟
10h	10 小时

对 GPU 内存进行分区

您可以在 GPU 内存的专用分区中运行您的应用程序，其他应用程序可以使用 GPU 的剩余内存。

GPU 分区参数是可选的 `run` 命令参数。可以通过参数，限定为仅在 GPU 内存的一个分区中运行您的应用程序。

- GPU 分区过程是动态的。启动带参数的 `run` 命令时，vSphere Bitfusion 会在应用程序运行之前分配分区，随后再解除分配分区。
- 同时共享 GPU 的应用程序通过使用单独的客户端进程、网络流、服务器进程和内存分区彼此隔离。
- vSphere Bitfusion 仅对 GPU 的内存（而不是计算资源）进行分区。应用程序严格包含在分配的内存分区中，但如果需要，它可以访问完整的计算资源。当需要相同的计算单元时，应用程序会争用计算资源，否则应用程序将同时运行。

可以使用 MB 为单位指定分区大小，也可以将其指定为总 GPU 内存的一部分。

对 GPU 内存大小进行分区，分成多个部分（数字 > 0.0 且 <= 1.0，例如 0.37）

```
bitfusion run -n num_gpus -p gpu_fraction -- applications and arguments
```

以 MB 为单位对 GPU 的内存大小进行分区

```
bitfusion run -n num_gpus -p MBs_per_gpu -- applications and arguments
```

有关详细信息，请参见 GPU 分区示例。

GPU 分区示例

多个并发应用程序可能会比单个应用程序更高效地使用 GPU 的计算容量。可以通过多种方法对 GPU 的内存进行分区。

如果使用的是模型规模较小或具有小批量工作任务（例如，图像数量）的推理应用程序，则可以在分区的 GPU 上同时运行这些应用程序。

可以执行实证测试以了解应用程序所需的内存大小。某些应用程序扩展为使用所有可用内存，但其性能在超出特定阈值后可能无法再提高。

以下示例假设您了解不同批处理大小的可接受内存要求。

- 预计批处理大小为 64 的应用程序需要使用 66% 的 GPU 内存时，请运行 `bitfusion run -n 1 -p 0.66 -- python asimov_i.py --num_gpus=1 --batchsz=64`
- 预计批处理大小为 32 的应用程序需要使用 5461 MB 的 GPU 内存时，请运行 `bitfusion run -n 1 -m 5461 -- python asimov_i.py --num_gpus=1 --batchsz=32`

请求多个 GPU 时，所有 GPU 都分配相同的内存量。部分大小规格必须基于内存量最小的 GPU。

在以下示例中，`-p` 参数会请求两个已请求 GPU 各自内存的 33%。GPU 必须在物理上位于同一个服务器上。如果 GPU 为 16 GB 设备，或者最小 GPU 为 16 GB 设备，则每个 GPU 上大约分配 5461 MB。当没有任何其他应用程序运行时，`asimov_i.py` 可以利用两个 GPU 的全部计算能力。

运行 `bitfusion run -n 2 -p 0.33 -- python asimov_i.py --num_gpus=1 --batchsz=64`

可以在同一个 GPU 上从一个客户端同时运行多个应用程序。

例如，要在后台启动两个并发应用程序实例，请运行以下两个命令。

1 `bitfusion run -n 1 -p 0.66 -- python asimov_i.py --num_gpus=1 --batchsz=64 &`

2 `bitfusion run -n 1 -p 0.33 -- python asimov_i.py --num_gpus=1 --batchsz=32 &`

NVIDIA System Management Interface (`nvidia-smi`)

可以通过运行 NVIDIA System Management Interface (`nvidia-smi`) 监控应用程序执行各种操作，例如，查看 GPU 分区大小或验证 vSphere Bitfusion 服务器上的可用资源。通常，在安装 NVIDIA 驱动程序时，服务器上会提供该应用程序。

在 vSphere Bitfusion 客户端中运行的应用程序不需要 NVIDIA 驱动程序，但可能需要 `nvidia-smi` 等应用程序，以了解 GPU 的功能或确定 GPU 内存大小。为了支持此类操作，从 vSphere Bitfusion 3.0 起，所有 `nvidia-smi` 客户端上均提供 vSphere Bitfusion 应用程序。vSphere Bitfusion 会将应用程序从服务器复制到客户端。

例如，要在 GPU 上请求 1024 MB 分区，请运行 `bitfusion run -n 1 -m 1024 -- nvidia-smi`。

nvidia-smi 应用程序的输出显示请求的分区值 1024MiB。

```
Requested resources:
Server List: 172.16.31.241:56001
Client idle timeout: 0 min
Wed Sep 23 15:21:17 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.100      Driver Version: 440.64.00    CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| 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 T4            Off  | 00000000:13:00.0 Off |                    0 |
| N/A   36C    P8     9W /  70W |      0MiB /  1024MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

在特定 GPU 和服务器上启动应用程序

自 vSphere Bitfusion 4.0 起，可以使用 CLI 命令参数筛选资源池中的 GPU，并在一组特定的 GPU 上启动应用程序。

可以在 run、request_gpus 和 list_gpus 命令中使用 --filter 参数，并对一组特定的 GPU 或服务器运行命令。还可以组合使用筛选器，列出满足多个条件的服务器和 GPU。对于每种数据类型，必须使用适当的运算符，例如，<、>、>=、<=、= 或 !=。

表 3-1. 可用 GPU 和服务器筛选器列表

筛选器	数据类型	描述
device.index	整数	GPU 的系统索引。例如，1。要查看 GPU 的索引，请运行 nvidia-smi 命令。
device.name	String	GPU 设备的型号名称。例如，NVIDIA Tesla T4。
device.memory	整数	GPU 设备的物理内存大小（以 MB 为单位）。例如，对于内存大小为 16 GB 的 GPU 设备，输入 16384。
device.capability	版本	NVIDIA 设备 CUDA 计算功能。CUDA 计算能力是一种机制，即 NVIDIA 与 CUDA API 配合使用以指定 GPU 支持的功能。值必须以 X.Y 格式输入。例如，8.0。有关详细信息，请参见 NVIDIA CUDA GPU 文档。

表 3-1. 可用 GPU 和服务器筛选器列表 （续）

筛选器	数据类型	描述
server.addr	String	vSphere Bitfusion 服务器的 IP 地址。
server.hostname	String	vSphere Bitfusion 服务器的主机名。
server.has-rdma	布尔	vSphere Bitfusion 服务器使用 RDMA 网络连接。
server.cuda-version	版本	vSphere Bitfusion 服务器上安装的 CUDA 版本。值必须以 X.Y 格式输入。例如，11.3。
server.driver-version	版本	vSphere Bitfusion 服务器上安装的 NVIDIA 驱动程序版本。值必须以 X、X.Y 或 X.Y.Z 格式输入。例如，460.73。

例如，要列出内存大小大于 16 GB 的 GPU 设备，请运行 `bitfusion list_gpus --filter "device.memory>16384"` 命令。

例如，要在仅具有 Ampere GPU 微架构的 GPU 设备上运行 AI 或 ML 工作负载，请运行 `bitfusion run -n 1 --filter "device.capability=8.0" -- workload` 命令。同样，要仅在具有 Volta GPU 微架构的 GPU 设备上运行工作负载，请运行 `bitfusion run -n 1 --filter "device.capability=7.0" -- workload` 命令。

注 具有 Ampere GPU 微架构的 GPU 设备的 CUDA 计算能力相当于 CUDA 版本 8.0，具有 Volta GPU 微架构的 GPU 设备的 CUDA 计算能力相当于 CUDA 版本 7.0。有关详细信息，请参见 NVIDIA CUDA GPU 文档。

通过预留 GPU 启动应用程序

您可以分配多个 GPU，并在同一 GPU 上运行多个应用程序。

虽然 `run` 命令可以分配 GPU、运行应用程序以及集中取消分配 GPU，但 vSphere Bitfusion 提供了三个命令来分别执行这些任务。通过使用单个命令，可将同一 GPU 用于多个应用程序，并且可在将 vSphere Bitfusion 集成到其他工具和工作流（如调度软件 SLURM）时更好地进行控制。

- 要分配 GPU，请运行 `request_gpus`。
- 要在应用程序执行 CUDA 调用时可访问 GPU 的环境中启动应用程序，请运行 `client`。
- 要解除分配 GPU，请运行 `release_gpus`。

  注 `request_gpus` 命令用于创建可转发到其他工具的文件和环境变量。这些工具可以使用相同的分配配置运行 `client` 命令。

`run` 命令的参数拆分到 `request_gpus` 和 `client` 命令中。

要了解各个命令的用法，请参见以下使用 AI 应用程序 `asimov_i.py` 的示例工作流。

1 要分配 GPU 以启动多个连续的应用程序，请运行 `bitfusion request_gpus -n 1 -m 5461`。

```
Requested resources:
Server List: 172.16.31.241:56001
Client idle timeout: 0 min
```

2 要通过运行 `client` 命令启动应用程序，请运行 `bitfusion client nvidia-smi`。

```
Wed Sep 23 15:26:02 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.100      Driver Version: 440.64.00    CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| 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 T4            Off  | 00000000:13:00.0 Off |                    0 |
| N/A   36C    P8    10W /  70W |      0MiB /  5461MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

3 要通过运行 `client` 命令启动另一个应用程序，请运行 `bitfusion client -- python asimov_i.py --num_gpus=1 --batchsz=64`。

4 要解除分配 GPU，请运行 `bitfusion release_gpus`。