WHITE PAPER–OCTOBER 2019

Using Bitfusion FlexDirect

Bitfusion Guide

vmware®

Table of Contents

Introduction

FlexDirect is Bitfusion's intuitive GPU virtualization software. Your GPU applications can run seamlessly and concurrently from all the nodes in your cluster, whether they have physically attached GPUs or not. Multiple configurations and use-cases are supported:

- Scheduling GPU resources for multiple applications and users
- Remote access of GPU resources
- Partitioning GPU resources (memory) for shared (simultaneous) use
- Aggregation of multiple GPUs for applications that can use them

This document assumes you have successfully installed and initialized FlexDirect on the GPU nodes and on the CPU nodes of your cluster, and that you have run the health check and fixed any problems it uncovered. (Start with the *System Requirements* and *Installation* guides if you have not.)

Starting FlexDirect Servers from the CLI

Normally the FlexDirect Daemon runs as a `systemd` service, but if you wish to execute it from the command line, follow these instructions. Launch the FlexDirect daemon on all GPU servers in your cluster by running the following (do not run this on servers without GPUs):

Shell

```
% flexdirect resource_scheduler [[-n num_gpus] | [-d devices]] [-l listen_address] [-s
SRS_listening_port] [-p server_listening_port]
```

The FlexDirect daemon is also known as the resource scheduler or SRS (Simple Resource Scheduler). SRS acts as a server for clients needing GPU services. It manages a set of its local, physically attached GPUs. It listens to FlexDirect clients on a specified TCP port. You may launch multiple instances of SRS, with unique TCP ports, on a server to manage distinct, separate subsets of GPUs. Clients may consume the GPU resources from multiple servers (SRS instances).

FLEXDIRECT RESOURCE_ SCHEDULER OPTIONS	EXPLANATION
`-n, --num_gpus <num>`	Use/manage the first <num> GPUs. Conflicts with --devices. Default: all local GPUs
`-d, --devices <list>`	A comma-separated list of GPU devices (by index) to be managed by this instance of SRS. If a server had three devices, then the complete list would be 0,1,2. Conflicts with --num_gpus. Default: defer to --num_gpus.

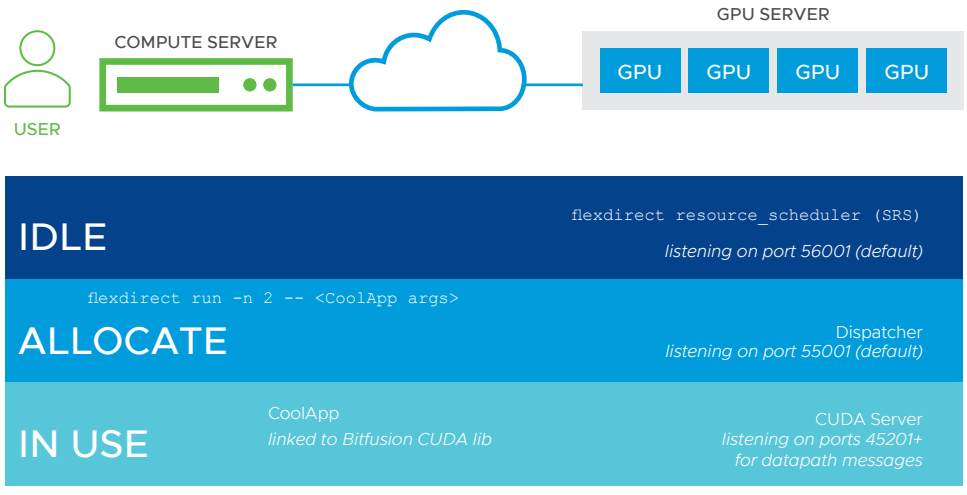
`-l, --listen <IP address>`	The IP address that this instance of SRS will listen to for messages from server. Default: 0.0.0.0 (listen on any address)
`-s, --srs_port <number>`	The TCP port that SRS will listen to for client requests. Default: 56001
`-p, --po`	The TCP port that a dispatcher will listen to for messages from the client application. Default: 55001

Sample Output

```
Running resource scheduler on 0.0.0.0:56001 using 1 GPUs (0) with 1024 MiB of memory
each
```

Be sure the ports in the range of 55001–55100, 56001–56100 and 45201–46225 are open on the GPU servers. By default, if no parameters are provided, all available GPUs are used, and FlexDirect SRS will listen on port 56001. This default configuration is suitable for many use-cases.

The drawing below shows the five processes that are running on a client (or CPU) node and on a server (or GPU) node when FlexDirect is running an application. It should help you understand the concepts, commands and usage that this manual discusses. Only two processes are directly launched by the user. These are the ones shown in a fixed font as you would type them in a command shell. The drawing also shows the TCP ports used by the GPU server processes.



Client Configuration and Cluster Health

Configure each client with a file, `/etc/bitfusionio/servers.conf` (or `~/.bitfusionio/servers.conf`), listing the IP addresses (with optional port) of all of the FlexDirect daemons that the client can access. Here is an example servers.conf showing different formats:

servers.conf

```
$ cat ~/.bitfusionio/servers.conf
172.31.51.20
172.31.51.26:56003
172.31.51.42 56003
```

A servers.conf file may use hostnames instead of IP address, but this is not a best practice. Since there is no easy way to ensure the IP that is resolved to is a valid, correct, performant or routed host, a hostname should be considered unsafe.

With the servers.conf file in place, you can verify the installation and find a list of available GPUs by running:

Shell

```
$ flexdirect list_gpus
- server 0 [100.98.0.16:56001]: running 0 tasks
       |- GPU 0: free memory 12000 MiB / 12000 MiB
       |- GPU 1: free memory 12000 MiB / 12000 MiB
       |- GPU 2: free memory 12000 MiB / 12000 MiB
       |- GPU 3: free memory 12000 MiB / 12000 MiB
```

Verifying Setup with Health Check

Running a health check from a client node will check all GPU servers as well as the client node:

Shell

```
$ flexdirect health
```

Using GPUs from the Client

To launch an application and use the shared GPUs, use `flexdirect run`. This may be invoked from any client node, including the GPU nodes as long as they have a `server.conf` file configured (See the *Installation* guide).

Shell

```
$ flexdirect run -n num_gpus [ [-p fraction_of_gpu] | [-m memory_per_gpu] ] --
<application and arguments>
```

FLEXDIRECT RUN OPTIONS	EXPLANATION
`-s, --servers <path>`	The location of the servers configuration file. Defaults to (in order of precedence): ~/.VMware Bitfusionio/servers.conf /etc/VMware Bitfusionio/servers.conf
`-w, --wait <time>`	Timeout in seconds or time+unit (m, h) when waiting for enough GPU to be available. Default: "10m" // (ten minutes)
`-m, --memory <size>`	Amount of GPU memory in MiB to be requested. Default: 0
`-p, --partial <value>`	Proportion of GPU memory to be requested. Default: 1.0 // 100%
`-n, --num_gpus <number>`	The number of shared GPUs to use. Default: 0
`-r, --rdma`	Connect through Infiniband (IB) RDMA.
`-l, --server_list <list>`	Specify the SRS servers. If <list> is non-empty, then the configuration file is not used. Format: "<ip_address>:[port];<ip_address>:[port];<ip_address>:[port]
`--`	Standard POSIX convention to ensure that any following arguments are not interpreted by FlexDirect itself, but passed to the application.

To verify that the GPUs are allocated as requested, you can use the `nvidia-smi` monitoring command with `flexdirect run <options>` at any point (from a separate terminal). For example, to verify the number and memory size of allocated GPUs, you can run the following which should show 1 GPU with 1024 MiB of available memory:

Shell

```
$ flexdirect run -n 1 -m 1024 nvidia-smi
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
    |-------------------------------+----------------------+----------------------+
    | 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 K80    Off  | 0000:01:00.0     Off |                  N/A |
    | N/A%   53C    P8    29W /  149W |      0MiB /  1024MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+

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

Alternatively, you could invoke a shell and allow for any applications to automatically use the configured GPU configuration, with RDMA support. The commands below have some comments inserted for clarity:

Shell

```
$ flexdirect run -n 1 -r bash
### -n 1 requests 1 GPU, -r requests an RDMA connection
### Now you are in a new shell with an allocated GPU available
### for any commands that follow.
$ nvidia-smi
...
$ python my_tensorflow_model.py
...
$ exit
### Now you are back in the original shell and the GPU has been freed.
$
```

Some of the output skipped above with ellipses above is shown below:

Shell

```
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
    |-------------------------------+----------------------+----------------------+
    | 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 K80           Off  | 0000:01:00.0     Off |                  N/A |
    | N/A%   53C   P8    29W / 149W |      0MiB / 11439MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+

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

my_tensorflow_model

step 0: error 100
    step 1: error 90
    step 2: error 91
    step 3: error 85
    step 4: error 73
    step 5: error 56
    step 6: error 53
    step 7: error 55
    step 8: error 42
    step 9: error 36
```

Running Applications on the Local Server

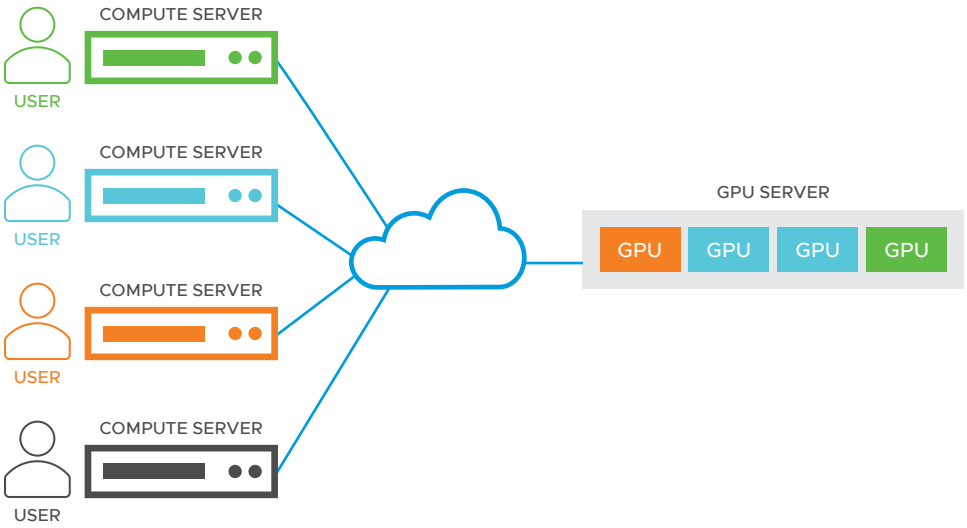
If you want to use the GPUs on the local server, just substitute the `local` command in place of the `run` command:

Shell

```
$ flexdirect local -p 0.5 nvidia-smi
```

NOTE
The number of GPUs option, `-n <num_gpus>`, is not supported with the `local` command.

Note:



Reserving GPUs

Alternatively, you can request GPUs and the configurations you need: Any number of GPUs and the amount of memory per GPU specified by either the `-m` or `-p` argument.

Shell

```
$ flexdirect request_gpus -n <num_gpus> [ [-p fraction_of_gpu] | [-m memory_per_gpu]
Requesting 1 GPUs with partial memory 1...
```

The above command reserves the specified number and configuration of GPUs and creates an adaptor configuration file (`~/.bitfusionio/adaptor.conf` by default). For example, here is the format of `adaptor.conf`:

adaptor.conf

```
$ cat ~/.bitfusionio/adaptor.conf
172.31.51.20
172.31.51.26
```

The configuration file can be used for running applications with:

Shell

```
$ flexdirect client -- <application>
```

To verify that the GPUs are allocated as requested, you can use the `nvidia-smi` monitoring command with `flexdirect client` at any point.

Shell

```
$ flexdirect client nvidia-smi
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
    |-------------------------------+----------------------+----------------------+
    | 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 K80    Off  | 0000:01:00.0             Off |                  N/A |
    | N/A%   53C   P8    29W / 149W |      0MiB / 11439MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+

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

See *Advanced Usage* for documentation of the `flexdirect client` command. Additionally, when using reserved GPUs, you should not specify the list of servers, as that was provided by the `flexdirect request_gpus` command.


```
flexdirect request_gpus -n 1 -p 0.5
flexdirect client -- <app>
```

Running `request_gpus` multiple times will automatically release the previous configuration to the pool before requesting a new configuration.

Releasing GPUs

After you are done using the reserved GPUs, you can release them with:

Shell

```
$ flexdirect release_gpus
Releasing GPUs from config file '/root/.bitfusionio/adaptor.conf'...
```

Advanced Networking Configuration

When starting a job, FlexDirect will explore the network interfaces, paths, and bandwidths available for CPU to GPU communication. Then it will select the most efficient means of communication, balancing the characteristics of one request against any concurrent requests.

By default, FlexDirect will explore all interfaces and transport layers. By means of a network configuration file, however, you can specify which network interfaces and which transport layers FlexDirect is allowed to use. If you wish, you may set the path to this file with an environmental variable like so:

BF_NETWORK_CONFIG=/path/to/the/config/file

If this variable is not set, FlexDirect will look first for the file /etc/bitfusionio/network.conf, and then for ~/.bitfusionio/network.conf. The file is optional; by default, FlexDirect will explore everything.

A network file only specifies ports and transports for the local machine, not for other servers in a cluster. Every CPU server and GPU server may have one, which limits which of its local ports or transport layers may be used.

The file itself is a YAML file that can specify two lists: a list of enabled interfaces named `enabled_interfaces` and a list of transports named `enabled transports`. Interfaces can be listed by their assigned IP addresses, or by name. Infiniband interfaces should be listed by device name only (use the `ibv_devices` command to list names). The legal values in the transports list are `tcp` and `infiniband`. Here is an example network.conf file listing two interfaces and two transports:

YAML

```yaml
---
# A list of ports FlexDirect can use
# (by ip address, 192.168.1.42, e.g.,
# or by port, e.g. eth0.
# Infiniband interfaces should be listed only by device name, e.g., usnic_0)
enabled_interfaces:
    - 10.10.10.42
    - 10.10.10.43
    - mlx5_0

# A list of transports FlexDirect can use
# (from tcp and infiniband).
enabled_transports:
    - tcp
    - infiniband
...
```

Detailed Usage

Run `flexdirect help [command]` for more information. Two examples are shown below:

Shell

```
NAME:
    flexdirect - Run application with Bitfusion FlexDirect.

USAGE:
   flexdirect <command> <options> “application”
   flexdirect <command> <options> -- [application]
   flexdirect help [command]

   For more information system requirements and advanced usage please visit
   docs.bitfusion.io

COMMANDS:
        init, i          Initialize configuration. Requires root privileges.
        version, v       Display full FlexDirect version.
        localhealth, LH  Run health check on current node only.
        upgrade, U       Upgrade version. Requires root privileges.
        uninstall        Uninstall FlexDirect. Requires root privileges.
        dealloc          Deallocate license certificate. Requires root privileges.
        crashreport      Send crash report to Bitfusion.
        license          Check license status.
        list_gpus        List the available GPUs in a shared pool.
        help, h          Shows a list of commands or help for one command.

   Client Commands:
        client, c     Run application.
        health, H     Run health check on all specified servers and current node.
        request_gpus  Request GPUs from a shared pool.
        release_gpus  Release GPUs back into a shared pool. Options must match a
                      previous request_gpus command.
```

```
        run           Request GPUs from a shared pool, run a client command, then
                      release the GPUs.
        stats         Gather stats from all servers.
        smi           Display smi-like info for all servers.
        local         Run a CUDA application locally.
        net_perf      Gather network performance data from all SRS servers.
   Server Commands:
        server, s           Run server.
        resource_scheduler  Run FlexDirect resource scheduler (SRS) on GPU server.


EXAMPLES:
   $ sudo flexdirect init -l <license_key>

   $ flexdirect resource_scheduler --srs_port 50001

   $ flexdirect run -n 4 -- <application>
```

Shell

```
$ flexdirect help run
NAME:
   flexdirect run - Request GPUs from a shared pool, run a client command, then release
   the GPUs

USAGE:
   flexdirect run [command options] [arguments...]

CATEGORY:
   Client Commands

OPTIONS:
 --capture value                Capture debug info: combination of logs, tracing,
                                stats and coredump, or all to include all.
 --servers value, -s value      Location of the servers config file. By default,
                                ~/.bitfusionio/servers.conf is used. If it doesn't
                                exist, /etc/bitfusionio/servers.conf is used.
 --timeout value, -t value      Timeout in sec when waiting for enough GPU to be
                                available, defaults to 10min (default: "10m")
 --memory value, -m value       Amount of GPU memory in MiB to be requested. Default is
                                all GPU memory. (default: 0)
 --partial value, -p value      Proportion of GPU memory to be requested. Default is
                                1.0. (default: 1)
 --num_gpus value, -n value     The number of shared GPUs to use (default: 0)
 --rdma, -r                     Connect through infiniband RDMA.
 --server_list value, -l value  Specify the SRS servers. If it is non-empty, then
                                config file is not used. Format:"<ip_address>:<port>;<ip_
                                address>:<port>;...". If no port is provided for an
                                address, 56001 is assumed.
 --verbosity value, -v value    Specify the level of verbosity from 0-4 inclusive
                                (default: 1)
```

Dividing GPU Resources by GPU Type

Support for a heterogenous cluster of GPUs is natively supported. A user can request GPUs from a different *pool* which has the desired type or generation of GPUs. Simply edit and use different versions of the /etc/bitfusion/services.conf file mentioned earlier:

Shell

```
$ cat K80_servers.conf
10.10.10.1
10.10.10.2

$ cat M60_servers.conf
10.10.20.1
10.10.20.2
10.10.20.3

$ cat P100_servers.conf
10.10.30.1
10.10.30.2
10.10.30.3
10.10.30.4
```

Then use the -s parameter when requesting or using GPUs from the client side:

Shell

```
$ flexdirect run -n 2 -s M60_servers.conf [--] <commmand>
$ flexdirect run -n 1 -s K80_servers.conf [--] <commmand>

$ flexdirect request_gpus -n 2 -s M60_servers.conf
$ flexdirect client [-]- <command>
$ flexdirect request_gpus -n 1 -s K80_servers.conf
$ flexdirect client [--] <command>
```

For example:

Shell

```
$ flexdirect run -n 2 -s M60_servers.conf nvidia-smi
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 384.111                Driver Version: 384.111                   |
    |-------------------------------+----------------------+----------------------+
    | 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 M60           On   | 00000000:00:1D.0 Off |                    0 |
    | N/A   23C    P8    13W / 150W |      1MiB /  7613MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+
    |   1  Tesla M60           On   | 00000000:00:1E.0 Off |                    0 |
    | N/A   28C    P8    13W / 150W |      1MiB /  7613MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+

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

$ flexdirect run -n 1 -s K80_servers.conf nvidia-smi
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
    |-------------------------------+----------------------+----------------------+
    | 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 K80           Off  | 0000:01:00.0     Off |                  N/A |
    | N/A%   53C   P8    29W / 149W |      0MiB / 11439MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+

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

Shell

```
$ flexdirect request_gpus -n 2 -s M60_servers.conf
Requesting 2 GPUs ...
$ flexdirect client nvidia-smi
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 384.111                Driver Version: 384.111                   |
    |-------------------------------+----------------------+----------------------+
    | 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 M60           On   | 00000000:00:1D.0 Off |                    0 |
    | N/A   23C    P8    13W / 150W |       1MiB / 7613MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+
    |   1  Tesla M60           On   | 00000000:00:1E.0 Off |                    0 |
    | N/A   28C    P8    13W / 150W |      1MiB /  7613MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+

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

$ flexdirect request_gpus -n 1 -s K80_servers.conf
Requesting 1 GPU ...
$ flexdirect client nvidia-smi
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
    |-------------------------------+----------------------+----------------------+
    | 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 K80           Off  | 0000:01:00.0     Off |                  N/A |
    | N/A%   53C   P8    29W / 149W |      0MiB / 11439MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+

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

Dividing GPU Resources into Multiple Pools

You may wish to divide the available resources on the same GPU server among multiple users. In this case, you must watch out for two things:

- Make sure that the GPUs used by each resource scheduler (SRS process instance) do not overlap. This is controlled by the `--devices` flag. It is not enough to just use the `--num_devices` flag, as that will always use the first n devices.
- Make sure the ports used by each scheduler do not overlap. This includes the main FlexDirect port, controlled by `--srs_port`, but also the port range used by spawned server processes, controlled by `--port`.

For example, the following commands will divide four GPUs between two resource schedulers (two SRS process instances) on a single server. They have to be run either in separate terminals or in the background.

Shell

```
$ flexdirect resource_scheduler --devices 0,1 --srs_port 56001 --port 56002 &
Running resource scheduler on 0.0.0.0:56001 using 2 GPUs (0,1) with 1024 MiB of memory
each

$ flexdirect resource_scheduler --devices 2,3 --srs_port 57001 --port 57002 &
Running resource scheduler on 0.0.0.0:57001 using 2 GPUs (2,3) with 1024 MiB of memory
each
```

Usage Examples

Run a command with one GPU:

Shell

```
$ flexdirect run -n 1 nvidia-smi
```

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
|-------------------------------+----------------------+----------------------+
| 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 K80           Off  | 0000:01:00.0     Off |                  N/A |
| N/A%   53C   P8    29W / 149W |      0MiB / 11439MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

Run a command with one GPU with 1/2 of total GPU memory:

Shell

```
$ flexdirect run -n 1 -p 0.5 nvidia-smi
```

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
|-------------------------------+----------------------+----------------------+
| 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 K80           Off  | 0000:01:00.0     Off |                  N/A |
| N/A%   53C   P8    29W / 149W |      0MiB /  5719MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

Run a command with one GPU with 1GB of memory:

Shell

```
$ flexdirect run -n 1 -m 1024 nvidia-smi
```

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
|-------------------------------+----------------------+----------------------+
| 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 K80           Off  | 0000:01:00.0     Off |                  N/A |
| N/A%   53C   P8    29W / 149W |      0MiB /  1024MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

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

Run a command using eight GPUs with 1/3 of total memory each:

Shell

```
$ flexdirect run -n 8 -p 0.33 python my_tensorflow_model.py
...
step 0: error 100
    step 1: error 90
    step 2: error 91
    step 3: error 85
    step 4: error 73
    step 5: error 56
    step 6: error 53
    step 7: error 55
    step 8: error 42
    step 9: error 36
...
```

Reserve four GPUs with 2GB of memory each, run multiple commands, then release the GPUs. Reserving the GPUs instead of using `flexdirect run` ensures that the same four GPUs are used for all commands.

Shell

```
$ flexdirect request_gpus -m 2048 -n 4

...

$ flexdirect client nvidia-smi
    +-----------------------------------------------------------------------------+
    | NVIDIA-SMI 375.26                 Driver Version: 375.26                    |
    |-------------------------------+----------------------+----------------------+
    | GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
    | Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
    |===============================+======================+======================|
    |   0  GeForce GTX 1050    Off  | 0000:01:00.0     Off |                  N/A |
    | 35%   28C    P8    35W /  75W |      0MiB /  2048MiB |      0%      Default |
    |   0  GeForce GTX 1050    Off  | 0000:01:00.0     Off |                  N/A |
    | 35%   28C    P8    35W /  75W |      0MiB /  2048MiB |      0%      Default |
    |   0  GeForce GTX 1050    Off  | 0000:01:00.0     Off |                  N/A |
    | 35%   28C    P8    35W /  75W |      0MiB /  2048MiB |      0%      Default |
    |   0  GeForce GTX 1050    Off  | 0000:01:00.0     Off |                  N/A |
    | 35%   28C    P8    35W /  75W |      0MiB /  2048MiB |      0%      Default |
    +-------------------------------+----------------------+----------------------+
    +-----------------------------------------------------------------------------+
    | Processes:                                                       GPU Memory |
    |  GPU       PID  Type  Process name                               Usage      |
    |=============================================================================|
    |  No running processes found                                                 |
    +-----------------------------------------------------------------------------+

$ flexdirect client -r python my_tensorflow_model.py
...
step 0: error 100
    step 1: error 90
    step 2: error 91
    step 3: error 85
    step 4: error 73
    step 5: error 56
    step 6: error 53
    step 7: error 55
    step 8: error 42
    step 9: error 36
...
$ flexdirect release_gpus
```

Upgrading FlexDirect

FlexDirect provides the `upgrade` command to install a new version. By default, it installs the latest stable version. As of the v1.9 release, you may specify any other released version.

Shell

```
# Install the latest stable release.
$ sudo flexdirect upgrade

# Install a specific release. May be older than the current installation.
# In the example below, install version 1.8.3.
$ sudo flexdirect upgrade -v 1.8.3.
```

NOTE

Remember to kill the resource scheduler first when changing FlexDirect versions on a GPU server, and then to launch it again afterwards.

Licensing

FlexDirect is a licensed software product. When you purchase the software, Bitfusion will send you one or more license keys (a string of characters), each of which can license several CPU or GPU servers. Any server running FlexDirect authorizes features by contacting an external Bitfusion license server and by means of three local files in `/opt/bitfusionio/lic`.

- `license_key` contains the key
- `license.txt` contains a license certificate generated by the external Bitfusion server
- `host_id` contains an identity number, unique to the host, generated by the FlexDirect software

The Bitfusion server will validate certificates when FlexDirect runs.

The Bitfusion license server also generates new certificates when FlexDirect sends requests containing valid host IDs and license keys. The server will not generate certificates beyond the number authorized by the key.

If you need to migrate a FlexDirect installation to a new server, you should deallocate its existing certificate on the old server, prior to requesting a certificate on the new server. Deallocation informs the Bitfusion server that the old certificate is invalid and no longer counts toward the total number authorized by the key.

When FlexDirect is installed, it will automatically try to obtain a license certificate, but you may also explicitly request a license with the `init` command. You might do this if, for example, your old license has expired, and you purchase a new key.

Text

```
$ # Allocate a new license from the Bitfusion server and copy to
$ # /opt/bitfusionio/lic/license.txt
$ # Using “-l <license key string>” to avoid interactive prompt
$ flexdirect init -l 0123456780ABCDEF    # example, but invalid, key,
```

You may deallocate a license certificate with the `dealloc` command:

Text

```
$ # Deallocate the license certificate from the local machine and from
$ # the Bitfusion server, thus I can request/re-allocate this license on a new host.
$ flexdirect dealloc
```

If you need to license servers that cannot connect to the external license server, please contact your Bitfusion representative or *askbitfusion@vmware.com*

Crashreport

FlexDirect has a crash report command that gathers run-time log files and installation log files as well as health check output from your cluster and sends everything in an email to *askbitfusion@vmware.com*. The email will open a support ticket at Bitfusion. Feel free to use this command as a means of reporting issues you encounter using FlexDirect. You might want to supplement this report with your own email, also to *askbitfusion@vmware.com*, describing your issue.

Text

```
$ # The following will report local data as well as data from
$ # the servers in /etc/bitfusionio/servers.conf.
$ # It will run in interactive mode, so you will need to
$ # respond to prompts
$
$ flexdirect crashreport


$ # The following will report local data as well as data from
$ # the servers in /my/random/servers.conf.
$ # (-y option) Does not seek confirmation from prompt.
$
$ flexdirect crashreport -y -s /my/random/servers.conf
```

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 vmware.com