VMware vSphere Bitfusion User Guide

22 JAN 2021

VMware vSphere Bitfusion 2.5

vmware®

You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

About *VMware vSphere Bitfusion User Guide*

1

The *VMware vSphere Bitfusion User Guide* provides information about using and configuring VMware vSphere® Bitfusion®.

The *VMware vSphere Bitfusion User Guide* describes how to allocate, partition, and attach GPUs to workloads, and how to configure and monitor vSphere Bitfusion.

Intended Audience

This guide is intended for advanced users who are familiar with ESXi, vCenter Server, and command-line interface (CLI).

Updated Information

This *VMware vSphere Bitfusion User Guide* is updated with each release of the product or when necessary.

This table provides the update history of the *VMware vSphere Bitfusion User Guide*.

Revision	Description
22 JAN 2021	■ Added a new topic how to create and download server logs by using command-line interface: Create vSphere Bitfusion Server Logs. ■ Added information about managing VMware vSphere Bitfusion in Chapter 4 Managing vSphere Bitfusion with the vSphere Bitfusion Plug-In. ■ Added information how to reuse a virtual machine or the underlying hardware in Remove a vSphere Bitfusion Server. ■ Minor update to Restore a vSphere Bitfusion Cluster.
03 DEC 2020	■ Minor updates to reflect the graphic user-interface in Remove a vSphere Bitfusion Server, Disable or Delete a vSphere Bitfusion Client, and Back Up a vSphere Bitfusion Cluster. ■ Minor update to vSphere Bitfusion Commands Reference.
05 NOV 2020	Initial release.

Understanding VMware vSphere Bitfusion

2

VMware vSphere Bitfusion virtualizes hardware accelerators such as graphical processing units (GPUs) to provide a pool of shared, network-accessible resources that support artificial intelligence (AI) and machine learning (ML) workloads.

vSphere Bitfusion Architecture and Components

vSphere Bitfusion has a client-server architecture. The product allows multiple client virtual machines (VMs) running artificial intelligence (AI) and machine learning (ML) applications to share access to remote GPUs on virtual machines running vSphere Bitfusion server software. You run the applications on the vSphere Bitfusion client machines, while the GPUs that provide acceleration are installed on the vSphere Bitfusion server machines across a network.

vSphere Bitfusion Server

vSphere Bitfusion server runs on an ESXi host with locally installed GPUs as a VMware appliance, which is a preconfigured virtual machine (VM) with prepackaged software and services. The server requires access to the local GPUs, usually through VMware vSphere® DirectPath I/O™.

vSphere Bitfusion Client

vSphere Bitfusion client runs on VMs which run the AI and ML applications.

vSphere Bitfusion Plug-In

The vSphere Bitfusion servers register a vSphere Bitfusion Plug-in with VMware vCenter Server. The plug-in provides monitoring and management of vSphere Bitfusion clients and servers.

vSphere Bitfusion Cluster

vSphere Bitfusion cluster is the set of all vSphere Bitfusion servers and clients in a vCenter Server instance.

vSphere Bitfusion Group

The vSphere Bitfusion client creates a vSphere Bitfusion group during the installation process. Only the members of the group can use vSphere Bitfusion. Certain configuration files are set

up with appropriate permissions and the members of the group inherit appropriate limits to work effectively with vSphere Bitfusion.

vSphere Client

The vSphere Client lets you connect to vCenter Server instances by using a Web browser, so that you can manage your vSphere infrastructure. You access the vSphere Bitfusion Plug-in through the vSphere Client.

Command-Line Interface (CLI)

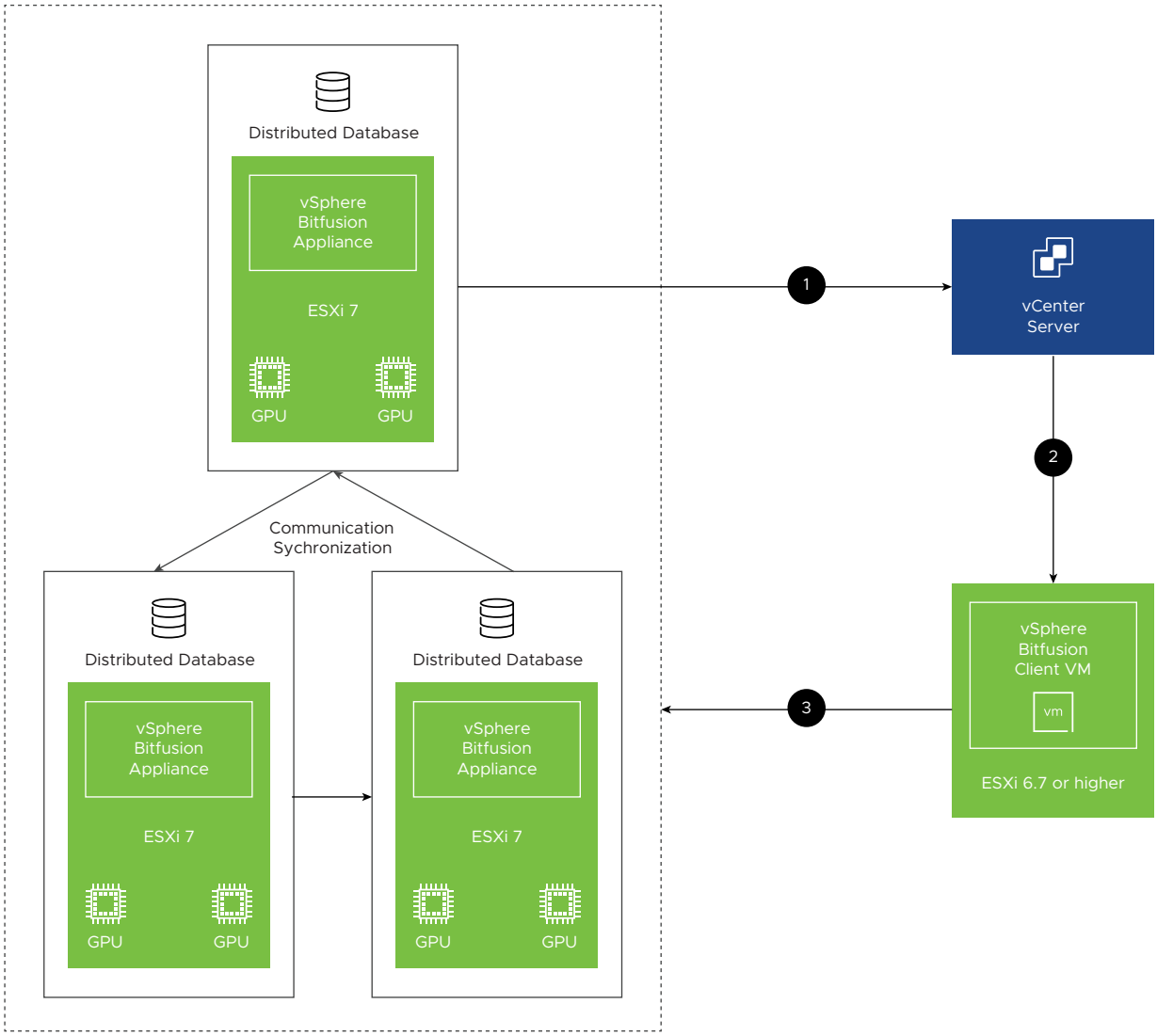
You can manage vSphere Bitfusion servers and clients by using command-line interface (CLI) commands.

vCenter Server

vCenter Server is the server management software that provides a centralized platform for controlling your vSphere environment.

The following figure is an example of a small vSphere Bitfusion cluster, such as a set of vSphere Bitfusion server-client machines and vCenter Server on a switched network. A minimal vSphere Bitfusion cluster configuration is one client, one server, and one vCenter Server. You can create large clusters with multiple clients and multiple servers.

Figure 2-1. Example of a small vSphere Bitfusion cluster



1 The primary vSphere Bitfusion server registers a vSphere Bitfusion Plug-in with vCenter Server.

2 The vSphere Bitfusion Plug-in enables a vSphere Bitfusion client VM.

3 The vSphere Bitfusion Client has authorized access to all vSphere Bitfusion servers in the vSphere Bitfusion cluster.

Note Before using VMware vSphere Bitfusion, you must deploy a vSphere Bitfusion server, and install and enable a vSphere Bitfusion client. For more information, see the *VMware vSphere Bitfusion Installation Guide*.

Benefits of vSphere Bitfusion

To run AI and ML applications, vSphere Bitfusion can perform the following tasks.

- Dynamically allocate and access GPU resources from vSphere Bitfusion servers.

  Applications can share GPU resources that are not dedicated to individual machines and you can run each application on a configured machine, container, and environment. Applications consume GPU acceleration services from a pool of vSphere Bitfusion servers across a network, and consume the resources only for the length of time that an application or session runs. GPUs return to the pool when applications or sessions complete.

- Access partitions of GPU resources for concurrent sharing with other applications.

  Another option to share GPUs is by partitioning the GPUs. The memory of a physical GPU can be divided into fractions of an arbitrary size and allocated to different applications at the same time. vSphere Bitfusion performs sharing with an interposition technology. vSphere Bitfusion intercepts API calls that normally address a local accelerator on a PCIe host bus and sends the API calls and related data across a network. vSphere Bitfusion provides sharing services for AI and ML applications, and supports the CUDA API to target NVIDIA GPUs.

Using vSphere Bitfusion with the Command-Line Interface

3

You can use, manage, and configure vSphere Bitfusion by using command-line interface (CLI) commands on the vSphere Bitfusion client.

You can use CLI commands to run applications in vSphere Bitfusion, partition GPU memory, and manage the vSphere Bitfusion service in several ways.

This chapter includes the following topics:

Starting Applications with the Run Command

The vSphere Bitfusion client can run machine learning applications on remote shared GPUs. By using the `run` command, you can start a single application in vSphere Bitfusion.

The vSphere Bitfusion command to start an application is `run` with a mandatory argument for the number of the GPUs. To distinguish vSphere Bitfusion arguments from applications, you use a double-hyphen separator or place the application within quotes. You start an application in vSphere Bitfusion by replacing the placeholder values with actual values and running one of the following commands.

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

By running the `run` command, you can perform the following three tasks.

1 Allocate GPUs from the shared pool

2 Start an application in an environment that can access the GPUs when the application makes CUDA calls

3 Deallocate the GPUs when the application closes

The `run` command encapsulates the `request_gpus`, `client`, and `release_gpus` commands. You can use the individual commands to allocate GPUs and run multiple applications on the same GPUs. For more information, see Starting Applications with Reserved GPUs.

Allocating GPUs with the RUN Command

You can run the `run` command to allocate GPUs for a single application. The application runs in the entire memory resource of the GPUs.

All GPUs that are requested by using the `run` command must be allocated from a single vSphere Bitfusion server, and the server must list the GPUs as separate devices with different PCIe addresses.

For example, the AI application, `asimov_i.py`, takes two arguments: the number of GPUs and a batch size.

- When the application expects 1 GPU, run `bitfusion run -n 1 -- python asimov_i.py --num_gpus=1 --batchsz=64`
- When the application expects 2 GPUs, run `bitfusion run -n 2 -- python asimov_i.py --num_gpus=2 --batchsz=64`

By default, vSphere Bitfusion waits for 30 minutes for enough GPUs to be available. To modify the default interval, use the`--timeout` *`value`*`, -t` *`value`* argument. Enter the timeout in seconds or time and unit, such as seconds (s), minutes (m), and hours (h).

For example, you can define the following values for the *value* argument.

10	10 seconds
10s	10 seconds
10m	10 minutes
10h	10 hours

Partitioning GPU Memory

You can run your application in a dedicated partition of a GPU's memory, and other applications can use the remaining GPU's memory.

The GPU partitioning arguments are optional `run` command arguments. You use the arguments to run your application only in a partition of a GPU memory.

- GPU partitioning is dynamic. A partition is allocated before an application runs and deallocated after.
- The applications that are sharing GPUs concurrently are isolated from each other by using separate client processes, network streams, server processes, and memory partitions.

- vSphere Bitfusion does not partition the GPU compute resource. The applications compete for compute resources when the same compute cells are required. Otherwise the applications run concurrently.

The partition size can be specified in MB or as a fraction of the total GPU memory.

Partitioning GPU memory size by fraction (number > 0.0 and <= 1.0, for example, 0.37)

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

Partitioning GPU's memory size by MB

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

For more information, see GPU Partitioning Examples.

GPU Partitioning Examples

Multiple concurrent applications might use a GPU's computational capacity more efficiently than a single application. There are several ways you can partition the memory of your GPUs.

If you are using inference applications with smaller model sizes or small batches of work, such as number of images, you can run the applications concurrently on partitioned GPUs.

You can perform empirical testing to understand the memory size an application requires. Some applications expand to use all available memory, but they might not achieve better performance beyond a certain threshold.

The following examples presume knowledge of acceptable memory requirements with different batch sizes.

- When you expect that an application with a batch size of 64 requires 66% of GPU memory, run `bitfusion run -n 1 -p 0.66 -- python asimov_i.py --num_gpus=1 --batchsz=64`
- When you expect that an application with a batch size of 32 requires 5461 MBs of GPU memory, run `bitfusion run -n 1 -m 5461 -- python asimov_i.py --num_gpus=1 --batchsz=32`

When you request multiple GPUs, all GPUs allocate the same amount of memory. The fraction size specification must be based on the GPU with the smallest amount of memory.

In the following example, the `-p` argument requests 33% of the memory of each of the two requested GPUs. The GPUs must physically reside on the same server. If the GPUs are 16 GB devices or if the smallest GPU is a 16 GB device, then approximately 5461 MB is allocated on each GPU. While no other applications are running, `asimov_i.py` can access the full compute power of the two GPUs.

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

You can run multiple applications from a single client on the same GPU concurrently.

For example, to start two concurrent application instances in the background, run both these commands.

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`)

You can run the NVIDIA System Management Interface `nvidia-smi` monitoring application, for example, to check your GPU partition size or verify the resources available on a vSphere Bitfusion server. The application is provided by the NVIDIA driver.

For example, to request a 1024 MB partition on a GPU, run `bitfusion run -n 1 -m 1024 -- nvidia-smi`.

The output of the `nvidia-smi` application displays the requested partition value of `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                                                 |
+-----------------------------------------------------------------------------+
```

Starting Applications with Reserved GPUs

You can allocate a number of GPUs and run multiple applications on the same GPUs.

While the `run` command allocates GPU, runs applications, and deallocates GPU collectively, vSphere Bitfusion has three individual commands to perform the same tasks. By using the individual commands, you can use the same GPU for multiple applications and have greater control when you are integrating vSphere Bitfusion into other tools and workflows, such as the scheduling software, SLURM.

- To allocate GPUs, run `request_gpus`.

- To start applications in an environment that can access the GPUs when the application makes CUDA calls, run `client`.
- To deallocate the GPUs, run `release_gpus`.

Note The `request_gpus` command creates a file and environment variables that can be forwarded to other tools. The tools can run the `client` command with the same allocation configuration.

The arguments of the `run` command are split between the `request_gpus` and `client` commands.

To understand the use of the individual commands, see the following example workflow that is using the AI application `asimov_i.py`.

1 To allocate GPUs to start multiple and sequential applications, run `bitfusion request_gpus -n 1 -m 5461`.

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

2 To start an application by running the `client` command, run `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 To start another application by running the `client` command, run `bitfusion client -- python asimov_i.py --num_gpus=1 --batchsz=64`.

4 To deallocate the GPUs , run `bitfusion release_gpus`.

Start and Stop the vSphere Bitfusion Service

You can stop and start vSphere Bitfusion to make a configuration change or perform debugging.

vSphere Bitfusion runs as a regular application on both vSphere Bitfusion servers and clients. A `systemd` service starts the vSphere Bitfusion server software when the vSphere Bitfusion server starts. To stop, start, and restart the vSphere Bitfusion service or check the service log, you must access a vSphere Bitfusion server by using command line. The `systemd` file is in `/lib/systemd/system/bitfusion-manager.service`.

Note Typically, administrators and users do not interact with the vSphere Bitfusion server from the CLI. The interaction must be performed by using the vSphere Bitfusion Plug-in.

Procedure

1 Open a terminal application and run `ssh customer@`*`ip_address`*.

You can obtain the vSphere Bitfusion server IP address from the vSphere Bitfusion Plug-in.

2 Enter the customer password that you specified during the deployment of the vSphere Bitfusion open virtual appliance (OVA).

3 Start, stop, or monitor the vSphere Bitfusion service.

You can use the alias `bitfusion` for `bitfusion-manager.service`.

Action	CLI Command
Check the Bitfusion service	`sudo systemctl status bitfusion`
Stop the Bitfusion service	`sudo systemctl stop bitfusion`
Start the Bitfusion service	`sudo systemctl start bitfusion`
Restart the Bitfusion service	`sudo systemctl restart bitfusion`
Check the Bitfusion service log	`sudo journalctl -u bitfusion-manager.service` **Note** You cannot use an alias.

Create vSphere Bitfusion Server Logs

Since vSphere Bitfusion 2.5, you can run a support script that gathers essential server information and creates a bundle of the server logs. These logs provide important information when troubleshooting a server.

The support script gathers information about the setup and configuration of vSphere Bitfusion, driver versions, health check output, status of all servers and the server's hardware, Cassandra configuration and status, and others. Typically, you need the support bundle when working with VMware support.

Procedure

1 Log into the terminal of a running vSphere Bitfusion server.

2 Run `sudo bitfusion-supportbundle.sh`.

Results

You created a support bundle in `/tmp/bitfusion-supportbundle.tar.gz`.

vSphere Bitfusion Configuration Files

After you start a vSphere Bitfusion server instance, vSphere Bitfusion creates and maintains `servers.conf` and `bitfusion-limits.conf` configuration files on the client virtual machines (VMs). The client VMs must be deployed on ESXi hosts that are part of the same vCenter Server environment as the vSphere Bitfusion server instance.

Servers Configuration File

vSphere Bitfusion creates a high-priority user-specific file in `~/.bitfusion/servers.conf`. Alternatively, you can create a system file `/etc/bitfusion/servers.conf`, which vSphere Bitfusion uses with a lower priority than the user-specific file. You use the `cat` command to display a server list.

To understand the command use, see the following example.

```
cat ~/.bitfusion/servers.conf
```

The servers configuration file lists the IPv4 addresses of all vSphere Bitfusion servers and ports that a vSphere Bitfusion client can access. The default port 56001 is not listed.

```
172.31.51.20
172.31.51.26:56003
172.31.51.42 56003
```

You can run the `run` command with an alternative vSphere Bitfusion server list that is a subset of the primary server list of GPU servers maintained by vSphere Bitfusion in the `~/.bitfusion/servers.conf` file. To create a subset list of vSphere Bitfusion servers, you can perform one of the following steps. vSphere Bitfusion supports IPv4 addresses only.

- You can use `--servers` *`value`*`, -s` *`value`* and supply a subset of the primary server list in a file of your choice. You must change the *value* argument with a filepath to a `servers.conf` file.
- You can use `--server_list` *`value`*`, -l` *`value`* and supply a subset of the primary list of servers in the command line. You must change the *value* argument to a "*`ip_address`*`:`*`port`*`;`*`ip_address`*`:`*`port`*" format.

  You must enclose the list within quotes, because a semicolon is used as a separator when you list multiple addresses and the command-line interpreter can parse the list as multiple commands.

Limits Configuration File

The following limits apply to members of the vSphere Bitfusion group. Any user of the vSphere Bitfusion client must be a member of the vSphere Bitfusion group.

The `bitfusion-limits.conf` configuration file is installed on the vSphere Bitfusion client in `/etc/security/limits.d/bitfusion-limits.conf` by the client package. The file contains the following settings, which you can view and enforce by using the standard Linux utility, ulimit.

- Maximum number of open files

```
@bitfusion soft nofile 100000
@bitfusion hard nofile 100000
```

- Unlimited locked-in-memory address space

```
@bitfusion soft memlock unlimited
@bitfusion hard memlock unlimited
```

- Unlimited maximum resident set size

```
@bitfusion soft rss unlimited
@bitfusion hard rss unlimited
```

Note If the resource limit for open files is too low, vSphere Bitfusion might receive a connection error: `Cannot allocate memory error`. To resolve this issue, set the open files limit to 4096 or higher by running the `ulimit -n 4096` command.

vSphere Bitfusion Commands Reference

This section lists the most important vSphere Bitfusion CLI commands and their tasks. Additional CLI commands can be provided by the VMware support team.

Allocate GPUs

To allocate a number of GPUs for a single application, run the `run` command.

To allocate a number of GPUs and start a session, wherein you can run multiple applications on the same GPUs, run the `request_gpus`.

Start Applications in the vSphere Bitfusion Environment Accessing the GPUs

To start a single application, run the `run` command.

To start multiple applications in a session started with the `request_gpus` command, run the `client` command.

Deallocate the GPUs

To deallocate GPUs in a session started with the `request_gpus` command, run the `release_gpus` command.

List Available GPUs

To verify a vSphere Bitfusion server installation and find a list of available GPUs, run the `list_gpus` command.

```
- server 0 [172.31.51.20:56001]: 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
- server 1 [172.31.51.26:56003]: running 0 tasks
|- GPU 0: free memory 12000 MiB / 12000 MiB
|- GPU 1: free memory 12000 MiB / 12000 MiB
- server 2 [172.31.51.42:56003]: running 0 tasks
|- GPU 0: free memory 12000 MiB / 12000 MiB
|- GPU 1: free memory 12000 MiB / 12000 MiB
```

Run a Health Check

You can access the health check from the command line.

- To check the health of all vSphere Bitfusion servers and the Bitfusion client, run `bitfusion health`.
- To check the health of a single vSphere Bitfusion client or server, run `bitfusion localhealth`.

Check vSphere Bitfusion Version

To check the version of vSphere Bitfusion that is installed, run the `version` command.

```
Bitfusion version: 2.5.0 release
```

Display GPU Information

To display GPU information, run the `smi` command. Alternatively, to receive a similar output, you can start the `nvidia-smi` application with the `run` command.

```
+-----------------------------------------------------------------------------------------+
| 172.16.31.243:56001                                    Driver Version: 440.64.00 |
+-------------------------------+----------------------+----------------------+
| GPU  Name              Persistence-M | Virt Mem    Alloc / All | BusId  Vol Uncorr ECC |
| Fan  Temp  Perf        Pwr:Usage/Cap | Phy Mem     Used  / All | GPU-Util   Compute M. |
|===============================+======================+======================|
| 0    Tesla T4               Disabled | 0       MB / 15109   MB | 00000000:13:00.0    0 |
| 0 %   36C  P8             10W /  70W | 11      MB / 15109   MB |   0%          Default |
+-------------------------------+----------------------+----------------------+
+-----------------------------------------------------------------------------------------+
| 172.16.31.241:56001                                                                     |
+-----------------------------------------------------------------------------------------+
```

Test the Bandwidth

To test the bandwidth and latency between the vSphere Bitfusion client and servers, run the `net_perf` command.

Single network interface

```
Displayed results are calculated from round-trip measurements
BW(1MB) = 1000/(LAT(1MB) - LAT(1B))

[ <client>] ens160 => [10.202.8.169] net1 ( tcp) Single packet lat = 51 us, bw(1MB) = 1.71
GB/s
[ <client>] ens160 => [10.202.8.185] net1 ( tcp) Single packet lat = 48 us, bw(1MB) = 1.09
GB/s
[ <client>] ens160 => [10.202.8.233] net1 ( tcp) Single packet lat = 50 us, bw(1MB) = 0.87
GB/s
```

Multiple network interfaces

```
Displayed results are calculated from round-trip measurements
BW(1MB) = 1000/(LAT(1MB) - LAT(1B))

[ <client>] ens160 => [10.202.8.169] net1 ( tcp) Single packet lat = 51 us, bw(1MB) = 1.71
GB/s
[ <client>] ens160 => [10.202.8.185] net1 ( tcp) Single packet lat = 48 us, bw(1MB) = 1.09
GB/s
[ <client>] ens160 => [10.202.8.233] net1 ( tcp) Single packet lat = 50 us, bw(1MB) = 0.87
GB/s
[ <client>] ens192f0 => [10.202.8.169] net2 ( tcp) Single packet lat = 47 us, bw(1MB) = 2.14
GB/s
[ <client>] ens192f0 => [10.202.8.185] net2 ( tcp) Single packet lat = 49 us, bw(1MB) = 1.11
GB/s
[ <client>] ens192f0 => [10.202.8.233] net2 ( tcp) Single packet lat = 50 us, bw(1MB) = 1.15
GB/s
[ <client>] vmw_pvrdma0 => [10.202.8.169] vmw_pvrdma0 (infiniband) Single packet lat = 19 us,
bw(1MB) = 3.66 GB/s Single packet Write lat = 8 us, bw = 10.101 GB/s
[ <client>] vmw_pvrdma0 => [10.202.8.185] vmw_pvrdma0 (infiniband) Single packet lat = 21 us,
bw(1MB) = 3.45 GB/s Single packet Write lat = 8 us, bw = 10.5263 GB/s
[ <client>] vmw_pvrdma0 => [10.202.8.233] vmw_pvrdma0 (infiniband) Single packet lat = 21 us,
bw(1MB) = 3.46 GB/s Single packet Write lat = 8 us, bw = 10.4167 GB/s
```

Request Help

To get the full list of vSphere Bitfusion CLI commands or more information about a specific command, run the `help` command.

```
NAME:
    bitfusion - Run application with VMware Bitfusion

USAGE:
   bitfusion <command> <options> "application"
   bitfusion <command> <options> -- [application]
   bitfusion help [command]
```

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

COMMANDS:
        tls-certs, TC    Manage TLS certificates used by bitfusion server.  Requires root
privileges.
        version, v       Display full Bitfusion version
        localhealth, LH  Run health check on current node only
        dealloc          Deallocate license certificate.  Requires root priviledges.
        crashreport      Send crash report to bitfusion
        license          Check license status
        list_gpus        List the available GPUs in a shared pool
        initdb           Init database setup
        token            Fetch and manipulate tokens
        register         Register remote server as the plugin
        unregister       Unregister remote plugin
        removenode       Remove unavailable nodes
        user             Manage bitfusion users
        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 dispatcher service - listens for 'bitfusion client'
commands
        resource_scheduler, srs  Run Bitfusion resource scheduler (SRS) on GPU server
        analytics                Run Bitfusion analytics server
        manager                  Run Bitfusion manager server

EXAMPLES:
   $ sudo bitfusion init -l <license_key>

   $ bitfusion resource_scheduler --srs_port 50001

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

Managing vSphere Bitfusion with the vSphere Bitfusion Plug-In

4

You can manage, configure, and monitor vSphere Bitfusion by using the vSphere Bitfusion Plug-in.

After a vSphere Bitfusion server starts for the first time, the server registers a plug-in with vCenter Server. Any additional vSphere Bitfusion servers and clients must be enabled to join a vSphere Bitfusion cluster and to use the vSphere Bitfusion Plug-in.

The vSphere Bitfusion Plug-in provides a graphical user interface (GUI) in the main navigation pane and the drop-down menu of vCenter Server. The GUI displays the following data.

- GPU allocation
- Memory and compute resources use
- Network traffic
- Logging reports
- Health reports

You can use the plug-in to manage allocation limits and idle intervals. You can also perform other management functions, such as ending client connections, gracefully taking servers offline, and removing hosts from the vSphere Bitfusion cluster.

Managing vSphere Bitfusion with the vSphere Client

- You can create a snapshot of a vSphere Bitfusion server virtual machine (VM), but first you must power off the VM. Taking a snapshot while the VM is powered on may result in failure of the operation due to the pass-through devices that are connected to the server.
- You can perform a graceful shutdown or restart of a server VM by using the **Shut Down Guest OS** and **Restart Guest OS** options in the vSphere Client. Using the power on, power off, suspend, and reset options may result in failure of the vSphere Bitfusion appliance.

This chapter includes the following topics:

- View vSphere Bitfusion Server Logs
- Perform a Health Check of a vSphere Bitfusion Server
- vSphere Bitfusion Health Checks List
- View GPU Information for a vSphere Bitfusion Client
- View GPU Information for a vSphere Bitfusion Server
- Set a Global Display Refresh Interval
- Change the Settings of a vSphere Bitfusion Client
- Change the Settings of a vSphere Bitfusion Server

Add Additional vSphere Bitfusion Servers

You can add additional servers to your vSphere Bitfusion cluster when you require more GPU resources. To add a new server in a vSphere Bitfusion cluster, first you deploy the vSphere Bitfusion appliance on a virtual machine (VM), enable passthrough of the GPU to the vSphere Bitfusion server VM, customize the vSphere Bitfusion OVF template, and enable the VM as vSphere Bitfusion server.

After the first vSphere Bitfusion server starts, vSphere Bitfusion registers a vSphere Client Plug-in in the vCenter Server, resulting in a single vSphere Bitfusion cluster containing one vSphere Bitfusion server. To add a new server to your cluster, you must perform the steps that are listed as prerequisite, and enable the server in vCenter Server before you power on the VM. Additional vSphere Bitfusion servers must be part of the same vCenter Server instance as the first vSphere Bitfusion server.

Prerequisites

- Verify that all existing vSphere Bitfusion servers are in a healthy state.
- Verify that you deployed the vSphere Bitfusion appliance.
- Verify that you customized the vSphere Bitfusion OVF template.
- Verify that you enabled passthrough of the GPUs to the vSphere Bitfusion server VM.
- Power off the vSphere Bitfusion server VM.

Procedure

1. From the **Hosts and Clusters** view in vCenter Server, right-click the server VM.
2. Select **Bitfusion > Enable Bitfusion**.
3. In the **Bitfusion Enablement** dialog box, select the **For a server, this will allow it to be used as a GPU server** option, and click **Enable**.

4 Power on the server VM.

When you add multiple vSphere Bitfusion servers to a cluster, you must power on the VMs in a sequential order.

Note Do not power on the VM before enabling the VM in vCenter Server. Otherwise the new vSphere Bitfusion server replaces the vSphere Client Plug-in, which eliminates the cluster, and creates a new cluster.

Results

When a new vSphere Bitfusion server joins the cluster, vCenter Server supplies a token, a certificate, and a configuration to access the vSphere Bitfusion cluster.

Remove a vSphere Bitfusion Server

To perform troubleshooting or maintenance on a vSphere Bitfusion server, you must remove the server from the vSphere Bitfusion cluster.

When powering off a vSphere Bitfusion server for maintenance or to perform troubleshooting, the health status of the vSphere Bitfusion cluster changes. When the cluster is not in a healthy state, you cannot add vSphere Bitfusion servers or perform a cluster backup operation. If half of the servers are powered off, the cluster is inoperable. When powering off a server for a longer period of time, you can prevent any potential risk by removing the server from the cluster.

Performing the following procedure immediately removes the server from the vSphere Bitfusion cluster. Any running applications that are using the GPUs receive an immediate GPU failure and usually return an error condition.

Prerequisites

- Prevent new client connections to the specific server in the server settings.
- Verify that there are no running applications on the server.

Procedure

1 In the vSphere Client, select **Menu > Bitfusion**.

2 On the **Servers** tab, select a server from the list.

3 From the **Actions** drop-down menu, select **Delete**.

4 In the confirmation dialog box, click **Delete**.

5 Wait until the server is no longer listed on the **Servers** tab.

The delete operation can take up to 10 minutes and longer. During this time, the backing storage rebalances. Alternatively, you can verify that the delete operation is finished by running the `nodetool status` command in the terminal of a running server.

6 (Optional) Delete the server virtual machine (VM).

Accidentally powering on the removed VM may result in the vSphere Bitfusion plug-in and cluster information being overwritten.

Results

You have deleted the selected server from the vSphere Bitfusion cluster.

What to do next

To reuse the VM or the underlying hardware, you can perform one of the following tasks.

- If you deleted the server from the cluster without deleting the VM, you can reenable the VM as a vSphere Bitfusion server and power it back on to add it to the cluster. For more information, see *Add Additional Bitfusion Servers* in the *VMware vSphere Bitfusion Installation Guide*.
- If you deleted the server VM, you can reuse the underlying hardware as a vSphere Bitfusion server by creating a VM and deploying the vSphere Bitfusion server appliance. See *Deploying the vSphere Bitfusion Appliance* in the *VMware vSphere Bitfusion Installation Guide*.

Disable or Delete a vSphere Bitfusion Client

You can stop a client from starting new application jobs or immediately prevent the client from accessing all vSphere Bitfusion servers.

Procedure

1 In the vSphere Client, select **Menu > Bitfusion**.

2 On the **Clients** tab, select a client from the list.

3 Disable or delete a vSphere Bitfusion client.

- a From the **Actions** drop-down menu, select **Disable**

  b In the confirmation dialog box, click **Disable**.

  This option prevents the client from starting new applications and allows running applications to finish. After the client is disabled, you can still view the historical client data and re-enable the client later.

- a From the **Actions** drop-down menu, select **Delete**.

  b In the confirmation dialog box, click **Delete**.

  This option immediately stops the client from accessing all vSphere Bitfusion servers. After the client is deleted, you can view only the historical client data in the vSphere Bitfusion server's database.

View vSphere Bitfusion Server Logs

Server logs can provide useful insights when troubleshooting a vSphere Bitfusion server.

To investigate any possible issues with vSphere Bitfusion, you can view the activity log of a specific vSphere Bitfusion server. For example, you can check the logs for thumbprint problems or vCenter Server GUID problems, that have occurred during the vSphere Bitfusion Plug-in registration process.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Servers** tab, select a server from the list.
3. From the **Actions** drop-down menu, select **Logs**.

Perform a Health Check of a vSphere Bitfusion Server

You can check the performace, stability, system resources, and software versions of a vSphere Bitfusion server by performing a health check.

You can check the health status of a selected vSphere Bitfusion server and if needed, perform troubleshooting. The health check examines the performace, stability, system resources, and software versions of a selected vSphere Bitfusion server and the server's surrounding vCenter Server environment. Each health check can return a pass, marginal, or fatal status.

For example, the health check verifies that all nodes are running, that there is enough free space, and that the connection to vCenter Server is working. To view the list of all available health checks, see vSphere Bitfusion Health Checks List.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Servers** tab, select a server from the list.
3. From the **Actions** drop-down menu, select **Health**.

   The **Health logs** dialog box appears and the results of the health checks are displayed. You see the status, type, name, and details of the check.
4. (Optional) Disable a specific health check by clicking the toggle button.

   The disabled health check is still performed in the background, but the status of the check is not changing the overall health status of the server on the **Servers** tab.
5. Click **Save and Exit**.

What to do next

- View vSphere Bitfusion Server Logs
- Back Up a vSphere Bitfusion Cluster

vSphere Bitfusion Health Checks List

vSphere Bitfusion performs the following checks when a health check of a server is initiated from the vSphere Bitfusion Plug-in.

Health Checks List

Name	Type	Description
`cass_buckets`	Stability	Validates the bucketing used by Cassandra to store data for utilization and other items.
`cass_node_num`	Stability	Confirms that Cassandra and Bitfusion see the same number of servers in the cluster.
`cass_nodetool`	Stability	Confirms that Cassandra sees that the cluster is in a healthy state.
`cass_replication`	Stability	Confirms the replication factor.
`compute_mode`	Stability	Confirms that the GPUs have compute mode set appropriately.
`network`	Stability	Verifies if there are dropped packets on the network.
`ecc`	Stability	Verifies if there are any ECC errors on the GPUs.
`gpu_api`	Stability	Confirms that the GPU APIs are matching.
`pci_nvml`	Stability	Confirms that all GPUs can be enumerated.
`pci_p2p`	Stability	Verifies that PCIe P2P is supported.
`temperature`	Stability	Verifies that the GPUs temperature is below 100 degrees celsiuses.
`vcenter_check`	Stability	Validates that the server can connect to vCenter Server.
`xid`	Stability	Verifies if there are any GPU Xid failures.
`bogomips`	Performance	Validates performance. The metric is used by the Linux kernel.
`hostmem`	Performance	Validates that there is enough host memory on the system.
`iface_compat`	Performance	Validates that the network configuration is valid.
`memops`	Performance	Verifies that `memops` is enabled for the GPUs.
`mtu`	Performance	Verifies that jumbo frames are enabled for the network.
`nvidia_stats`	Performance	Validates the statistics for the GPUs.
`nvidia_topo`	Performance	Validates the host topology.
`pci_width`	Performance	Validates that the GPUs are using the maximum PCIe lane capacity.
`ulimit_n`	Performance	Verifies that the maximum file descriptors limit is appropriate.
`diskspace`	System Resource	Confirms the free space on the server.
`install`	System Resource	Validates the Bitfusion installation.

Name	Type	Description
`pciinfo`	System Resource	Validate the PCI configuration.
`shadow_mem`	System Resource	Verifies that there is at least the same amount of system memory as there is frame buffer memory on the GPUs.
`cuda_version`	Software Version	Verifies the CUDA version.
`libdep`	Software Version	Verifies that the software dependencies for Bitfusion are installed.
`driver_version`	Software Version	Verifies the NVIDIA driver version.

View GPU Information for a vSphere Bitfusion Client

You can view the number of GPUs, that are allocated fully and partially for a specific vSphere Bitfusion client. Also, the GPU model and allocated memory are displayed.

To view GPU information related to a specific server, see View GPU Information for a vSphere Bitfusion Server.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Clients** tab, select a client from the list.
3. From the **Actions** drop-down menu, select **GPUs**.

View GPU Information for a vSphere Bitfusion Server

You can view GPU-related information, such as driver version, partition size, and available resources for your vSphere Bitfusion servers.

The information displayed is similar to the output of the `nvidia-smi` application. For example, you can view the GPU temperature, fan speed, the currently running processes, and the resources available on a vSphere Bitfusion server.

If you want to view the allocated and partial GPUs for a specific vSphere Bitfusion client, see View GPU Information for a vSphere Bitfusion Client.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Servers** tab, select a server from the list.
3. From the **Actions** drop-down menu, select **GPUs**.

Set a Global Display Refresh Interval

You can configure the vSphere Bitfusion Plug-in to refresh the data that it displays for clusters, servers, and clients regularly.

The refresh interval controls how often the vSphere Bitfusion Plug-in refreshes the displayed information. Alternatively, you can disable the automatic refresh in the GUI and manually press the **Refresh** button or navigate to a new tab.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Settings** tab, click **Application settings**.
3. Set a global refresh interval.
   a. Select the **Enable Refresh** check box.
   b. Enter a **Refresh Interval**.

      The value is in seconds.
4. Click **Save**.

Change the Settings of a vSphere Bitfusion Client

You can change client-specific settings from the vSphere Bitfusion Plug-in, such as current GPU quota, auto disconnect, and auto-shutdown idle interval.

The following procedure changes the settings for a specific vSphere Bitfusion client only. You can change the global settings for all vSphere Bitfusion clients in the **Settings > Global Client Defaults** tab.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Clients** tab, select a client from the list.
3. From the **Actions** drop-down menu, select **Settings**.
4. Change one or more client settings as required.
   - Enter a **Current GPU Quota**.

     The quota is the maximum number of GPUs that a vSphere Bitfusion client can allocate for all client applications. You can use non-integer values. For example, a quota of 3.5 allows a client to run simultaneously one application on two GPUs and a second application on 3 half-sized GPUs. The default value of 64 is enough to approximate an unlimited quota.
   - Select the **Auto Disconnect Idle GPUs** check box and enter an idle interval in minutes.

     This option allows vSphere Bitfusion to deallocate client GPUs and return the GPUs to the pool if the auto-shutdown idle interval is reached.

- To use the global client settings for this vSphere Bitfusion client, click **Match Defaults**.

5 Click **Save**.

Change the Settings of a vSphere Bitfusion Server

You can change server-specific settings from the vSphere Bitfusion Plug-in, such as allowing new client connections and entering a metrics interval.

The following procedure changes the settings for a specific vSphere Bitfusion server only. To change the global settings for all vSphere Bitfusion servers in the **Settings > Global Server Defaults** tab.

Procedure

1 In the vSphere Client, select **Menu > Bitfusion**.

2 On the **Servers** tab, select a server from the list.

3 From the **Actions** drop-down menu, select **Settings**.

4 Change one or more server settings as required.

- Select or deselect the **Allow new client connections** check box.

  To shut down a vSphere Bitfusion server gracefully and perform maintenance or troubleshooting, you can deselect the **Allow new client connections** check box. This action prevents vSphere Bitfusion clients from running new applications on the GPUs of the vSphere Bitfusion servers while you wait for all running applications to finish.

- To set a frequency to collect server statistics, enter a **Metrics interval** value in seconds.
- To use the global server settings for this vSphere Bitfusion server, click **Match Defaults**.

5 Click **Save**.

Monitoring the vSphere Bitfusion Environment

5

You can monitor your vSphere Bitfusion environment from the vSphere Bitfusion Plug-in and the CLI. You can also download the monitoring data of your cluster, servers, and clients.

This chapter includes the following topics:

Monitoring vSphere Bitfusion in the vSphere Bitfusion Plug-In

You can view IP addresses, host names, GPU allocation, memory use, and other data of your vSphere Bitfusion cluster, servers, and clients in the vSphere Bitfusion Plug-in.

Monitoring vSphere Bitfusion Cluster

You can use the vSphere Bitfusion Plug-in to view the following data for your cluster.

- The IP address of the primary vSphere Bitfusion server. The vSphere Bitfusion Plug-in uses the IP for communication.
- The allocation history of GPUs, shown in the Cluster GPU Allocation chart. The chart covers a range from the last 5 minutes to the last 30 days, the number of GPUs populating the cluster, and the number of GPUs allocated from all vSphere Bitfusion servers.
- All vSphere Bitfusion servers in the vSphere Bitfusion cluster, including servers that have been disabled or powered off, shown in the Servers table. Each entry displays a host name, IP address, and the number of the allocated GPUs.
- All vSphere Bitfusion clients that have run applications on the vSphere Bitfusion servers, shown in the Clients table. Each entry lists a host name, ID, and the number of GPUs currently allocated to the client.

Monitoring vSphere Bitfusion Servers

You can use the vSphere Bitfusion Plug-in to view the following data for your servers.

- All vSphere Bitfusion servers in the vSphere Bitfusion cluster, shown in the Servers table. You can select any server to display the server details. The table displays each server's host name, IP address, current GPU allocation, and the current health state.
- A heat map with an entry for each GPU on the server, shown in the Allocation chart. Each cell displays by intensity of color how engaged the GPU is during the selected time interval . The level of engagement is a weighted sum of memory allocation and CUDA cell use.
- Memory and core use charts, one pair for each GPU. The Memory charts also show the memory capacity.
- The outgoing and incoming traffic for each network interface.

Monitoring vSphere Bitfusion Clients

You can use the vSphere Bitfusion Plug-in to view the following data for your clients.

- All vSphere Bitfusion clients in the vSphere Bitfusion cluster, shown in the Clients table. A new entry appears on the list after a new client runs a vSphere Bitfusion command that requires a server connection for the first time. You can select a client to display the client details. The table displays each client's host name, ID, current GPU allocation, and version.
- The GPUs that are allocated to a client, shown in the GPU Assignment chart. A client can run multiple applications, each allocating separate GPUs, but they are displayed together. Allocations of partial GPUs add the fractional value to the sum.

Monitoring vSphere Bitfusion in the CLI

By using CLI commands, you can check the shadow memory of a vSphere Bitfusion client, the MTU size of your network, and the network interfaces for error statistics and dropped packet counts.

Shadow Memory Check

The vSphere Bitfusion client uses a part of its memory space as a shadow memory of the allocated remote GPU memory. The precise amount of memory required on the client host varies between applications. The shadow memory check determines if the host's memory is as large as the GPU memory. For more information about memory requirements, see the *System Requirements for vSphere Bitfusion* topic in the *VMware vSphere Bitfusion Installation Guide*.

You can see the amount of memory on your client from the `MemTotal` line of the `pseudo` file `/proc/meminfo`. To calculate the GPU memory, from a GPU server, you can run the `bitfusion smi` or `nvidia-smi` command, and add up the memory sizes of all GPUs.

You can add more memory to the vSphere Bitfusion client to meet the requirement. Alternatively, when you run applications, do not allocate more GPUs than you can shadow in the memory of the vSphere Bitfusion client.

MTU Size Check

The vSphere Bitfusion performance relies on a healthy, low-latency, and high-speed network. Applications perform better when they send a few large packets instead of many small packets. The maximum transfer unit (MTU) check determines whether you have a large (³4K) MTU setting for all high-speed (³10 Gbps) interfaces. Ignore this check for interfaces you do not use with vSphere Bitfusion.

Note For best performance of applications running under vSphere Bitfusion, set the MTU to 4096 or higher and set vSphere Bitfusion clients to match the MTU size of the deployed vSphere Bitfusion servers. If the MTU is above 1500, enable jumbo frames in the network switches.

To obtain and set the MTU size, see the following examples.

- To check the MTU size, you can run the `ifconfig` command.
- To change the MTU size on network interface enp175s to 4096 bytes, you can run `ifconfig enp175s mtu 4096`.

For more information on MTUs, see Determine maximum MTU.

Network Errors Check

You can check the network interfaces for error statistics and dropped packet counts. The files are in the following locations.

```
/sys/class/net/<interface>/statistics/*errors
```

```
/sys/class/net/<interface>/statistics/*dropped
```

If your network is healthy, the error count between the checks does not increase, new error messages do not occur, and no packets are dropped. The files are zeroed out only after a reboot.

Download vSphere Bitfusion Monitoring Data

You can download monitoring data of your vSphere Bitfusion cluster, servers, and clients in the vSphere Bitfusion Plug-in.

By exporting monitoring data, you can use external tools to review and troubleshoot your vSphere Bitfusion environment. The **Download CSV** button on each tab in the vSphere Bitfusion Plug-in, provides you with a different monitoring data set.

Procedure

1 In the vSphere Client, select **Menu > Bitfusion**.

2 Download the required monitoring data.

Option	Action
Cluster data	To save the cluster GPU allocation data, on the **Cluster** tab, click **Download CSV**.
Servers data	To save the data that is displayed for the selected server and pane, on the **Servers** tab, click **Download CSV**.
Clients data	To save the data that is displayed for the selected client and pane, on the **Clients** tab, click **Download CSV**.

3 (Optional) Select a location for the `.csv` file on your local machine.

Back Up and Restore a vSphere Bitfusion Cluster

6

You can back up and restore your vSphere vSphere Bitfusion database. The database includes the configuration, connectivity, health state, and history data of your vSphere Bitfusion cluster.

vSphere Bitfusion servers have a distributed database with the configuration, connectivity, state, and history data of the cluster. The backup operation saves a snapshot of the database information. If your cluster fails, the restore operation recovers the cluster to a previously healthy state.

This chapter includes the following topics:

Back Up a vSphere Bitfusion Cluster

You can back up your vSphere Bitfusion database and save a snapshot of the configuration, connectivity, health state, and history data of your vSphere Bitfusion cluster.

You can save a snapshot of your vSphere Bitfusion cluster database and download the backup copy to a local machine.

Prerequisites

Verify that your cluster is in a healthy state.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Settings** tab, click **Backup/Restore**.
3. Click **New Backup**, enter a backup name, and click **Take Backup**.

   The backup is listed in the table.
4. Select the backup and click **Download**.
5. (Optional) Select a location for the backup file on your local machine.

Restore a vSphere Bitfusion Cluster

You can restore your vSphere Bitfusion database and recover the cluster to a previously healthy state.

To recover your vSphere Bitfusion cluster from a failure or unhealthy state, you can use a backup file to restore the configuration, connectivity, health state, and history data of the cluster.

Prerequisites

- Verify that you have a backup of your vSphere Bitfusion environment.
- Verify that your vSphere Bitfusion cluster has one vSphere Bitfusion server in a healthy state.

Procedure

1. In the vSphere Client, select **Menu > Bitfusion**.
2. On the **Settings** tab, click **Backup/Restore**.
3. Restore the cluster.
   a. Click **Restore From Download**.
   b. Select a backup file.
   c. Click **Restore From Backup**.

Results

The restore operation might need several minutes to complete. During the process, a **Restore in progress** notification is displayed in the **Backup/Restore** pane.

What to do next

After the operation is completed and the vSphere Bitfusion plug-in is registered again with vCenter Server, click the refresh icon () to update all data in the current vSphere Client view.