NVIDIA Topology-Aware GPU Selection 0.1.0 (Early Access)

User Guide

DU-09998-001_v0.1.0 (Early Access) | April 2023

Table of Contents

List of Tables

Chapter 1. Introduction

Many NVIDIA® graphical processing units (GPU)-accelerated HPC applications that use Message Passing Interface (MPI) spend a substantial portion of their runtime in non-uniform GPU-to-GPU communications. These expensive communications prevent users from maximizing performance from their existing hardware.

To ensure that GPU-to-GPU communication in these applications is efficient, you need to make informed decisions when assigning GPUs to MPI processes. The assigning of GPUs to processes depends on the following factors:

- System GPU topology

  Shows how different GPUs are linked and the communication channel they use to connect. Different communication channels exist in multi-GPU servers, which results in some GPU pairs using faster communication links than other GPU pairs.
- Application GPU profiling

  Shows the total volume of communication between different GPUs in the system. This topology shows the application's communication pattern and also shows that some GPU pairs can have a higher communication volume than other pairs.

NVIDIA® Topology-Aware GPU Selection (NVTAGS) is a toolset for high performance computing (HPC) applications that uses MPI to enable faster solve times for applications with a high GPU communication time and a communication pattern that does not fit the underlying system GPU topology.

NVTAGS does the following:

- Profiles application GPU communication by using a PMPI-based profiler.
- Extracts system GPU communication topology that leverages NVIDIA's System Management Interface (nvidia-smi).
- Finds an efficient way of assigning GPUs to processes to minimize GPU communication congestion.
- Intelligently and automatically assigns GPUs to MPI processes.

  This reduces the overall GPU-to-GPU communication time of HPC applications that run on a multi-GPU system.

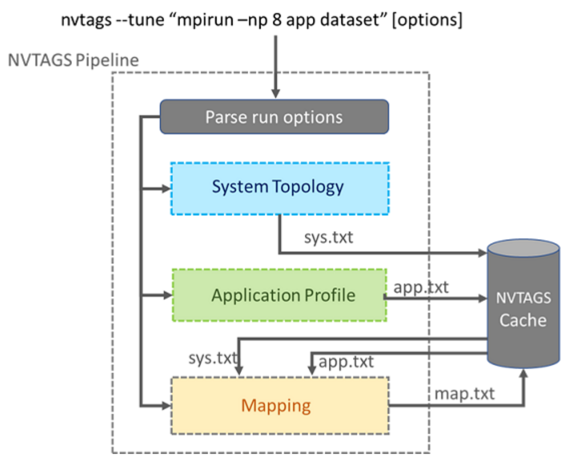
Here is the two-step process that NVTAGS follows to identify and apply efficient GPU assignments:

1. NVTAGS Tune

In this step, NVTAGS does the following:

- Gathers, or uses already available, application and system profiling data to understand how GPU-to-GPU communication is performed for a target application.
- Leverages this profiling information to identify and recommend a GPU assignment solution that better suits your application on the target system, so this assignment solution can be used in subsequent runs.

The following figure shows the NVTAGS tuning pipeline and the workflow to find an efficient GPU assignment. As you can see, the system GPU topology and application GPU profiling are extracted and cached in the `sys.txt` and the `app.txt` files. These files are fed to the NVTAGS mapping component and are used by a graph mapping algorithm to look for a better GPU assignment and store the result in the `map.txt` file in the NVTAGS cache.



2. NVTAGS Run

   In this step, NVTAGS applies the suggested GPU assignment from the tuning step and initiates your application run command. Optionally, based on how GPUs are selected in your application, NVTAGS also automatically sets the proper CPU and NIC affinity.

> Note: Both NVTAGS Tune and Run steps are light weight and impose negligible overhead for most of the MPI applications.

Systems that NVTAGS Benefits

NVTAGS leverages the GPU communication pattern of an application and the GPU topology of a system to generate efficient GPU assignments for an application that runs on the system.

NVTAGS benefits systems with asymmetric system topologies where some GPU pairs share stronger communication links than other pairs. Examples of these systems include NVIDIA DGX-1™ and PCIe servers that use different GPU communication channels to connect GPUs. Systems with symmetric system topologies, where all GPU pairs use the same communication links with equal capacity, will not benefit from custom GPU assignment because shuffling GPUs do not guide processes to use GPU pairs with stronger communication links. Examples of thse systems include NVIDIA DGX-2™ and NVIDIA DGX™ A100. Systems with symmetric topologies do not benefit from NVTAGS because all GPU assignments on such systems are equally optimal.

Chapter 2. Getting Started

This section provides information about the requirements to install NVTAGS, the installation instructions, and how to use NVTAGS.

2.1. Prerequisites

This section provides information about installing and using NVTAGS.

Ensure that you have completed the following prerequisites:

- Have a Linux operating system.
- Have an x86 system architecture.
- Installed a working NVIDIA Graphics Driver.

  To download the driver, go to Download Drivers.
- Have at least 3 GPUs installed on your machine.
- Verified that your application uses one GPU per MPI process and runs with at least 3 processes (3 GPUs).
- Have a CUDA-aware Open MPI to run your application.
- Verified that your Open MPI version matches the NVTAGS profiler library version.

  A different version might work but is not recommended. NVTAGS currently includes a profiler library that supports Open MPI 4.0.
- Optional: If you decide to use NVTAGS with CPU binding, `numactl` needs to be available on your machine.

  Depending on your OS, `numactl` can be installed by using `apt-get install numactl` , `yum install numactl` or other method.

2.2. Installing NVTAGS

Complete these steps to install NVTAGS.

About this task

Before you install NVTAGS, read Prerequisites.

Procedure

1. Download the latest NVTAGS release from the NVTAGS releases page.
2. To extract the NVTAGS archive, run:
```
tar -xzvf nvtags-ea-0.1.0.tar.gz
```
3. Copy the NVTAGS directory into the default NVTAGS path on your machine:
```
cp -r nvtags-ea-0.1.0 /opt/nvtags
```
4. Update `PATH` to make the NVTAGS binaries discoverable:
```
export PATH=/opt/nvtags/bin:${PATH}
```
5. (Optional) Although the NVTAGS binaries and scripts that are bundled in the NVTAGS release archive are executable, depending on your system, you might need to update your permissions.
```
chmod +x /opt/nvtags/bin/*
```
6. (Optional) If you do not have permission to copy the NVTAGS package into `/opt/nvtags/`, complete the following tasks:
   a). Adjust `PATH` to point to the NVTAGS binaries on the appropriate path.
   b). Set `NVTAGS_DEF_LIB_DIR` to a directory path where the NVTAGS library (for example, `libmpi_prof_x.y.so` or `libmpi_prof.so`) exists.
```
export PATH=/MY/PATH/TO/nvtags/bin:${PATH}
export NVTAGS_DEF_LIB_DIR=/MY/PATH/TO/nvtags/libs
```

Chapter 3. Using NVTAGS

This section provides additional information about the two NVTAG modes, NVTAGS Tune and NVTAGS Run.

3.1. NVTAGS Tune Mode

In the NVTAGS Tune mode, the application and system profiling data is used to recommend an efficient GPU assignment.

The Tune mode requires application profiling data to evaluate the efficiency of default GPU assignments and search for a better GPU assignment by using mapping algorithms. Depending on whether application profiling data exists, tuning can be completed with or without profiling.

After the tuning is complete, subsequent application runs can be used with NVTAGS in the Run mode.

3.1.1. Tune with profiling

To tune with profiling, application profiling data is used to extract the GPU communication pattern of the application.

If you do not know the GPU communication pattern, NVTAGS must be used in the Tune with profiling mode. You can also manually provide the pattern and use the Tune without profiling option. See Tune without Profiling for more information.

NVTAGs uses an MPI profiler library that dynamically links to your MPI application and intercepts MPI calls to build a GPU communication pattern. After the profiling is complete, NVTAGS looks for a better GPU assignment solution by using the application and system GPU topology information. The profiling results and recommended GPU assignments are cached in the local NVTAGS cache that defaults to `./.nvtags/.cache`.

Although NVTAGS can provide an efficient GPU assignment by using the default settings, NVTYAGS might provide a better GPU assignment by using non-default settings. This process can be achieved by changing the default profiling and mapping settings with input arguments. The profiling information is cached after each tuning step, so when you tune the settings again, you do not need to profile your application again. See About the NVTAGS CLI for more information.

3.1.2. Run NVTAGS in Tune with Profiling Mode

You can run NVTAGS in Tune with Profiling mode.

Procedure

To run NVTAGS in the Tune with Profiling mode, prepend your application run command with `nvtags --tune`:

```
nvtags [options] --tune "application run cmd"
```

3.2. Tune NVTAGS without Profiling Mode

You can run NVTAGS in Tune without Profiling mode.

After the application profiling data is available once, if a better GPU assignment exists, you can search for this assignment by using the cached data and without profiling your application again. NVTAGS runs quickly in this mode because it already has access to the profiling data for your application.

NVTAGS supports different mapping and profiling options, and if an efficient mapping exists, the default options usually successfully finds it. However, this might not be the case for all applications. For complete list of mapping and profiling options check Mapping Options and Application Profiling Options.

When the tuning step is complete, and a better GPU assignment is found, a message similar to the following is printed. A list of GPU IDs is stored in the `.nvtags/.cache/map.txt` file.

```
Better mapping found!
Max Congestion_improvement: 10.00%
Avg Congestion_improvement: 17.27%
0,1,3,2,7,6,4,5
```

> Note: GPU IDs are only stored when the congestion improvement is greater than the NVTAGS threshold value. The default value is 5%.

If no better GPU assignment is found, nothing is stored in the `.nvtags/.cache/map.txt` file, and NVTAGS outputs the following message:

```
No Better mapping found!
```

3.2.1. Run NVTAGS in Tune without Profiling Mode

You can tune NVTAGS in Tune without Profiling Mode.

Procedure

To tune NVTAGS without profiling, use the `--rebuild-prof` option:

```
nvtags [options] --rebuild-prof
```

3.3. NVTAGS Run Mode

Here is some information about the NVTAGS Run mode.

In the Run mode, NVTAGS applies the recommended efficient GPU assignment from the tuning process by setting `CUDA_VISIBLE_DEVICES` and executing your application run command. NVTAGS can also pin the CPU and the NIC based on their affinity information and the GPU assignment.

3.3.1. Run Mode with Binding

You can configure automatic CPU and NIC binding by using the `nvtags_run.sh` script, which can be found in the `/opt/nvtags/bin/nvtags_run.sh` directory.

This script automatically detects the CPU and NIC affinity, and based on the GPU assignment, binds them to each process.

Here is an example of how to apply NVTAGS to the `mpirun -np 8 app args` run command:

```
mpirun -np 8 --bind-to none -x EXE=app -x ARGS=args nvtags_run.sh
```

> Note: To run NVTAGS in this mode:
>
> - Pass the `--bind-to none` flag to the `mpirun` command, so that MPI does not attempt to handle the setting affinity.
> - Ensure that `numactl` is available on your system.

3.3.2. Run NVTAGS in Run Mode without Binding

You can run NVTAGs in Run mode without binding.

Procedure

To run your application with NVTAGS, add `nvtags --run` before your application run command:

For example:

```
nvtags [options] --run "application run cmd"
```

In this mode, there is no CPU or NIC pinning.

3.4. About the NVTAGS CLI

This section provides additional information about the two NVTAGS modes, NVTAGS Tune and NVTAGS Run.

3.4.1. CLI Options for the Tune Mode

You can tune the CLI with or without profiling.

Procedure

Complete one of the following options:

- To tune with application profiling, use the `--tune` option and pass the application run command to it.

  ```
  nvtags [options] --tune application run cmd  # tune with profiling
  ```

- To tune without application profiling, and use the existing cached data, run the `--rebuild-prof` option.

  ```
  nvtags [options] --rebuild-prof    # tune without profiling
  ```

3.4.1.1. System profiling options

Here is some information about the options that are used to modify NVTAGS system profiling parameters.

The system profiling options are `-m, --manual` .

By default, NVTAGS assigns predefined values to system GPU communication channels, which are calculated by using the channels' bandwidth and latency. Table 1 shows the list of GPU links that are recognized by `nvidia-smi` and their corresponding default values.

To better represent the strength of the communication links on your system, you can modify these values by setting the environment variable that NVTAGS associates with the link. The environment variable name that is used by NVTAGS is constructed by adding `NVTAGS_PROF_` to the name of the link. For example, `NVTAGS_PROF_SYS` is used to change the SYS default link value, and `NVTAGS_PROF_NV1` is used to change the NV1 default link value.

3.4.1.2. Supported Link Names

This table provides a list of the supported link names and their default values.

Table 1. List of Supported Link Names in NVTAGS and their Default Value and Description

Link Name	Link Description	Link Value
SYS	Connection traversing the PCIe and SMP interconnect	10

Link Name	Link Description	Link Value
	between NUMA nodes (Inter-socket)	
NODE	Connection traversing the PCIe and interconnect between Host Bridges in a NUMA node	19
PHB	Connection traversing PCIe and a PCIe Host Bridge	18
PXB	Connection traversing multiple PCIe bridges without traversing the Host Bridge	20
PIX	Connection traversing a maximum of one PCIe bridge	20
NV1	Connection traversing a bonded set of 1 NVLinks	25
NV2	Connection traversing a bonded set of 2 NVLinks	25
NV3	Connection traversing a bonded set of 3 NVLinks	25
NV4	Connection traversing a bonded set of 4 NVLinks	25
NV5	Connection traversing a bonded set of 5 NVLinks	25
NV6	Connection traversing a bonded set of 6 NVLinks	25
NV12	Connection traversing a bonded set of 6 NVLinks	25

After you set your new values to the link names by using their environment variables, use the `--manual` argument so their values can be applied by NVTAGS.

Although NV1 to NV12 have different bandwidth capacities, experiments on various systems and MPI applications shows that using the same value for all NVLinks leads to better mapping results. The mapping algorithm uses this value to select NVLinks, over non-NVLinks instead of selecting NVLinks with different bonded sets over each other. If it does not apply to your application and/or system, you can use a manual assignment to change the default link values.

3.4.1.3. Application Profiling Options

This section provides information about the options that are used to modify the application profiling parameters.

`-d, --disable-normalized:`

By default, NVTAGS normalizes raw application GPU communication pattern values, represented in bytes, because some mapping algorithms work better when normalized values are used. To disable this feature, and use raw communication pattern values, pass `--disable-normalize` (or `-d`) to the NVTAGS Tune command.

`-e, --enable-symmetric:`

This option allows you to make application profiling values symmetric. By default, application communication patterns are not symmetric, but sometimes mapping algorithms can find a better solution if a symmetric profiling value is used.

`-f, --prof-lib-path <path to dir>`

By default, NVTAGS uses a default porfiler that exists in the `/opt/nvtags/libs` directory or in the directory that is set by `NVTAGS_DEF_LIB_DIR`. However, you can provide the exact path to your custom profiler by using the `-prof-lib-path` argument with the profiler path.

`-v, ---normalized-value <value>`

The default normalization value is 100, which results in scaling raw GPU communication data that ranges between 0 and 100. You can change this default normalization value by using the `--normalized-value (or -v)` argument with the new value.

3.4.1.4. Mapping Options

These mapping group options can be used to modify mapping parameters.

`-i, --improvement-threshold`

NVTAGS uses a congestion metric to compare new GPU assignment candidates against your application's default assignment. Only GPU assignments that can improve the default assignment congestion by more than the threshold value are stored. By default, this threshold value is set to 5%, but it can be changed by using the `--improvement` (or `-i`) argument with the new threshold value. The new value must be between 0 and 100.

`-m, --map-alg map alg name`

Here are the options for the *map alg name* variable:

- **`greedy`**
- **`rb`**
- **`all`**

Currently, NVTAGS supports the Greedy (**`greedy`**), Recursive-bipartitioning (**`rb`**), and All (**`all`**) mapping algorithms. The `All` mapping algorithm is the default mapping, which is a combination of the `Greedy` and `RB` algorithms. You can change the `All` mapping algorithm to the `RB` or the `Greedy` algorithm by using `--map-alg` (or `-m`) and the mapping name.

`-o, --opt-time time in ms`

By default, NVTAGS spends 1000 ms (1 second) to evaluate and optimize different mapping solutions. If an efficient GPU assignment solution exists, the solution is found during this period. To change this value, use the `--opt-time` (or `-o`) argument with the new optimization period.

3.4.2. CLI options for the Run Mode

This section provides information about the CLI options that you can use to run NVTAGS in Run mode with or without binding.

- To run NVTAGS with the binding, use the `nvtags_run.sh` script.
- To run NVTAGS without binding, use the NVTAGS binary.

3.4.2.1. Run Mode with the Binding CLI

To use the NVTAGS Run mode with binding, pass the application run command to the `nvtags_run.sh` script.

To run the script, you must set the `EXE` and `ARGS` values to the application executable and other application arguments. For example, to run the `mpirun -np 8 app all other args` command with `nvtags_run.sh` script, run the following command:

```
mpirun -np 8 --bind-to none -x EXE="app" -x ARGS="all other args" nvtags_run.sh
```

This script reads the new potential GPU assignment from the `./.nvtags/.cache/map.txt` file and, before starting the application run command, sets `CUDA_VISIBLE_DEVICES`. It also extracts the system affinity information and the CPU and NIC affinity setting. By default, this script uses 1 thread per process and binds the process to core. When you run the script by using `N` processes, the script assumes that you are using `GPU 0` to `GPU N-1` on your system. You can change these default values by setting the associated environment variables before running your application.

- To change the number of threads, before running the NVTAGS run commend, set `OMP_NUM_THREADS`.
- To change the bind target, set `NVTAGS_BIND_TARGET` to `socket` (for socket binding) or `core` (for core binding).
- To change the GPU list that your application uses, set `NVTAGS_GPU_LIST` to comma-separated list of GPUs.

Note: By default, applications use `GPU0` to `GPU N-1` when running with the `N` process. You do not need to change this environment variable frequently.

3.4.2.2. Examples

Here are some examples of running NVTAGS in Run mode with binding.

Example: Use NVTAGS Run with Binding to the Socket

Here is an example where the Run mode is used with binding to `socket`:

```
export NVTAGS_BIND_TARGET=socket
```

```
mpirun -np 8 --bind-to none -x EXE="app" -x ARGS="all other args" nvtags_run.sh
```

Example: Use NVTAGS Run with Binding to Core using Four Threads per Process

Here is an example where the Run mode is used with binding to `core`:

```
export NVTAGS_BIND_TARGET=core
export OMP_NUM_THREADS=4
mpirun -np 8 --bind-to none -x EXE="app" -x ARGS="all other args" nvtags_run.sh
```

If NVTAGS cannot find a better mapping in the tuning step, running the `nvtags_run.sh` script exits the process. Since no mapping file exists, by default, the application will not run.

To change this behavior and allow NVTAGS to run your application with the default GPU assignment and apply CPU and NIC bindings, before you run the `nvtags_run.sh` script, set `NVTAGS_ALLOW_DEFAULT_RUN` to **1**.

3.4.2.3. Run Mode without the Binding CLI

You can use the Run mode with the binding CLI.

To use NVTAGS without binding, pass the application run command to the `--run` argument. If no mapping file is found from the tuning step, and there is no binding CLI, NVTAGS will not run the application.

To change this behavior and allow the application to run NVTAGS with the default GPU assignment, set `NVTAGS_ALLOW_DEFAULT_RUN` to **1**.

3.5. Generic CLI Options

Here is a list of generic options that can be used with the NVTAGS binary in the Tune and Run modes.

`-h, --help`

Prints a help message that includes a description of how to use NVTAGS and its options.

`-l, --log-level`

Enables debug logs that are, by default, disabled. To enable the logs, use the `--log-level` DEBUG option.

`-p, --path`

The path to a directory where NVTAGS caches the profiling and mapping files. The default path is `./nvtags/.cache/`.

Chapter 4. NVTAGS Examples

This section contains information and sample code to help you understand NVTAGS.

4.1. Examples: NVTAGS Tune Mode

The following examples show a variety of tuning options.

Tune with Profiling

Example 1: Tune `app2` with `dataset2` by using the default tuning options with a normalization value of `50`:

```
nvtags --tune "mpirun -np 8 app dataset" --normalized-value 50
```

Tune Without Profiling

Example 2: Using the cached profiling data in Example 2, complete retuning for `app2` with a normalization value of `200`:

```
nvtags --rebuild-prof --normalized-value 200
```

Tune with Custom Profiling Options

Example 3: Tune `app3` with `args3` by using custom manual system profiling link values. In this example, a DGX-1 server is used with the `SYS`, `NV1`, `NV2` link names, and you want to manually assign `1`, `2`, and `3` to these names:

```
export NVTAGS_PROF_SYS=1
export NVTAGS_PROF_NV1=2
export NVTAGS_PROF_NV2=3
nvtags --tune "mpirun -np 8 app3 args3" --manual
```

Tune with Custom Mapping Options

Example 4: Tune `app4` with `dataset4` by using symmetric, not normalized, application profiling with an improvement threshold value of `2.5`%:

```
nvtags --tune "mpirun -np 8 app4 dataset4" --disable-normalized --enable-symmetric
 --improvement-threshold 2.5
```

Example 5: Retune `app4` from *Example 4* by changing the default mapping to `"greedy"` and the optimization time to `3000` milliseconds (3 seconds):

> Note: When retuning an app, navigate to the same folder where you previously tuned the app. This step ensures that the `./.nvtags/.cache` directory content from the previous tuning is accessible.

```
nvtags --rebuild-prof --map-alg "greedy" --opt-time 3000
```

Tune with the Custom Cache Path

Example 6: Tune `app5` with `dataset5` by using the custom NVTAGS cache path:

```
nvtags "mpirun -np 8 app5 dataset5" --path /home/nvtags/mycache
```

When you use custom cache for tuning, if you do not provide the custom cache path, the default cache path is selected during an NVTAGS run:

```
nvtags "mpirun -np 8 app5 dataset5" --path /home/nvtags/mycache
```

4.2. Examples: NVTAGS Run Mode with Binding

Here are some examples that show the Run mode with binding.

In this mode, by using the `nvtags_run.sh` script, `CUDA_VISIBLE_DEVICES` is set by using the `./nvtags/.cache/map.txt` content, and `CUDA_VISIBLE_DEVICES` is based on new GPU assignment. This assignment is also used to perform CPU and NIC pinning based on their affinity information.

Example 7: Run `app4` with `dataset4` (tuned in *Example 6* in Examples: NVTAGS Tune Mode) by using the default setting. This setting binds the CPUs to core and uses 1 thread per CPU core:

```
mpirun -np 8 --bind-to none -x EXE="app4" -x ARGS="dataset4" nvtags_run.sh
```

Example 8: Run `app4` with `dataset4` (tuned in *Example 6* in Examples: NVTAGS Tune Mode) using socket for binding:

```
export NVTAGS_BIND_TARGET=socket
mpirun -np 8 --bind-to none -x EXE="app4" -x ARGS="dataset4" nvtags_run.sh
```

Example 9: Run `app4` with `dataset4` (tuned in *Example 6* in Examples: NVTAGS Tune Mode) using core binding and 4 threads per process:

```
export NVTAGS_BIND_TARGET=core
export OMP_NUM_THREADS=4
mpirun -np 8 --bind-to none -x EXE="app4" -x ARGS="dataset4" nvtags_run.sh
```

In this mode, when a better GPU assignment is found from previous step(s) in the `./nvtags/.cache/map.txt` file, `CUDA_VISIBLE_DEVICES` is set before starting the application command. Otherwise, NVTAGS skips running the application unless `NVTAGS_ALLOW_DEFAULT_RUN` is set to 1.

Example 10: Run `app4` with `dataset4` (tuned in *Example 6* in Examples: NVTAGS Tune Mode) with the log-level debug:

```
nvtags --run "mpirun -np 8 app dataset" --log-level debug
```

4.3. End-to-End Usage Example

This section includes an example to complete NVTAGS tuning and running with the Jacobi kernel.

Here is the standard Jacobi run command for this example:

```
mpirun -np 8 ./jacobi -t 4 2
```

NVTAGS Tune

1. Run the tuning step to profile your application and system topology:

   ```
   nvtags --tune "mpirun -np 8 ./jacobi -t 4 2"
   ```

2. Review the logs that indicate by how much communication congestion will improve with the NVTAGS-recommended GPU assignment:

   ```
   NVTAGS: 2020-06-16 08:36:07 info : Detected number of processes from profiling
    file is "8"!
   NVTAGS: 2020-06-16 08:36:08 info : Better mapping found!
   NVTAGS: 2020-06-16 08:36:08 info : Max Congestion improvement: 0.00%
   NVTAGS: 2020-06-16 08:36:08 info : Avg Congestion improvement: 11.54%
   NVTAGS: 2020-06-16 08:36:08 info : mapping result is stored in "./.nvtags/.cache/
   map.txt"!
   ```

NVTAGS Run

To launch your application with the improved GPU assignment that was recommended by NVTAGS:

Run Mode with CPU/NIC binding

```
mpirun -np 8 --bind-to none -x EXE="./jacobi" -x ARGS="-t 4 2" nvtags_run.sh
```

Run Mode without Binding

```
nvtags --run "mpirun -np 8 ./jacobi -t 4 2"
```

Chapter 5. Licensing

This section includes the license for NVTAGS and some third-party licenses.

5.1. NVTAGS License

Here is the license for NVTAGS.

```
Apache License
 Version 2.0, January 2004
 http://www.apache.org/licenses/

 TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

 1. Definitions.

 "License" shall mean the terms and conditions for use, reproduction,
 and distribution as defined by Sections 1 through 9 of this document.

 "Licensor" shall mean the copyright owner or entity authorized by
 the copyright owner that is granting the License.

 "Legal Entity" shall mean the union of the acting entity and all
 other entities that control, are controlled by, or are under common
 control with that entity. For the purposes of this definition,
 "control" means (i) the power, direct or indirect, to cause the
 direction or management of such entity, whether by contract or
 otherwise, or (ii) ownership of fifty percent (50%) or more of the
 outstanding shares, or (iii) beneficial ownership of such entity.

 "You" (or "Your") shall mean an individual or Legal Entity
 exercising permissions granted by this License.

 "Source" form shall mean the preferred form for making modifications,
 including but not limited to software source code, documentation
 source, and configuration files.

 "Object" form shall mean any form resulting from mechanical
 transformation or translation of a Source form, including but
 not limited to compiled object code, generated documentation,
 and conversions to other media types.

 "Work" shall mean the work of authorship, whether in Source or
 Object form, made available under the License, as indicated by a
 copyright notice that is included in or attached to the work
 (an example is provided in the Appendix below).

 "Derivative Works" shall mean any work, whether in Source or Object
 form, that is based on (or derived from) the Work and for which the
 editorial revisions, annotations, elaborations, or other modifications
 represent, as a whole, an original work of authorship. For the purposes
```

```
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,
publicly display, publicly perform, sublicense, and distribute the
Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
(except as stated in this section) patent license to make, have made,
use, offer to sell, sell, import, and otherwise transfer the Work,
where such license applies only to those patent claims licensable
by such Contributor that are necessarily infringed by their
Contribution(s) alone or by combination of their Contribution(s)
with the Work to which such Contribution(s) was submitted. If You
institute patent litigation against any entity (including a
cross-claim or counterclaim in a lawsuit) alleging that the Work
or a Contribution incorporated within the Work constitutes direct
or contributory patent infringement, then any patent licenses
granted to You under this License for that Work shall terminate
as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
Work or Derivative Works thereof in any medium, with or without
modifications, and in Source or Object form, provided that You
meet the following conditions:

(a) You must give any other recipients of the Work or
Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices
stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works
that You distribute, all copyright, patent, trademark, and
attribution notices from the Source form of the Work,
excluding those notices that do not pertain to any part of
the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its
distribution, then any Derivative Works that You distribute must
include a readable copy of the attribution notices contained
within such NOTICE file, excluding those notices that do not
pertain to any part of the Derivative Works, in at least one
```

```
of the following places: within a NOTICE text file distributed
as part of the Derivative Works; within the Source form or
documentation, if provided along with the Derivative Works; or,
within a display generated by the Derivative Works, if and
wherever such third-party notices normally appear. The contents
of the NOTICE file are for informational purposes only and
do not modify the License. You may add Your own attribution
notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS
```

```
APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following
boilerplate notice, with the fields enclosed by brackets "[]"
replaced with your own identifying information. (Don't include
the brackets!) The text should be enclosed in the appropriate
comment syntax for the file format. We also recommend that a
file or class name and description of purpose be included on the
same "printed page" as the copyright notice for easier
identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

5.2. FPMPI License

Here is the third-party license for FPMI.

```
COPYRIGHT NOTIFICATION
(C) COPYRIGHT 2000 UNIVERSITY OF CHICAGO
This program discloses material protectable under copyright laws of the United
 States. Permission to copy and modify this software and its documentation is
 hereby granted, provided that this notice is retained thereon and on all copies
 or modifications. The University of Chicago makes no representations as to the
 suitability and operability of this software for any purpose. It is provided "as
 is" without express or implied warranty. Permission is hereby granted to use,
 reproduce, prepare derivative works, and to redistribute to others, so long as this
 original copyright notice is retained.
This software was authored by:
William D. Gropp: (630) 252-4318
Mathematics and Computer Science Division
Argonne National Laboratory,
Argonne IL 60439 FAX: (630) 252-5986
Any questions or comments on the software may be directed to fpmpi@mcs.anl.gov.
Argonne National Laboratory with facilities in the states of Illinois and Idaho, is
 owned by The United States Government, and operated by the University of Chicago
 under provision of a contract with the Department of Energy.
DISCLAIMER
THIS PROGRAM WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY AN AGENCY OF THE UNITED
 STATES GOVERNMENT. NEITHER THE UNITED STATES GOVERNMENT NOR ANY AGENCY THEREOF,
 NOR THE UNIVERSITY OF CHICAGO, NOR ANY OF THEIR EMPLOYEES OR OFFICERS, MAKES ANY
 WARRANTY, EXPRESS OR IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR
 THE ACCURACY, COMPLETENESS, OR USEFULNESS OF ANY INFORMATION, APPARATUS, PRODUCT,
 OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS USE WOULD NOT INFRINGE PRIVATELY
 OWNED RIGHTS. REFERENCE HEREIN TO ANY SPECIFIC COMMERCIAL PRODUCT, PROCESS, OR
 SERVICE BY TRADE NAME, TRADEMARK, MANUFACTURER, OR OTHERWISE, DOES NOT NECESSARILY
 CONSTITUTE OR IMPLY ITS ENDORSEMENT, RECOMMENDATION, OR FAVORING BY THE UNITED
 STATES GOVERNMENT OR ANY AGENCY THEREOF. THE VIEW AND OPINIONS OF AUTHORS EXPRESSED
 HEREIN DO NOT NECESSARILY STATE OR REFLECT THOSE OF THE UNITED STATES GOVERNMENT OR
 ANY AGENCY THEREOF.
```

5.3. LipTopoMap License

Here is the license for LipTopoMap.

```
Copyright (c) 2010 The Trustees of the University of Illinois. All
 rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- Redistributions of source code must retain the above copyright
 notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
 notice, this list of conditions and the following disclaimer listed
 in this license in the documentation and/or other materials
 provided with the distribution.

- Neither the name of the copyright holders nor the names of its
 contributors may be used to endorse or promote products derived from
 this software without specific prior written permission.

The copyright holders provide no reassurances that the source code
provided does not infringe any patent, copyright, or any other
intellectual property rights of third parties. The copyright holders
disclaim any liability to any recipient for claims brought against
recipient by any third party for infringement of that parties
intellectual property rights.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

5.4. Metis License

Here is the license for Metis.

```
Copyright & License Notice
---------------------------

Copyright 1995-2013, Regents of the University of Minnesota

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied. See the License for the specific language governing
```

```
permissions and limitations under the License.

Apache 2.0 License Text from http://www.apache.org/licenses/LICENSE-2.0
Apache License

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and
 distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright
 owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that
 control, are controlled by, or are under common control with that entity. For the
 purposes of this definition, "control" means (i) the power, direct or indirect, to
 cause the direction or management of such entity, whether by contract or otherwise,
 or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or
 (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions
 granted by this License.

"Source" form shall mean the preferred form for making modifications, including but
 not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or
 translation of a Source form, including but not limited to compiled object code,
 generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made
 available under the License, as indicated by a copyright notice that is included in
 or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that
 is based on (or derived from) the Work and for which the editorial revisions,
 annotations, elaborations, or other modifications represent, as a whole, an
 original work of authorship. For the purposes of this License, Derivative Works
 shall not include works that remain separable from, or merely link (or bind by
 name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version
 of the Work and any modifications or additions to that Work or Derivative Works
 thereof, that is intentionally submitted to Licensor for inclusion in the Work by
 the copyright owner or by an individual or Legal Entity authorized to submit on
 behalf of the copyright owner. For the purposes of this definition, "submitted"
 means any form of electronic, verbal, or written communication sent to the Licensor
 or its representatives, including but not limited to communication on electronic
 mailing lists, source code control systems, and issue tracking systems that
 are managed by, or on behalf of, the Licensor for the purpose of discussing and
 improving the Work, but excluding communication that is conspicuously marked or
 otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of
 whom a Contribution has been received by Licensor and subsequently incorporated
 within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License,
 each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-
charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative
```

```
 Works of, publicly display, publicly perform, sublicense, and distribute the Work
 and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License,
 each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-
charge, royalty-free, irrevocable (except as stated in this section) patent license
 to make, have made, use, offer to sell, sell, import, and otherwise transfer
 the Work, where such license applies only to those patent claims licensable
 by such Contributor that are necessarily infringed by their Contribution(s)
 alone or by combination of their Contribution(s) with the Work to which such
 Contribution(s) was submitted. If You institute patent litigation against any
 entity (including a cross-claim or counterclaim in a lawsuit) alleging that
 the Work or a Contribution incorporated within the Work constitutes direct or
 contributory patent infringement, then any patent licenses granted to You under
 this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative
 Works thereof in any medium, with or without modifications, and in Source or Object
 form, provided that You meet the following conditions:

You must give any other recipients of the Work or Derivative Works a copy of this
 License; and
You must cause any modified files to carry prominent notices stating that You
 changed the files; and
You must retain, in the Source form of any Derivative Works that You distribute, all
 copyright, patent, trademark, and attribution notices from the Source form of the
 Work, excluding those notices that do not pertain to any part of the Derivative
 Works; and
If the Work includes a "NOTICE" text file as part of its distribution, then
 any Derivative Works that You distribute must include a readable copy of the
 attribution notices contained within such NOTICE file, excluding those notices
 that do not pertain to any part of the Derivative Works, in at least one of the
 following places: within a NOTICE text file distributed as part of the Derivative
 Works; within the Source form or documentation, if provided along with the
 Derivative Works; or, within a display generated by the Derivative Works, if and
 wherever such third-party notices normally appear. The contents of the NOTICE file
 are for informational purposes only and do not modify the License. You may add Your
 own attribution notices within Derivative Works that You distribute, alongside
 or as an addendum to the NOTICE text from the Work, provided that such additional
 attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide
 additional or different license terms and conditions for use, reproduction, or
 distribution of Your modifications, or for any such Derivative Works as a whole,
 provided Your use, reproduction, and distribution of the Work otherwise complies
 with the conditions stated in this License.
5. Submission of Contributions. Unless You explicitly state otherwise, any
 Contribution intentionally submitted for inclusion in the Work by You to the
 Licensor shall be under the terms and conditions of this License, without any
 additional terms or conditions. Notwithstanding the above, nothing herein shall
 supersede or modify the terms of any separate license agreement you may have
 executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names,
 trademarks, service marks, or product names of the Licensor, except as required for
 reasonable and customary use in describing the origin of the Work and reproducing
 the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to
 in writing, Licensor provides the Work (and each Contributor provides its
 Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
 KIND, either express or implied, including, without limitation, any warranties or
 conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR
 PURPOSE. You are solely responsible for determining the appropriateness of using
 or redistributing the Work and assume any risks associated with Your exercise of
 permissions under this License.
```

```
8. Limitation of Liability. In no event and under no legal theory, whether in tort
 (including negligence), contract, or otherwise, unless required by applicable law
 (such as deliberate and grossly negligent acts) or agreed to in writing, shall any
 Contributor be liable to You for damages, including any direct, indirect, special,
 incidental, or consequential damages of any character arising as a result of this
 License or out of the use or inability to use the Work (including but not limited
 to damages for loss of goodwill, work stoppage, computer failure or malfunction, or
 any and all other commercial damages or losses), even if such Contributor has been
 advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or
 Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance
 of support, warranty, indemnity, or other liability obligations and/or rights
 consistent with this License. However, in accepting such obligations, You may
 act only on Your own behalf and on Your sole responsibility, not on behalf of
 any other Contributor, and only if You agree to indemnify, defend, and hold each
 Contributor harmless for any liability incurred by, or claims asserted against,
 such Contributor by reason of your accepting any such warranty or additional
 liability.

END OF TERMS AND CONDITIONS
```

5.5. SPDLOG License

Here is the license for SPDLOG license.

```
The MIT License (MIT)

Copyright (c) 2016 Gabi Melman.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.
```

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA products are not designed, authorized, or warranted to be suitable for use in medical, military, aircraft, space, or life support equipment, nor in applications where failure or malfunction of the NVIDIA product can reasonably be expected to result in personal injury, death, or property or environmental damage. NVIDIA accepts no liability for inclusion and/or use of NVIDIA products in such equipment or applications and therefore such inclusion and/or use is at customer's own risk.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com