Third Party Software: How can I use CUDA MPS with Bright?

Introduction

CUDA MPS is a feature that allows multiple CUDA processes to share a single GPU context. A CUDA program runs in MPS mode, if the MPS control daemon is running on the system. When a CUDA program starts, it tries to connect to the MPS control daemon, which will then create an MPS server for the connecting client if one does not exist for the user (UID) who launched the client. Each user (UID) has its own MPS server. The MPS server creates the sharedGPU context, manages its clients, and issues work to the GPU on behalf of its clients. CUDA MPS should be transparent to CUDA programs.

Currently, CUDA MPS is available on 64-bit Linux only, requires a device that supports Unified Virtual Address (UVA) and has compute capability SM 3.5 or higher. Applications requiring pre-CUDA 4.0 APIs are not supported under CUDA MPS.

NVIDIA officially supports configurations with a single GPU, but it is possible to run it on systems with multiple GPUs creating the multiple MPS servers. Bright provides an init script to start the MPS control daemon for each GPU, as well as a utility to detect the GPUs the support MPS. CMDaemon can be configured to start and monitor the MPS daemon on a cluster.

The cuda-mps int script and for systemd systemd the cuda-mps service file are included in the cuda-driver package. Please make sure that you're using a cuda-driver package version 396.44-393 or higher. Please keep in mind that the cuda-mps service is disabled by default.

To enable MPS on your cluster follow the steps below:

1) Reboot the nodes so that they will get provisioned from the updated software image.

2) Add the cuda-mps service to Bright so that it can be started stopped automatically:

cmsh

category use default

services

add cuda-mps

set autostart on

(c) 2019 Bright Computing <kb@brightcomputing.com> | 2019-11-14 20:40

URL: http://kb.brightcomputing.com/faq/index.php?action=artikel&cat=18&id=161&artlang=en

Third Party Software: How can I use CUDA MPS with Bright?

set monitored on

commit

Notes:

** The mps.conf file is used by the init/systemd script.

** Don't enable the cuda-mps service in the software images to avoid having it started on boot time and leave it to Bright to start the service so that you can use the image for other nodes on which you don't want the CUDA MPS server to be running.

** You can add the cuda-mps service per node and not per category so that you'll have more control on which nodes will be starting the CUDA MPS service.

Using MPS

The MPS pipe directories are located in /var/spool/cuda_mps.

/var/spool/cuda_mps/mps_i is the pipe directory and /var/spool/cuda_mps/mps_log_i is the log directory for GPU i e.g /var/spool/cuda_mps/mps_0 for GPU 0

The exact numbering of the MPS-compatible GPUs can be obtained by running mps_dev_Query

To run a program outside the workload management system:

```
export CUDA_VISIBLE_DEVICES=0
export CUDA_MPS_CLIENT=1
export CUDA_MPS_PIPE_DIRECTORY=/var/spool/cuda_mps/mps_0
./prog
```


(c) 2019 Bright Computing <kb@brightcomputing.com> | 2019-11-14 20:40
URL: http://kb.brightcomputing.com/faq/index.php?action=artikel&cat=18&id=161&artlang=en

Third Party Software: How can I use CUDA MPS with Bright?

./prog

The cluster's administrator can enable the above settings for all users e.g by creating a file /cm/images/<your software image>/etc/profile.d/cuda_mps.sh and exporting the environment variables listed above.

MPS can be used in conjunction with MPI with the following restrictions:

- Only one job per user

- There is no isolation between MPI ranks

- One GPU per rank. this the reason that the only CUDA device 0 is visible.

- The MPI implementation needs to support UVA: currently OpenMPI 1.7 and MVAPICH2 1.8 or later. This means that a cudaMemcpy() is not required for MPI_Send/MPI_Recv

- Can use up to 32 concurrent hardware working queues (CUDA streams)

Example using the SLURM workload manager:

```
#!/bin/bash
#SBATCH -A <account>
#SBATCH -n 1
# Spread the tasks evenly among the nodes
#SBATCH --ntasks-per-node=4
#SBATCH --time=35:15:00
# Want the node exlusively
#SBATCH --exclusive

echo "Starting at `date`"
echo "Running on hosts: $SLURM_NODELIST"
echo "Running on $SLURM_NNODES nodes."
echo "Running on $SLURM_NPROCS processors."
echo "Current working directory is `pwd`"
```

```
export CUDA_VISIBLE_DEVICES=0
```

```
export CUDA_MPS_CLIENT=1
```

```
module load openmpi/gcc
```

```
srun ./myscript.sh
```

The myscript.sh file contains:

(c) 2019 Bright Computing <kb@brightcomputing.com> | 2019-11-14 20:40
URL: http://kb.brightcomputing.com/faq/index.php?action=artikel&cat=18&id=161&artlang=en

Third Party Software: How can I use CUDA MPS with Bright?

```
#!/bin/bash

lrank=$OMPI_COMM_WORLD_LOCAL_RANK

case ${lrank} in

[0])

  export CUDA_MPS_PIPE_DIRECTORY=/var/spool/cuda_mps/mps_0

  mpirun ./prog

  ;;

[1])

  export CUDA_MPS_PIPE_DIRECTORY=/var/spool/cuda_mps/mps_1

  mpirun ./prog

  ;;



esac
```

Unique solution ID: #1160
Author: Panos Labropoulos
Last update: 2018-09-04 12:50

(c) 2019 Bright Computing <kb@brightcomputing.com> | 2019-11-14 20:40
URL: http://kb.brightcomputing.com/faq/index.php?action=artikel&cat=18&id=161&artlang=en