Caffe-MPI Installation Best Practices

1. Introduction:

The following best practices document is provided as courtesy of the HPC Advisory Council.

2. Application Description

Caffe-MPI is a deep learning framework designed for both efficiency and flexibility, developed by HPC development team of Inspur. This version of Caffe supports running Caffe on a distributed cluster of GPU compute nodes. It is designed and developed based on the BVLC single GPU version (https://github.com/BVLC/caffe). For more information about Caffe, please visit http://caffe.berkeleyvision.org).

3. Version Information:

Download Caffe-MPI at github:
https://github.com/Caffe-MPI/Caffe-MPI.github.io

4. Prerequisites:

The instructions from this best practice have been tested on the following configuration:
Hardware:

- Colfax CX2660s-X6 2U 4-node “Odin” cluster
- Dual-Socket 16-Core Intel E5-2697v4 @ 2.60 GHz CPUs
- Mellanox ConnectX-4® EDR InfiniBand and 100Gb/s Ethernet VPI adapters
- Mellanox Switch-IB SB7700 36-Port 100Gb/s EDR InfiniBand switches
- GPU: NVIDIA Kepler K80 GPUs
- Memory: 64GB DDR4 2133MHz RDIMMs per node
- NVIDIA Kepler K80 and Pascal P100 GPUs

OS and software:

- OS: Ubuntu 14.04
- InfiniBand driver: MLNX_OFED_LINUX-3.4-1.0.0.0 InfiniBand SW stack
- MPI: Mellanox HPC-X v1.7.0-406
- Compilers: GNU compilers 4.8.4
- CUDA Library: 8.0, CUDNN version 5.1.5
- Application:
  - Caffe-MPI master (6c2c347)
- Benchmarks:
  - CIFAR-10 Object Recognition in Images
  - ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)

5. Building HPC-X (Open MPI) with Thread Support

The following script is used for compiling and installing HPC-X with thread support for Caffe:

```
# cd /opt/hpcx-v1.7.406-gcc-MLNX_OFED_LINUX-3.4-1.0.0.0-ubuntu14.04-
x86_64/sources

# cat rebuild-ompi-threads-cuda.sh
#!/bin/bash

#module purge
#module load intel/compiler/2016.4.258

BASE=$PWD
#export CC=icc
#export CXX=icpc
#export FC=ifort
#export F77=ifort

rm -rf /dev/shm/openmpi-gitclone
tar xfp openmpi-gitclone.tar.gz -C /dev/shm
cd /dev/shm/openmpi-gitclone

export CUDA_INSTALL_PATH=/usr/local/cuda
export PATH=$PATH:$CUDA_INSTALL_PATH/bin
export LD_LIBRARY_PATH=$CUDA_INSTALL_PATH/lib64:$LD_LIBRARY_PATH

module use $BASE/../modulefiles
module load hpcx

./configure --prefix=${HPCX_HOME}/ompi-v1.10-threads-cuda \
--with-knem=${HPCX_HOME}/knem \
            --with-fca=${HPCX_HOME}/fca \
            --with-mxm=${HPCX_HOME}/mxm \
            --with-hcoll=${HPCX_HOME}/hcoll \
            --with-platform=contrib/platform/mellanox/optimized \
            --with-slurm \
            --enable-mpi-thread-multiple \
            --with-verbs --with-cuda 2>&1 | tee config-output.log

make -j16 all 2>&1 |tee build.log
make -j16 install 2>&1| tee install.log

# ./rebuild-ompi-threads-cuda.sh
```

6.0 Building Caffe-MPI

6.1 Changes to Caffe-MPI

First, load HPC-X, and CUDA driver:

```
module use /opt/hpcx-v1.7.406-gcc-MLNX_OFED_LINUX-3.4-1.0.0.0-
ubuntu14.04-x86_64/modulefiles
module load hpcx-ompi-v1.10-threads-cuda
module load cuda/8.0
```

6.2. Makefile.config

Modify the Makefile.config to enable CUDNN, and define generalization of GPU architecture and specify the name of the NVIDIA GPU which assemble and optimize the PTX for the GPU.

For Kepler K80, use:

```
-gencode arch=compute_37,code=sm_37 \
```

For Pascal P100, use:

```
-gencode arch=compute_60,code=sm_60 \
```

```
$ cp Makefile.config.example Makefile.config
$ vim Makefile.config
USE_CUDNN := 1
CUSTOM_CXX := mpic++
CUDA_ARCH := -gencode arch=compute_20,code=sm_20 \
                -gencode arch=compute_20,code=sm_21 \
                -gencode arch=compute_30,code=sm_30 \
                -gencode arch=compute_35,code=sm_35 \
                -gencode arch=compute_37,code=sm_37 \
                -gencode arch=compute_50,code=sm_50 \
                -gencode arch=compute_50,code=compute_50
```

6.3. Makefile

Modify Makefile to add include path for HPC-X or Open MPI:

```
# Debugging
ifeq ($(DEBUG), 1)
        COMMON_FLAGS += -DDEBUG -g -O0
        NVCCFLAGS += -G
else
        COMMON_FLAGS += -DNDEBUG -O2
        COMMON_FLAGS += -std=c++11
        COMMON_FLAGS += -I/opt/hpcx-v1.8.0-gcc-MLNX_OFED_LINUX-3.4-
1.0.0.0-ubuntu14.04-x86_64/ompi-v2.x/include
Endif
```

Otherwise it would run into problem that mpi.h not found during compile time, because Caffe uses nvcc to compile both CUDA and MPI code.

Also comment out the line that warns about the C++ compiler cannot create static link:

```
else ifneq (,$(findstring mpicxx,$(CXX)))
        STATIC_LINK_COMMAND := -Wl,--whole-archive $(STATIC_NAME) -Wl,-
-no-whole-archive
+#  # The following line must not be indented with a tab, since we are
not inside a target
+#  $(error Cannot static link with the $(CXX) compiler)
```

6.4. Compile

```
$ make
$ make test
$ make runtest
```

7.0 Preparing Input data for Caffe-MPI

Download the input data for ImageNet:

```
Chmod a+x data/ilsvrc12/get_ilsvrc_aux.sh
data/ilsvrc12/get_ilsvrc_aux.sh
```

Download the input data for CIFAR-10:

Uncomment the line in this file:

```
-#wget --no-check-certificate http://www.cs.toronto.edu/~kriz/cifar-10-
binary.tar.gz
+wget --no-check-certificate http://www.cs.toronto.edu/~kriz/cifar-10-
binary.tar.gz
```

Run script to download CIFAR-10:

```
./data/cifar10/get_cifar10.sh
```

8.0. Running Caffe-MPI

To run the CIFAR-10 training:

```
export CUDA_VISIBLE_DEVICES=0,1

mpirun \
-x LD_LIBRARY_PATH \
-x CUDA_VISIBLE_DEVICES=0,1 \
-hostfile ~/hostfile/hostfile.2.20 \
/home/caffe/Caffe-MPI.github.io/build/tools/caffe train \
--solver=examples/cifar10/cifar10_quick_solver.prototxt
```