Designing Shared Address Space MPI libraries in Many-core Era

Jahanzeb Maqbool Hashmi and Dhabaleswar K. Panda (Advisor)
Department of Computer Science and Engineering,
The Ohio State University
hashmi.29@osu.edu, panda@cse.ohio-state.edu

A ARTIFACT DESCRIPTION

A.1 Abstract

The artifact contains software package for the designs that we have developed for this work. The software implements the shared address-space based communication designs for MPI point-to-point and collective communication primitives. The proposed designs are implemented in open-source MPI implementation, specifically MVAPICH2-X and will be available for the public use with the upcoming release version 2.3-Xrc1. The software supports various CPU architectures including Intel SkyLake, IBM OpenPOWER and ARM CortexA series of processors. The multi-node support is currenlty limited to only Mellanox InfiniBand network adapters, however, support for OmniPath is underway and will be available soon. The proposed designs employ XPMEM kernel module to achieve address-space sharing and thus, XPMEM module is required to be installed and loaded on the systems.*

A.2 Description

A.2.1 Check-list (artifact meta information).

- **Algorithm: Zero-copy Reduce and Allreduce. XPMEM based MPI rendezvous communication**
- **Program: Shared Address Space based MPI library (based on MVAPICH2X-2.3rc1, available soon)**
- **Binary: C executables for miniAMR, and OSU Micro Benchmarks**
- **Run-time environment: RHEL7, CentOS7, kernel version 3.12 or higher (for XPMEM)**
- **Hardware: Intel Xeon/KNL/OpenPOWER CPU, InfiniBand Network**
- **Output: Text output of the different applications and benchmarks**
- **Experiment workflow: Download the tarball and install the optimized MPI library. Then run tests with different modes**
- **Experiment customization: Change the input file of the application under test and set the desired mode of optimization**
- **Publicly available?: Yes**

A.2.2 How software can be obtained (if available). The MPI library containing all the proposed designs e.g., MVAPICH2-X can be downloaded from: http://mvapich.cse.ohio-state.edu/downloads/. The release version containing the designs mentioned in this work is 2.3rc1.

A.2.3 Hardware dependencies. The proposed design is independent of any CPU but restricted to InfiniBand network adapters for a short while.

A.2.4 Software dependencies. The proposed designs depend on XPMEM software package which is available at: https://github.com/hjelmn/xpmem. The instructions for installing and loading the XPMEM module are explained below.

A.3 Installation

Download the appropriate tarball of MVAPICH2-X 2.3rc1 from the project website available at: http://mvapich.cse.ohio-state.edu/downloads/.
We do not include the Intel MPI library due to licensing reasons. It has to be installed by the administrator/user.
Install the MVAPICH2X RPM by using the script provided in the downloaded tarball.
Download, install, and load XPMEM kernel module available at: `$ git clone https://github.com/hjelmn/xpmem.git`.
Either run `$ ./install-xpmem.sh` script available in the MVAPICH2X tarball, or you can install XPMEM by using following steps:

```
1. $ ./configure --prefix=/opt/xpmem \
            --with-default-prefix=/opt/xpmem \
          --with-module=/opt/xpmem/share/modules/xpmem

2. $ make -j 8

3. $ sudo make install

4. $ sudo insmod /opt/xpmem/lib/module/xpmem.ko ; \
       sleep 1 ; \
       sudo chmod 666 /dev/xpmem

5. $ make check
```

Once XPMEM is installed and loaded, you can use MVAPICH2X library with all the XPMEM based designs.
After installing MVAPICH2X and XPMEM libraries, the applications (here OSU Micro Benchmarks, miniAMR, and CNTK) need to be installed against these libraries. To do so, mpicc and mpicxx flags in the Makefile of these applications should be pointed to the installation of the MPI libraries. After that, these applications need to be configured and compiled:

```
  $ ./configure --prefix=path/to/install
CC=mpicc CXX=mpicxx
  $ make clean; make; make install
```

A.4 Experiment workflow

For these experiments, an script will be available that sets the proper environment variables to enable the the optimization that one would like to use. For the time being, we are providing required parameters and commands to trigger the proposed designs for given applications.
To enable efficient XPMEM based point-to-point communication, following parameter needs to be used:

```
MV2_SMP_USE_XPMEM=1
```

Example:

```
$ mpirun_rsh -np 2 n0 n0 MV2_SMP_USE_XPMEM=1 ./a.out
```

To enable efficient collectives designs, following paramters can be used:

```
MV2_USE_XPMEM_COLL=1
```

Furthermore, the threshold after which XPMEM based collectives are to be used, can be tweaked via following paramter:

```
MV2_XPMEM_COLL_THRESOLD=<nbytes>
```

Example: `$ mpirun_rsh -np 28 -hostfile hosts MV2_USE_XPMEM_COLL=1 MV2_XPMEM_COLL_THRESOLD=4096 ./a.out`

A.5 Evaluation and expected result

OSU Micro Benchmarks/miniAMR/CNTK

As the results of the jobs are redirected to sdtout, one can find the results in the output file provided by the job scheduler. The results file contains the latency numbers for different set of the experiments.

OSU Micro Benchmarks have several inputs which give the user ability to change the number of the MPI_Allreduce iterations and change the output format of it.

To get more information please refer to: http://mvapich.cse.ohio-state.edu/benchmarks/

miniAMR has several input variables which are set as below for job with 32 processes per node with 1K as the number of mesh refinements:

```
  --num_refine 1024 --max_blocks 4000 --init_x 1 --init_y 1 --init_z 1 --npx $num_nodes --npy 4 --npz 8 --nx 8 --ny 8 --nz 8 --num_objects 2 --uniform_refine 0 --object 2 0 -1.10 -1.10 -1.10 0.030 0.030 0.030 1.5 1.5 1.5 0.0 0.0 0.0 --object 2 0 0.5 0.5 1.76 0.0 0.0 -0.025 0.75 0.75 0.75 0.0 0.0 0.0 --num_tsteps 100 --checksum_freq 4 --stages_per_ts 16
```

To get more information about miniAMR please refer to: https://mantevo.org/packages/

For CNTK, we used ImageNet dataset and AlexNet model. CNTK with AlexNet can be run as follows:

```
$ mpirun_rsh -np XX -hostfile hosts $XPMEM_ENV_VARS configFile=./cntk/Examples/Image/Classification/AlexNet/BrainScript/AlexNet_ImageNet.cntk
```

A.6 Notes

We will provide all the scripts as a single tarball and make it publicly available for the camera-ready draft. For the time being, we have provided sufficient details of compiling and running the necessary programs.