# Run MPI with containers

## Simple test

Simple container with ucx and openmpi.

```
Bootstrap: docker
From: debian:trixie

%environment
    export LD_LIBRARY_PATH=/usr/local/lib

%post
    apt-get update && apt-get install -y build-essential wget rdma-core libibverbs-dev
    wget https://github.com/openucx/ucx/releases/download/v1.18.1/ucx-1.18.1.tar.gz
    tar xzf ucx-1.18.1.tar.gz
    cd ucx-1.18.1
    mkdir build
    cd build
    ../configure --prefix=/opt/
    make -j4
    make install
    cd ..
    export OPENMPI_VERSION="4.1.6"
    export OPENMPI_MAJOR_VERSION="v4.1"
    export OPENMPI_MAKE_OPTIONS="-j4"
    mkdir -p /openmpi-src
    cd /openmpi-src
    wget https://download.open-mpi.org/release/open-mpi/${OPENMPI_MAJOR_VERSION}/openmpi-
${OPENMPI_VERSION}.tar.gz \
        && tar xfz openmpi-${OPENMPI_VERSION}.tar.gz
    cd openmpi-${OPENMPI_VERSION} && ./configure --with-ucx=/opt --without-verbs
    make all ${OPENMPI_MAKE_OPTIONS}
    make install
```

```
cd /
rm -rf /openmpi-src
```

To build it:

```
singularity build -f openmpitest.sif openmpi.def
```

Then we compile an MPI application inside the container. For example osu-benchmarks.

```
wget https://mvapich.cse.ohio-state.edu/download/mvapich/osu-micro-benchmarks-7.5-1.tar.gz
tar -xvf osu-micro-benchmarks-7.5-1.tar.gz
```

```
singularity shell openmpitest.sif
```

```
cd osu-micro-benchmarks-7.5-1
./configure CC=/usr/local/bin/mpicc CXX=/usr/local/bin/mpicxx --prefix=/scratch/$USER/osu_install
make install
```

Then you can use the following job:

```
#!/bin/bash

#SBATCH -N 2
#SBATCH -n 2
#SBATCH -o mpi-%j.out
#SBATCH -e mpi-%j.err

module purge
module load singularityce
module load openmpi
export PMIX_MCA_psec=native
export PMIX_MCA_gds=^ds12

export SINGULARITY_BINDPATH=/scratch

srun --mpi=pmix singularity run openmpitest.sif /scratch/$user/osu-install/libexec/osu-micro-benchmarks/mpi/collective/osu_alltoall
```

# Some possible errors

if the option `--mpi=mpix` is not used, you will have the following error:

```
[dna067:2560172] OPAL ERROR: Unreachable in file pmix3x_client.c at line 111
--------------------------------------------------------------------------
The application appears to have been direct launched using "srun",
but OMPI was not built with SLURM's PMI support and therefore cannot
execute. There are several options for building PMI support under
SLURM, depending upon the SLURM version you are using:

  version 16.05 or later: you can use SLURM's PMIx support. This
  requires that you configure and build SLURM --with-pmix.

  Versions earlier than 16.05: you must use either SLURM's PMI-1 or
  PMI-2 support. SLURM builds PMI-1 by default, or you can manually
  install PMI-2. You must then build Open MPI using --with-pmi pointing
  to the SLURM PMI library location.

Please configure as appropriate and try again.
```

By default it uses PMI2 to launch the process but the OpenMPI on the container does not have support for it. From OpenMPI 3.1.0 PMIx is included by default.

# Psec error

You can also have this error:

```
A requested component was not found, or was unable to be opened.  This
means that this component is either not installed or is unable to be
used on your system (e.g., sometimes this means that shared libraries
that the component requires are unable to be found/loaded).  Note that
PMIX stopped checking at the first component that it did not find.

Host:      dna075
Framework: psec
Component: munge
```

Here, the application will run. This is related to the PMIX_SECURITY_MODE. When `srun` is executed, it will setup previous variable to: `munge,native`. Which means that munge protocol will be used for authentication. As the PMIx library on the container (client side) does not have that component, it will failed but it will then use the `native` component. You can read here for more explanations. You have to use `export PMIX_MCA_psec=native` to avoid this message.

# gds error

You can also see this error:

```
[dna075:373342] PMIX ERROR: ERROR in file gds_ds12_lock_pthread.c at line 168
```

This is an OpenPMIx bug related to the 'Generalized DataStore for storing job-level and other data' component. You can blacklist it by setting: `export PMIX_MCA_gds=^ds12`.

> This is fixed in OpenMPI 5

# Running OpenMPI 5.0

This works, there is no any compatibilty problem with the host version. If you want to test, set the version of OpenMPI to `5.0.7`. Other versions have problems to compile.

---

Révision #10
Créé 19 mai 2025 11:09:14 par Cristian Ruiz
Mis à jour 21 mai 2025 12:37:23 par Cristian Ruiz