

## Open MPI 4.1.8

### ウェブページ

<https://www.open-mpi.org/>

### バージョン

- 4.1.8

### ビルド環境

- gcc 8.5.0, 9.2.1 (gcc-toolset-9), 10.3.1 (gcc-toolset-10), 11.2.1 (gcc-toolset-11), 12.2.1 (gcc-toolset-12), 13.1.1 (gcc-toolset-13)
- Intel oneAPI 2023.2.0
  - (oneAPI 2024.2.1, 2025.0.1 環境でも同様にビルド可能)
- AOCC 4.2, 5.0
- NVIDIA HPC SDK 23.9, 24.9
  
- OpenPBS 22.05.11

### 必要なファイル

- openmpi-4.1.8.tar.bz2

### ビルド手順

#### gcc

```
#!/bin/sh

VERSION=4.1.8
WORKDIR=/gwork/users/${USER}
TARBALL=/home/users/${USER}/Software/OpenMPI/${VERSION}/openmpi-${VERSION}.tar.bz2
PBSROOT=/apl/pbs/22.05.11

PARALLEL=12

#-----
umask 0022

cd ${WORKDIR}
if [ -d openmpi-${VERSION} ]; then
  mv openmpi-${VERSION} openmpi-erase
  rm -rf openmpi-erase &
fi

tar jxf ${TARBALL}
cd openmpi-${VERSION}

# GCC 8
module purge
INSTALLDIR=/apl/openmpi/${VERSION}/gcc8
mkdir rccs-gcc8 && cd rccs-gcc8
./configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../

# GCC 9
module purge
module load gcc-toolset/9
```

```
INSTALLDIR=/apl/openmpi/${VERSION}/gcc9
mkdir rccs-gcc9 && cd rccs-gcc9
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../
```

```
# GCC 10
module purge
module load gcc-toolset/10
INSTALLDIR=/apl/openmpi/${VERSION}/gcc10
mkdir rccs-gcc10 && cd rccs-gcc10
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../
```

```
# GCC 11
module purge
module load gcc-toolset/11
INSTALLDIR=/apl/openmpi/${VERSION}/gcc11
mkdir rccs-gcc11 && cd rccs-gcc11
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../
```

```
# GCC 12
module purge
module load gcc-toolset/12
INSTALLDIR=/apl/openmpi/${VERSION}/gcc12
mkdir rccs-gcc12 && cd rccs-gcc12
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../
```

```
# GCC 13
module purge
module load gcc-toolset/13
INSTALLDIR=/apl/openmpi/${VERSION}/gcc13
mkdir rccs-gcc13 && cd rccs-gcc13
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../
```

```

VERSION=4.1.8
WORKDIR=/gwork/users/${USER}
TARBALL=/home/users/${USER}/Software/OpenMPI/${VERSION}/openmpi-${VERSION}.tar.bz2
PBSROOT=/apl/pbs/22.05.11

PARALLEL=12

#-----
umask 0022

cd ${WORKDIR}
if [ -d openmpi-${VERSION} ]; then
  mv openmpi-${VERSION} openmpi-erase
  rm -rf openmpi-erase &
fi

tar jxf ${TARBALL}
cd openmpi-${VERSION}

# intel compiler classic 2023.2.0
module purge
. ~/intel/oneapi/compiler/2023.2.0/env/vars.sh
module -s load compiler-rt/2023.2.0 # this line can be omitted

export CC=icc
export CXX=icpc
export FC=ifort

INSTALLDIR=/apl/openmpi/${VERSION}/intelclassic2023
mkdir rccs-intel && cd rccs-intel
./configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../

```

## oneAPI Compiler 2023.2.0

```

#!/bin/sh

VERSION=4.1.8
WORKDIR=/gwork/users/${USER}
TARBALL=/home/users/${USER}/Software/OpenMPI/${VERSION}/openmpi-${VERSION}.tar.bz2
PBSROOT=/apl/pbs/22.05.11

PARALLEL=12

#-----
umask 0022

cd ${WORKDIR}
if [ -d openmpi-${VERSION} ]; then
  mv openmpi-${VERSION} openmpi-erase
  rm -rf openmpi-erase &
fi

tar jxf ${TARBALL}
cd openmpi-${VERSION}

# ad hoc patch to avoid error
sed -i -e "s/UINTPTR_MAX/(void *) (uintptr_t)UINTPTR_MAX/" \
  oshmem/mca/sshmem/base/sshmem_base_open.c

# intel compiler icx/icpx 2023.2.0
module purge

```

```

~/intel/oneapi/compiler/2023.2.0/env/vars.sh
module -s load compiler-rt/2023.2.0

export CC=icx
export CXX=icpx
export FC=ifx

INSTALLDIR=/apl/openmpi/${VERSION}/intel2023
mkdir rccs-intel && cd rccs-intel
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../

```

## AOCC

```

#!/bin/sh

VERSION=4.1.8
WORKDIR=/gwork/users/${USER}
TARBALL=/home/users/${USER}/Software/OpenMPI/${VERSION}/openmpi-${VERSION}.tar.bz2
PBSROOT=/apl/pbs/22.05.11

PARALLEL=12

#-----
umask 0022

cd ${WORKDIR}
if [ -d openmpi-${VERSION} ]; then
  mv openmpi-${VERSION} openmpi-erase
  rm -rf openmpi-erase &
fi

tar jxf ${TARBALL}
cd openmpi-${VERSION}

export CC=clang
export CXX=clang++
export FC=flang

# AOCC 4.2.0
module -s purge
module -s load aocc/4.2.0
INSTALLDIR=/apl/openmpi/${VERSION}/aocc4.2
mkdir rccs-aocc42 && cd rccs-aocc42
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \
  --with-ucx \
  --without-libfabric \
  --enable-mpi1-compatibility
make -j ${PARALLEL} && make install && make check
cd ../

# AOCC 5.0.0
module -s purge
module -s load aocc/5.0.0
INSTALLDIR=/apl/openmpi/${VERSION}/aocc5.0
mkdir rccs-aocc50 && cd rccs-aocc50
../configure --prefix=${INSTALLDIR} \
  --with-tm=${PBSROOT} \
  --enable-mpi-cxx \

```

```
--with-ucx \  
--without-libfabric \  
--enable-mpi1-compatibility  
make -j ${PARALLEL} && make install && make check  
cd ../
```

## NVIDIA HPC SDK

```
#!/bin/sh  
  
VERSION=4.1.8  
WORKDIR=/gwork/users/${USER}  
TARBALL=/home/users/${USER}/Software/OpenMPI/${VERSION}/openmpi-${VERSION}.tar.bz2  
PBSROOT=/apl/pbs/22.05.11  
  
PARALLEL=12  
  
#-----  
umask 0022  
  
cd ${WORKDIR}  
if [ -d openmpi-${VERSION} ]; then  
mv openmpi-${VERSION} openmpi-erase  
rm -rf openmpi-erase &  
fi  
  
tar jxf ${TARBALL}  
cd openmpi-${VERSION}  
  
export CC=nvc  
export CXX=nvc++  
export FC=nvfortran  
  
export CFLAGS="-fPIC"  
export CXXFLAGS="-fPIC"  
export FCFLAGS="-fPIC"  
export LDFLAGS="-fPIC"  
  
# nvidia hpc sdk 23.9  
module purge  
module load nvhpc/23.9-nompi  
INSTALLDIR=/apl/openmpi/${VERSION}/nv23  
export CUDA_HOME=${NVHPC_ROOT}/cuda  
mkdir rccs-nv23 && cd rccs-nv23  
../configure --prefix=${INSTALLDIR} \  
--with-tm=${PBSROOT} \  
--enable-mpi-cxx \  
--with-ucx \  
--with-cuda=${CUDA_HOME} \  
--enable-mpi1-compatibility  
make -j ${PARALLEL} && make install && make check  
cd ../  
  
# nvidia hpc sdk 24.9  
module purge  
module load nvhpc/24.9-nompi  
INSTALLDIR=/apl/openmpi/${VERSION}/nv24  
export CUDA_HOME=${NVHPC_ROOT}/cuda  
mkdir rccs-nv24 && cd rccs-nv24  
../configure --prefix=${INSTALLDIR} \  
--with-tm=${PBSROOT} \  
--enable-mpi-cxx \  
--with-ucx \  
--with-cuda=${CUDA_HOME} \  
--enable-mpi1-compatibility  
make -j ${PARALLEL} && make install && make check
```

```
cd ../
```

## メモ

- icx を使う場合、oshmem/mca/sshmem/base/sshmem\_base\_open.c でコンパイルエラーが発生する(oneAPI 2023, 2024, 2025 で確認)。コンパイルオプションを変えるか、上記のように cast すれば回避できると思われる。
  - 4.x 系では Open MPI 4.1.8 で入ったコードと思われる。5.x 系でも 5.0.6 以前には入っていない。
- oneAPI 2024, 2025 を使う場合は 2023 と同じ手順でビルド可能。
- nvhpc sdk は PBS-aware な OpenMPI を作るためのビルド。