VIC-BORG-MPI User Manual

1. Install MPICH on each node

The process of compiling and deploying MPICH typically includes the following steps:

1.1 Download Source Code

Visit the MPICH official website (mpich.org) and download the latest source code tarball. In the terminal, use the following command to download the source code.

```
v=3.2
wget http://www.mpich.org/static/downloads/${v}/mpich-${v}.tar.gz
```

1.2. Extract the Source Code

In the terminal, use the following command to extract the downloaded file:

```
tar –xzf mpich-${v}.tar.gz
cd mpich-${v}
export MPICH2_3_2_DIR="/Download Path/mpich-${v}"
./configure --prefix=$MPICH2_3_2_DIR
```

1.3. Compile and Install

```
make
sudo make install
```

1.4 Set Environment Variables

Once installed, you need to set environment variables to use MPICH. Add the following lines to your .bashrc or .bash_profile

```
export PATH=/path/to/install/bin:$PATH
export LD_LIBRARY_PATH=/path/to/install/lib:$LD_LIBRARY_PATH
```

2. Prepare a comprehensive case study on each node

2.1 Download case study data

```
# Enter the 'example' directory

#go to https://github.com/JinfengM/VIC-Borg/tree/main/example,
cd example directory and download example.tar.gzaa and example.tar.gzab

#To ensure that MPICH 3.2 has been installed

# Unzip the example file 'example.tar.gz*' to obtain the 'run_lh' directory.
cat example.tar.gz* | tar -xzv
tar -zxvf example_files.tar.gz

# Create a directory at '/home/VIC'
mkdir /home/VIC

#To ensure that the '/home/VIC' directory exists
ls /home/VIC

#To copy the run_lh directory to the /home/VIC/ directory
cp -r run_lh /home/VIC/

# To ensure the '/home/VIC/run_lh' exists
ls /home/VIC/run_lh

# Download VIC-Borg-MPI project to the local directory '/home/VIC'

# Move https://github.com/JinfengM/VIC-Borg-MPI/src/main directory to local /home/VIC/VIC-Borg-MPI directory

#Move https://github.com/JinfengM/VIC-Borg-MPI/src/routMPI directory to local /home/VIC/VIC-Borg-MPI/routMPI directory

cd local /home/VIC/VIC-Borg-MPI/routMPI

# compile streamflow routing module, copy routMPI.so to /home/VIC/VIC-Borg-MPI/
make
cp routMPI.so ../

# Add the current directory to the LD_LIBRARY_PATH environment variable
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:`pwd`
```

# compile BorgMS at /home/VIC/VIC-Borg-MPI/
cd ..
make

# Execute VIC_BORG_MPI.X on a single node
mpiexec -n 51 ./VIC_BORG_MPI.X -g /home/VIC/run_lh/chanliu_input.txt

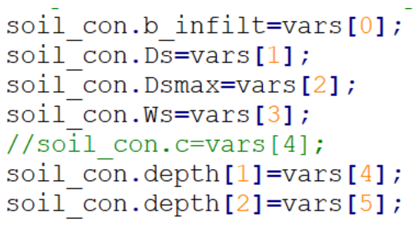
# Execute VIC_BORG_MPI.X on a cluster using machinefile
mpiexec -f machinefile -n 51 ./VIC_BORG_MPI.X -g /home/VIC/run_lh/chanliu_input.txt

# Notice
VIC model's source code is from https://github.com/UW-Hydro/VIC, users can access the source code here.

Borg algorithm's source code is from http://borgmoea.org/, users are required to complete the Google form to request access to the source code.

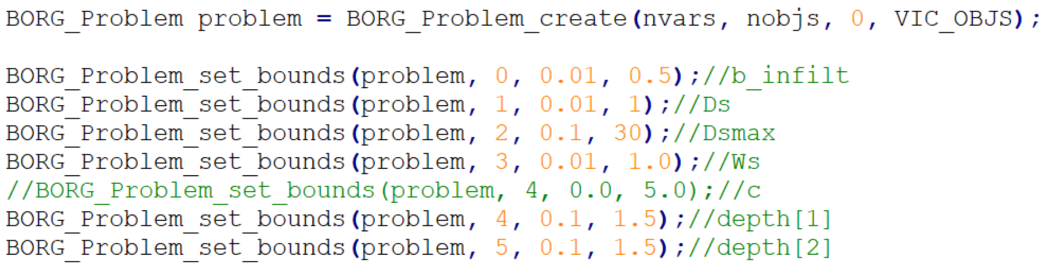
3. Select your target parameters and Customize your objective function

#Lines 155-161 of the vicNl.c code illustrate how to select the target parameters



```
soil_con.b_infilt=vars[0];
soil_con.Ds=vars[1];
soil_con.Dsmax=vars[2];
soil_con.Ws=vars[3];
//soil_con.c=vars[4];
soil_con.depth[1]=vars[4];
soil_con.depth[2]=vars[5];
```

#Lines 346-356 of the dtlz2_ms.c code illustrate how to customize the target parameters



```
BORG_Problem problem = BORG_Problem_create(nvars, nobjs, 0, VIC_OBJS);

BORG_Problem_set_bounds(problem, 0, 0.01, 0.5);//b_infilt
BORG_Problem_set_bounds(problem, 1, 0.01, 1);//Ds
BORG_Problem_set_bounds(problem, 2, 0.1, 30);//Dsmax
BORG_Problem_set_bounds(problem, 3, 0.01, 1.0);//Ws
//BORG_Problem_set_bounds(problem, 4, 0.0, 5.0);//c
BORG_Problem_set_bounds(problem, 4, 0.1, 1.5);//depth[1]
BORG_Problem_set_bounds(problem, 5, 0.1, 1.5);//depth[2]
```

#Lines 20 and 295 of the dtlz2_ms.c code illustrate how to customize the object functions

int nobjs = 3;

void VIC_OBJS(double* vars, double* objs, double* consts,int argc, char* argv[],int rank)