Multi-Processor Computing framework *(MPC version 2.5.1)*

General Installation guide

Please refer to the GettingStarted.pdf document for general installation.

Cross-compilation Installation guide

Prerequisites

- **General**:
  - You need to install MPC for the architecture you will target *AND* for the architecture from which you will launch your programs. For instance, **For ARM architecture**: a cross-compiled gcc which generate code for ARM architecture.
  - The prefix for both installations has to be on the same.
- **For ARM architecture**: a cross-compiled gcc which generate code for ARM architecture
- **For MIC architecture**: the Intel Compiler suite and the Intel libraries (Intel MPSS)

MIC Compilation

1. Load Intel modules (*icc*, *libraries*, etc) in your environment.
2. Launch the **installmpc** script for the MIC:

   ```
   $ ./installmpc --prefix=$HOME/install-mpc --target=mic --compiler=icc
     --arch-library-path=(Path to Intel lib for mic architecture) --disable-mpc-gcc
     --disable-mpc-gdb
   ```

   --target=mic: specify the target architecture. You can either use *--target=mic* or *--target=k1om*.
   --compiler=icc: specify the compiler to be used by MPC. Here *icc* has to be selected.
   --arch-library-path=path: specify the path of libraries used for the target architecture.

   Note: Do not forget to add *--disable-mpc-gcc* and *--disable-mpc-gdb* options to the installmpc script.
   You could have errors installing these two programs with icc.
3. Source the *mpcvars* script located in the root directory of your MPC installation prefix

   ```
   $ . $HOME/install-mpc/mpcvars.sh for sh shells
   ```

   $ source $HOME/install-mpc/mpcvars.sh for bash shells $ source $HOME/install-mpc/mpcvars.csh for csh or tcsh shells

   This will load the MPC environment for the current architecture. You can force to load the environment for the MIC architecture by typing:

   ```
   $ source $HOME/install-mpc/mpcvars.sh k1om
   ```
4. To compile your first MPC program for a MIC architecture, you may execute the mpc_cc compiler:

   If you loaded the MIC environment:

   ```
   $ mpc_cc main.c -o main.mic
   ```

   If you loaded the host environment:

   ```
   $ mpc_cc -target=k1om main.c -o main.mic
   ```
5. Execute your MPC program:

- **Homogeneous launch:**
  - Compile your code for the MIC architecture (main.mic)
  - Create config.cfg file:

    ```
    -host mic0 -p 1 ./main.mic
    ```
  - Launch the binary with mpcrun script:

    ```
    $ mpcrun -p=1 -n=4 -net=tcp -l=mic_hybrid --mic-config=config.cfg
    ```

    -l=mic_hybrid: load the mic launcher
    --mic-config=*: load the config file for launch
    Note that your process number have to be the same in *config.cfg* and launch command
- **Heterogeneous launch:**

- Compile your code for the MIC architecture (*main.mic*) as well as for the host architecture (*main.host*)
- Create config.cfg file:

```
-host knc02 -p 4 ./main.host
-host mic0  -p 3 ./main.mic
-host mic1  -p 2 ./main.mic
```

- Launch the binaries with mpcrun script:

```
$ mpcrun -p=9 -n=16 -net=tcp -l=mic_hybrid
  --mic-config=config.cfg --mic-nb-task=5 --nb-mic=2 --nb-host=1 --host-nb-task=6
```

--nb-mic=*: number of MIC devices for the launch (optional)
--nb-host=*: number of host devices for the launch (optional)
--mic-nb-task=*: number of tasks per MIC device (optional)
--host-nb-task=*: number of tasks per host device (optional)
If these options are not specified, the repartition of the tasks on the processes is homogeneous.

Cross-Compilation, ARM example

1. Cross-compile gcc for ARM architecture
2. Launch the *installmpc* script for the ARM architecture:

```
$ ./installmpc --prefix=$HOME/install-mpc --with-mpc-gcc=prefix --target=arm
```

   prefix is the path of your cross-compiled gcc for ARM architecture

3. Source the *mpcvars* script at the root of your MPC installation prefix

```
$ . $HOME/install-mpc/mpcvars.sh for sh shells
$ source $HOME/install-mpc/mpcvars.csh for csh, tcsh or bash shells
```

   This will load the MPC environment for the current architecture you are using.
   You can force to load the environment for the ARM architecture:

```
$ source $HOME/install-mpc/mpcvars.sh arm
```

4. To compile your first MPC program for ARM architecture, you may execute the *mpc_cc* compiler on the host:

   If you loaded the ARM environment:

```
$ mpc_cc main.c -o main
```

   If you loaded the host environment:

```
$ mpc_cc -target=arm main.c -o main.mic
```

5. Execute your ARM binary with mpcrun command:

```
$ mpcrun -n=4 ./main
```

Arguments of the installmpc installation script

```
 ___  _________ _____    __                                                _
|  \/  || ___ |  __ \  / _|                                              | |
| .  . || |_/ / /  \/ | |_ _ __ ___ _ _ ___ ___   _____   _____ _ __| | __
| |\/| ||  __/| |     |  _| '__/ _ \ '_ ` _ \ / _ \ \ /\ / / _ \| '__| |/ /
| |  | || |   | \__/\ | | | | | (_| | | | | | | |  __/\ V  V / (_) | |  |   <
\_|  |_/\_|    \____/ |_| |_|  \__,_|_| |_| |_|_|\___| \_/\_/ \___/|_|  |_|\_\

  Build script - MPC Distribution 2.5.1 to adapt to many kinds of systems.

Usage: ./installmpc [OPTION]... [VAR=VALUE]...

Defaults for the options are specified in brackets.

# Information
  --help|-h|-?                          : Display this help and exit
  --version                             : Report version number and exit

# Installation
  --prefix=PREFIX                       : Install architecture-independent files in PREFIX [/usr/local]
  --disable-check-install               : Override installation if it already exists in the prefix
  --disable-check-deps                  : Disable dependency checking

# Build
  --compiler                            : Default compiler
  clean                                 : Delete directories inside build directory
  distclean                             : Delete directories and makefiles inside build directory
```

```
# Download missing deps
  --download-missing-deps                : Download dependencies
  --mirror={1|2|3|4}                     : Choose a mirror for downloading dependencies

# Disable sub packages
  --disable-mpc-gdb                      : Disable gdb
  --disable-mpc-gcc                      : Disable gcc
  --disable-mpc-binutils                 : Disable binutils
  --disable-mpc-fortran                  : Disable fortran

# Specify system subpackages
  --with-mpc-gdb=*                       : Specify gdb prefix on the system
  --with-mpc-gcc=*                       : Specify gcc prefix on the system
  --with-sctk-arch=*                     : Specify sctk_arch prefix on the system
  --with-openpa=*                        : Specify openpa prefix on the system
  --with-mpfr=*                          : Specify mpfr prefix on the system
  --with-gmp=*                           : Specify gmp prefix on the system
  --with-mpc-binutils=*                  : Specify binutils prefix on the system
  --with-hwloc=*                         : Specify hwloc prefix on the system
  --with-libxml2=*                       : Specify libxml2 prefix on the system

# Options to transmit to subpackages
  --mpc-gcc-*                            : Add options to gcc configure
  --mpc-gdb-*                            : Add options to gdb configure
  --sctk-arch-*                          : Add options to sctk-arch configure
  --openpa-*                             : Add options to openpa configure
  --gmp-*                                : Add options to gmp configure
  --mpfr-*                               : Add options to mpfr configure
  --mpc-*                                : Add options to mpc multiprecision library configure
  --mpc-binutils-*                       : Add options to binutils configure
  --libxml2-*                            : Add options to libxml2 configure
  --hwloc-*                              : Add options to hwloc configure
  --mpc-option=*                         : Add options to mpc framework configure

# Cross-compilation
  --target=*                             : Specify architecture for target
  --host=*                               : Specify architecture for host
  --arch-library-path                    : Specify path for architecture libraries

# Features
  --disable-color                        : Disable colors in display
  --verbose=1|2|3                        : Level of verbosity
  -v|-vv|-vvv                            : Level of verbosity
  -jN                                    : Allow N jobs at once (parallel install)
```