Compiling code

https://gitlab.tuwien.ac.at/vsc-public/training/vsc-intro/-/blob/main/compiling/compiling.md

Compiling code

CPU architectures Skylake and Cascadelake:

These architectures are available in the following nodes/partitions:

- VSC-4 Login nodes l4[0-9]
- skylake_0096
- skylake_0384
- skylake_0768
- cascadelake_0384

You can choose between these compilers:

- Intel
- GNU

Usually on these architectures the Intel compiler delivers the best performance.

Compiling code

CPU architectures Zen2 and Zen3:

These architectures are available in the following nodes/partitions:

- VSC-5 Login nodes l5[0-6]
- zen3_0512
- zen3_1024
- zen3_2048
- zen3_0512_a100x2
- zen2_0256_a40x2

You can choose between these compilers:

- Intel
- GNU
- AOCC

Usually on these architectures the AOCC compiler delivers the best performance.

Compiling code

Skylake

Modules:

Intel compilers:

```
skylake VSC-4$ module load intel/19
skylake VSC-4$ module load compiler/latest
skylake VSC-4$ module load intel-oneapi-compilers/2022.2.1-gcc-9.5.0-xg435ds
```

GNU compilers:

```
skylake VSC-4 $ module load --auto gcc/12.2.0-gcc-9.5.0-aegzcbj
skylake VSC-4 $ module load --auto gcc/9.5.0-gcc-8.5.0-3wfbr74
```

Compiling code

Zen

Modules:

Intel compilers:

```
zen VSC-5 $ module load intel/19
zen VSC-5 $ module load compiler/latest
zen VSC-5 $ module load intel-oneapi-compilers/2023.1.0-gcc-8.5.0-6y5tcji
```

GNU compilers:

```
zen VSC-5 $ module load --auto gcc/12.2.0-gcc-9.5.0-ohbahza
zen VSC-5 $ module load --auto gcc/9.5.0-gcc-8.5.0-r47lqjh
```

AOCC compiler:

```
zen VSC-5 $ module load --auto aocc/3.2.0-gcc-8.5.0-hc2lh5d
zen VSC-5 $ module load --auto aocc/4.0.0-gcc-8.5.0-lh4stmz
```

Compiling code

Compiler names

Intel:

```
icc/icx   # C compiler
icpc/icpx # C++ compiler
ifort/ifx # Fortran compiler
```

GNU:

```
gcc       # C compiler
g++/c++   # C++ compiler
gfortran  # Fortran compiler
```

AOCC:

```
clang     # C compiler
clang++   # C++ compiler
flang     # Fortran compiler
```

Compiling code

Examples: compiling serial code

- Compiler: Intel
- Language: C
- *~training/examples/15_compiling/hello.c*

```
skylake/zen VSC-4/5 $ module load compiler/latest
```

Compile without optimization:

```
skylake/zen VSC-4/5 $ icc -O0 hello.c -o hello_c
skylake/zen VSC-4/5 $ ./hello_c
Hello World
```

Compile with optimization:

```
skylake/zen VSC-4/5 $ icc -O3 -xHost hello.c -o hello_c
skylake/zen VSC-4/5 $ ./hello_c
Hello World
```

Compiling code

Examples: compiling serial code

- Compiler: GNU
- Language: C
- *~training/examples/15_compiling/hello.c*

```
skylake VSC-4 $ module load --auto gcc/12.2.0-gcc-9.5.0-aegzcbj
zen VSC-5 $ module load --auto gcc/12.2.0-gcc-9.5.0-ohbahza
```

Compile without optimization:

```
skylake/zen VSC-4/5 $ gcc -O0 hello.c -o hello_c
skylake/zen VSC-4/5 $ ./hello_c
Hello World
```

Compiling code

Examples: compiling serial code

- Compiler: GNU
- Language: C
- *~training/examples/15_compiling/hello.c*

  Compile with optimization:

```
skylake VSC-4 $ gcc -O2 -march=skylake hello.c -o hello_c
zen VSC-5 $ gcc -O2 -march=znver3 hello.c -o hello_c
```

```
skylake/zen VSC-4/5 $ ./hello_c
Hello World
```

Compiling code

Examples: compiling serial code

- Compiler: AOCC
- Language: C
- *~training/examples/15_compiling/hello.c*

```
zen VSC-5 $ module load --auto aocc/3.2.0-gcc-8.5.0-hc2lh5d
```

Compile without optimization:

```
zen VSC-5 $ clang -O0 hello.c -o hello_c
zen VSC-5 $ ./hello_c
Hello World
```

Compiling code

Examples: compiling serial code

- Compiler: AOCC
- Language: C
- *~training/examples/15_compiling/hello.c*

  Compile with optimization:

```
zen VSC-5 $ clang -O3 -march=znver3 hello.c -o hello_c
```

```
zen VSC-5 $ ./hello_c
Hello World
```

Compiling code

Examples: compiling OPENMP code:

- Compiler: Intel
- Language: C
- *~training/examples/15_compiling/hello-openmp.c*

```
skylake/zen VSC-4/5 $ icc -qopenmp hello_openmp.c -o hello_openmp_c
skylake/zen VSC-4/5 $ export OMP_NUM_THREADS=2
skylake/zen VSC-4/5 $ ./hello_openmp_c
Hello World... from thread = 0
Hello World... from thread = 1
```

Compiling code

Examples: compiling OPENMP code:

- Compiler: GNU
- Language: C
- *~training/examples/15_compiling/hello_openmp.c*

```
skylake/zen VSC-4/5 $ gcc -fopenmp hello_openmp.c -o hello_openmp_c
skylake/zen VSC-4/5 $ export OMP_NUM_THREADS=2
skylake/zen VSC-4/5 $ ./hello_openmp_c
Hello World... from thread = 0
Hello World... from thread = 1
```

Compiling code

Examples: compiling OPENMP code:

- Compiler: AOCC
- Language: C
- *~training/examples/15_compiling/hello_openmp.c*

```
skylake/zen VSC-4/5 $ clang -fopenmp hello_openmp.c hello_openmp_c
skylake/zen VSC-4/5 $ export OMP_NUM_THREADS=2
skylake/zen VSC-4/5 $ ./hello_openmp_c
Hello World... from thread = 0
Hello World... from thread = 1
```

Compiling code

Compiling MPI code:

- Use compiler wrappers
- Various MPI implementations available:
  - Intel MPI
  - OpenMPI

Compiling code

Compiling MPI code: Intel MPI

Compiler wrappers for Intel compiler:

```
mpiicc              # C
mpiicpc             # C++
mpiifort            # Fortran
```

Compiler wrappers for GNU compiler:

```
mpicc/mpigcc        # C
mpicxx/mpigxx       # C++
mpif90/mpif77/mpifc # Fortran
```

Compiling code

Compiling MPI code: Intel MPI

```
skylake VSC-4 $ module purge
skylake VSC-4 $ module load compiler/latest intel-oneapi-mpi/2021.7.1-
oneapi-2022.2.1-qosid72
skylake VSC-4 $ mpiicc -show
skylake VSC-4 $ module purge
skylake VSC-4 $ module load --auto gcc/12.2.0-gcc-9.5.0-aegzcbj intel-
oneapi-mpi/2021.7.1-oneapi-2022.2.1-qosid7
skylake VSC-4 $ mpicc -show
```

Compiling code

Compiling MPI code: OpenMPI

Compiler specific installations, e.g.:

```
skylake VSC-4 $ module load openmpi/4.1.4-intel-2021.7.1-idwkgjx
skylake VSC-4 $ module load openmpi/4.1.4-gcc-12.2.0-c7xovdh
```

```
zen VSC-5 $ module load openmpi/4.1.4-gcc-12.2.0-ehj6isf
zen VSC-5 $ module load openmpi/4.1.4-aocc-4.0.0-2canz76
```

Compiler wrappers:

```
mpicc                   # C
mpic++/mpiCC/mpicxx     # C++
mpif77/mpif90/mpifort   # Fortran
```

Compiling code

Examples: compiling MPI code

- Compiler: Intel
- MPI: Intel
- Language: C
- *~training/examples/15_compiling/hello_mpi.c*

```
skylake VSC-4 $ module purge
skylake VSC-4 $ module load compiler/latest intel-oneapi-mpi/2021.7.1-
oneapi-2022.2.1-qosid72
```

Compile without optimization:

```
skylake VSC-4 $ mpiicc -O0 hello-mpi.c -o hello-mpi_c
skylake VSC-4 $ mpirun -np 2 ./hello-mpi_c
Hello world from processor l44, rank 1 out of 2 processors
Hello world from processor l44, rank 0 out of 2 processors
```

Compiling code

Examples: compiling MPI code

- Compiler: Intel
- MPI: Intel
- Language: C
- *~training/examples/15_compiling/hello_mpi.c*

```
skylake VSC-4 $ module purge
skylake VSC-4 $ module load compiler/latest intel-oneapi-mpi/2021.7.1-
oneapi-2022.2.1-qosid72
```

Compile with optimization:

```
skylake VSC-4 $ mpiicc -O3 -xHost hello-mpi.c -o hello-mpi_c
skylake VSC-4 $ mpirun -np 2 ./hello-mpi_c
Hello world from processor l44, rank 1 out of 2 processors
Hello world from processor l44, rank 0 out of 2 processors
```

Compiling code

Examples: compiling MPI code

- Compiler: GNU
- MPI: Intel
- Language: C
- *~training/examples/15_compiling/hello_mpi.c*

```
skylake VSC-4 $ module purge
skylake VSC-4 $ module load --aut gcc/12.2.0-gcc-9.5.0-aegzcbj intel-oneapi-
mpi/2021.7.1-oneapi-2022.2.1-qosid72
```

Compile without optimization:

```
skylake VSC-4 $ mpicc -O0 hello-mpi.c -o hello-mpi_c
skylake VSC-4 $ mpirun -np 2 ./hello-mpi_c
Hello world from processor l44, rank 1 out of 2 processors
Hello world from processor l44, rank 0 out of 2 processors
```

Compiling code

Examples: compiling MPI code

- Compiler: GNU
- MPI: Intel
- Language: C
- *~training/examples/15_compiling/hello_mpi.c*

```
skylake VSC-4 $ module purge
skylake VSC-4 $ module load --auto gcc/12.2.0-gcc-9.5.0-aegzcbj intel-
oneapi-mpi/2021.7.1-oneapi-2022.2.1-qosid72
```

Compile with optimization:

```
skylake VSC-4 $ mpicc -O2 -march=skylake hello-mpi.c -o hello-mpi_c
skylake VSC-4 $ mpirun -np 2 ./hello-mpi_c
Hello world from processor l44, rank 1 out of 2 processors
Hello world from processor l44, rank 0 out of 2 processors
```

Compiling code

Examples: compiling MPI code

- Compiler: AOCC
- MPI: Openmpi
- Language: C
- *~training/examples/15_compiling/hello_mpi.c*

```
zen VSC-5 $ module purge
zen VSC-5 $ module load openmpi/4.1.4-aocc-4.0.0-2canz76
```

Compile without optimization:

```
zen VSC-5 $ mpicc -O0 hello-mpi.c -o hello-mpi_c
zen VSC-5 $ mpirun -np 2 ./hello-mpi_c
Hello world from processor l54.vsc.xcat, rank 1 out of 2 processors
Hello world from processor l54.vsc.xcat, rank 0 out of 2 processors
```

Compiling code

Examples: compiling MPI code

- Compiler: AOCC
- MPI: Openmpi
- Language: C
- *~training/examples/15_compiling/hello_mpi.c*

```
zen VSC-5 $ module purge
zen VSC-5 $ module load openmpi/4.1.4-aocc-4.0.0-2canz76
```

Compile with optimization:

```
zen VSC-5 $ mpicc -O2 -march=znver3 hello-mpi.c -o hello-mpi_c
zen VSC-5 $ mpirun -np 2 ./hello-mpi_c
Hello world from processor l54.vsc.xcat, rank 1 out of 2 processors
Hello world from processor l54.vsc.xcat, rank 0 out of 2 processors
```