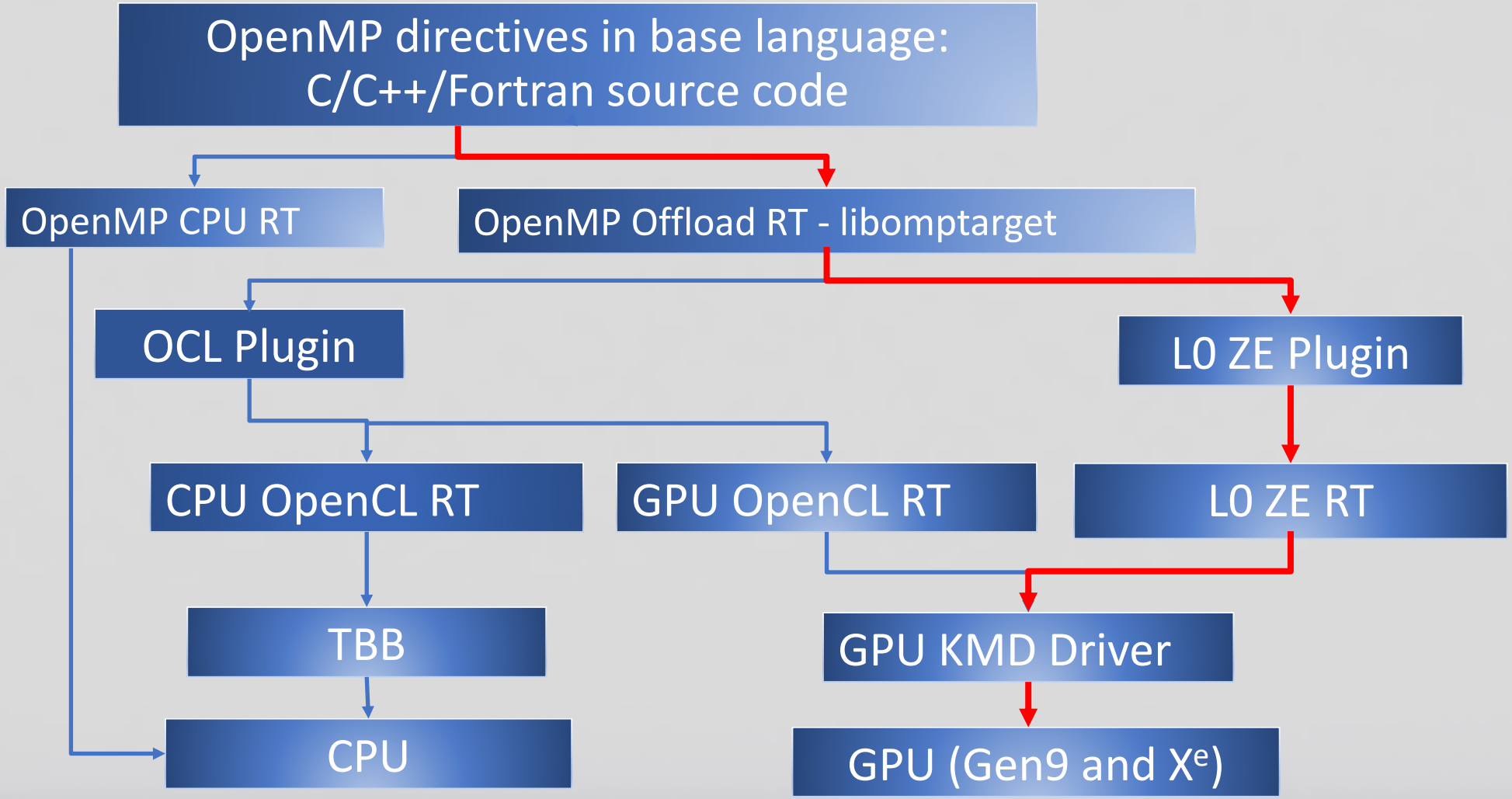
Fortran, OpenMP offload to GPU Demo

Supporting OpenMP Standard, Intel’s OMP Runtime Implementation

High level architecture of OpenMP Offload to GPU

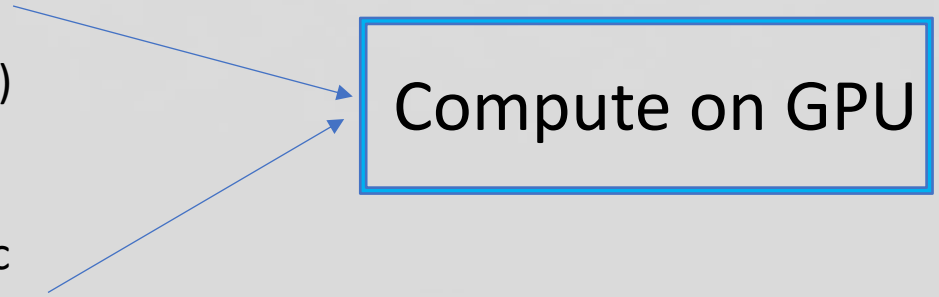


Auto offload “Do Concurrent” to GPU

There is no need to change the source code, only by adding the following compiler flag **`-fopenmp-target-do-concurrent`**, the native Fortran language parallelism feature, which is the do concurrent loop, will run on GPU.

```
ifx -xhost -qopenmp -fopenmp-targets:spir64 -fopenmp-target-do-concurrent source.f90
```

```
subroutine add_vec
    do concurrent (i=1:N)
        c(i)=a(i)+b(i)
    enddo
end subroutine add_vec
```



Just-In-Time (JIT) and Ahead-of-Time (AOT) Compilation

JIT compilation

```
ifx –qopenmp –fopenmp-targets=spir64 source.f90
```

AOT compilation : the target device binary is generated during compilation, not runtime **NEW**

```
ifx -qopenmp -fopenmp-targets=spir64_gen -Xopenmp-target-backend “-device <dev>” source.f90
```

-fopenmp-targets=spir64 instructs OMP Offload Generates SPIRV code fat binary for offload kernels

_gen suffix spir64 instructs OMP to perform AOT **NEW**

`-Xopenmp-target-backend “-device <dev_name>” specifies the target device model name` NEW

<dev_name> is your target, use ‘ocloc compile –help’ for list of targets. If you can access the target machine, the device model name can be queried by sycl-ls, such as [0x0bd6] for the PVC GPU.

Essential OpenMP Environment Variables

export LIBOMPTARGET_PLUGIN_PROFILE=T

LLVM OpenMP Runtime ENV vars are accepted. Performance profiling for tracking on GPU kernel start/complete time and data-transfer time.

export LIBOMPTARGET_DEBUG=1

Dumps offload runtime debug information. Default value is 0 indicates no offloading runtime debugging information dump.

Auto offload “Do Concurrent” to GPU

```
shiquans@c001n0037:~/testFortranOpenMPGPU> export LIBOMPTARGET_PLUGIN_PROFILE=T
shiquans@c001n0037:~/testFortranOpenMPGPU> cat demo.f90
program auto_offload_do_concurrent
  integer, parameter :: N=100000
  integer :: i
  real :: a(N)=1.0d0, b(N)=2.0d0, c(N)=0.0d0, sumc=0.0d0

  call addVec(a,b,c,N)

  do i=1,N
    sumc=sumc+c(i)
  enddo
  print *,' sumc=300,000=',sumc
end program auto_offload_do_concurrent

subroutine addVec(a,b,c,nv)
  integer::nv
  real::a(nv),b(nv),c(nv)
  do concurrent (i=1:nv)
    c(i)=a(i)+b(i)
  enddo
end subroutine addVec

shiquans@c001n0037:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets:spir64 -fopenmp-target-do-concurrent demo.f90
shiquans@c001n0037:~/testFortranOpenMPGPU> ./a.out
  sumc=300,000=   300000.0
=========================================================================================================
LIBOMPTARGET_PLUGIN_PROFILE(LEVEL0) for OMP DEVICE(0) Intel(R) Graphics [0x0bd6], Thread 0
---------------------------------------------------------------------------------------------------------
Kernel 0                  : __omp_offloading_3e_487d8356_addvec__l17
---------------------------------------------------------------------------------------------------------
                          : Host Time (msec)                          Device Time (msec)
Name                      :      Total   Average       Min       Max     Total   Average       Min       Max     Count
---------------------------------------------------------------------------------------------------------
Compiling                 :     447.60    447.60    447.60    447.60      0.00      0.00      0.00      0.00      1.00
DataAlloc                 :       0.99      0.09      0.00      0.30      0.00      0.00      0.00      0.00     11.00
DataRead (Device to Host) :       0.37      0.12      0.12      0.13      0.05      0.02      0.02      0.02      3.00
DataWrite (Host to Device):       1.07      0.36      0.21      0.59      0.08      0.03      0.02      0.03      3.00
Kernel 0                  :       3.30      3.30      3.30      3.30      0.06      0.06      0.06      0.06      1.00
Linking                   :       0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00      1.00
OffloadEntriesInit        :       2.03      2.03      2.03      2.03      0.00      0.00      0.00      0.00      1.00
=========================================================================================================
shiquans@c001n0037:~/testFortranOpenMPGPU>
```

Auto offload “Do Concurrent” to GPU

```
shiquans@c001n0037:~/testFortranOpenMPGPU> export LIBOMPTARGET_PLUGIN_PROFILE=T
shiquans@c001n0037:~/testFortranOpenMPGPU> cat demo.f90
program auto_offload_do_concurrent
  integer, parameter :: N=100000
  integer :: i
  real :: a(N)=1.0d0, b(N)=2.0d0, c(N)=0.0d0, sumc=0.0d0

  call addVec(a,b,c,N)

  do i=1,N
    sumc=sumc+c(i)
  enddo
  print *,' sumc=300,000=',sumc
end program auto_offload_do_concurrent

subroutine addVec(a,b,c,nv)
  integer::nv
  real::a(nv),b(nv),c(nv)
  do concurrent (i=1:nv)
    c(i)=a(i)+b(i)
  enddo
end subroutine addVec

shiquans@c001n0037:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets:spir64 -fopenmp-target-do-concurrent demo.f90
shiquans@c001n0037:~/testFortranOpenMPGPU> ./a.out
  sumc=300,000=   300000.0
=========================================================================================================
LIBOMPTARGET_PLUGIN_PROFILE(LEVEL0) for OMP DEVICE(0) Intel(R) Graphics [0x0bd6], Thread 0
---------------------------------------------------------------------------------------------------------
Kernel 0                  : __omp_offloading_3e_487d8356_addvec__l17
---------------------------------------------------------------------------------------------------------
                          : Host Time (msec)                          Device Time (msec)
Name                      :      Total   Average       Min       Max     Total   Average       Min       Max     Count
---------------------------------------------------------------------------------------------------------
Compiling                 :     447.60    447.60    447.60    447.60      0.00      0.00      0.00      0.00      1.00
DataAlloc                 :       0.99      0.09      0.00      0.30      0.00      0.00      0.00      0.00     11.00
DataRead (Device to Host) :       0.37      0.12      0.12      0.13      0.05      0.02      0.02      0.02      3.00
DataWrite (Host to Device):       1.07      0.36      0.21      0.59      0.08      0.03      0.02      0.03      3.00
Kernel 0                  :       3.30      3.30      3.30      3.30      0.06      0.06      0.06      0.06      1.00
Linking                   :       0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00      1.00
OffloadEntriesInit        :       2.03      2.03      2.03      2.03      0.00      0.00      0.00      0.00      1.00
=========================================================================================================
shiquans@c001n0037:~/testFortranOpenMPGPU>
```

Auto offload “Do Concurrent” to GPU

```
shiquans@c001n0037:~/testFortranOpenMPGPU> export LIBOMPTARGET_PLUGIN_PROFILE=T
shiquans@c001n0037:~/testFortranOpenMPGPU> cat demo.f90
program auto_offload_do_concurrent
  integer, parameter :: N=100000
  integer :: i
  real :: a(N)=1.0d0, b(N)=2.0d0, c(N)=0.0d0, sumc=0.0d0

  call addVec(a,b,c,N)

  do i=1,N
    sumc=sumc+c(i)
  enddo
  print *,' sumc=300,000=',sumc
end program auto_offload_do_concurrent

subroutine addVec(a,b,c,nv)
  integer::nv
  real::a(nv),b(nv),c(nv)
  do concurrent (i=1:nv)
    c(i)=a(i)+b(i)
  enddo
end subroutine addVec

shiquans@c001n0037:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets:spir64 -fopenmp-target-do-concurrent demo.f90
shiquans@c001n0037:~/testFortranOpenMPGPU> ./a.out
  sumc=300,000=   300000.0
================================================================================
LIBOMPTARGET_PLUGIN_PROFILE(LEVEL0) for OMP DEVICE(0) Intel(R) Graphics [0x0bd6], Thread 0
--------------------------------------------------------------------------------
Kernel 0                  : __omp_offloading_3e_487d8356_addvec_l17
--------------------------------------------------------------------------------
                          : Host Time (msec)                          Device Time (msec)
Name                      :      Total   Average       Min       Max     Total   Average       Min       Max     Count
--------------------------------------------------------------------------------
Compiling                 :     447.60    447.60    447.60    447.60      0.00      0.00      0.00      0.00      1.00
DataAlloc                 :       0.99      0.09      0.00      0.30      0.00      0.00      0.00      0.00     11.00
DataRead (Device to Host) :       0.37      0.12      0.12      0.13      0.05      0.02      0.02      0.02      3.00
DataWrite (Host to Device):       1.07      0.36      0.21      0.59      0.08      0.03      0.02      0.03      3.00
Kernel 0                  :       3.30      3.30      3.30      3.30      0.06      0.06      0.06      0.06      1.00
Linking                   :       0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00      1.00
OffloadEntriesInit        :       2.03      2.03      2.03      2.03      0.00      0.00      0.00      0.00      1.00
================================================================================
shiquans@c001n0037:~/testFortranOpenMPGPU>
```

Auto offload “Do Concurrent” to GPU

```
shiquans@c001n0037:~/testFortranOpenMPGPU> export LIBOMPTARGET_PLUGIN_PROFILE=T
shiquans@c001n0037:~/testFortranOpenMPGPU> cat demo.f90
program auto_offload_do_concurrent
  integer, parameter :: N=100000
  integer :: i
  real :: a(N)=1.0d0, b(N)=2.0d0, c(N)=0.0d0, sumc=0.0d0

  call addVec(a,b,c,N)

  do i=1,N
    sumc=sumc+c(i)
  enddo
  print *,' sumc=300,000=',sumc
end program auto_offload_do_concurrent

subroutine addVec(a,b,c,nv)
  integer::nv
  real::a(nv),b(nv),c(nv)
  do concurrent (i=1:nv)
    c(i)=a(i)+b(i)
  enddo
end subroutine addVec

shiquans@c001n0037:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets:spir64 -fopenmp-target-do-concurrent demo.f90
shiquans@c001n0037:~/testFortranOpenMPGPU> ./a.out
  sumc=300,000=   300000.0
================================================================================
LIBOMPTARGET_PLUGIN_PROFILE(LEVEL0) for OMP DEVICE(0) Intel(R) Graphics [0x0bd6], Thread 0
--------------------------------------------------------------------------------
Kernel 0                  : __omp_offloading_3e_487d8356_addvec_l17
--------------------------------------------------------------------------------
                          : Host Time (msec)                        Device Time (msec)
Name                      :      Total   Average       Min       Max     Total   Average       Min       Max     Count
--------------------------------------------------------------------------------
Compiling                 :     447.60    447.60    447.60    447.60      0.00      0.00      0.00      0.00      1.00
DataAlloc                 :       0.99      0.09      0.00      0.30      0.00      0.00      0.00      0.00     11.00
DataRead (Device to Host) :       0.37      0.12      0.12      0.13      0.05      0.02      0.02      0.02      3.00
DataWrite (Host to Device):       1.07      0.36      0.21      0.59      0.08      0.03      0.02      0.03      3.00
Kernel 0                  :       3.30      3.30      3.30      3.30      0.06      0.06      0.06      0.06      1.00
Linking                   :       0.00      0.00      0.00      0.00      0.00      0.00      0.00      0.00      1.00
OffloadEntriesInit        :       2.03      2.03      2.03      2.03      0.00      0.00      0.00      0.00      1.00
================================================================================
shiquans@c001n0037:~/testFortranOpenMPGPU>
```

export LIBOMPTARGET_DEBUG=1

```
Libomptarget --> Launching target execution __omp_offloading_3e_487d8356_addvec__l17 with pointer 0x00000000035e7900 (index=0).
Target LEVEL0 RTL --> Executing a kernel 0x00000000035e7900...
Target LEVEL0 RTL --> Assumed kernel SIMD width is 32
Target LEVEL0 RTL --> Preferred team size is multiple of 64
Target LEVEL0 RTL --> Loop 0: lower bound = 0, upper bound = 99999, Stride = 1
Target LEVEL0 RTL --> Team sizes = {64, 1, 1}
Target LEVEL0 RTL --> Number of teams = {1563, 1, 1}
Target LEVEL0 RTL --> Kernel Pointer argument 0 (value: 0xff002aaaaa400000) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Pointer argument 1 (value: 0xff002aaaaa480000) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Pointer argument 2 (value: 0xff002aaaaa500000) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Scalar argument 3 (value: 0x00000000000186a0) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Scalar argument 4 (value: 0x00000000000186a0) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Scalar argument 5 (value: 0x00000000000186a0) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Scalar argument 6 (value: 0x00000000000186a0) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Scalar argument 7 (value: 0x000000000001869f) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Scalar argument 8 (value: 0x0000000000000000) was set successfully for device 0.
Target LEVEL0 RTL --> Kernel Scalar argument 9 (value: 0x00000000000186a0) was set successfully for device 0.
Target LEVEL0 RTL --> Setting indirect access flags 0x0000000000000002
Target LEVEL0 RTL --> Created a command list 0x0000000004451840 (Ordinal: 0) for device 0.
Target LEVEL0 RTL --> Submitted kernel 0x0000000004399dd0 to device 0
Target LEVEL0 RTL --> Executed kernel entry 0x00000000035e7900 on device 0
```

Ahead-of-Time (AOT) Compilation

```
shiquans@c001n0033:~/testFortranOpenMPGPU> sycl-ls | grep "ext_oneapi_level_zero:gpu:0"
[ext_oneapi_level_zero:gpu:0] Intel(R) Level-Zero, Intel(R) Graphics [0x0bd6] 1.3 [1.3.23937]
shiquans@c001n0033:~/testFortranOpenMPGPU>
shiquans@c001n0033:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets=spir64 demo.f90
shiquans@c001n0033:~/testFortranOpenMPGPU>
shiquans@c001n0033:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets=spir64_gen -Xopenmp-ta
rget-backend "-device 0x0bd6" -o a.out-pvc demo.f90
Auto-detected target based on 0x0bd6 device id: pvc
Compilation from IR - skipping loading of FCL
Build succeeded.
shiquans@c001n0033:~/testFortranOpenMPGPU> ls -l a.out*
-rwxr-xr-x 1 shiquans intelall  1979640 Dec  3 18:06 a.out
-rwxr-xr-x 1 shiquans intelall 94287096 Dec  3 18:06 a.out-pvc
shiquans@c001n0033:~/testFortranOpenMPGPU>
```

Ahead-of-Time (AOT) Compilation

```
shiquans@c001n0033:~/testFortranOpenMPGPU> sycl-ls | grep "ext_oneapi_level_zero:gpu:0"
[ext_oneapi_level_zero:gpu:0] Intel(R) Level-Zero, Intel(R) Graphics [0x0bd6] 1.3 [1.3.23937]
shiquans@c001n0033:~/testFortranOpenMPGPU>
shiquans@c001n0033:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets=spir64 demo.f90
shiquans@c001n0033:~/testFortranOpenMPGPU>
shiquans@c001n0033:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets=spir64_gen -Xopenmp-ta
rget-backend "-device 0x0bd6" -o a.out-pvc demo.f90
Auto-detected target based on 0x0bd6 device id: pvc
Compilation from IR - skipping loading of FCL
Build succeeded.
shiquans@c001n0033:~/testFortranOpenMPGPU> ls -l a.out*
-rwxr-xr-x 1 shiquans intelall  1979640 Dec  3 18:06 a.out
-rwxr-xr-x 1 shiquans intelall 94287096 Dec  3 18:06 a.out-pvc
shiquans@c001n0033:~/testFortranOpenMPGPU>
```

Ahead-of-Time (AOT) Compilation

```
shiquans@c001n0033:~/testFortranOpenMPGPU> sycl-ls | grep "ext_oneapi_level_zero:gpu:0"
[ext_oneapi_level_zero:gpu:0] Intel(R) Level-Zero, Intel(R) Graphics [0x0bd6] 1.3 [1.3.23937]
shiquans@c001n0033:~/testFortranOpenMPGPU>
shiquans@c001n0033:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets=spir64 demo.f90
shiquans@c001n0033:~/testFortranOpenMPGPU>
shiquans@c001n0033:~/testFortranOpenMPGPU> ifx -qopenmp -fopenmp-targets=spir64_gen -Xopenmp-ta
rget-backend "-device 0x0bd6" -o a.out-pvc demo.f90
Auto-detected target based on 0x0bd6 device id: pvc
Compilation from IR - skipping loading of FCL
Build succeeded.
shiquans@c001n0033:~/testFortranOpenMPGPU> ls -l a.out*
-rwxr-xr-x 1 shiquans intelall  1979640 Dec  3 18:06 a.out
-rwxr-xr-x 1 shiquans intelall 94287096 Dec  3 18:06 a.out-pvc
shiquans@c001n0033:~/testFortranOpenMPGPU>
```

New Intel® Fortran Compiler (ifx) Product Highlights:

- Available in the Intel® oneAPI HPC Toolkit 2022.3.
- Production-ready for CPUs and GPUs.
- Based on ifort frontend and runtime libraries, and uses LLVM backend compiler technology.
- In addition to Fortran 2018, ifx also supports from FORTRAN 77 to Fortran 2008, all main versions of Fortran language standards.
- Supports OpenMP 4.5 and OpenMP 5.x directives and GPU offloading features.

This robust implementation provides Fortran programmers access to many capabilities of Intel Data Center GPUs right from their native language.