**INTRODUCTION**

This study explores a practical way to mix two primary paradigms for accelerators – OpenACC and OpenMP4.5 – in the same code. There are several reasons to do so:

- Incremental porting without starting from the host codes (error-prone, and sometimes impossible)
- To use libraries written in another paradigm
- Performance studies for different paradigms
- Debugging

OpenACC and OpenMP4.5 share a similar programming model, and many directives have a prerequisite for any code which uses both paradigms. This study explores a practical way to mix two primary paradigms for accelerators – OpenACC and OpenMP4.5. The Makefile looks like:

```
run : 

$(objects): $(modules)
```

The challenges come from three fronts:

- Interoperability of Fortran implementations and runtimes
- Interoperability of OpenACC and OpenMP4.5 runtimes
- Complications from compilation and source files

**THE ORIGINAL OPENACC CODE**

For simplicity, this study is focused on PGI (for OpenACC) and IBM XL (for OpenMP4.5) on IBM “Minsky” systems (POWER8, Nvidia P100). The original OpenACC Fortran code looks like:

```fortran
program mixaccomp4
    implicit none
    real n :: sub1(n,arr)
    integer i
    implicit none
    integer(i) int(i)
    integer intent(i) int(i)
    subroutine sub1(n,arr)
        real*8 a(i)
        integer(i)
        interface
            subroutine sub1(n,arr)
                implicit none
                integer intent(i) int(n)
                integer(i) int(n)
            end subroutine
        end interface
```

The expected output is:

```
1.0 2.0 3.0 4.0 5.0
6.0 7.0 8.0 9.0
10.0
```

**INCREMENTAL PORTING**

To demonstrate the possibility of incremental porting, only one of the two OpenACC kernels are ported to OpenMP4.5. Data management is still handled by OpenACC.

Incompatible Fortran object name mangling by the two compiler suits can be resolved by using the BIND(C)

Currently, the XL Fortran compiler must be used as the host linker. The original PGI main program is turned into a subroutine, and then called by a wrapper XL C main program. A dummy OpenMP4.5 kernel is placed before the PGI main subroutine, so that the OpenMP4.5 offloading environment is initialized before the initialization of OpenACC. If not done, OpenMP4.5 kernels will fail inside OpenACC data regions.

**Mix-COMPILING AND RESULTS**

PGI Fortran and OpenACC runtime libraries need to be linked manually. Also, “nordc” must be specified when compiling OpenACC kernels. The Makefile looks like:

```
run : 

$(objects): $(modules)
```

The expected output is:

```
1.0 2.0 3.0 4.0 5.0
6.0 7.0 8.0 9.0
10.0
```

**CONCLUSION AND KNOWN LIMITATIONS**

- It is possible to mix OpenACC and OpenMP4.5 in one code.
- OpenACC variables can be used in OpenMP4.5 kernels.
- Careful use of vendor-independent coding practice allows incremental porting from either direction.

Several known limitations:

- Implementation-dependent data objects (such as OOP objects) cannot be shared across compiler boundary.
- PGI main program cannot access command line arguments
- “nordc” disables the use of external device functions.