C++ on GPUs Using OpenACC and the PGI Accelerator Compilers

Michael Wolfe, Compiler Engineer
The Portland Group
www.pgroup.com
What is OpenACC?

A set of directive-based extensions to C, C++ and Fortran that allow you to annotate regions of code and data for offloading from a CPU host to an attached Accelerator

OpenACC Directives

```c
#pragma acc data copy(a[0:n]) copyin(b[0:n])
{
    ...
    axpy( a, b, 1.0, n );
    ...
}

... axpy( float* y, float* x, float a, int n ){
    #pragma acc parallel loop present(a[0:n],b[0:n])
    for (int i = 0; i < n; i++)
        y[i] += a*x[i];
}
```
NVIDIA Kepler Overall Block Diagram*

CPU+Accelerator
Abstract Machine Architecture

Multicore CPU

Latency Optimized Host Memory

Execution Queues

PE 0

PE 1

PE n-1

SIMD/SMT

Hardware/Software Cache

Hardware/Software Cache

Hardware/Software Cache

Stream Optimized Device Memory

©2014 NVIDIA Corp.
OpenACC Directives

```c
#pragma acc data copy(a[0:n]) copyin(b[0:n])
{
    ...
    axpy( a, b, 1.0, n );
    ...
}
...
void axpy( float* y, float* x, float a, int n ){
    #pragma acc parallel loop present(y[0:n],x[0:n])
    for (int i = 0; i < n; i++)
        y[i] += a*x[i];
}
```
template<typename vtype>
class myvector{
    vtype* data;
    size_t size;
public:
    void devcopyin(){
        #pragma acc enter data create(this)
        #pragma acc enter data copyin(data[0:size])
    }
    void updatedev(){
        #pragma acc update device(data[0:size])
    }
    void axpy( myvector<vtype>& x, vtype a ){
        #pragma acc parallel loop present(this,x)
        for( int i = 0; i < size; ++i )
            data[i] += a*x[i];
    }
}
Building OpenACC Programs

% pgc++ -acc -Minfo=accel -c foo.c
% pgc++ -acc -Minfo=accel -o phoo foo.o
% phoo

Useful command line options:

-acc
-ta -ta=tesla:cc35 -ta=tesla:nordc
-Minfo -Minfo=accel
-help
OpenACC C++ Usage

- Move data to the device
  - data construct or enter data / exit data directives
  - must create / copyin the class before the dynamic data members
  - in the class, must create ‘this’ before members
- Compute on the device
  - acc parallel
- Maintain data coherence
  - update directives to copy data host->device->host
- Deep copy will come to allow std::vector<>
C++ on GPUs Using OpenACC and the PGI Accelerator Compilers

- trs@pgroup.com for bug reports
- www.pgroup.com/userforum
- www.pgroup.com/openacc
- www.openacc.org