

GPU implementation using PGI directives

- Kernel generated at compilation time by adding directives in the code.
Example of a matrix multiply to be compiled for an accelerator

!\$acc region

```
do k = 1,n1
  do i = 1,n3
    c(i,k) = 0.0
    do j = 1,n2
      c(i,k) = c(i,k) + a(i,j) * b(j,k)
    enddo
  enddo
enddo
```

!\$acc end region

- Grid and block sizes are automatically set by the compiler or can be manually tuned using the **parallel** and **vector** keywords
- Mirror** and **reflected** keywords enable to declare GPU resident data arrays, thus avoiding data transfer between multiple kernel calls.
- Based on other codes experiences (WRF, fluid dynamics) ¹, directly adding directives to existing code may not be very efficient : still requires some re-writing to get better performance (loop reordering ...)
- Limitations : calls to subroutines within acc region need to be inlined, ...

¹ <http://www.pgroup.com/resources/accel.htm>