OpenMP for Computational Scientists

Wrap up

Dr Tom Deakin
University of Bristol

Tuesday 1 December, 2020

University of BRISTOL

OpenMP 5 adds features to make writing performance portable programs simpler. Highlighting some applicable to target:

- Loop construct
- Mappers
- Unified Shared Memory (USM)
- Function variants
- Reverse offload
- `OMP_TARGET_OFFLOAD`
- Reduction variables now implicitly `map(tofrom)`

Loop

University of BRISTOL

- Assert that the iterations in a loop nest may execute in any order, including concurrently
- Let the compiler figure our how to best utilize parallel resources

```
!$omp target
!$omp loop
do i = 1, N
  a(i) = b(i)
end do
!$omp end loop
!$omp end target
```

Unified shared memory

Code requires specific features, e.g. shared memory between host and devices.

```
real(kind=8), dimension(:), allocatable :: A
allocate(A(1024))

!$omp requires unified_shared_memory

!$omp target
  call do_something_with_A(A)
!$omp end target
```

No map clauses. Data is shared between the host and device.

OpenMP resources

University of BRISTOL

- Two brilliant books from MIT Press:
  - The OpenMP Common Core: Making OpenMP Simple Again — Tim Mattson, Yun (Helen) Ye and Alice Koniges.
  - Using OpenMP - The Next Steps — Ruud van de Pas, Eric Stotzer and Christian Terboven.
- OpenMP website: `https://www.openmp.org`
  - The specification (not for the faint hearted).
  - Download summary cards.
  - List of compiler support.
  - Example code for all the directives.
  - List of books: `https://www.openmp.org/resources/openmp-books/`