Australian Government
Bureau of Meteorology

PSyclone and Existing Applications

Rupert Ford, Andy Porter, Sergi Siso, STFC Hartree Centre
Iva Kavcic, Chris Maynard, Andrew Coughtrie, UK Met Office
Joerg Henrichs, Australian Bureau of Meteorology

ESIWACE2 training course on Domain-specific Languages in Weather and Climate, 23rd-27th November 2020

Overview

- Using NEMO API for other program - Evolution
  - Creating a hybrid version of ROMS
  - Config file settings
- Porting an application to PSyclone - Revolution
  - Introduction to dl_esm_inf
  - Best practice

Australian Government
Bureau of Meteorology

Recap NEMO API - Evolution

- Use PSyclone with existing Fortran program
- Automatically detect kernels
  - Based on coding style (e.g. usage of loop variables):
    - Do loop with *jk* indicate loops over level
    - …
- No infrastructure library is used
  - NEMO provides distributed memory functionality etc.
- Once kernels are identified, PSyclone transformations can be used

Australian Government
Bureau of Meteorology

The Config File

- Contains settings for PSyclone
  - Read at startup
  - Default version installed when installing PSyclone
  - You can use a custom version for each project
    - Use the `-c` command line options
- Config file has different sections
  - One default section
    - Generic setting for all APIs
  - One for each API
    - Only the one you use is required

The NEMO Section in the Config File

```
[nemo]
# The valid types of loop and associated loop variable
and bounds:
mapping-lon = var: ji, start: 1, stop: jpi
mapping-lat = var: jj, start: 1, stop: jpj
mapping-levels = var: jk, start: 1, stop: jpk
mapping-tracers = var: jt, start: 1, stop:
mapping-unknown = var: , start: 1, stop:

# Used for converting implicit loops to explicit loops
index-order = lon, lat, levels, tracers
```

Australian Government
Bureau of Meteorology

The NEMO Section in the Config File

```
[nemo]
# The valid types of loop and associated loop variable
and bounds:
mapping-lon = var: ji, start: 1, stop:
mapping-lat = var: jj, start: 1, sto
mapping-levels = var: jk, start: 1,
mapping-tracers = var: jt, start: 1,
mapping-unknown = var: , start: 1,

# Used for converting implicit loops to explicit loops
index-order = lon, lat, levels, tracers
```

Start and stop are not used in detecting kernel loops

Australian Government
Bureau of Meteorology

Modified Config File

```
mapping-lon = var: i, start: 1, stop: jpi
mapping-lat = var: j, start: 1, stop: jpj
mapping-levels = var: k, start: 1, stop: jpk
mapping-grid = var: np, start: Ns, stop: Ne
mapping-grid2 = var: mp, start: Ns, stop: Ne
mapping-sources = var: is, start: 1, stop: nsrc
```

```
 DO j = JstrV, Jend
     DO i = Istr, Iend
         fac1 = 0.5_r8 * (VFe(i, j) + VFe(i, j - 1))
…
 DO np = Ns, Ne
     DO mp = Ms, Me
```

Outcome

- I developed one PSyclone script to add OpenMP directives to loop
  - Rather sophisticated script that automatically detects large OMP regions and declares private/shared variables
  - Uses internal dependency analysis
- While ROMS supports MPI or OpenMP, it does not support hybrid (both at the same time).
  - With the above script I created a hybrid version

Number of Nodes	Original ROMS	PSyclone ROMS 1 thread	PSyclone 2 threads	PSyclone 4 threads
1	995.37	1002.2	995.9	987.6
24	140.2	153.8	133.3	129.6

Australian Government
Bureau of Meteorology

Porting an Existing Application - Revolution

- A 'complete port' involves:
  - Use an infrastructure library supported by PSyclone
  - Refactor computations into kernels
    - PSyclone will create the loops for you
  - Use `invoke` to call kernels
- Based on porting NOAA's MOST (Method of Splitting Tsunamis) to use PSyclone
  - Using *Daresbury Laboratory Earth-System Modelling Infrastructure library* 'dl_esm_inf'
  - Adding OpenMP only (MPI work in progress)
  - Using GOcean API

Australian Government
Bureau of Meteorology

1 - Basics

- You should have (at least one) **reproducible** test case
- Clean up Fortran if required
  - Remove deprecated features
    - Computed goto, arithmetic if, common blocks, …
  - Use modules to get proper interface
- **Important:**
  Refactor to use assumed-shape arrays:
  …, `dimension(:,:)`
   Because the infrastructure library might align arrays and make them bigger!

Australian Government
Bureau of Meteorology

2 - Create 'Proper' Kernels

-
  ```
  do j=...
      do i=...
          a_fld(i, j) = f(i, j, b_fld, c_fld, …)
      enddo
  enddo
  ```
- Single point, no dependencies to other loop iteration
- Remove loop iteration dependencies:

-
  ```
  do i=1,nx
      if(i.lt.nx) dw2=dw(i+1)
      ! Not set for i=nx, so previous value
  ```

Take Care of Dependencies

```
do j=…
   do i=…
      qw(i)=…
   enddo
   do i=…
      q1(i) = f(qw(i-1), qw(i), qw(i+1), …)
   enddo
enddo
```

Take Care of Dependencies

```
do j=…
   do i=…
      qw(i)=…


      q1(i) = f(qw(i-1), qw(i), qw(i+1), …)
   enddo
enddo
```

Take Care of Dependencies

```
do j=…
   do i=…
      qw(i)=…



      q1(i) = f(qw(i-1), qw(i), qw(i+1), …)
   enddo
enddo
```

Australian Government
Bureau of Meteorology

Use 2d Array and Create Two Kernels

```
do j=…
   do i=…
      qw(i,j)=…
   enddo
enddo
do j=…
   do j=…
      q1(i,j) = f(qw(i-1,j), qw(i,j),   &
                  qw(i+1,j), …)
   enddo
enddo
```

- Must be split into two kernels and use 2d-arrays to satisfy dependencies!

3 - Introduce dl_esm_inf

- Infrastructure for 2d grids (3d grids in the future)
- Supported grid types: GO_ARAKAWA_B/C
- 2d-grids, different boundary conditions
- Different staggering of fields
- Use its field type instead of Fortran arrays

- See:
  https://github.com/stfc/dl_esm_inf
  for details

Australian Government
Bureau of Meteorology

Add Initialisation of dl_esm_inf

```
USE grid_mod,      only: grid_type, &
                         grid_init
use parallel_mod, only: parallel_init

TYPE(grid_type), target   :: grid

call parallel_init()

grid = grid_type(GO_ARAKAWA_C,  (/GO_BC_PERIODIC,    &
                                 GO_BC_PERIODIC,     &
                                 GO_BC_NONE/),       &
                 GO_OFFSET_SW)

call grid%decompose(nx, ny, num_domains, &
                    domX, domY,          &
                    halo_width=1)
call grid_init(grid, dx, dy)
```

Australian Government
Bureau of Meteorology

Change Data Structures

- Replace 2d arrays with dl_esm_inf fields
  - To create a field:

```
use field_mod, only : r2d_field, &
                      T_POINTS
a_field = r2d_field(grid, T_POINTS)
```

  - To use a field as 2d arraym use `field%data`:

```
do j… do i…
    da%data(i,j) = …
ua%data = 0
read (1,*) (da%data(i,j), i=1,nx)
```

- Creating fields is more expensive than 2d Fortran array

4- Use PSyclone

- Convert one kernel at a time
  - Move kernel computation into a module
  - Add meta-data declaration:
    - Parameters, iteration space
  - Create subroutine taking additionally `i, j` index parameters
  - Fields are passed in as simple 2d Fortran arrays

Australian Government
Bureau of Meteorology

GOcean Kernel Metadata

```
 type, extends(kernel_type) :: swlon_adjust
    type(arg), dimension(3) :: meta_args =  &
       (/ arg(READ,       CT, go_stencil(010,     &
                                         111,     &
                                         010)),   & ! da
          arg(READWRITE,  CT, POINTWISE),         & ! ua
          arg(WRITE,      CT, POINTWISE),         & ! qa
       /)
    integer :: ITERATES_OVER = INTERNAL_POINTS
    integer :: index_offset = OFFSET_NE
  contains
    procedure, nopass :: code => swlon_adjust_code
  end type swlon_adjust

contains
```

Australian Government
Bureau of Meteorology

Iteration Spaces

- Iteration space specifies the loop boundaries
- Two implemented by default:
  - GO_INTERNAL_PTS: inside (excluding boundary)
  - GO_ALL_PTS: inside+boundary
- The config file can specify additional iteration space, e.g. to iterate of N/S halo and inner points:

```
iteration-spaces:
…internal_ns_halo:{start}-1:{stop}+1
                 :{start}:{stop}
```

GOcean Kernel Implementation

```
subroutine swlon_adjust_code(i, j, da, ua, qa)

  implicit none
  integer, intent(in)                          :: i, j
  real*8, dimension(:, :), intent(in)          :: da,
  real*8, dimension(:, :), intent(inout)       :: ua
  real*8, dimension(:, :), intent(out)         :: qa

  ua(i, j) = ua(i, j)  + da(i-1,j)+da(i+1,j)  &
                       + da(i,j-1)+da(i,j+1)  &
                       - 4.0*da(i,j)
```

Replace Loops with Invoke

```
use swlon_adjust_mod, only : swlon_adjust

call invoke(name="timestep",                 &
            swlon_adjust(da, ua, qa)   )
```

Invoke PSyclone to create algorithm and psy-layer:

```
psyclone -l -oalg timestep_alg.f90
         -opsy timestep_psy.f90
         -api gocean1.0
         -s my_script.py
         timestep.f90
```

Optional

5 – Optimise/Parallelise

- Write a script that optimised or parallelises the code
    - Or use PSyData for its functionality
- Read the docs for details about GOcean API (including defining your own iteration spaces and more): https://psyclone.readthedocs.io/en/latest/gocean1p0.html

Australian Government
Bureau of Meteorology

Performance

	Intel 17	Cray 8.4.5	GNU Fortran 5.1.0
Original code	**111.0**	115.6	102.4
Kernel-field	258.1	263.6	275.4
PSyclone	**48.5**	31.4	88.7
1d-Halo loops	49.8	29.1	90.0
Optimised C version	*49.0*	*47.0*	-

Outcome

- Due to refactoring the code is easier to understand, and better for the compiler to optimise
  - Run time much better than original code, though mostly due to refactoring:
    Kernel structure enforce simpler code
  - OpenMP performance with PSyclone as good as a hand-parallelised version
  - Code much easier to understand (and shorter)
  - Potential port to distributed Memory (MPI) and GPU from same source code

Australian Government
Bureau of Meteorology

Summary

- The NEMO API allows working with existing code
  - Modify the config file to recognise nested loops as kernels
- PSyclone's support for dl_esm_inf allows to refactor existing 2d code to use PSyclone:
  1. Cleanup code
  2. Refactor to create kernel-like code structure
  3. Introduce dl_esm_inf / replace arrays with fields
  4. Create kernels, and use invoke() with PSyclone
  5. Optimise/parallelise using transformation scripts

Australian Government
Bureau of Meteorology

Thank you

Dr. Joerg Henrichs, BOM
joerg.henrichs@bom.gov.au