

Alexandru Calotoiu

Fortran challenges



Towards Fortran coverage – with code discussion (multi_sdfg branch)



Offsets

AST transformation
Array(-8:34, 45:50) Default: 1



Subroutines

Slices, array slices structure views, global contexts



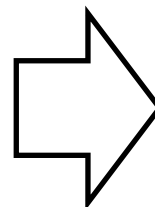
Nested structures

StructArrays in DaCe

Continues in loops

```

26      PROGRAM if_cycle_test
27      implicit none
28      double precision :: d(4)
29      CALL if_cycle_test_function(d)
30      end
31
32      SUBROUTINE if_cycle_test_function(d)
33      double precision d(4,5)
34      integer :: i
35      DO i=1,4
36      |  if (i .eq. 2) CYCLE
37      |  d(i)=5.5
38      ENDDO
39      if (d(2) .eq. 42) d(2)=6.5
40
41
42      END SUBROUTINE if_cycle_test_function
  
```



```

16      for (_for_it_0 = 1; (_for_it_0 <= 4); _for_it_0 = (_for_it_0 + 1)) {
17
18          if ((! (_for_it_0 == 2))) {
19              {
20
21                  {
22                      double d_out_1;
23
24                      ////////////////
25                      // Tasklet code (T_113_c13)
26                      d_out_1 = 5.5;
27                      ////////////////
28
29                      d[(_for_it_0 - 1)] = d_out_1;
30                  }
31              }
32          }
33      }
34
35
36
37
38
  
```

Offsets and placeholders

```

484 TYPE t_nh_state
485
486 !array of prognostic states at different timelevels
487 TYPE(t_nh_prog), ALLOCATABLE :: prog(:)      !< shape: (timelevels)
488 TYPE(t_nh_diag)    :: diag
489 TYPE(t_nh_ref)     :: ref
490 TYPE(t_nh_metrics) :: metrics
491
492 END TYPE t_nh_state

```

```

49 TYPE t_nh_prog
50
51 REAL(wp), POINTER, CONTIGUOUS :: &
52 w(:,:,:), & !> orthogonal vertical wind (nproma,nlevp1,nblks_c) [m/s]
53 vn(:,:,:), & !! orthogonal normal wind (nproma,nlev,nblks_e) [m/s]
54 rho(:,:,:), & !! density (nproma,nlev,nblks_c) [kg/m^3]
55 exner(:,:,:), & !! Exner pressure (nproma,nlev,nblks_c) [-]
56 theta_v(:,:,:), & !! virtual potential temperature (nproma,nlev,nblks_c) [K]
57 tracer(:,:,:), & !! tracer concentration (nproma,nlev,nblks_c,ntracer) [kg/kg]
58 tke (:,:,:) & !! turbulent kinetic energy [m^2/s^2]
59 | => NULL() !! (defined on half levels) with 2 time levels
60 TYPE(t_ptr_2d3d),ALLOCATABLE :: tracer_ptr(:) !< pointer array: one pointer for each tracer
61 TYPE(t_ptr_tracer),ALLOCATABLE :: conv_tracer(:,)
62 TYPE(t_ptr_tracer),ALLOCATABLE :: turb_tracer(:,)
63 END TYPE t_nh_prog

```

Offsets and placeholders

```

15  struct t_dist_dir {
16      int __f2dace_SA_owner_d_0_s_3713;
17      int __f2dace_SOA_owner_d_0_s_3713;
18      int comm;
19      int comm_rank;
20      int comm_size;
21      int global_size;
22      int local_start_index;
23  };

```

```

3117 int __f2dace_SA_ddxt_z_full_c_d_2_s_5247_p_metrics_43;
3118 int __f2dace_SOA_ddxt_z_full_v_d_0_s_5248_p_metrics_43;
3119 int __f2dace_SOA_ddxt_z_full_v_d_1_s_5249_p_metrics_43;
3120 int __f2dace_SOA_ddxt_z_full_v_d_2_s_5250_p_metrics_43;
3121 int __f2dace_SA_ddxt_z_full_v_d_0_s_5248_p_metrics_43;

```

```

9  DACE_EXPORTED velocity_tendencies_state_t *__dace_init_velocity_tendencies(t_nh_diag* p_diag, t_int_state* p_int,
10  {
11      int __result = 0;
12      velocity_tendencies_state_t *__state = new velocity_tendencies_state_t;
13
14
15      tmp_struct_symbol_0=p_patch->nlev;
16      tmp_struct_symbol_1=p_patch->nlevp1;
17      tmp_struct_symbol_2=p_patch->nlev;
18      tmp_struct_symbol_3=p_patch->nblks_c;
19      tmp_struct_symbol_4=p_patch->nlev;
20      tmp_struct_symbol_5=p_patch->nblks_e;
21      tmp_struct_symbol_6=p_patch->nlevp1;
22      tmp_struct_symbol_7=p_patch->nblks_v;
23      tmp_struct_symbol_8=p_patch->nlev;
24      tmp_struct_symbol_9=p_patch->nblks_v;
25      tmp_struct_symbol_10=p_patch->nlev;
26      tmp_struct_symbol_11=p_patch->nblks_c;
27      tmp_struct_symbol_12=p_patch->nblks_c;
28      tmp_struct_symbol_13=p_patch->nblks_c;
29      tmp_struct_symbol_14=p_patch->nlev;
30      tmp_struct_symbol_15=p_patch->nlev;
31      tmp_struct_symbol_16=p_patch->nlevp1;
32      __f2dace_SOA_neighbor_idx_d_0_s_3126_cells_p_patch_4 = p_patch->cells->__f2dace_SOA_neighbor_idx_d_0_s_3126;
33      __f2dace_SOA_neighbor_idx_d_1_s_3127_cells_p_patch_4 = p_patch->cells->__f2dace_SOA_neighbor_idx_d_1_s_3127;

```