



# The GungHo Dynamical Core

The Met Office  
STFC Daresbury Laboratory \*  
University of Manchester

Rupert Ford

\* STFC's effort was funded by the Hartree Centre

# GungHo

- Globally Uniform, Next Generation, Highly Optimised
- “To research, design and develop a new dynamical core suitable for operational, global and regional, weather and climate simulation on massively parallel computers of the size envisaged over the coming 20 years.”
- Remove the pole problem (replace lat-lon grid)
- Aimed at massively parallel computers –  $10^6$  way parallel → petaflop
- Split into two phases:
  - 2 years “research” (2011-13)
  - 3 years “development” (2013-2016)
- Met Office, STFC, Universities of: Bath, Exeter, Imperial, Leeds, Manchester, Reading, Warwick

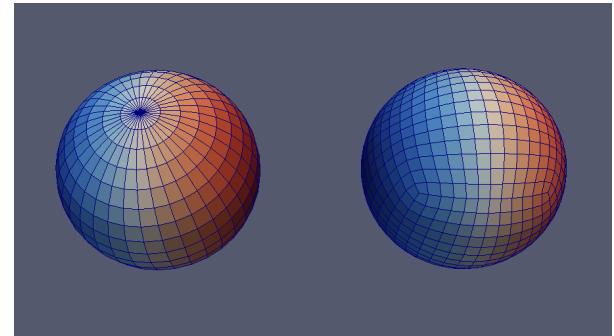
工合



Science & Technology  
Facilities Council

# GungHo

- (Most likely) Cubed Sphere
- Multi-grid
- Finite element approach (Order)
- Separate Science from HPC optimisation (PSyKAI)
  - Performance portability
  - Single science source
- 'Dynamo' prototype implementation
- PSyclone generation of PSy layer

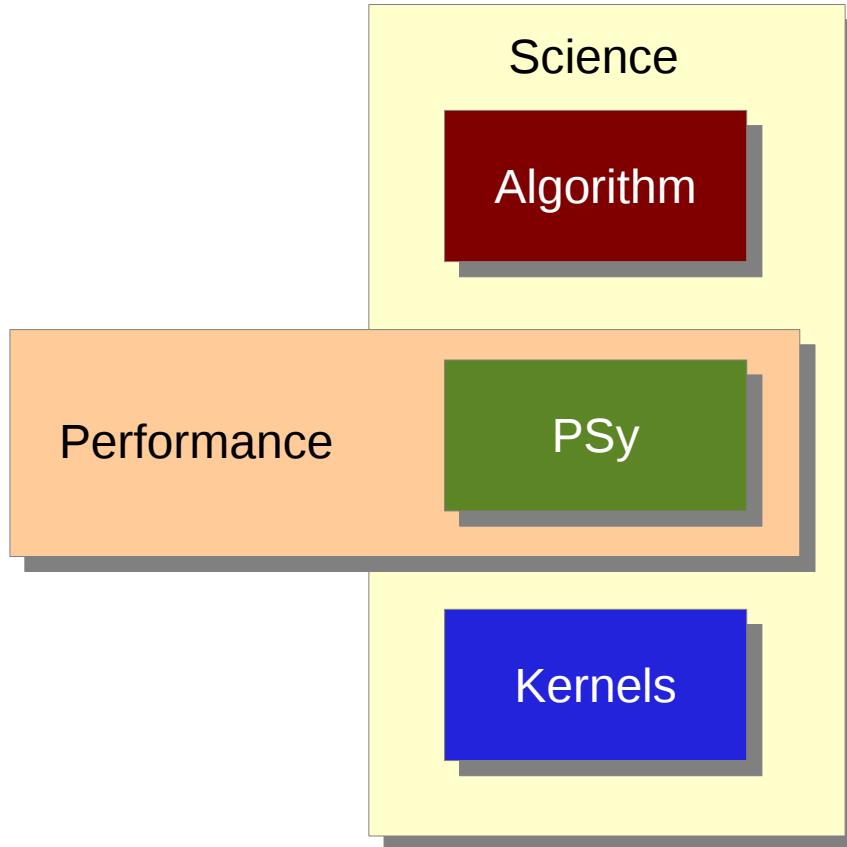


工合



Science & Technology  
Facilities Council

# PSyKAI



Science & Technology  
Facilities Council

# Dynamo algorithm example ...

- From `lhs_alg_mod.x90`
- Multiple kernels within an `invoke()`
- 'pointwise' not yet supported in PSyclone

```
! Use r_rho as a zero field so that exner' = 0 in pert_pressure_gradient_kernel
call invoke_set_field_scalar( 0.0_r_def, r_rho )
exner => state(4)
call invoke( exner_gradient_kernel_type( lhs_tmp(1), exner, theta_ref, qr ), &
            pert_pressure_gradient_kernel_type( lhs_tmp(1), r_rho, rho_ref, &
                                                theta, theta_ref, qr ), &
            matrix_vector_kernel_type( lhs(1), u, mm_w2 ) )
```



# Dynamo kernel example ...

- From `matrix_vector_kernel_mod.F90`

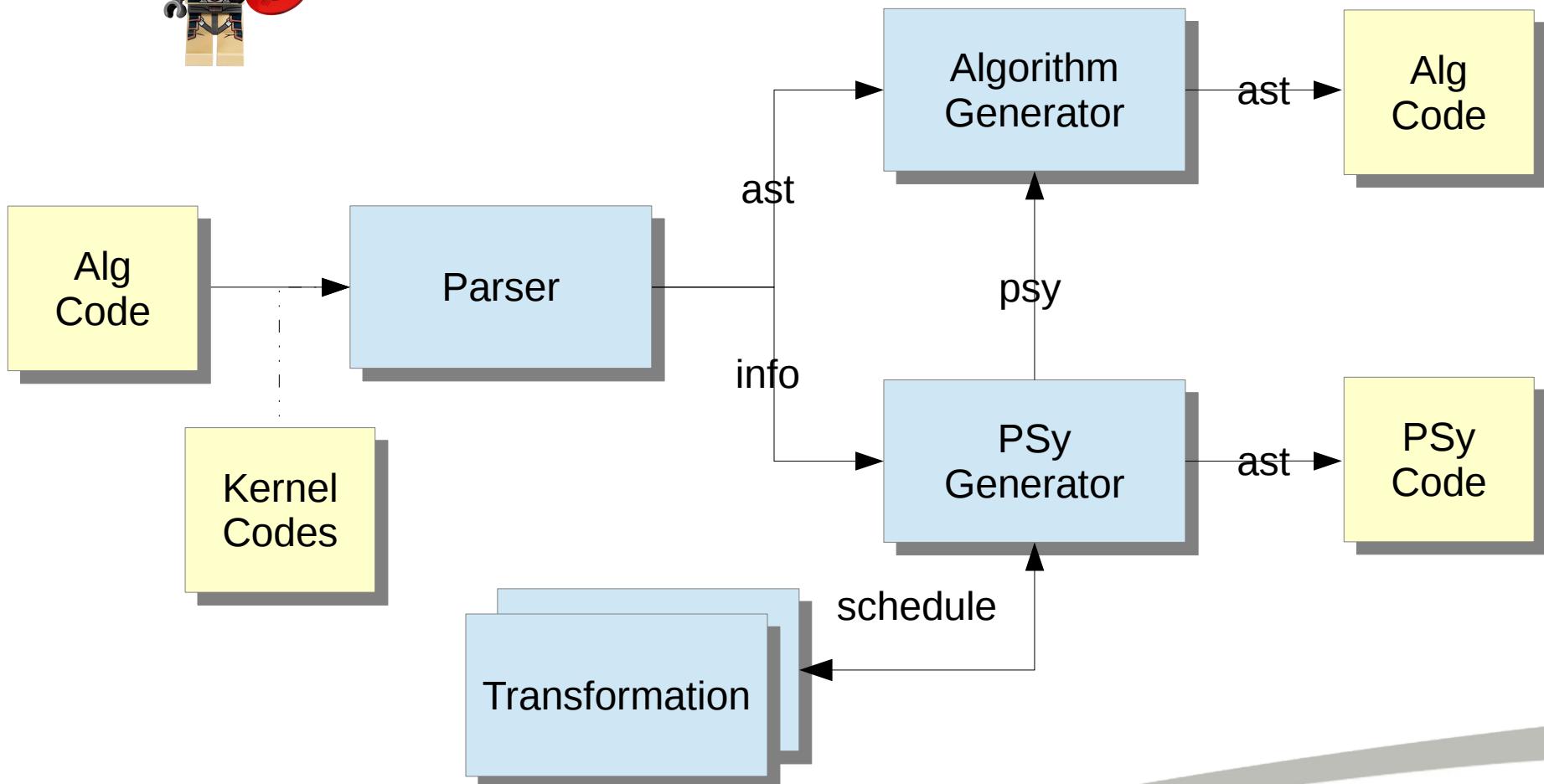
```
type, public, extends(kernel_type) :: matrix_vector_kernel_type
  private
    type(arg_type) :: meta_args(3) = (/&
      arg_type(GH_FIELD, GH_INC, ANY_SPACE_1), &
      arg_type(GH_FIELD, GH_READ, ANY_SPACE_2), &
      arg_type(GH_OPERATOR, GH_READ, ANY_SPACE_1, ANY_SPACE_2) &
    /)
    integer :: iterates_over = CELLS
contains
  procedure, nopass ::matrix_vector_code
end type
```

```
do k = 0, nlayers-1
  do df = 1, ndf2
    x_e(df) = x(map2(df)+k)
  end do
  ik = (cell-1)*nlayers + k + 1
  lhs_e = matmul(matrix(:,:,ik),x_e)
  do df = 1,ndf1
    lhs(map1(df)+k) = lhs(map1(df)+k) + lhs_e(df)
  end do
end do
```





# PSyclone



# PSyclone Scripts example ...

```
> python ../../psyclone/src/generator.py lhs_alg_mod.x90 \
-d ../../dynamo/kernel -oalg alg.f90 -opsy psy.f90 \
-nodm -s ./global.py
```

```
Schedule[invoke='invoke_2' dm=False]
  Loop[type='',field_space='w2',it_space='cells']
    KernCall exner_gradient_code(lhs_tmp,exner,theta_ref) [module_inline=False]
  Loop[type='',field_space='w2',it_space='cells']
    KernCall pert_pressure_gradient_code(lhs_tmp,r_rho,rho_ref,theta,theta_ref) [module_inline=False]
  Loop[type='',field_space='any_space_1',it_space='cells']
    KernCall matrix_vector_code(lhs,u,mm_w2) [module_inline=False]
```

Vanilla, no dm



Science & Technology  
Facilities Council

# PSyclone Scripts example ...

```
> python ../../psyclone/src/generator.py lhs_alg_mod.x90 \
-d ../../dynamo/kernel -oalg alg.f90 -opsy psy.f90 \
-nodm -s ./global.py
```

```
Schedule[invoke='invoke_2' dm=False]
  Loop[type='colours',field_space='w2',it_space='cells']
    Loop[type='colour',field_space='w2',it_space='cells']
      KernCall exner_gradient_code(lhs_tmp,exner,theta_ref) [module_inline=False]
  Loop[type='colours',field_space='w2',it_space='cells']
    Loop[type='colour',field_space='w2',it_space='cells']
      KernCall pert_pressure_gradient_code(lhs_tmp,r_rho,rho_ref,theta,theta_ref) [module_inline=False]
  Loop[type='colours',field_space='any_space_1',it_space='cells']
    Loop[type='colour',field_space='any_space_1',it_space='cells']
      KernCall matrix_vector_code(lhs,u,mm_w2) [module_inline=False]
```

Colours, no dm



Science & Technology  
Facilities Council

# PSyclone Scripts example ...

```
> python ../../psyclone/src/generator.py lhs_alg_mod.x90 \
-d ../../dynamo/kernel -oalg alg.f90 -opsy psy.f90 \
-nodm -s ./global.py
```

```
Schedule[invoke='invoke_2' dm=False]
  Loop[type='colours',field_space='w2',it_space='cells']
    Directive[OMP parallel do]
      Loop[type='colour',field_space='w2',it_space='cells']
        KernCall exner_gradient_code(lhs_tmp,exner,theta_ref) [module_inline=False]
  Loop[type='colours',field_space='w2',it_space='cells']
    Directive[OMP parallel do]
      Loop[type='colour',field_space='w2',it_space='cells']
        KernCall pert_pressure_gradient_code(lhs_tmp,r_rho,rho_ref,theta,theta_ref) [module_inline=False]
  Loop[type='colours',field_space='any_space_1',it_space='cells']
    Directive[OMP parallel do]
      Loop[type='colour',field_space='any_space_1',it_space='cells']
        KernCall matrix_vector_code(lhs,u,mm_w2) [module_inline=False]
```

Colours + OpenMP, no dm



Science & Technology  
Facilities Council

# PSyclone Scripts example ...

```
> python ../../psyclone/src/generator.py lhs_alg_mod.x90 \
-d ../../dynamo/kernel -oalg alg.f90 -opsy psy.f90 \
-s ./global.py
```

```
Schedule[invoke='invoke_2' dm=True]
    HaloExchange[field='lhs_tmp', type='region', depth=1, check_dirty=True]
    HaloExchange[field='exner', type='region', depth=1, check_dirty=True]
    HaloExchange[field='theta_ref', type='region', depth=1, check_dirty=True]
    Loop[type='', field_space='w2', it_space='cells']
        KernCall exner_gradient_code(lhs_tmp,exner,theta_ref) [module_inline=False]
    HaloExchange[field='lhs_tmp', type='region', depth=1, check_dirty=True]
    HaloExchange[field='r_rho', type='region', depth=1, check_dirty=True]
    HaloExchange[field='rho_ref', type='region', depth=1, check_dirty=True]
    HaloExchange[field='theta', type='region', depth=1, check_dirty=True]
    HaloExchange[field='theta_ref', type='region', depth=1, check_dirty=True]
    Loop[type='', field_space='w2', it_space='cells']
        KernCall pert_pressure_gradient_code(lhs_tmp,r_rho,rho_ref,theta,theta_ref) [module_inline=False]
    HaloExchange[field='lhs', type='region', depth=1, check_dirty=True]
    HaloExchange[field='u', type='region', depth=1, check_dirty=True]
    Loop[type='', field_space='any_space_1', it_space='cells']
        KernCall matrix_vector_code(lhs,u,mm_w2) [module_inline=False]
```

Vanilla, dm



Science & Technology  
Facilities Council

# PSyclone Scripts example ...

```
> python ../../psyclone/src/generator.py lhs_alg_mod.x90 \
-d ../../dynamo/kernel -oalg alg.f90 -opsy psy.f90 \
-s ./global.py
```

```
Schedule[invoke='invoke_2' dm=True]
    HaloExchange[field='lhs_tmp', type='region', depth=1, check_dirty=True]
    HaloExchange[field='exner', type='region', depth=1, check_dirty=True]
    HaloExchange[field='theta_ref', type='region', depth=1, check_dirty=True]
    Loop[type='colours',field_space='w2',it_space='cells']
        Loop[type='colour',field_space='w2',it_space='cells']
            KernCall exner_gradient_code(lhs_tmp,exner,theta_ref) [module_inline=False]
    HaloExchange[field='lhs_tmp', type='region', depth=1, check_dirty=True]
    HaloExchange[field='r_rho', type='region', depth=1, check_dirty=True]
    HaloExchange[field='rho_ref', type='region', depth=1, check_dirty=True]
    HaloExchange[field='theta', type='region', depth=1, check_dirty=True]
    HaloExchange[field='theta_ref', type='region', depth=1, check_dirty=True]
    Loop[type='colours',field_space='w2',it_space='cells']
        Loop[type='colour',field_space='w2',it_space='cells']
            KernCall pert_pressure_gradient_code(lhs_tmp,r_rho,rho_ref,theta,theta_ref) [module_inline=False]
    HaloExchange[field='lhs', type='region', depth=1, check_dirty=True]
    HaloExchange[field='u', type='region', depth=1, check_dirty=True]
    Loop[type='colours',field_space='any_space_1',it_space='cells']
        Loop[type='colour',field_space='any_space_1',it_space='cells']
            KernCall matrix_vector_code(lhs,u,mm_w2) [module_inline=False]
```

Colours, dm



Science & Technology  
Facilities Council

# PSyclone Scripts example ...

```
> python ../../psyclone/src/generator.py lhs_alg_mod.x90 \
-d ../../dynamo/kernel -oalg alg.f90 -opsy psy.f90 \
-s ./global.py
```

```
schedule[invoke='invoke_2' dm=True]
HaloExchange[field='lhs_tmp', type='region', depth=1, check_dirty=True]
HaloExchange[field='exner', type='region', depth=1, check_dirty=True]
HaloExchange[field='theta_ref', type='region', depth=1, check_dirty=True]
Loop[type='colours',field_space='w2',it_space='cells']
    Directive[OMP parallel do]
        Loop[type='colour',field_space='w2',it_space='cells']
            KernCall exner_gradient_code(lhs_tmp,exner,theta_ref) [module_inline=False]
HaloExchange[field='lhs_tmp', type='region', depth=1, check_dirty=True]
HaloExchange[field='r_rho', type='region', depth=1, check_dirty=True]
HaloExchange[field='rho_ref', type='region', depth=1, check_dirty=True]
HaloExchange[field='theta', type='region', depth=1, check_dirty=True]
HaloExchange[field='theta_ref', type='region', depth=1, check_dirty=True]
Loop[type='colours',field_space='w2',it_space='cells']
    Directive[OMP parallel do]
        Loop[type='colour',field_space='w2',it_space='cells']
            KernCall pert_pressure_gradient_code(lhs_tmp,r_rho,rho_ref,theta,theta_ref) [module_inline=False]
HaloExchange[field='lhs', type='region', depth=1, check_dirty=True]
HaloExchange[field='u', type='region', depth=1, check_dirty=True]
Loop[type='colours',field_space='any_space_1',it_space='cells']
    Directive[OMP parallel do]
        Loop[type='colour',field_space='any_space_1',it_space='cells']
            KernCall matrix_vector_code(lhs,u,mm_w2) [module_inline=False]
```

OpenMP + dm



Science & Technology  
Facilities Council

# PSyclone Scripts example ...

```
.  
IF (theta_ref_proxy%is_dirty(depth=1)) THEN  
    CALL theta_ref_proxy%halo_exchange(depth=1)  
END IF  
!  
CALL lhs_tmp_proxy%vspace%get_colours(ncolour, ncp_colour, cmap)  
!  
DO colour=1,ncolour  
    !$omp parallel do default(shared), private(cell,map_w2,map_w3,map_w0), schedule(static)  
    DO cell=1,ncp_colour(colour)  
        !  
        map_w2 => lhs_tmp_proxy%vspace%get_cell_dofmap(cmap(colour, cell))  
        map_w3 => r_rho_proxy%vspace%get_cell_dofmap(cmap(colour, cell))  
        map_w0 => theta_proxy%vspace%get_cell_dofmap(cmap(colour, cell))  
        !  
        CALL pert_pressure_gradient_code(nlayers, lhs_tmp_proxy%data, r_rho_proxy%data, rho_ref_proxy%data, theta_proxy%data,  
        theta_ref_proxy%data, ndf_w2, undf_w2, map_w2, basis_w2, diff_basis_w2, ndf_w3, undf_w3, map_w3, basis_w3, ndf_w0, undf_w0, map_w0,  
        basis_w0, diff_basis_w0, nqp_h, nqp_v, wh, wv)  
    END DO  
    !$omp end parallel do  
END DO  
!  
! Set halos dirty for fields modified in the above loop  
!  
CALL lhs_tmp_proxy%set_dirty()  
!
```

# GungHo, LFRic, PSyKAI, Dynamo and PSyclone Summary

- GungHo project
  - consortium to develop a new dynamical core
- LFRic project
  - Met Office project to deliver UM Atmosphere replacement using new dynamical core
- PSyKAI
  - separation of concerns suggested by GungHo project and implemented in Dynamo and PSyclone
- Dynamo software
  - prototype GungHo dynamical core
  - tests the science and software engineering
- PSyclone software
  - Code generation system used by Dynamo

工合



Science & Technology  
Facilities Council

# GungHo, LFRic, PSyKAI, Dynamo and PSyclone Summary

- GungHo project
  - consortium to develop a new dynamical core
- LFRic project
  - Met Office project to deliver UM Atmosphere replacement using new dynamical core
- PSyKAI
  - separation of concerns suggested by GungHo project and implemented in Dynamo and PSyclone
- Dynamo
  - GungHo
  - LFRic
  - **Dynamical Core**
  - **Infrastructure**
  - tests the science and software engineering
- PSyclone software
  - Code generation system used by Dynamo



# Dynamo Status

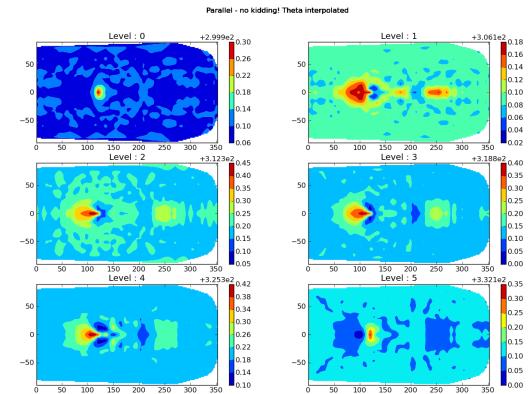
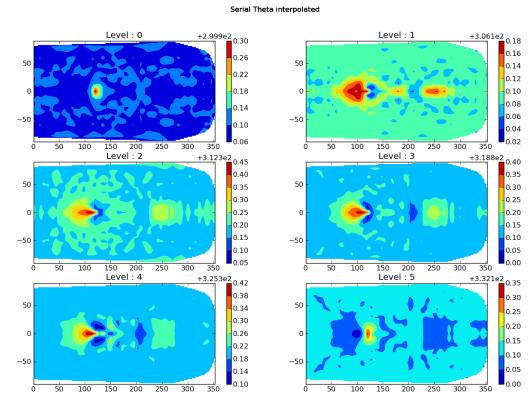
- Version 1.0 released 31<sup>st</sup> March : Whoop!
- Fully functional infrastructure
- Initial science : solver
- Uses PSyclone for parallelism
  - Sequential, OpenMP parallel, MPI parallel or mixed mode



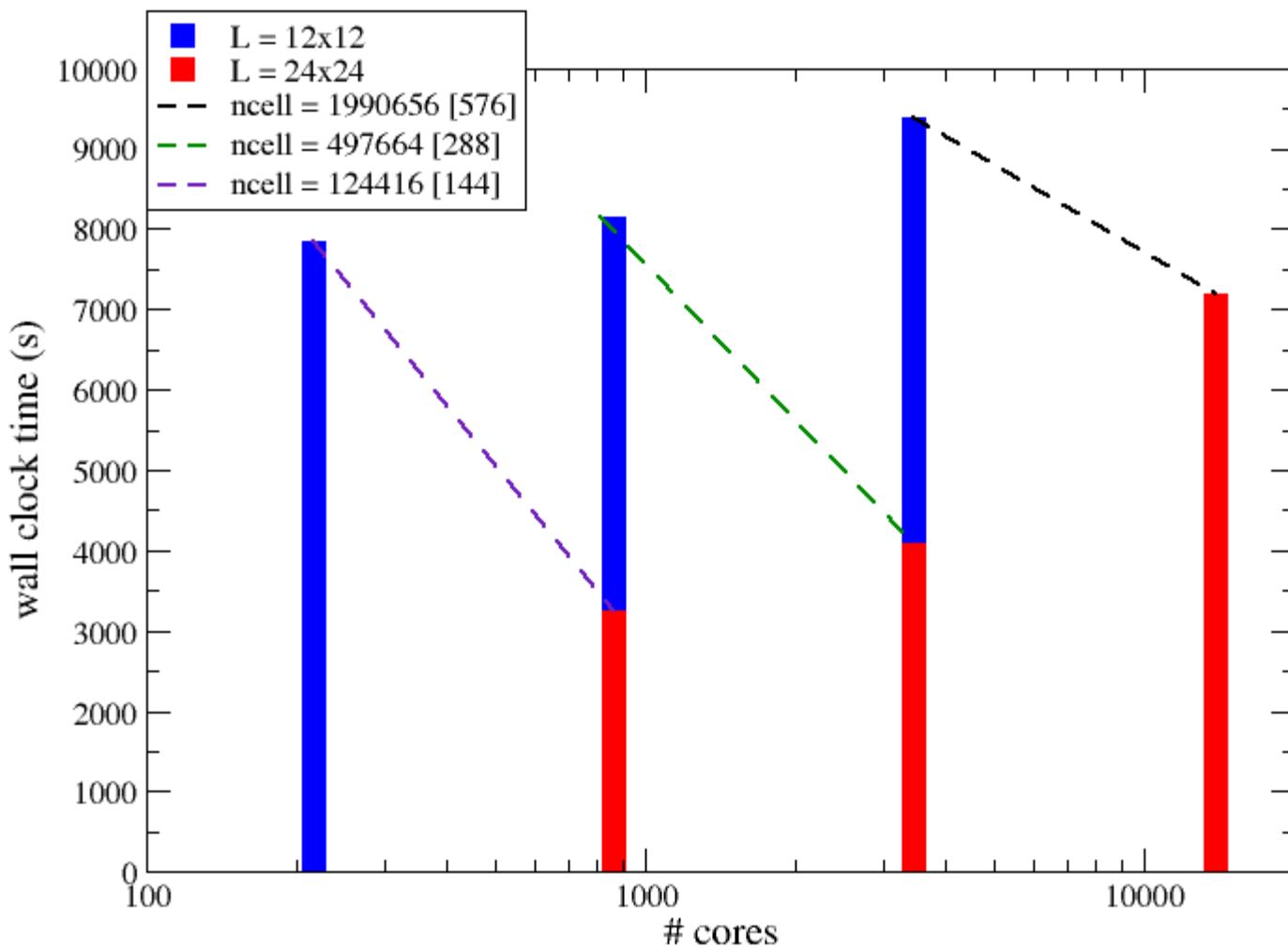
# Going MPI Parallel

- Dynamo infrastructure and Psyclone support
- PSyclone command line change

```
> python generator.py  
-oalg alg.f90 -opsy  
psy.f90 file.f90 -nodm
```

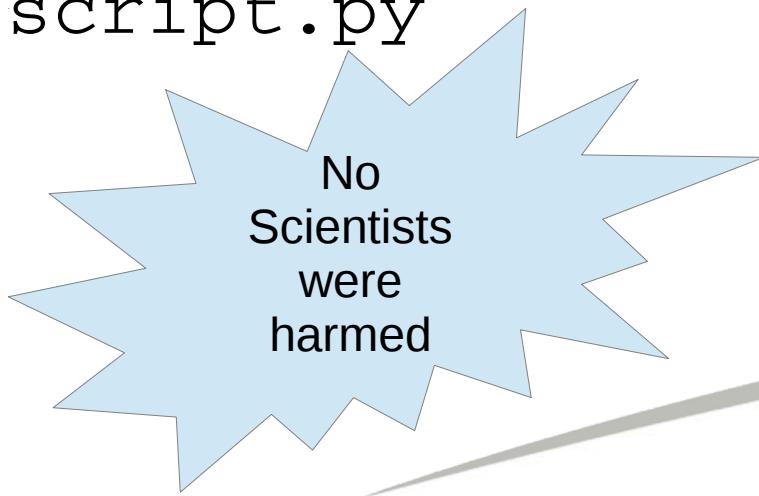


# Dynamo Initial MPI results



# Going OpenMP Parallel

- Dynamo infrastructure and Psyclone support
- PSyclone script
  - > python generator.py -oalg alg.f90 -opsy psy.f90 file.f90 -s script.py



```
from transformations import DynamoOp3ColourTrans, Dyna

def trans(psy):
    ctrans = DynamoOp3ColourTrans()
    otrans = DynamoOMPParallelLoopTrans()

    # Loop over all of the Invokes in the PSy object
    for invoke in psy.invokes.invoke_list:

        schedule = invoke.schedule

        # Colour loops unless they are on W3
        for loop in schedule.loops():
            if loop.field_space != "w3":
                schedule, _ = ctrans.apply(loop)

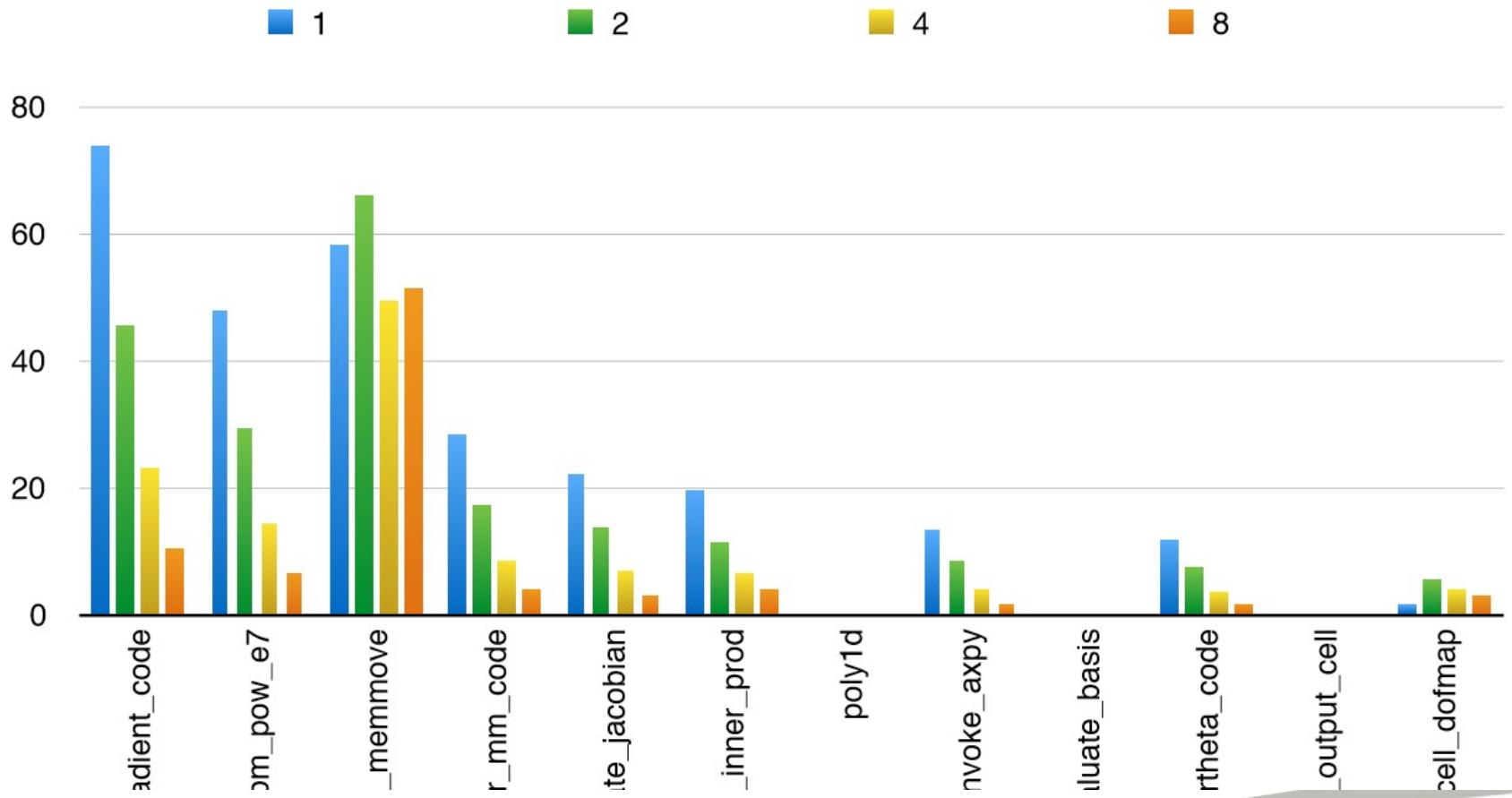
        # Add OpenMP to loops unless they are over co
        for loop in schedule.loops():
            if loop.loop_type != "colours":
                schedule, _ = otrans.apply(loop)

    return psy
```



Science & Technology  
Facilities Council

# Dynamo Initial OpenMP results



# Dynamo Future

- Dynamo => GungHo science + LFRic infrastructure
- Continued development under LFRic
- Continued collaboration with STFC
- Open source, new repository
- Regular releases (~3 monthly)
  - Full dynamical core + some physics
  - Optimised code using PSyclone
  - Determine accuracy, scalability and efficiency



# Wider collaboration?

- PSyKAI
- PSyclone FE and FD/FV
  - NEMO-lite 2D implementation
- Commonality with STELLA



Science & Technology  
Facilities Council