



AUTOMATING UKCA GPU PORTING EFFORTS USING PSYCLONE

Joe Wallwork¹ and Joe Abram¹

Thanks to: Mohit Dalvi¹, Alan Hewitt¹, Luke Abraham², Wayne Gaudin³, Seth Camp³,
Andy Porter⁴, Rupert Ford⁴, Sergi Siso⁴

¹Met Office

²University of Cambridge

³Nvidia

⁴STFC Hartree Centre

Met Office, 7th December 2023



UK Research
and Innovation



UK Atomic
Energy
Authority

Contents

1. PSyclone in transformation mode



2.

Replicating a manual GPU port

3.

Future-proofing against code changes

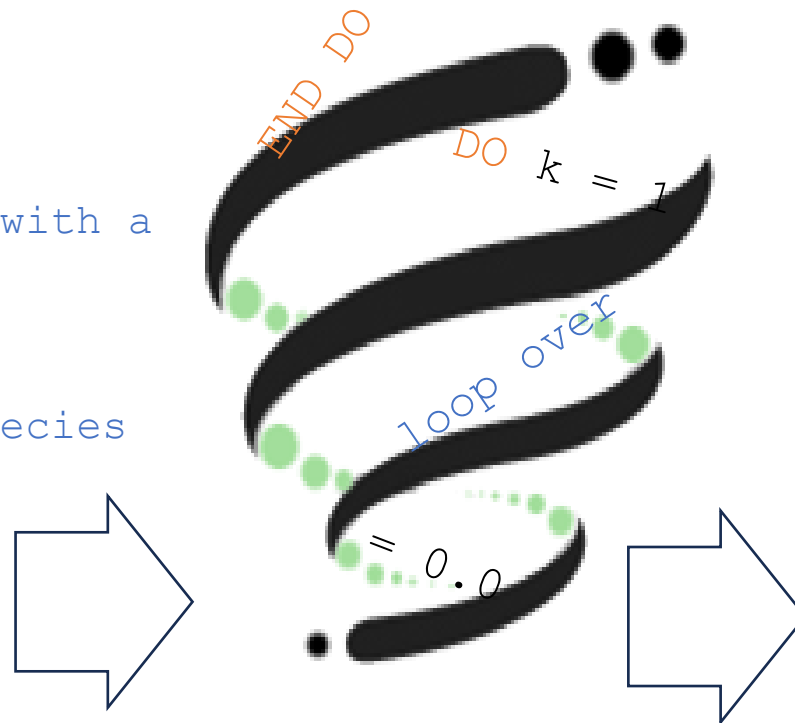
4.

PSyclone-driven debugging

PSyclone in code transformation mode

Without a transformation script

```
! Some loop over spatial points with a  
! convergence mask check  
DO k = 1, n_points  
  IF (mask(k)) THEN  
    ! Some loop over chemical species  
    DO j = 1, n_species  
      arr(j,k) = 0.0  
    END DO  
  END IF  
END DO
```

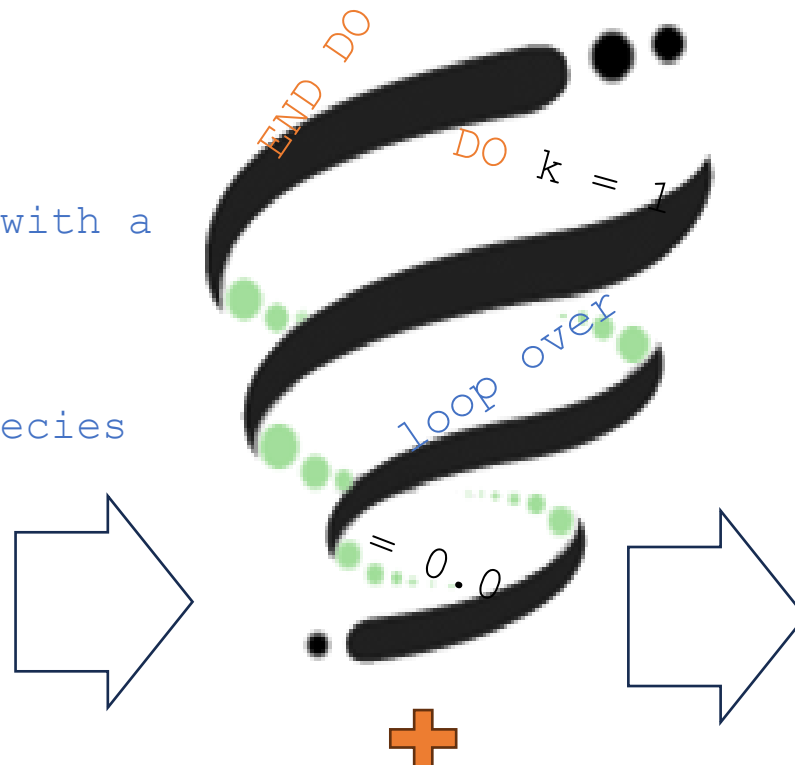


```
do k = 1, n_points, 1  
  if (mask(k)) then  
    do j = 1, n_species, 1  
      arr(j,k) = 0.0  
    end do  
  end if  
end do
```

PSyclone in code transformation mode

With a transformation script

```
! Some loop over spatial points with a  
! convergence mask check  
DO k = 1, n_points  
  IF (mask(k)) THEN  
    ! Some loop over chemical species  
    DO j = 1, n_species  
      arr(j,k) = 0.0  
    END DO  
  END IF  
END DO
```



```
!$acc kernels  
!$acc loop gang vector independent  
do k = 1, n_points, 1  
  if (mask(k)) then  
    !$acc loop seq  
    do j = 1, n_species, 1  
      arr(j,k) = 0.0  
    end do  
  end if  
end do  
!$acc end kernels
```

Replicating a manual GPU port

Manual Port

vs.

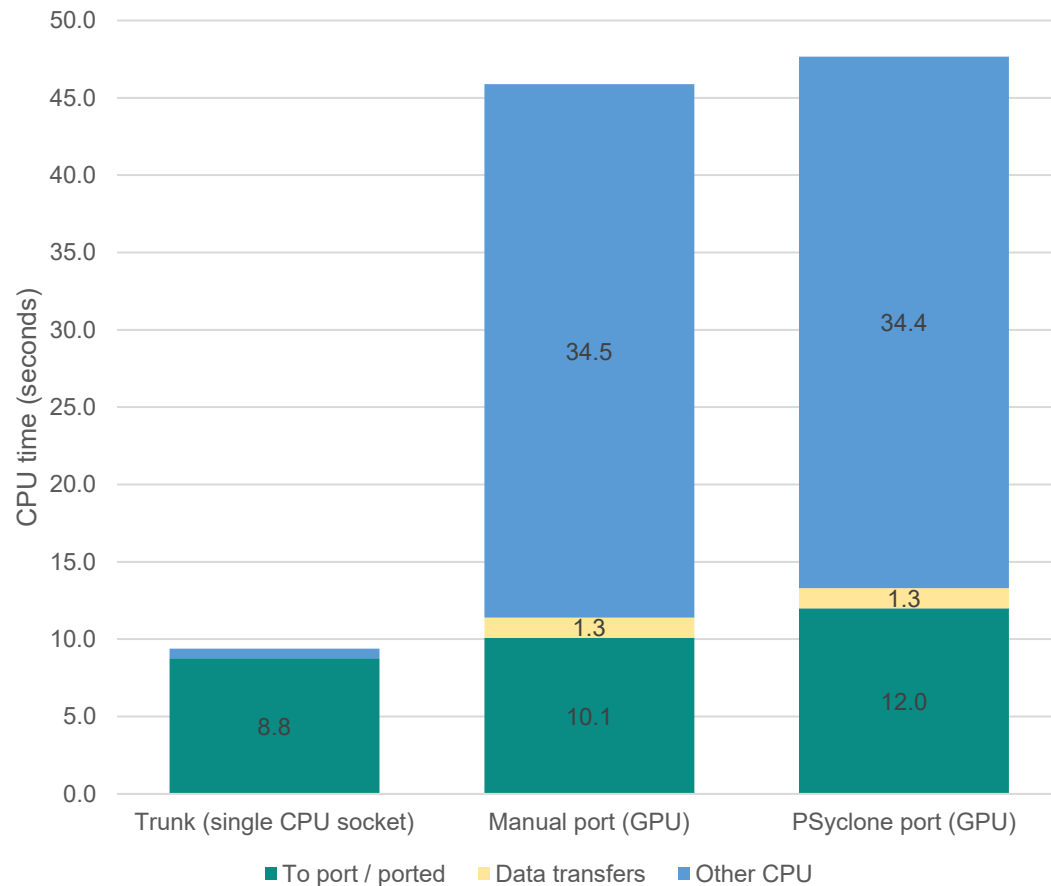
PSyclone Port

```
! Some loop over spatial points with a
! convergence mask check
!$acc kernels
!$acc loop gang vector independent
DO k = 1, n_points
  IF (mask(k)) THEN
    !$acc loop seq
    ! Some loop over chemical species
    DO j = 1, n_species
      arr(j,k) = 0.0
    END DO
  END IF
END DO
!$acc end kernels
```

```
!$acc kernels
!$acc loop gang vector independent
do k = 1, n_points, 1
  if (mask(k)) then
    !$acc loop seq
    do j = 1, n_species, 1
      arr(j,k) = 0.0
    end do
  end if
end do
!$acc end kernels
```

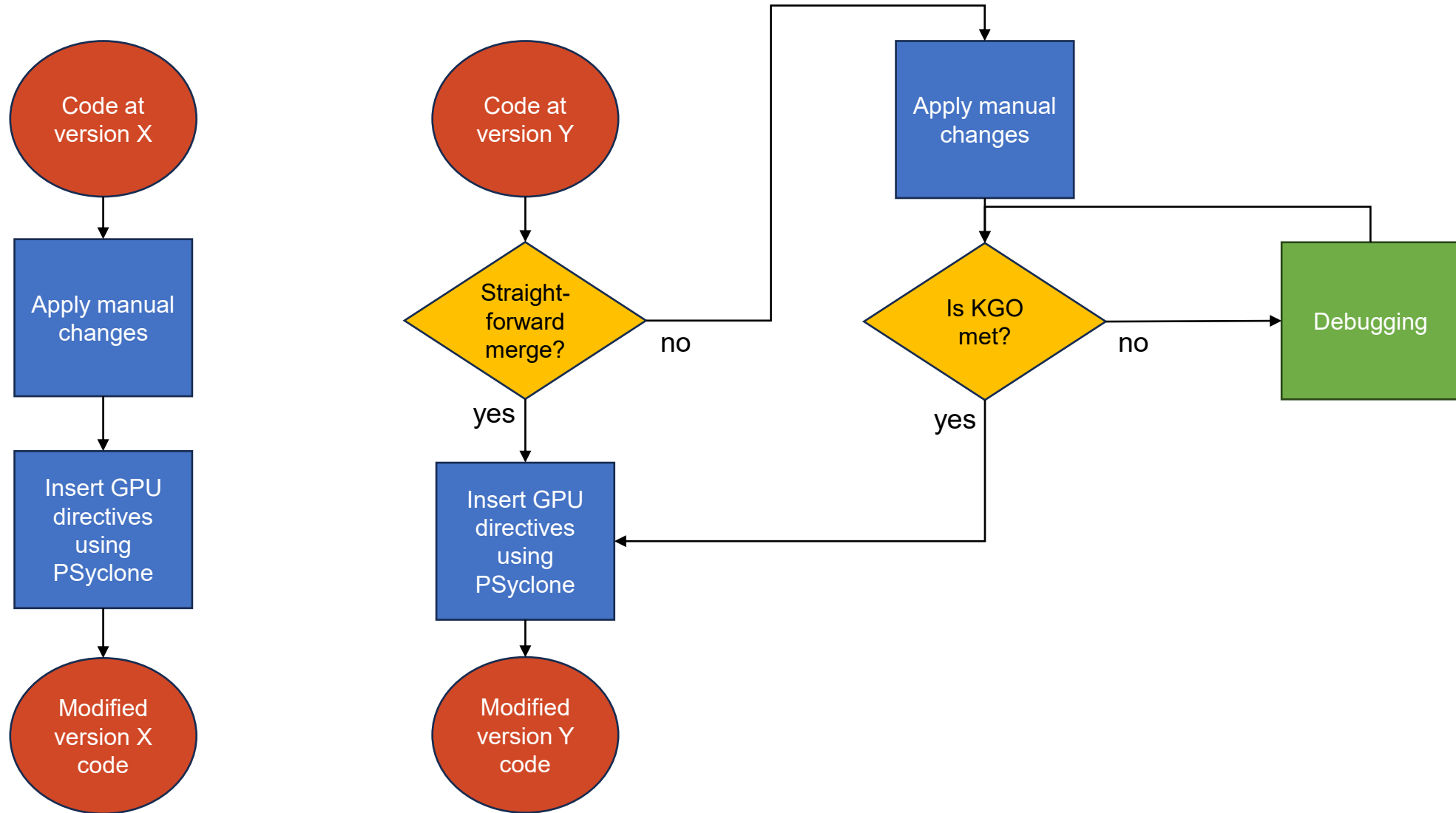
Computational performance

'N48' UKCA standard job – 262,656 spatial points and 3 chemical timesteps



Supported	Currently unsupported
independent	reduction
seq	async / wait
gang	
vector	
collapse	

Future-proofing against code changes



PSyclone-driven debugging

‘Canonicalisation’

```
DO k = 1, n_points
  IF (mask(k)) THEN
    ! Where the mask applies, initialise
    ! species to zero
    arr(k,:) = 0.0
  END IF
  ! Further comments
END DO
```



```
do k = 1, n_points, 1
  if (mask(k)) then
    do idx = LBOUND(arr,1), UBOUND(arr,1), 1
      arr(k,idx) = 0.0
    end do
  end if
end do
```

```
! Some loop over spatial points with a
! convergence mask check
WHERE mask
  ! Some loop over chemical species
  DO j = 1, n_species
    arr(j,k) = 0.0
  END DO
END WHERE
```



```
do idx1 = LBOUND(mask,1), UBOUND(mask,1), 1
  if (mask(idx1)) then
    do j = 1, n_species
      arr(j,idx1) = 0.0
    end do
  end if
end do
```

Conclusion

- **PSyclone has saved us *a lot* of time.**
- **We weren't expecting to get so much out of it, using it in so many ways.**
- **Understanding and writing a transformation script for PSyclone takes time but running it is very quick and easy once set up.**

Possible future uses

- **Portability to OpenMP.**

Thank You

Questions ?