

Hewlett Packard
Enterprise

Improve WRF IO performance with Parallel IO

Wei Huang

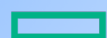
Hewlett Packard Enterprise

A Quick Case Review

- *2017.10.25 Hurricane Irma*
- *Grid: x-3100, y-1600, z-35*
- *Resolution: 5.4 km*
- *Forecast: 6 hours*
- *Output Frequency: 30 minutes*
- *Time step: 24 seconds*

Computer Review

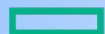
- *3.0 GHz Xeon Gold 6154*
- *Total of 288 Nodes*
- *18 cores per socket*
- *2 sockets per node*
- *Omni-path fabric*



Performance

➤ *using NetCDF 3 as IO option*

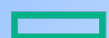
# of Nodes/Cores	Wall Clock (seconds)
9 / 324	3020
18 / 648	2223
36 / 1296	1384
72 / 2592	1171
128 / 4608	2089



Performance (cont.)

► *using NetCDF 3*

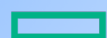
# of Nodes/Cores	Wall Clock	Computing Per Step
9 / 324	3020	1.576815
18 / 648	2223	0.805103
36 / 1296	1384	0.428261
72 / 2592	1171	0.218161
128 / 4608	2089	0.127812



Performance (cont.)

► *using NetCDF 3*

# of Nodes/Cores	Wall Clock	Computing Per Step	Output Time
9 / 324	3020	1.576815	1323.22
18 / 648	2223	0.805103	1316.74
36 / 1296	1384	0.428261	886.33
72 / 2592	1171	0.218161	880.63
128 / 4608	2089	0.127812	1870.94



Performance (cont.)

- ▶ *using NetCDF 3*
- ▶ *Parallel netcdf (pnetcdf), on lustre w/o stripe*

# of Nodes/Cores	Wall Clock (seconds)	
	NetCDF 3	Pnetcdf (no stripe)
9 / 324	3020	2768
18 / 648	2223	2441
36 / 1296	1384	741
72 / 2592	1171	873
128 / 4608	2089	3817

Performance (cont.)

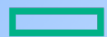
- ▶ *using NetCDF 3*
- ▶ *Parallel netcdf (pnetcdf), on lustre with and without stripe*

# of Nodes/Cores	Wall Clock (seconds)		
	NetCDF 3	pnetcdf (no stripe)	pnetcdf (with stripe 12)
9 / 324	3020	2768	1765
18 / 648	2223	2441	985
36 / 1296	1384	741	681
72 / 2592	1171	873	790
128 / 4608	2089	3817	1489

Performance (cont.)

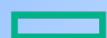
- *using NetCDF 3*
- *Parallel netcdf (pnetcdf), on lustre with and without stripe*
- *Integration time (with pnetcdf stripe set)*

# of Nodes/Cores	Wall Clock (seconds)			Integration time
	NetCDF 3	pnetcdf (no stripe)	pnetcdf (with stripe 12)	With pnetcdf stripe set
9 / 324	3020	2768	1765	1597.55
18 / 648	2223	2441	985	804.42
36 / 1296	1384	741	681	374.43
72 / 2592	1171	873	790	130.12
128 / 4608	2089	3817	1489	57.88



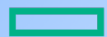
WRF I/O Options

- Sequential
 - NetCDF 3, NetCDF 4
 - Grib
- Paralell
 - HDF5
 - pnetcdf

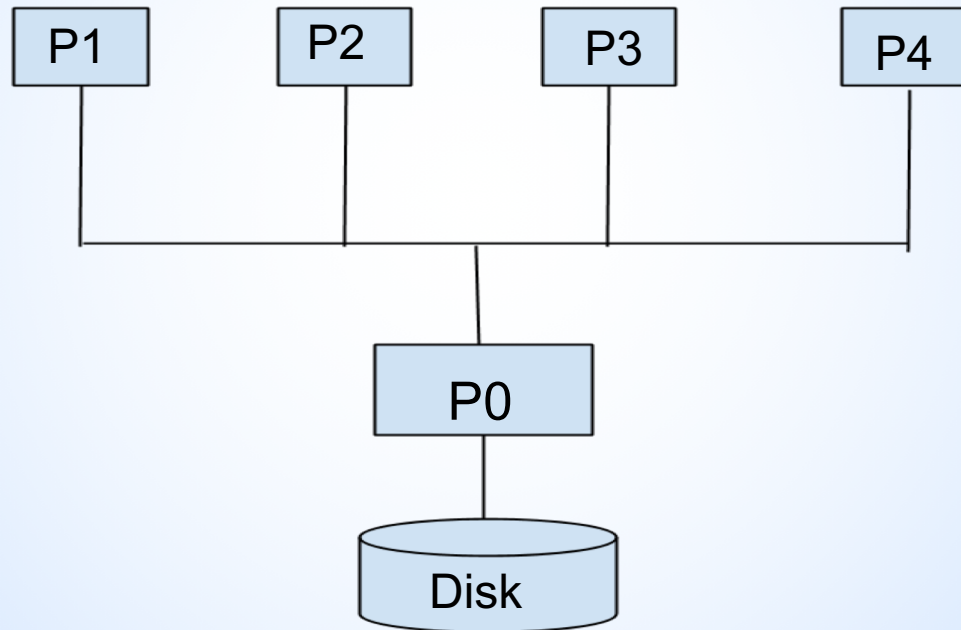


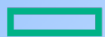
WRF I/O Types

- Collective (N to one)
 - NetCDF
 - Grib
 - HDF5
- Independent (N to N)
 - Pnetcdf (one file)
 - Splitter (N files)
- Hybrid
 - Quilt
 - **PIO**

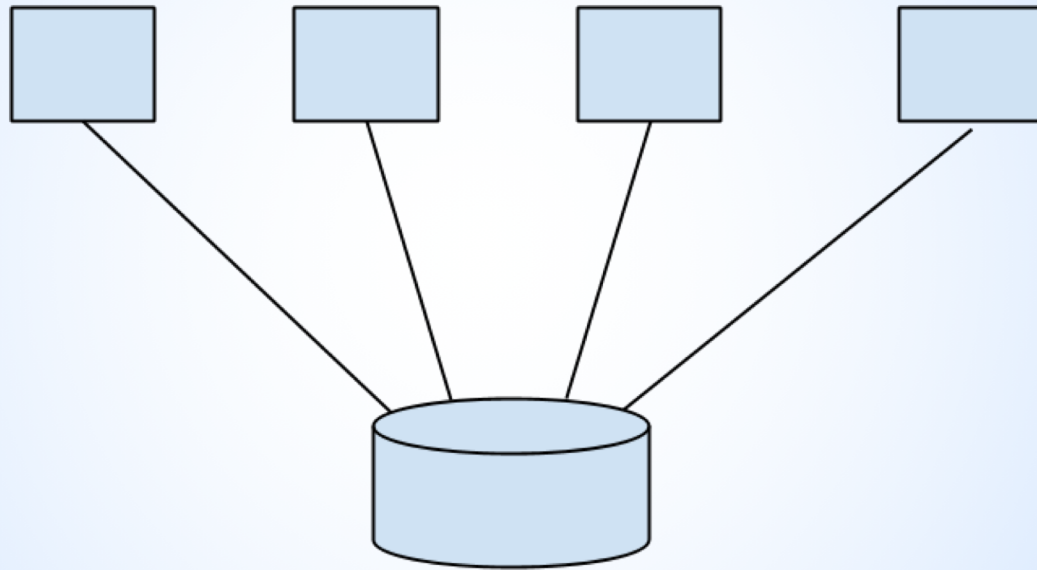


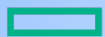
Collective I/O



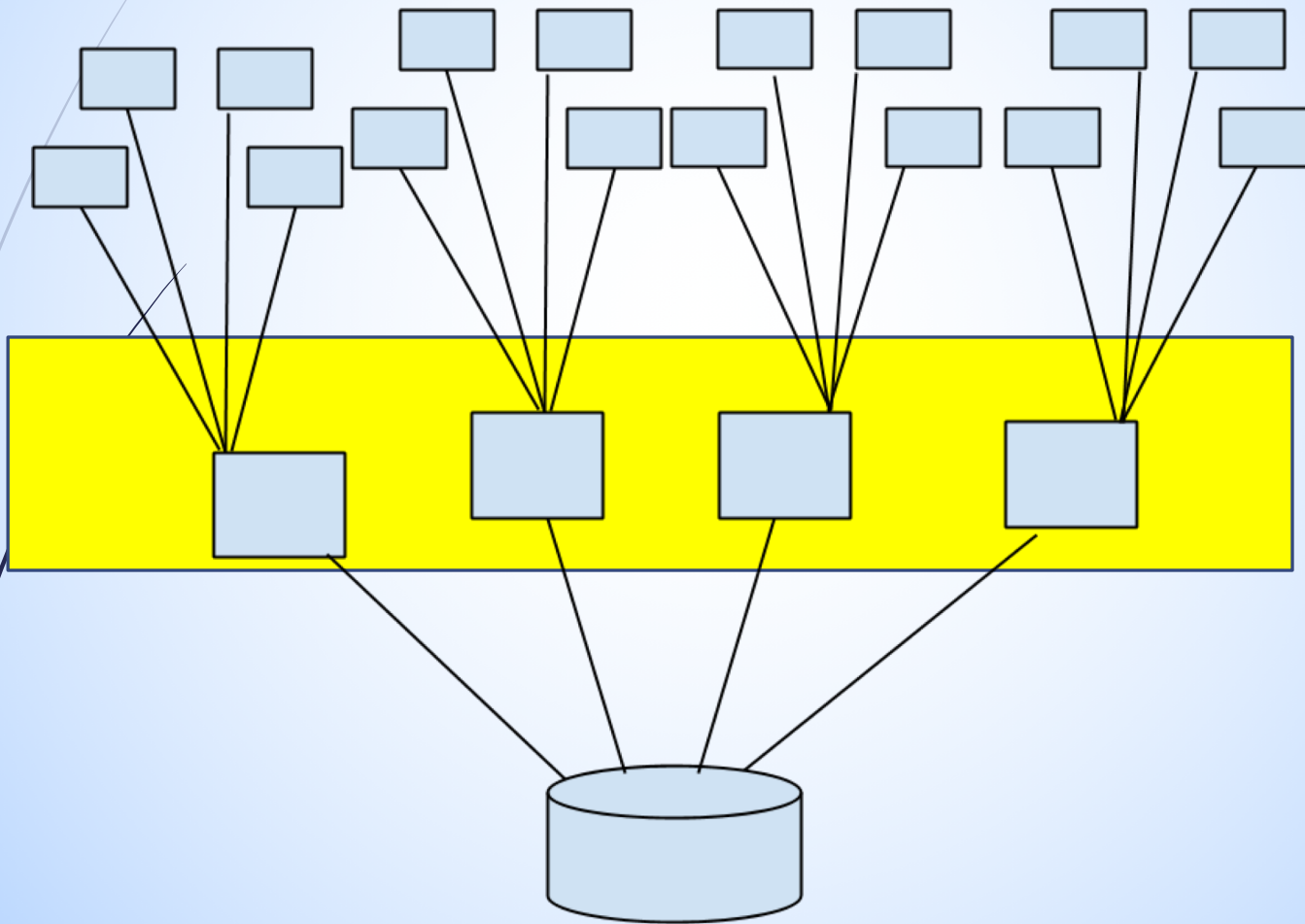


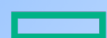
Independent IO





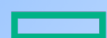
How quilt IO Works





Use NetCDF

- &time_control
 - io_form_history = 2,
- &domains (optional)
 - nproc_x = 10,
 - nproc_y = 10,
- &namelist_quilt
 - nio_tasks_per_group = 0,
 - nio_groups = 2,

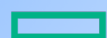


pnetcdf

- &time_control
 - io_form_history = 11,
- &domains (optional)
 - nproc_x = 10,
 - nproc_y = 10,
- &namelist_quilt
 - nio_tasks_per_group = 0,
 - nio_groups = 2,

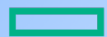
split

- &time_control
 - io_form_history = 102,
- &domains (optional)
 - nproc_x = 10,
 - nproc_y = 10,
- &namelist_quilt
 - nio_tasks_per_group = 0,
 - nio_groups = 2,



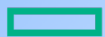
Quilt

- &time_control
 - io_form_history = 2,
- &domains
 - nproc_x = 9,
 - nproc_y = 10,
- &namelist_quilt
 - nio_tasks_per_group = 5,
 - nio_groups = 2,

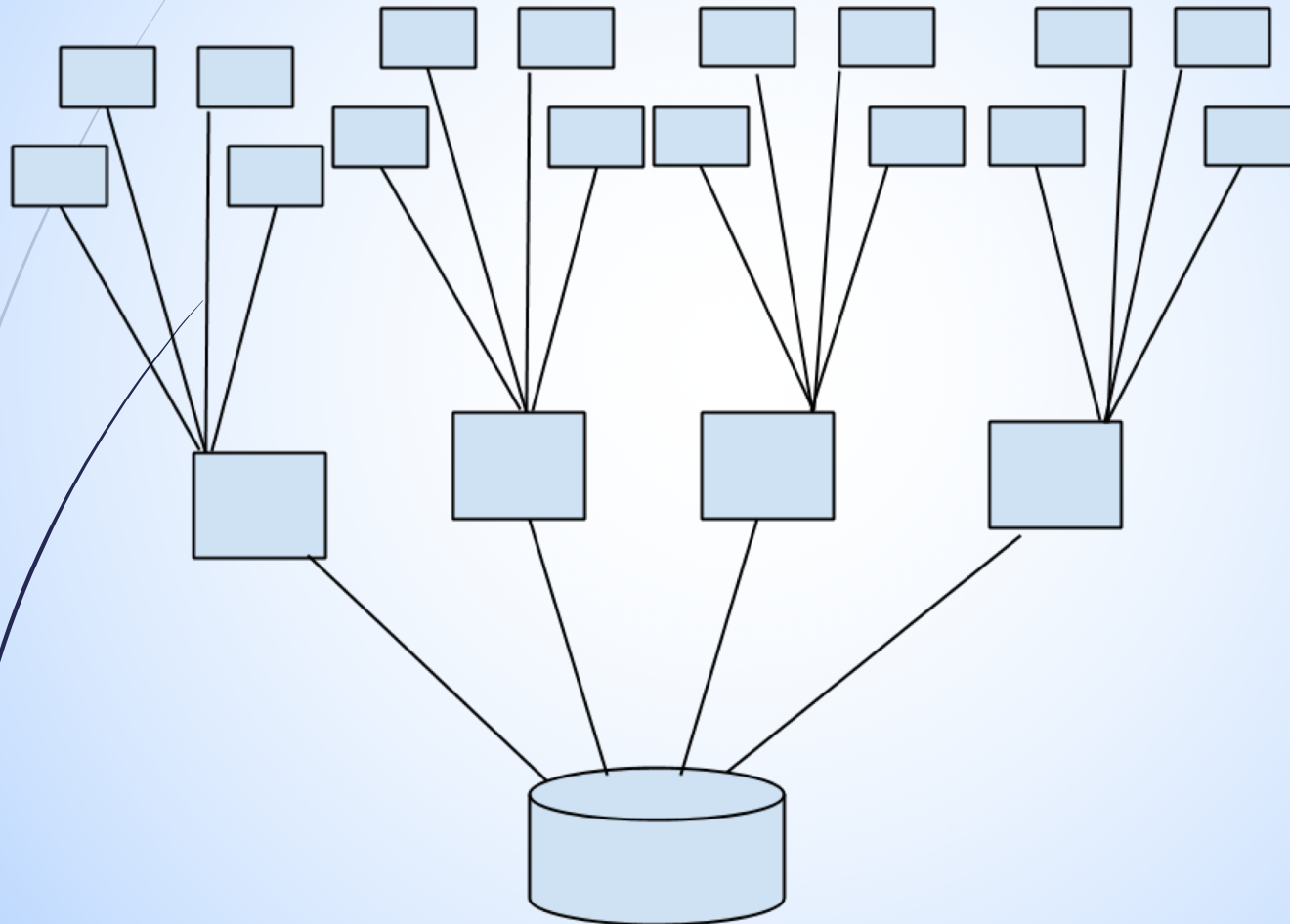


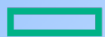
Paralell IO Library --PIO

- Developed by
 - John Dennis, Jim Edwards, Robert Jacob, Ray Loy, Art Mirin, Mariana Vertenstein
 - Web site
 - <http://www.cesm.ucar.edu/models/pio/>
- Has been used in
 - CESM
 - MPAS
 - WRF since 3.8.1

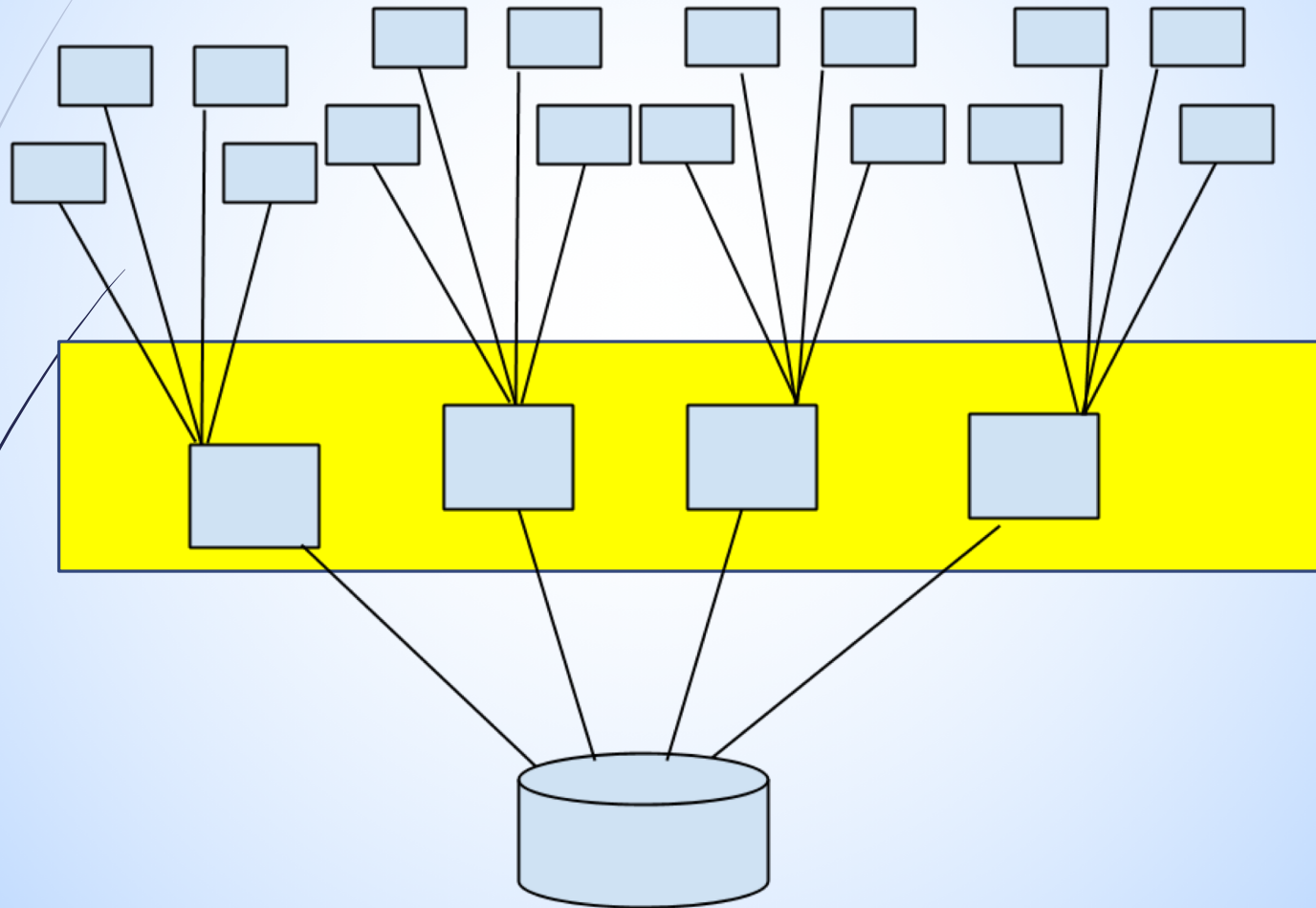


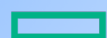
How PIO Works





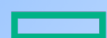
Quilt IO Vs PIO





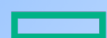
Implementation

- A new directory
 - **external/io_pio**
- Minimum code change
 - **Module_io.F, input_wrf.F, output_wrf.F, etc.**
- New namelist block
 - **pio_control**
 - pioprocs
 - piostride
 - piostart
 - pioshift



PIO Namelist

- &time_control
 - io_form_history = 12,
- &domains (optional)
 - nproc_x = 10,
 - nproc_y = 10,
- &namelist_quilt
 - nio_tasks_per_group = 0,
 - nio_groups = 2,



PIO Performance

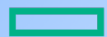
► Wall Clock Time

# of Nodes/Cores	Wall Clock (seconds)		
	NetCDF 3	pnetcdf (with stripe)	PIO
9 / 324	3020	1765	1883
18 / 648	2223	985	997
36 / 1296	1384	681	596
72 / 2592	1171	790	408
128 / 4608	2089	1489	294

PIO Performance (Cont.)

► *Total Output Time*

# of Nodes/Cores	Total Output Time (seconds)		
	NetCDF 3	pnetcdf (with stripe)	PIO
9 / 324	1323.22	129.82	231.92
18 / 648	1316.74	142.16	126.86
36 / 1296	886.33	198.05	119.53
72 / 2592	880.63	502.08	115.91
128 / 4608	1870.94	1283.95	86.25
256 / 9216	1845.88	2674.81	154.78



Conclusion

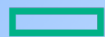
- Small Domain using less than 100 Processors
 - Do not bother with IO
 - NetCDF 3 (option 2) will work pretty well

Conclusion

- Small Domain using less than 100 processors
 - Do not bother with IO
 - NetCDF 3 (option 2) will work pretty well
- Medium Domain, few hundred processors
 - Use pnetcdf (option 11)

Conclusion

- Small Domain using less than 100 processors
 - Do not bother with IO
 - NetCDF 3 (option 2) will work pretty well
- Medium Domain, few hundred processors
 - Use pnetcdf (option 11)
- **Large Domain, thousands of processors**
 - **Need thinking of PIO (option 12)**

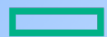


Future Work

- Make PIO an configurable options
- More testing/tuning
 - On different platforms
 - With variety of domain(s)

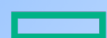
Acknowledgement

- Thanks to:
 - **Jim Edwards, CGD/NCAR**
 - Help implement PIO
 - **Dave Gill, Wei Wang, MMM/NCAR**
 - Giving me the chance to work on this project



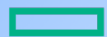
Questions ?

➡ Thank You!



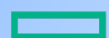
Initialize PIO

call PIO_init(&
myrank, &
communicator, &
pioprocs, &
piostart, &
piostride, &
PIO_rearr_box, &
DH%iosystem,&
pioshift)



Define PIO io-descriptor

- WRF grid domain
ids, ide, jds, jde, kds, kde, &
ims, ime, jms, jme, kms, kme, &
its, ite, jts, jte, kts, kte
- compdof_3d (3d io-descriptor)
dimension((ime - ims + 1) &
 * (jme - jms + 1) * &
 (kme - kms + 1))



Initialize io-descriptor

- Initialize io_descriptor

compdof_3d = 0

- Setup points inside tile

do j = jts, ljte

do k = kts, lkte

do i = its, lite

npos = (i - ims + 1) + (ime - ims + 1) &

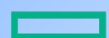
*** (k - kms + (kme - kms + 1) * (j - jms))**

compdof_3d(npos) = i + (ide-ids+1) * (j - 1 + (jde-jds+1) * (k - 1))

enddo

enddo

enddo



Setup io-descriptor

- call `PIO_initdecomp( & DH%iosystem, PIO_int, dims3d, & compdof_3d, DH%iodesc3d_m_int)`
- call `PIO_initdecomp(& DH%iosystem, PIO_real, dims3d, & compdof_3d, DH%iodesc3d_m_real)`
- call `PIO_initdecomp(& DH%iosystem, PIO_double, dims3d, & compdof_3d, DH%iodesc3d_m_double)`

Apply io-descriptor

- Select io-descriptor

Read a 3d real at Mass grid point

```
DH%ioVar(DH%CurrentVariable) = &  
    DH%iodesc3d_m_real
```

- Read data

call pio_read_darray(&
 DH%file_handle, &
 DH%descVar(DH%CurrentVariable), &
 DH%ioVar(DH%CurrentVariable), &
 Data, stat)

Performance on Cheyenne

► Domain size: 800x400x55

# of Nodes/Cores	Wall Clock (seconds)			
	NetCDF 3	pnetcdf	PIO	
4 / 144	4.82793	3.04686	15.27906	Input
	21.54631	2.82851	6.97871	Output
	108.19372	10.19792	17.21590	Restart
24 / 864	4.73887	10.56711	6.30086	Input
	24.22607	21.86972	4.94379	Output
	120.87177	63.85113	6.79355	Restart

Performance on Cheyenne (cont.)

► Domain size: 800x400x55

# of Nodes/Cores	Wall Clock (seconds)			
	NetCDF 3	pnetcdf	PIO	
72 / 2592	5.30372	12.16695	5.23244	Input
	25.17039	20.54580	10.95963	Output
	123.36142	62.05905	8.41547	Restart