NCI AUSTRALIA

Parallel IO in MOM5

Rui Yang and Marshall Ward

nci.org.au
@NCInews

IO patterns in MOM5

IO Patterns (write)	Number of Output Files	Run Time	Post-processing Time
Serial IO Single File per global domain	1	long	none
Serial IO Single File per IO domain	IO domains	moderate	long
Serial IO Single File per PE	PEs	short	long
Parallel IO	**1**	**scalable towards short**	**none**

Parallel IO can also reduce the memory requirement when creating a single output file.

Parallel IO Implementation in MOM5

- **Parallel API** to open/create the NetCDF files i.e. nf_create_par and nf_open_par.
- All fields are operated **as collective read/write** fields, via the NF_COLLECTIVE tag, as required for variables with an unlimited time axis and for good performance.
- **MPI-IO hints** are customized for performance fine-tuning.
- The root PE of each IO domain, i.e. IO tasks, are grouped in a **sub-communicator** via FMS subroutines for parallel IO purpose.
- All implementations was written in a way to take advantage of existing **FMS functionality**.
- **New namelist** variables have been introduced to enable parallel IO and choose libraries:

```
&mpp_io_nml
 parallel_netcdf = .true. #enable parallel IO
 pnetcdf = .true.         #choose PnetCDF or HDF5 lib.
/
```

Parallel IO Benchmark

Layer	Parameter	Value
Model	Configurations	1-day simulations with diagnostic output enabled. • 0.25° model (1440×1080) for IO tuning • 0.1° model (3600×2700) to verify IO performance
Output	Diagnostic	Diagnostic fields: T, S, u, v, t_{age} Diagnostic file write frequency: • 30 minutes for 0.25°, 48 steps, 70GB • 5 minutes for 0.1°, 288 steps, 2.7TB
Benchmark	PE	240, 960 for 0.25° model ,720,1440 for 0.1° model
	Domain Layout	16×15 for 240 PEs, 32×30 for 960 Pes (0.25°) 48×15 for 720 Pes,48×30 for 1440 Pes (0.1°)
	IO library & Format	NetCDF v4.6.1 with the following libs&formats: • HDF5 v1.8.20 & NC4 (default chunk) • HDF5 v1.10.2 & NC4, NC4-classic(default chunk) • Parallel NetCDF v1.9.0 & NC classic 64bit-offsets

Parallel IO tuning parameters

Layer	Parameter	Value
IO domain	IO layout (Ioy × Ioy)	iox=32,16,8,4,2,1 and ioy=30,15,5,3,1
NetCDF	chunk	default
MPI-IO	Cb_buffer_size	32MB
	Cb_nodes	number of PEs
	Naggr/node	1,2,4,8
Lustre	Stripe size	1MB
	Stripe counts	15,30,60,120,165(max)

Filtered out many other MPI-IO parameters based on our experiments and experiences.
Experiments are carried out at NCI Raijin supercomputer nodes with 16 cores/node.

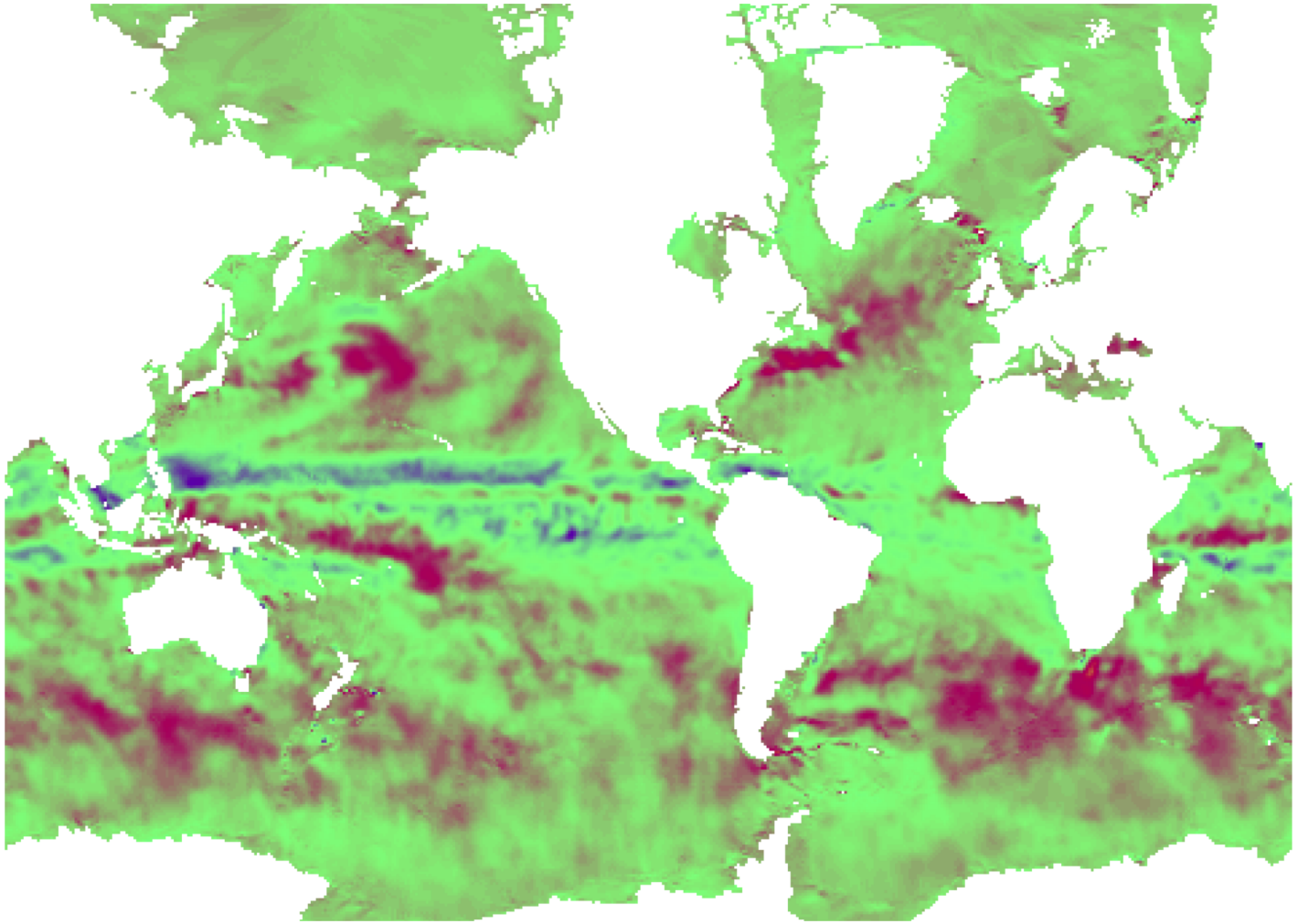
Serial single file IO (global domain)

0.25° Model	240 PEs			960 PEs		
Time (s)	64-bit offset (PnetCDF 1.9.0)	NC4 (HDF5 1.10.2)	NC4-classic (HDF5 1.10.2)	64-bit offset (PnetCDF 1.9.0)	NC4 (HDF5 1.10.2)	NC4-classic (HDF5 1.10.2)
Total runtime	637.82	687.20	675.93	629.33	671.95	675.40
mpp_open	7.46	6.39	6.31	15.62	14.97	14.41
mpp_read_meta	3.90	3.73	3.85	6.16	4.88	4.92
mpp_read	4.58	4.15	4.18	2.37	2.43	2.61
mpp_write_meta	0.01	0.01	0.01	0.00	0.01	0.01
mpp_write	**545.50**	**592.39**	**584.86**	**576.92**	**616.35**	**618.08**
mpp_write_3d	69.77	72.00	72.50	72.82	76.58	77.53
mpp_write_4d	475.72	520.22	512.27	504.09	539.58	540.44
mpp_close	0.65	0.96	1.10	1.23	2.37	2.47

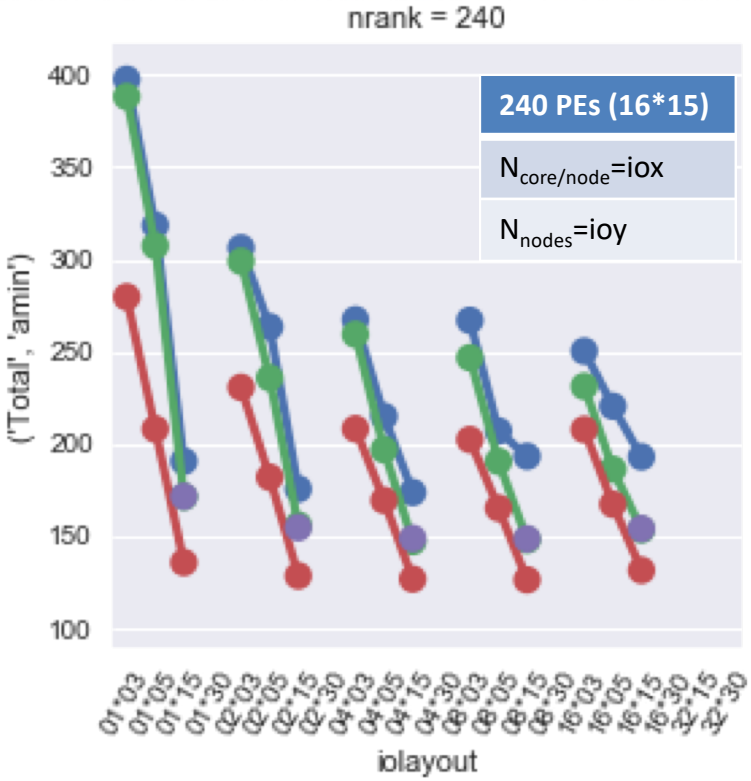
Serial single file IO is used as the reference.

Serial single file IO does NOT scale well to PEs.

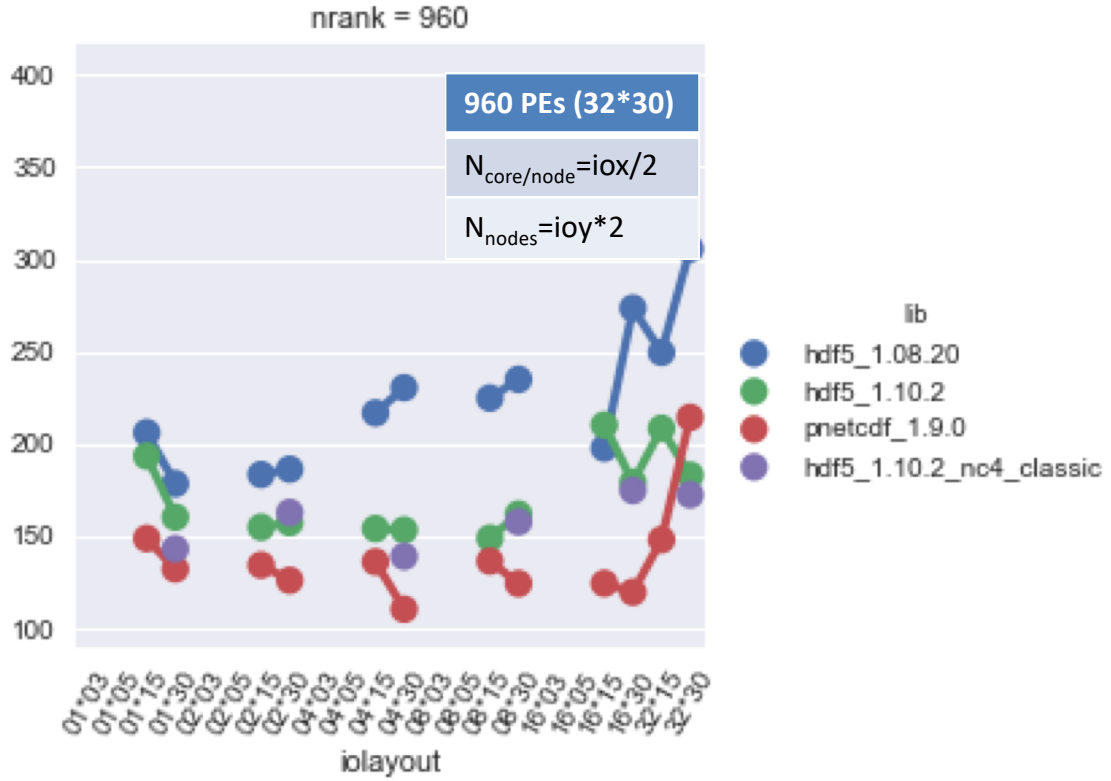
Diagnostic output (u) from Parallel IO



IO tuning: IO domain



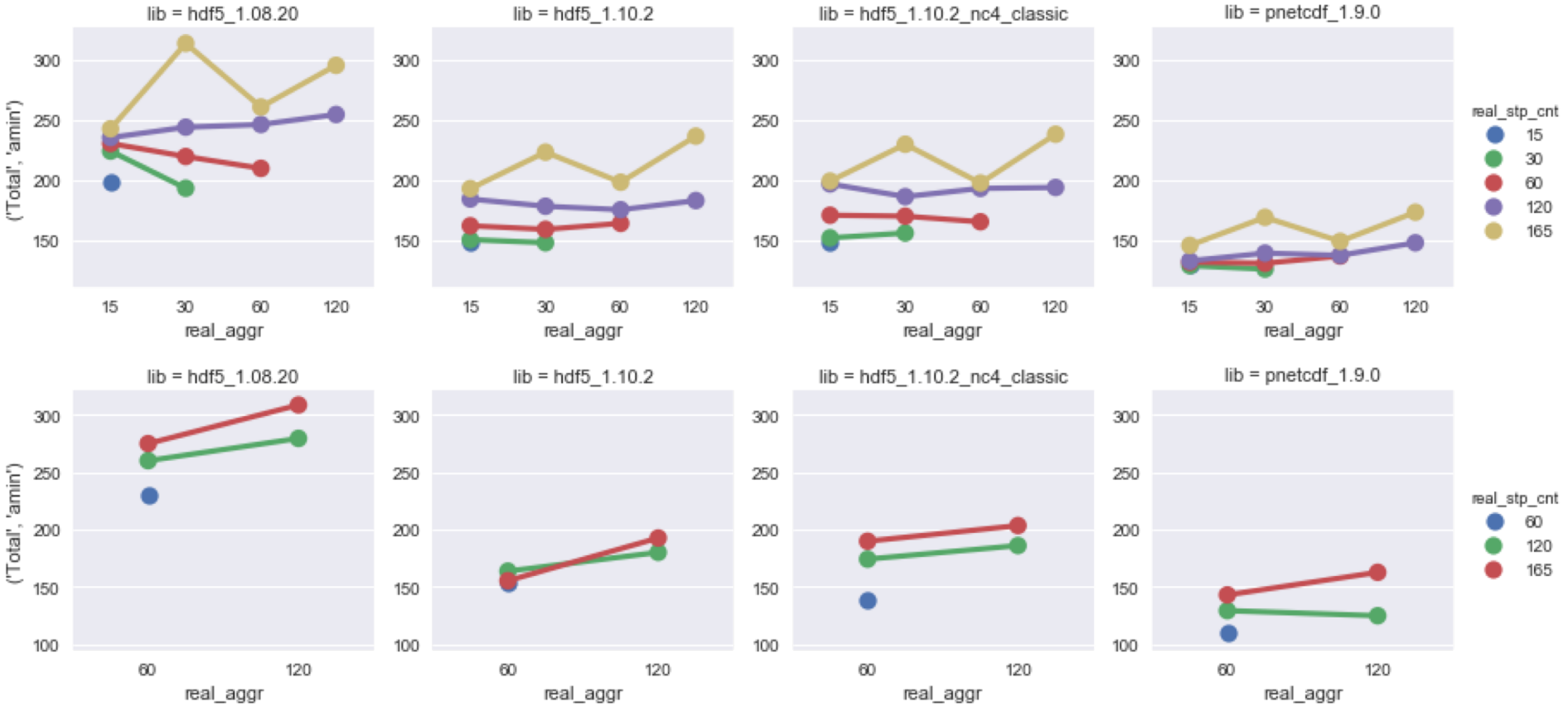
Serial single file IO:
HDF5_1.10.2:687
PnetCDF_1.9.0:638
HDF5_1.10.2_NC4_classic:676



Serial single file IO:
HDF5_1.10.2:672
PnetCDF_1.9.0:629
HDF5_1.10.2_NC4_classic:675

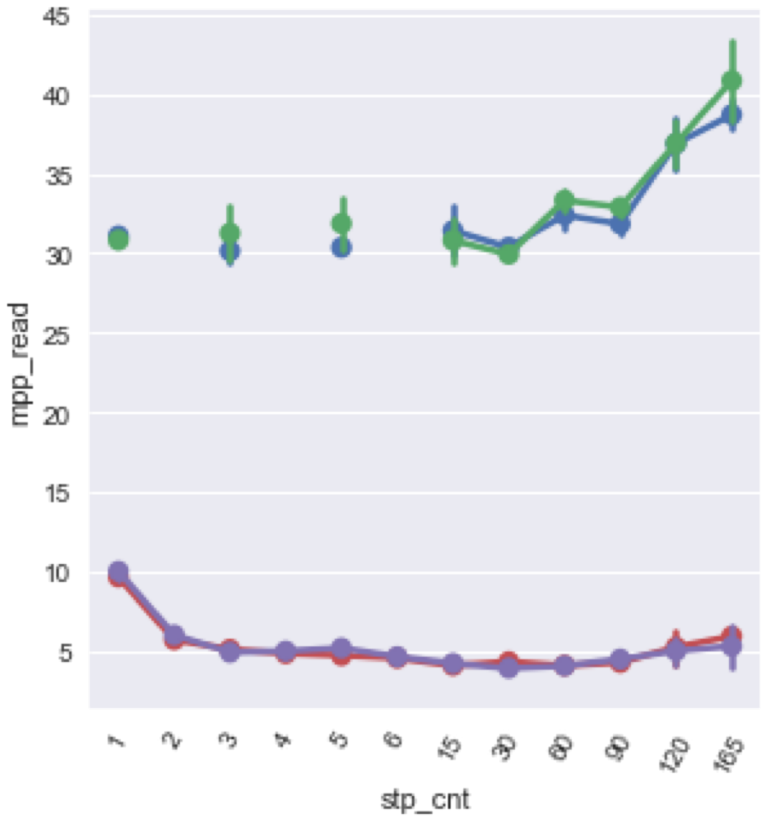
- Each node contains at least 1 IO domain to avoid inter-node communications.
- Each node uses 1~4 PEs to reduce the IO contentions.
- PnetCDF&NC Classic has the best performance.
- Parallel IO scales to PEs, and is much quicker than serial single file IO.

IO tuning: aggregator and stripe counts



Use 1~2 aggregators per node and Lustre stripe count should match total number of aggregators.

Serial Read .vs. Parallel Read



In MOM, Serial READ is more stable and faster than parallel READ.

0.10° Model Configuration based on optimal IO settings

Configurations	720 PEs	1440 PEs
Domain Layout	48×15 (720 PEs, 16 PEs/node, 45 nodes)	48×30 (1440 PEs, 16 PEs/node, 90 nodes)
IO Layout	3×15 (1 IO domain/node, 45 IO domains)	3×30 (1 IO domain/node, 90 IO domains)
Aggregator	1/node, 45 in total	1/node, 90 in total
Stripe count	45	90

IO performance in 0.1° Model

SIO: serial IO single file per global domain; PIO: parallel IO

Time (sec.)	NC4 (HDF5 1.10.2)			NC 64-bit offset (PnetCDF 1.9.0)		
IO Pattern	SIO	PIO	PIO	SIO	PIO	PIO
PEs	720	720	1440	720	720	1440
Total runtime	21689	1867	1118	19726	1535	812
mpp_open	8	78	128	9	29	74
mpp_read_meta	2	6	6	3	7	5
mpp_read	25	20	35	15	20	30
mpp_write	20826 (5.8 hrs)	909 (15 mins)	483 (8 mins)	18839 (5.2 hrs)	656 (11 mins)	325 (5.4 mins)
mpp_write_3d	458	27	12	455	69	9
mpp_write_4d	20369	880	469	18385	586	315
mpp_close	8	68	100	0	1	0

- Parallel write is much faster than serial write.
- Parallel IO is scalable to number of nodes/aggregators, but limits at 165 (2640 PEs).
- NC classic format is 20%~30% faster than NC4.

Summary and Future Work

Summary

- Parallel write wins both performance and management .vs. serial write.
- Use serial read in MOM.
- IO tuning is necessary to gain good performance.
- Parallel IO performance scales with aggregators.

To do

- NC4 format needs further tuning work, i.e. chunking.
- Parallel compression in HDF5 1.10.2 and NetCDF v???

Acknowledgements