Scientific Application performance and LUSTRE

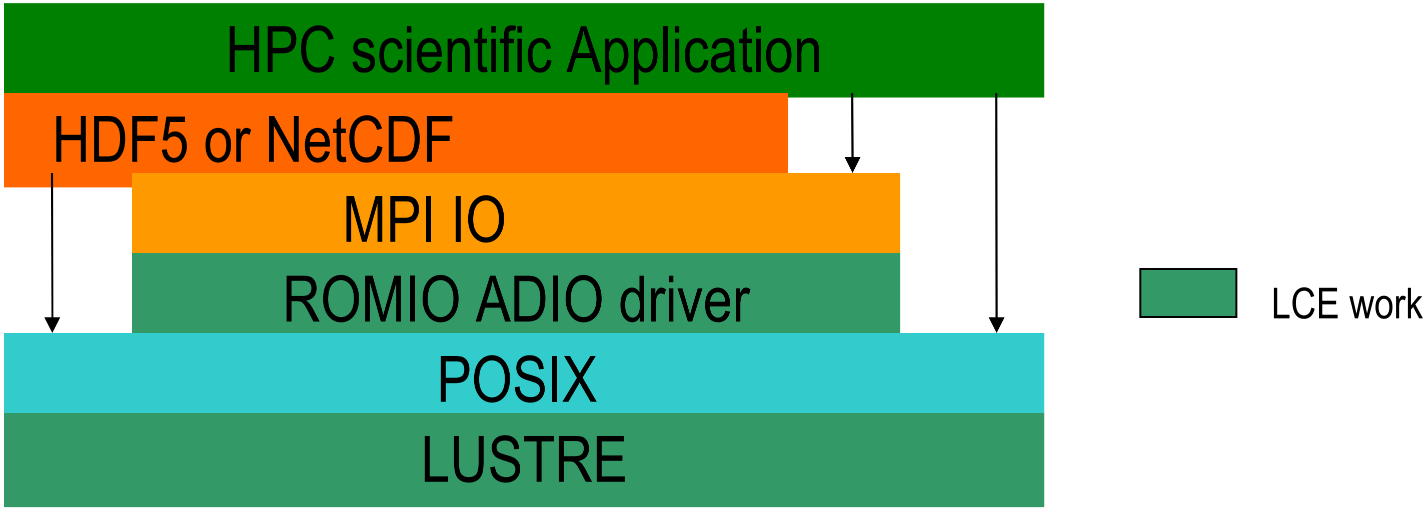
Tom Wang
Lustre Group

Agenda

- Scientific Application IO
- LUSTRE IO Tuning
  - General IO Tuning
  - Different IO API
  - HDF5
  - Examples
- LUSTRE ADIO driver

Scientific Application IO

- Scientific HPC application software stack



HPC scientific Application

HDF5 or NetCDF

MPI IO

ROMIO ADIO driver

POSIX

LUSTRE

🟩 LCE work

Scientific HPC application software stack

Scientific Application IO

- Required IO and Checkpoint IO
  - > Only writing or reading once then writing periodically.
  - > Most of META data operations are open/create.
- Contiguous IO and non-contiguous IO
- Implementation
  - > Some applications implement their IO by scientific IO lib (NetCDF or HDF5), some use MPIIO or POSIX directly.
  - > Some libraries support parallel IO pNetCDF and HDF5. Some do not, for example NetCDF.

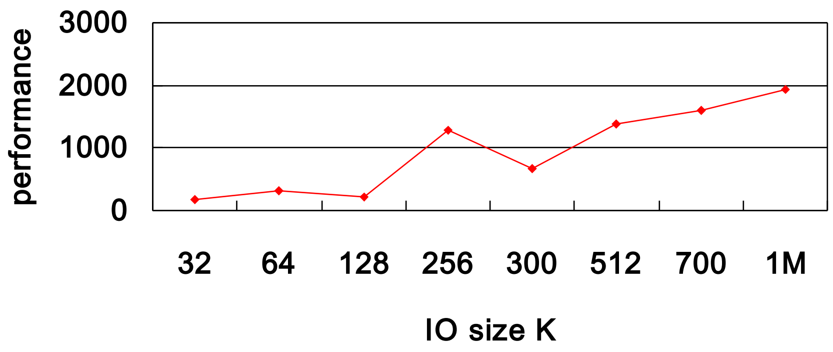
General IO Tuning

- Requirements for achieving good IO performance
  - > Balanced OST load.
    - – Choose right stripe size and stripe count according to the IO pattern.
  - > Efficient RPC between clients and servers.
    - – Saturate Network and disk IO
      - – Do stripe size IO
      - – Especially for Liblustre client.
    - – Less RPC and lock conflicts
      - – Stripe size aligned IO

General IO Tuning

- Different IO size comparison

IOR performance(MiB/sec) with different IO size 256 clients



Stripe_size = 1M.

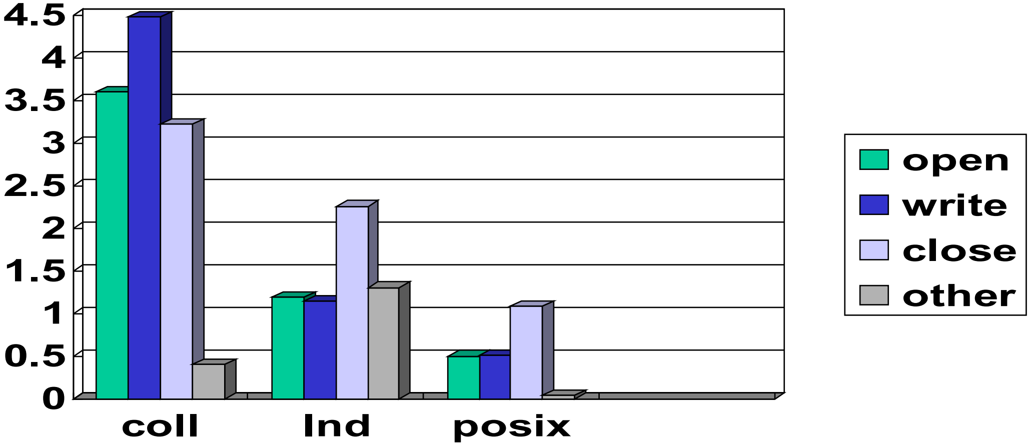
In some unaligned size points(300k and 700k), the performance dropped a bit.

Different IO API

- POSIX
  - Call POSIX system call directly, no optimization.
- Independent
  - Optimize the data pattern locally by data_sieving and stripe_size aligned.
- Collective
  - Optimize the data over multi-clients. Change interleave ,discontinuous and uneven IO load over multi clients into continuous and even IO load.
- Overhead of Independent and collective
  - Choose different API according to the application IO pattern.

HDF5

- HDF5 supports different low-level IO API



driver	coll	Ind	Posix
Total time(seconds)	11.7	5.85	2.13

Different layer performance with flash IO (256nodes)

HDF5

- Open
  - > Open costs abnormal high time in Flash IO sometimes
    - – 30%-40%time (1.3 seconds ---- 3.2 seconds)
    - – Reason: In HDF5, when open existing file with (TRUNC flags), all the clients will call MPI_SET_File_size to truncate the file to zero, which occupies about 95% open time.
- Write
  - > Improper read-modify-write for HDF5 collective IO
- Close
  - > HDF5 close includes flush(HDF5_mpio_flush).
    - – Which will cost about 40%-50% time.

Examples

- POP
  - The I/O client aggregates data from other computation clients. I/O size is about 60M.
    - Support Fortran POSIX IO, and NetCDF (non-parallel)
  - Optimization
    - Implement HDF5 parallel IO
    - Stripe_size for 60M IO
      - 60M IO size will hold too much client lock cache of multi-server on client, which will impact other clients access those server. So choose stripe_size to make each client access servers in parallel.

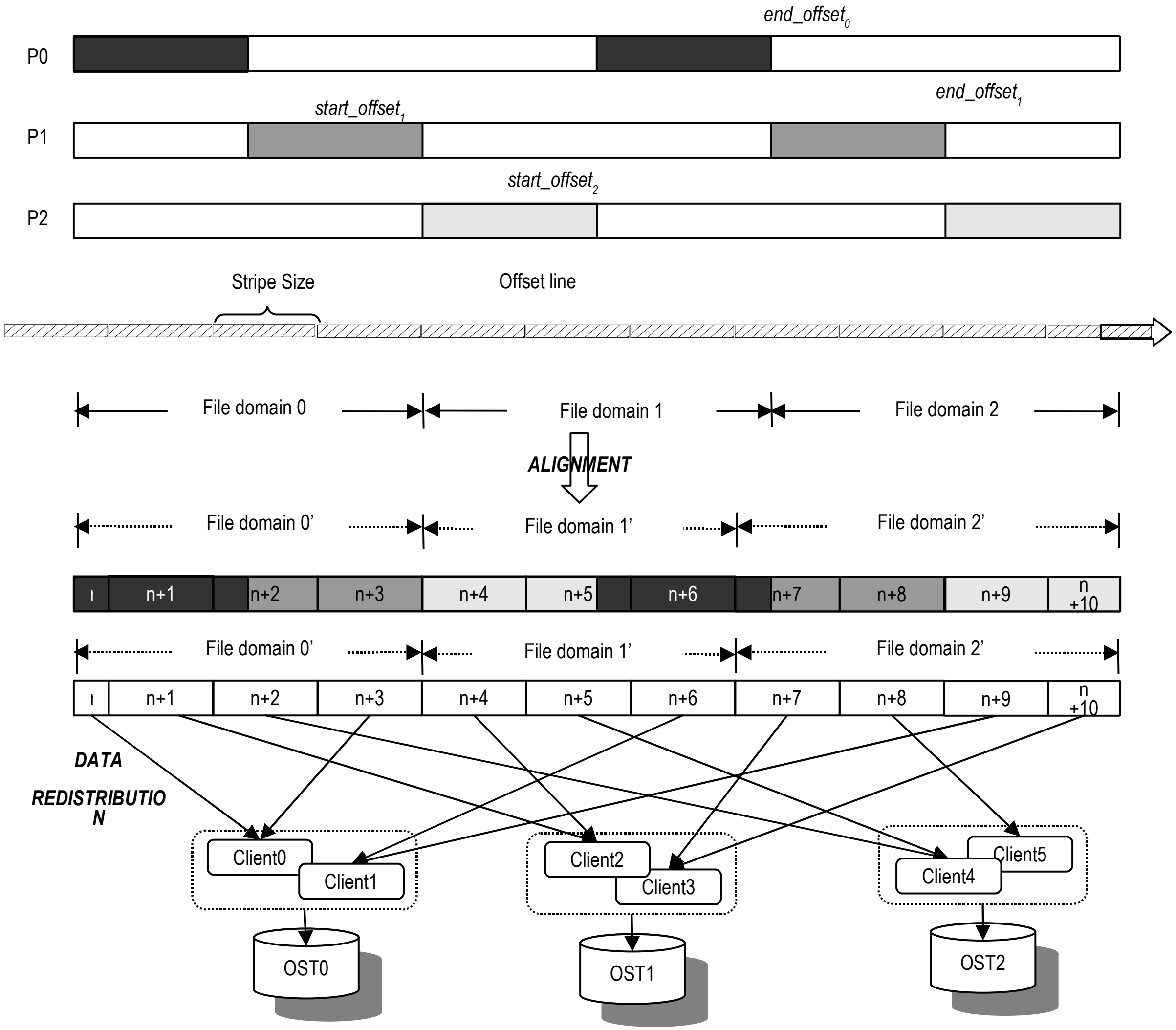
Examples

- WRF mode
  - > Produce a HDF5 file (about 8M)
    - – Each client writes several K bytes(small I/O size) to the shared data_set.
  - > Each client writes small and contiguous data segment
    - – Lustre does not like this I/O pattern.
    - – It is even worse for more clients.
  - > Optimization
    - – Optimize the WRF mode by the new Lustre ADIO driver.
    - – Aggregate the data from multi-clients and write big I/O size.

LUSTRE ADIO Driver

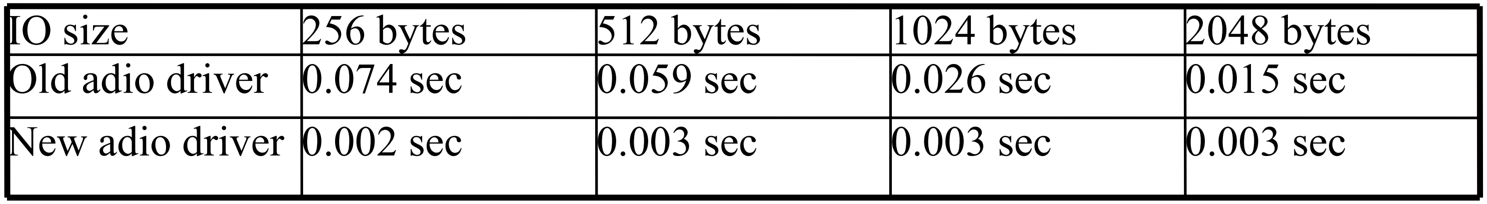
- Collective Write
  - Reorganize the data between the clients according to striping information.
    - Reorganize the data according to real data location on OST.
    - Choose IO clients to avoid unnecessary communication between clients.
    - Do stripe_size I/O
  - I/O patterns benefits from this driver.
    - Big size IO will be split to stripe_size IO(POP).
    - For small size IO, the data will be aggregated and do big size IO(WRF).

LUSTRE ADIO Driver



LUSTRE ADIO Driver

- Comparison



IO size	256 bytes	512 bytes	1024 bytes	2048 bytes
Old adio driver	0.074 sec	0.059 sec	0.026 sec	0.015 sec
New adio driver	0.002 sec	0.003 sec	0.003 sec	0.003 sec

- Overhead
  - In the ADIO driver, the time costs on communication increase a lot when IO size increases, which is unexpected.
    - The reason is being investigated.

Thanks & Questions

Tom.Wang@sun.com