ICHEC
Irish Centre for High-End Computing

Using DDN IME for Harmonie

Gilles Civario, **Marco Grossi**, Alastair McKinstry, Ruairi Short, Nix McDonnell

April 2016

DDN IME: "Infinite Memory Engine"

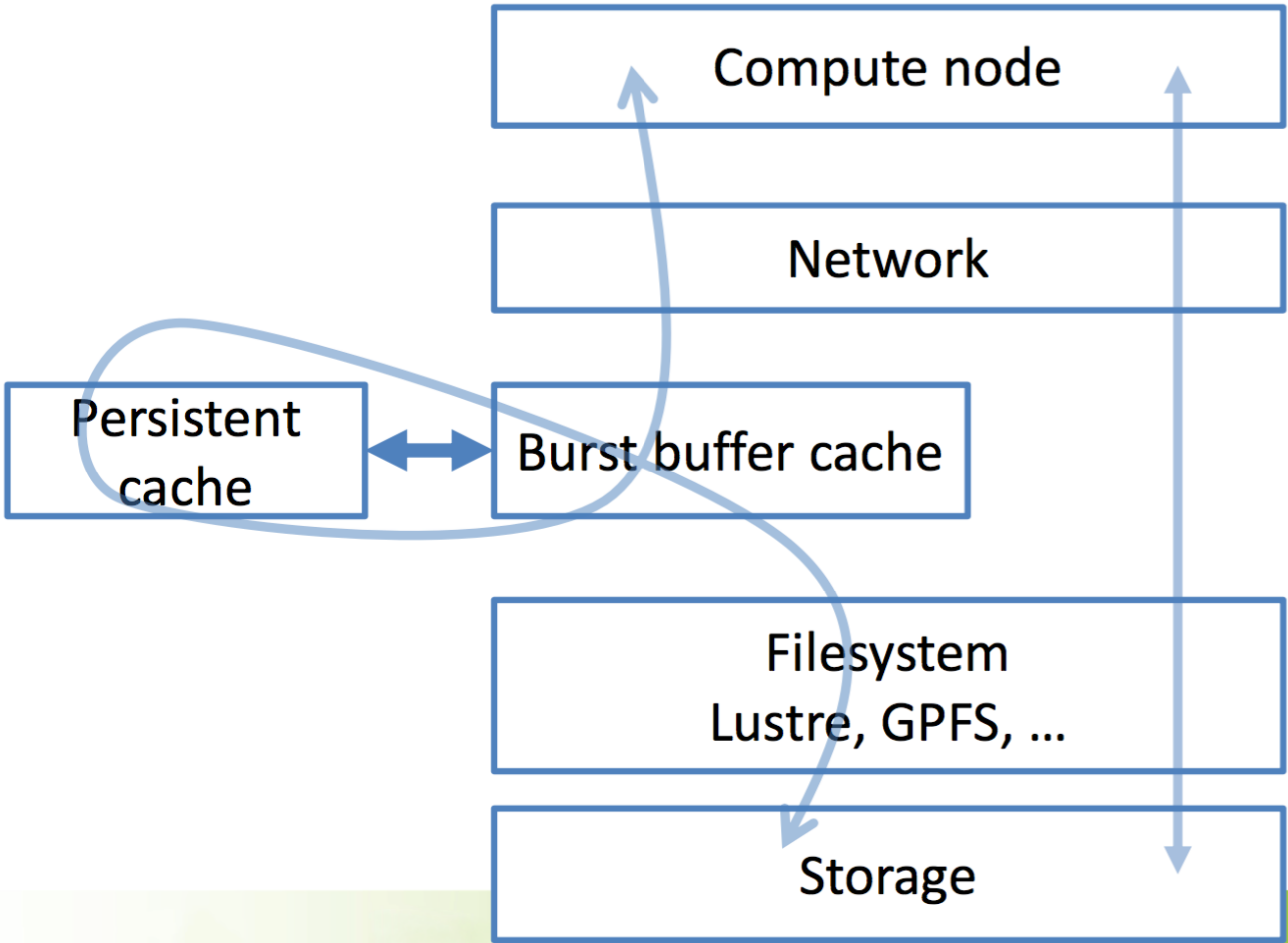
Burst buffer cache

- A burst buffer cache is a layer between the user and the filesystem
  - Absorb burst of write operations
  - Speedup read/write operations
  - Re-organize I/O requests in order to issue optimized operations to the filesystem

- Might not be completely transparent
  → multiple interfaces provided

- For the majority of interfaces the concept of a filesystem is still present
  - Not necessarily each interface will eventually flush to the underneath filesystem

What a burst buffer looks like?



Burst buffer interface

- Transparent
  - Any I/O passes through the buffer
  - No modification required to the application code
- Library-specific
  - MPI-IO, HDF5, …
- Low level API
  - E.g. dictionary style, key/value pair
  - In this case the concept of filesystem may not apply

IME: Major Features

- **Burst Buffer** – Takes bulk writes quicker than PFS.
- **High-performance Global Cache** – importing data into or file pinning data in the IME data tier,
- **I/O Accelerator** – Avoids POSIX locking bottleneck→ enhanced, low-level communications protocols, accelerating both reads and writes.
- **Application Workflows** – Integrates with job schedulers, enabling simultaneous job runs and shortening the job queue for faster application run time.
- **PFS Optimizer** – Dynamically reorganizes data into sequential writes, eliminating the latency, thrashing, and slow write times created by the fragmented I/O patterns of demanding mixed workload applications.
- Ag**gregated Storage Capacity** – Intelligently virtualizes disparate NVM devices into a single pool of shared memory, providing increased capacity and bandwidth across a cluster of IME Server nodes.
- **Scalable and Fault-tolerant Solution** – Provides scalability and redundancy at both the storage device and node level. If a server becomes saturated IME Client will automatically re-direct the data traffic

DDN | IME

Application I/O Workflow

COMPUTE

Diverse, high concurrency applications

IME

Fast Data NVM & SSD

SFA

Persistent Data (Disk)

Lightweight IME client passes fragments to application

IME server sends fragments to IME clients

IME servers write buffers to NVM and manage internal metadata

IME prefetches data based upon scheduler request

Parallel File system acts as persistent store for data

ICHECs experience

- Worked with IME in 2014:
  - Pre-GA code. Worked with MPI-IO, single program.
  - 30-60% speedup vs. Lustre on seismic code.

DDN - ICHEC | Whitepaper

ICHEC
Irish Centre for High-End Computing

Experimenting on IME with an Oil & Gas imaging code

Prepared by: Gilles Civario – Irish Centre for High-End Computing
Seán Óg Delaney – Tullow Oil plc
Marco Grossi – Irish Centre for High-End Computing
Date: November 10th 2014

ddn.com

INFORMATION IN MOTION

Todays work

- Test a full, NWP workflow:
  - Not just the forecast. Post-processing, dependent jobs
  - Simultaneous multiple jobs
  - Test MPI-IO and serial:
    - Adding post-processing file conversion to NetCDF to add MPI-IO job (using MVAPICH2)
    - Simultaneously read via POSIX

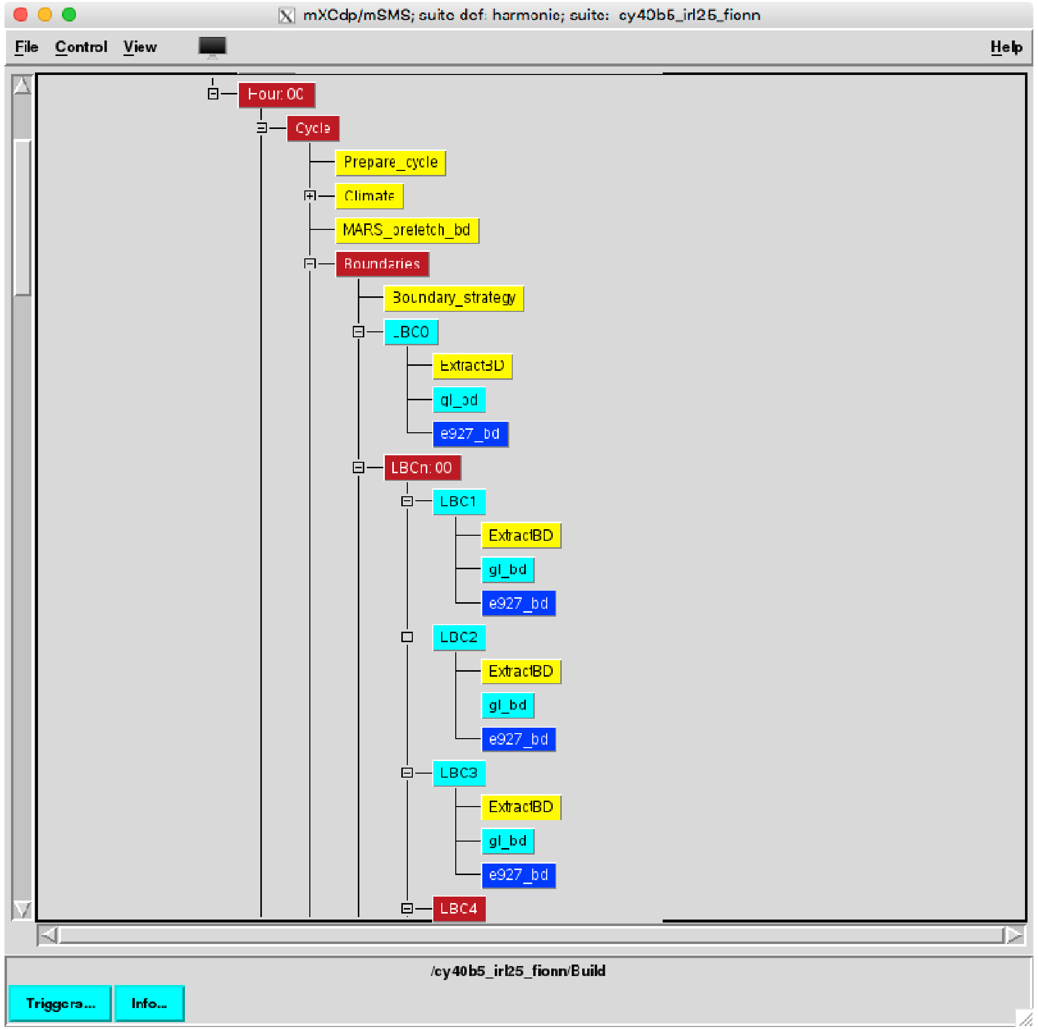
Harmonie

- Hirlam/ ALADIN consortium
  - In use in Met Éireann production at ICHEC
  - POSIX based I/O flow.
  - cy 40h1.1.5beta
  - FA/LFI files, postprocessed to GRIB
    - (Conversions to netCDF added to test MPI-IO)
  - IO_SERVER optional component

Harmonie workflow

Standard flow:

- Multiple (serial) pre-processing jobs
  - Populate cache
- Parallel forecast
- Post-processing (serial) jobs typically triggered on n-hour output
    - Read from cache



ICHEC Irish Centre for High-End Computing

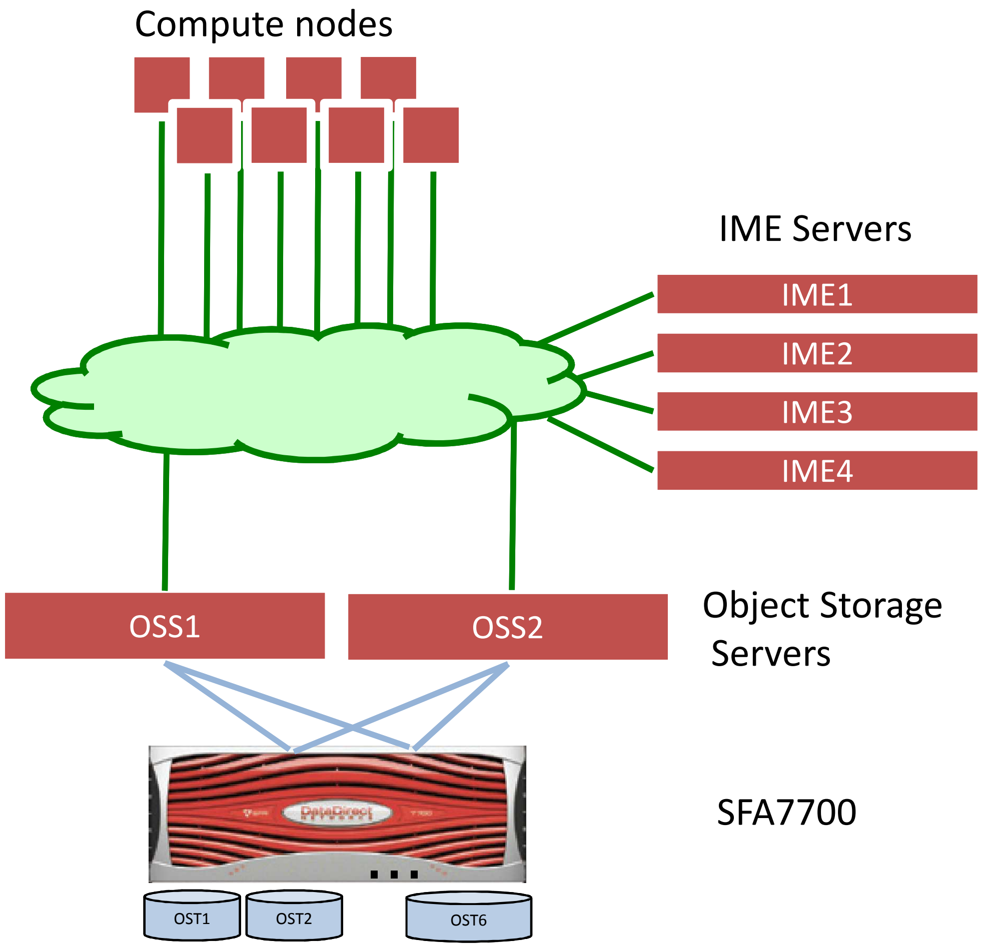
DDN STORAGE

Initial test system

8 x Compute Nodes:
- 2x Intel Xeon E5-2680v2
- 128GB RAM
- FDR InfiniBand

Filesystem Storage:
- DDN SFA 7700
- Lustre 2.5 with 2 x OSS servers
- 3.4GB/s Write, 3.3 GB/s Read

IME System:
- 4 servers with 24 x 240GB SSDs each
- 36GB/s Write, 39 GB/s Read



IME configuration

The persistence of the IME burst buffer is provided by SSD drives

Each IME Server:

- - 24 x 240GB SSD drive 2.5"
- - 2 x SAS2308 PCI-Express Fusion-MPT SAS-2

On each of the IME server are running two instances of the IME software service: -

each instance is pinned to a specific NUMA node and Infiniband port –

the maximum throughput per IME server is limited by the speed of the IB interface: dual port FDR in this case.

The MPI-IO request using IME as backend are automatically balanced between the configured IME server; data transfer between IME client and server is carried over RDMA.

- The MPI library to use for your run is a customized version of MVAPICH:
- MVAPICH version 2.0 that include the ROMIO driver for IME
- to use IME as backend for MPI-IO is simply as prepend the filename with 'im:/'
- the IME namespace replicate the underlining filesystem: in this particular case the Lustre one.
  - If - for any reason - your MPI task still need POSIX access in the context of an MPI-IO request: simply open the file as usual, omitting the 'im:/' prefix.

Test cases:

- IRL25:
  - 2.5 km domain over Ireland.
  - 500 x 540 grid, 45s timestep
  - Production run for Met Éireann
- IRL10:
  - 1 km: 1300 x 1300, 20s timestep
  - Also use IO_SERVER

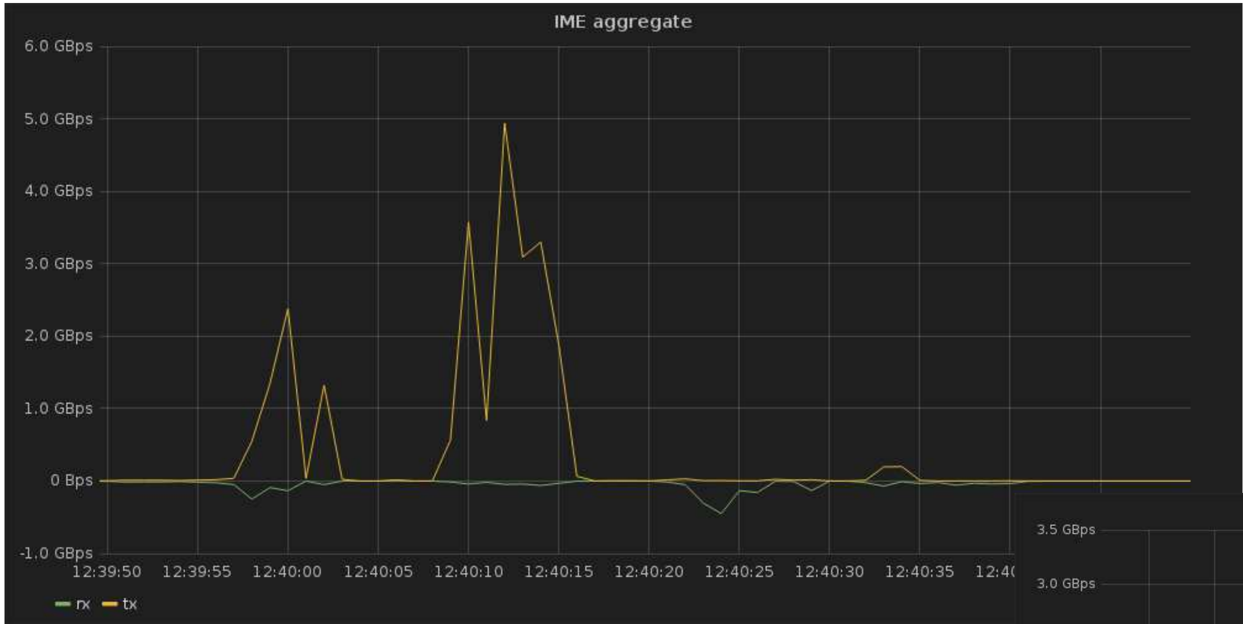
Results

- Harmonie workflow works ☺
- No significant speedup seen ☹

Results

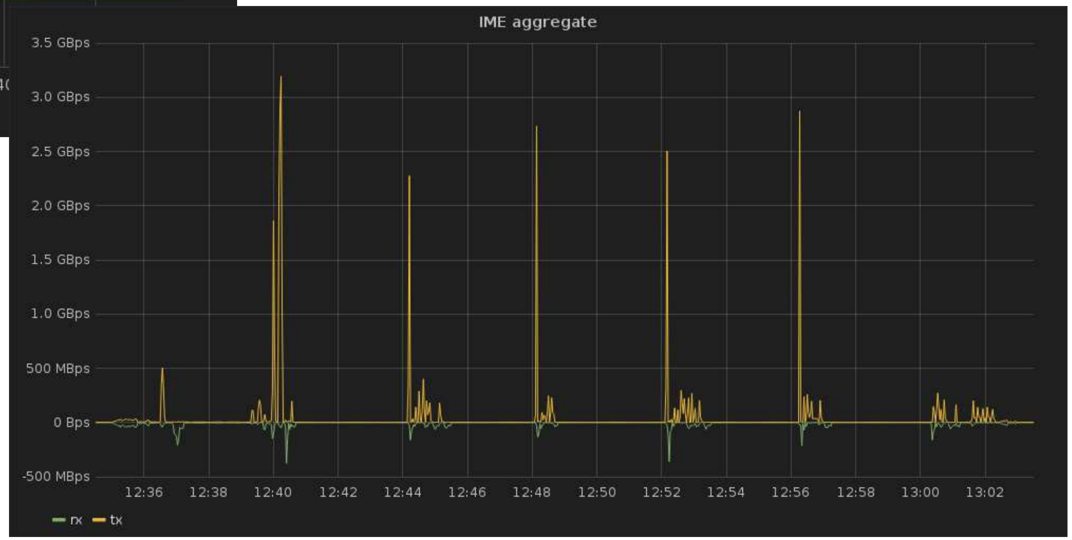
- Harmonie workflow works ☺
- No significant speedup seen ☹
  - Post-processing was tuned to minimize IO delays:
    - Minimal verification writes (every 6hr output)
    - Reduced variable set
- Tested IO server configuration, job size

Tracking via IB traffic



2.5km case,
No IO server.

8s IO step /each hr



Two stage writes,
including
SURFEX output

1km case (1300x1300)

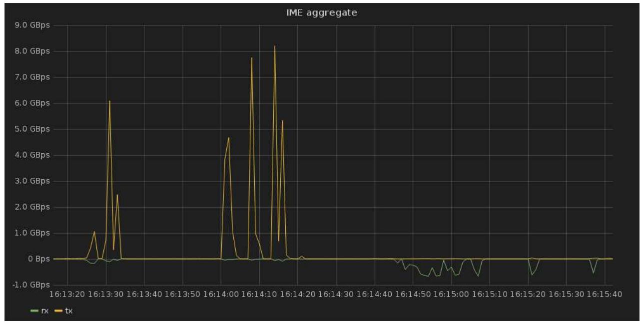
Limited by small test cluster size

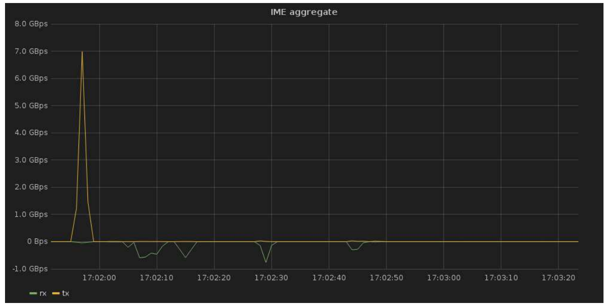
In IO Server case, 1-2 cores (pinned) reserved for server

20% drop in IO write time, (increased compute)

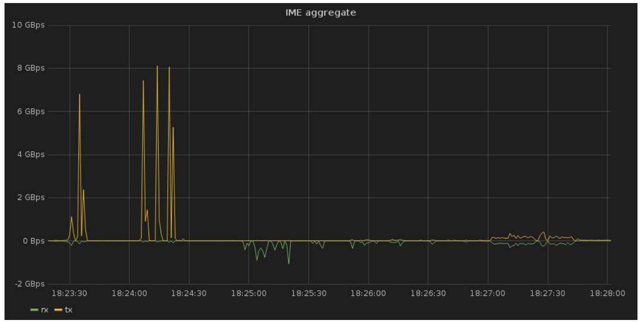
Same time to solution with IME

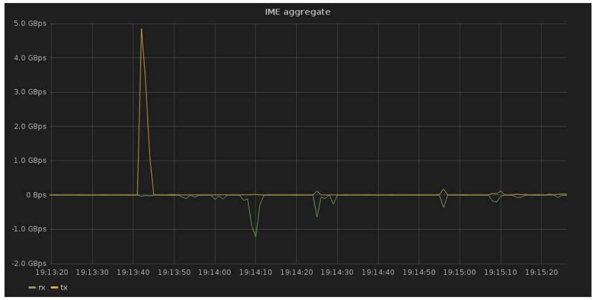
No IO Server





2 IO Server





startup

Typical step

Gotchas

- On crashes, inconsistent state seen in IME /Lustre
  - Scripts needed to delete and cleanup

/lustre/work/harmonie/hm_data/…
/ime/lustre/work/harmonie/hm_data/...

Current work

- Add extra compute nodes:
  - 30 extra nodes for running IRL25 + IRL10 **overlapped**
  - Testing performance of mixed postp + fcst jobs
- Based on work on fionn:
  - Should saturate test filesystem with:
    - 1km subdomain, 15 minute boundary updates
    - Serial verification workflow
  - Porting other postp tasks (hydrological model) to mix

Thank you!

Thanks to: James Coomer, DDN
Niall Wilson, ICHEC

Launching a profiling run with Darshan

```
/opt/mvapich-gcc/bin/mpirun \
    -envnone \
    -genv IM_CLIENT_DATA_PLACEMENT_TYPE=DETERMINISTIC \
    -genv IM_CLIENT_CFG_FILE=/opt/ddn/ime/config/ime_ichec.config \
    -genv IM_NETWORK_STACK=IM_NETWORK_CCI \
    -genv IM_CLIENT_NUM_IM_SERVERS=4 \
    -genv MV2_ENABLE_AFFINITY=0 \
    -genv OMP_NUM_THREADS=20 \
    -genv DARSHAN_LOG_DIR=$(pwd)/log/darshan \
    -genv DARSHAN_DISABLE_SHARED_REDUCTION=1 \
    -genv LD_PRELOAD=/lustre/ichec/packages/darshan/gcc/2.3.0-debug/lib/libdarshan.so \
    -prepend-rank -f ./mpi.hosts \
```

IME Architecture