WiP: MPI-IO In-Memory Storage with the Kove XPD

Julian M. Kunkel, Eugen Betke

German Climate Computing Center

2016-11-14
Introduction

- Burst-buffers optimize latency/throughput for I/O
- Desired characteristics for I/O:
  - I/O should be able to saturate the (compute node’s) network
  - Achieve network performance with small numbers of threads
  - Little variance (predictable performance) in production
- Kove®’s XPD® offers pooled memory for cluster systems
  - Capacity and performance scales with the number of servers/connections
- Can we leverage the XPD’s memory for MPI-IO?
Approach

- Development of an MPI-IO driver using the Kove KDSA API
  - KDSA is a lower-level API on top of Infiniband libraries
  - Implemented as LD_PRELOAD loadable library for (any) MPI
  - Provides many MPI-IO calls
  - Enabled by the filename prefix “xpd:”, others use the normal MPI

- Limitations of the driver
  - Only partial support of file views
  - Collectives are implemented as independent (actually that is a plus!)
  - No initialization of the storage space
    - Init. is actually not needed if all data is written, e.g., NetCDF/HDF
    - A tool is provided to format the space

- Evaluation
  - Benchmark: unmodified IOR (using MPI-IO backend)
  - System: Cooley visualization cluster of ALCF with 3 XPD’s; total: 14 FDR links
  - Variation of block size, PPN, client nodes, ...
  - Open/close time investigated separately (not discussed here)
Scaling Behavior with the Number of Clients (Using 14 FDR IB Links)

Random performance varying client node count and PPN w/o open/close time. The graph contains fitting curves for 100 KiB and 1 MiB blocks.
Scaling Behavior with XPD Connections

- Varying PPN and the number of XPD (FDR) connections on 14 nodes each

Read performance with variable number of server connections and PPN. Isolines for multiples of 5k MiB/s are shown.
Conclusions

- We introduced an MPI-IO driver for the Kove XPD
- Performance evaluation shows user-friendly performance behavior
  - Good single thread performance
  - Able to achieve nearly network performance (in many cases)
- Recent work (to be published):
  - Support of (typical) file views
  - Evaluation of NetCDF4/HDF5 performance
  - Investigation of performance variance
  - Performance optimization