Benefit of In-Memory Storage for MPI-IO Applications

(Double blind)

(Double blind)

DKRZ DEUTSCHES KLIMARECHENZENTRUM

Abstract

In contrast to disk or flash based storage solutions, throughput and latency of in-memory storage promises to be close to the best performance. Kove®'s XPD® offers pooled memory for cluster systems. For I/O intensive HPC applications, in particular for those with inefficient I/O access pattern, this technology provides a number of benefits.

Our MPI independent file driver enables high-level I/O libraries (HDF5, NetCDF) to utilize the XPD's pooled memory. We evaluate the benefit of this driver for synthetic and for user-relevant workloads.

Contributions of this poster are:

1. Description of I/O capabilities of the XPD
2. Elaboration of benefits for shared file access with MPI-IO and NetCDF

Approach

The developed MPI-IO file driver[a] is selectable at runtime via LD_PRELOAD. It checks the file name for the prefix "xpd:" and routes the accesses otherwise to the underlying MPI. Important MPI-IO functions for HDF5 and IOR are implemented. During the MPI_open/close the Infiniband connections to the XPD's are established/destroyed.

Benchmark tools

- **IOR** is used for benchmarking performance and barriers between the phases are used to synchronize the processes.
- **NetCDF-Bench** mimics behavior of scientific applications from earth-science.

The performance analysis varies the parameters:

- Access granularity: 16 KiB, 100 KByte[b], 1 MiB, 10 MiB
- Processes-per-node (PPN): 1 to 12
- Nodes: 1 to 98
- Connections: 1 to 14
- Access pattern: sequential and random[c]
- File size: 20 GiB per connection [d]

Performance metrics:

- M1. Throughput read/write reported by benchmark tools
- M2. Throughput read/write (computed based on the time for the read/write phase)

Each configuration is run at least three times.

A subset of measurements is run on the Lustre of DKRZ's supercomputer Mistral.

[a] `http://github.com/JulianKunkel/XPD-MPIIO-driver`
[b] Base 10 has been used on purpose as this leads to unaligned access for file systems, i.e., 100 KByte = 10^5 Bytes. All other cases are base 2.
[c] As expected for a DRAM based storage system, they did not show significant differences. Thus, the poster only contains values for random I/O.
[d] The capacity of the XPD is shared among all users.

Overview

Performance of all (7500) conducted IOR runs:

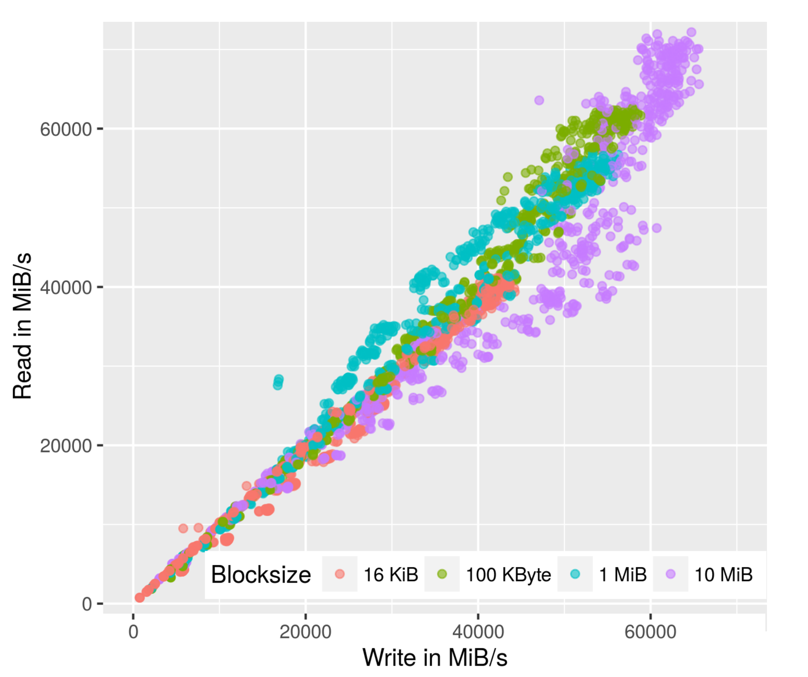


Fig. 1: Observed throughput computed based on the read/write phase (M2.)

Observations:

- Read/write behaves symmetric
  Pearson correlation coef.: 0.969
- Open/close overhead reduces throughput of $M1 \sim= 0.9 \cdot M2$
- Best performance:
  - 65,600 MiB/s (write)
  - 72.200 MiB/s (read)
  - ⇒ 5155 MiB/s per IB FDR link (read)

IOR Performance with Increasing Connections

Understanding the performance behavior when increasing the number of connections reveals scale-out behavior. The test uses always 14 client nodes. Results for reads are shown, write is similar.

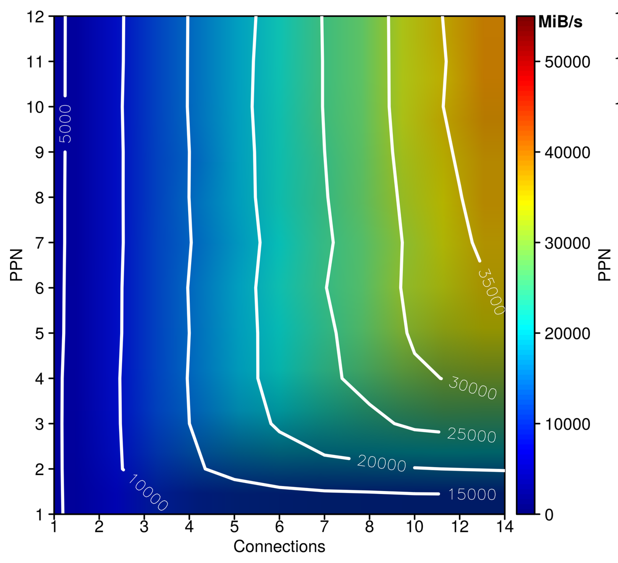


Fig. 2: Granularity: 16 KiB

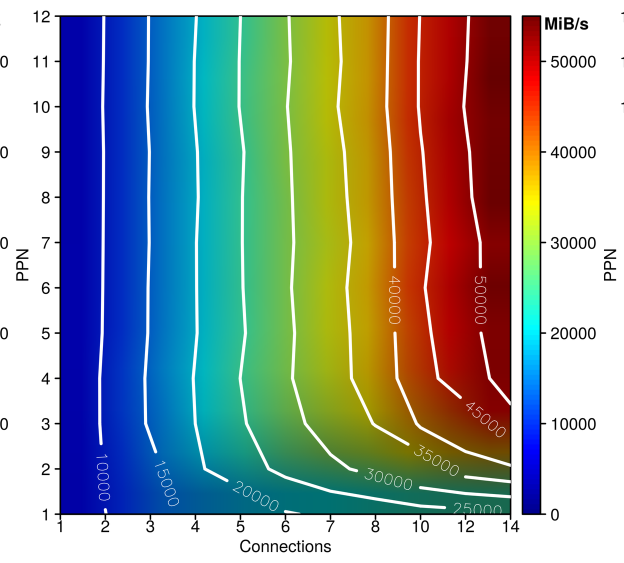


Fig. 3: Granularity: 100 KB

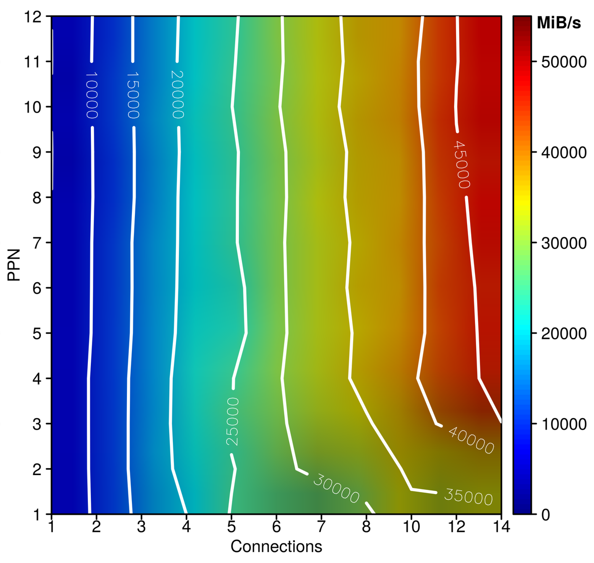
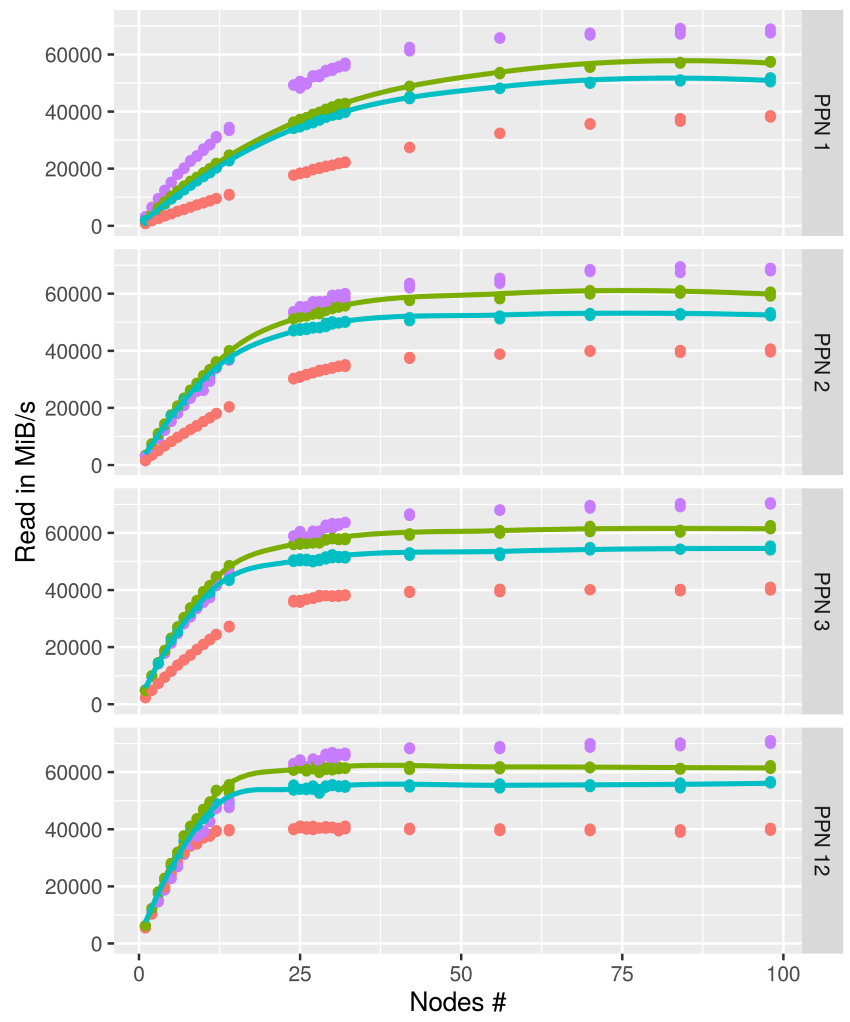


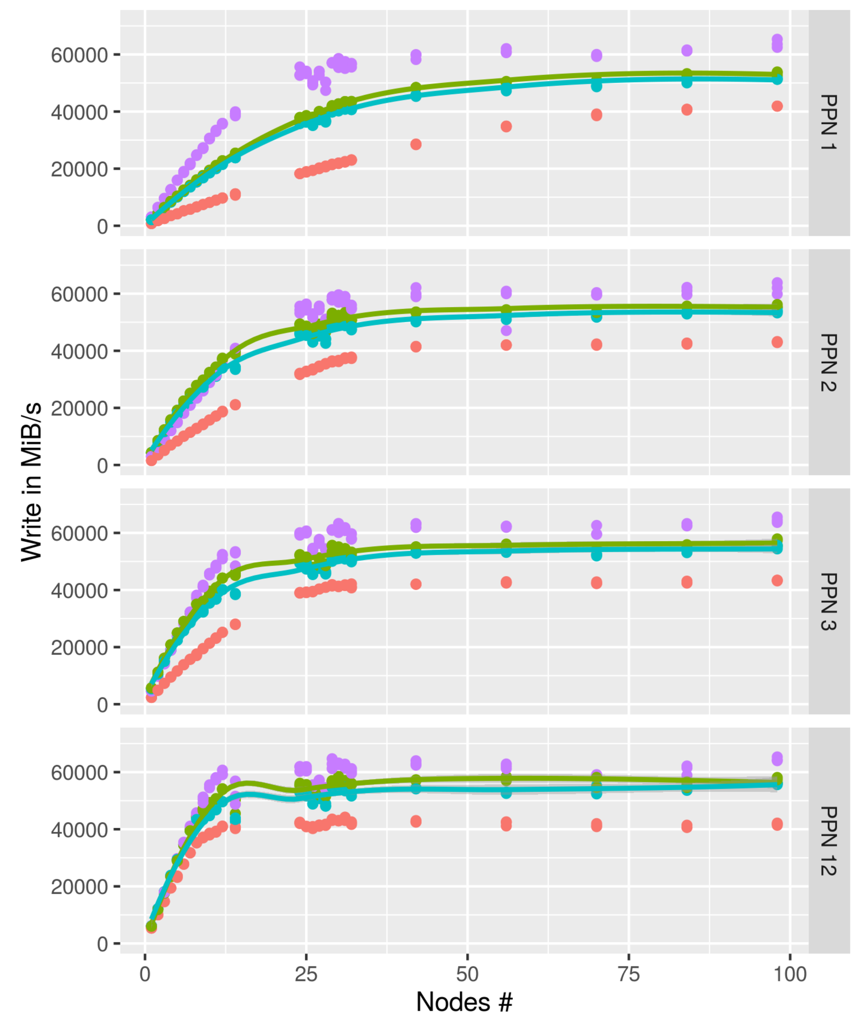
Fig. 4: Granularity: 10 MiB

IOR Scaling Behavior

Results for measuring performance varying blocksize (10 MiB, 1 MiB, 100 KB, 16 KiB), nodes and PPN. Each point on the graph represents a measurement with IOR.



(a) Read



(b) Write

The graphs contain fitting curves for 1 MiB and 100 KB. Graphs for PPN=5 and PPN=8 look similar.

Observations:

- With small block sizes, I/O becomes limited by network latency and CPU speed
- An increase of PPN or client nodes improves overall throughput until hardware is saturated
- Robust scaling behavior, with PPN=12 and 14 client nodes, peak performance is achieved
- Regardless of PPN, with 14 nodes (== 14 IB links), the 14 server links are at > 50% saturated

NetCDF Performance Examples

Results of similar experiments conducted on Cooley's GPFS, Mistral's Lustre and XPDs:

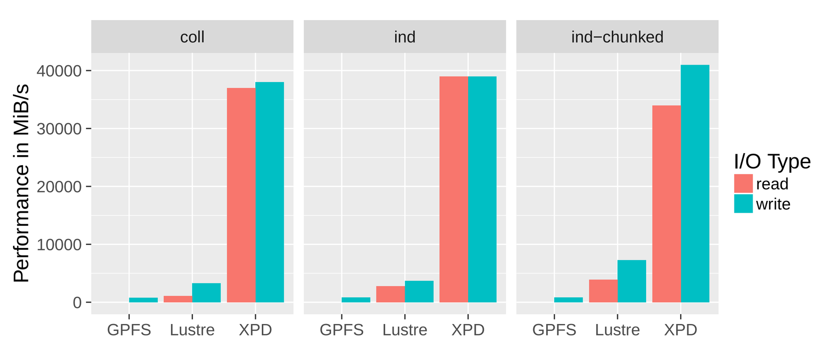
NN	PPN	Type	Write (XPD)	Read (XPD)	Write (GPFS)	Read[a] (GPFS)	Write (Lustre)	Read (Lustre)
1	4	ind	4,500	4,700	290	NA	960	860
2	10	col	11,000	11,000	370	NA	2,000	1,100
5	1	ind	15,000	15,000	690	NA	2,400	2,700
5	4	ind	21,000	22,000	700	NA	4,400	270
5	4	col	20,000	21,000	710	NA	2,500	1,100
5	10	ind	22,000	23,000	610	NA	4,200	5,100
10	10	ind	37,000	40,000	850	NA	7,100	2,900
10	1	ind	27,000	28,000	940	NA	3,600	2,500
20	10	ind	43,000	60,000	210	NA	10,100	9,600
20	1	ind	43,000	43,000	730	NA	3,500	2,900

[a] The values for GPFS read I/O performance were dropped, since they were influenced by page cache.

Collective vs. Independent vs. Chunked

Experiments with different NetCDF I/O modes: collective I/O, independent I/O and NetCDF chunking. The default settings for MPIO on GPFS were used and ROMIO on Lustre was optimized.

Nodes	10
Processes per node	1 (10 if chunked)
Pre-Filling	yes



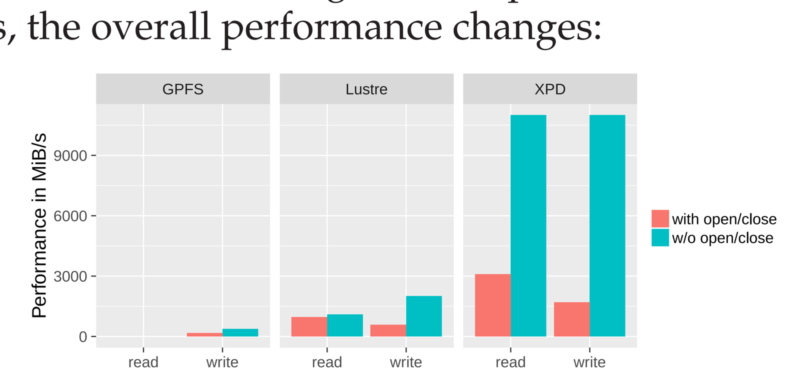
Observations:

- XPDs seem to be insensitive to collective, indendent and independent-chunked I/O, showing always best performance. (Collective-chunked mode is not supported by NetCDF.)

Impact of Open/Close Times

The driver establishes connections to the XPDs which is time consuming in this experiment with rather small data. When considering the open/close times, the overall performance changes:

Nodes	2
Processes per node	1
Test filesize	37.25 GB
XPD connections	14



Observations:

- The open/close time has a large influence. For large files it should not matter.

System Description

The XPD/GPFS test system is Cooley, the visualization cluster of Mira on ALCF:

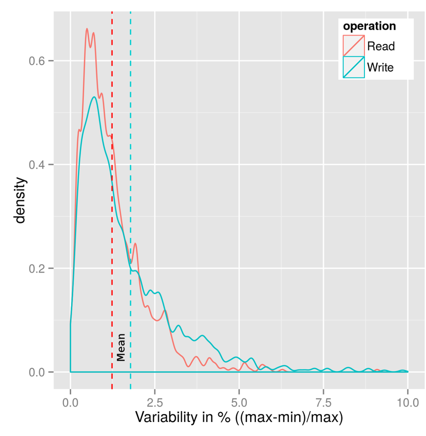
- 126 compute nodes equipped with two 2.4 GHz Haswell E5-2620
- FDR Infiniband
- Kove® XPD® L3
- 3 XPDs with 6+4+4=14 FDR connections

DKRZ's phase2 Lustre system consisting of 68 OSS and 33 PByte of storage capacity. Theoretical peak: 367 GiB/s. Metadata: 210.000 Ops/s

Performance Variability

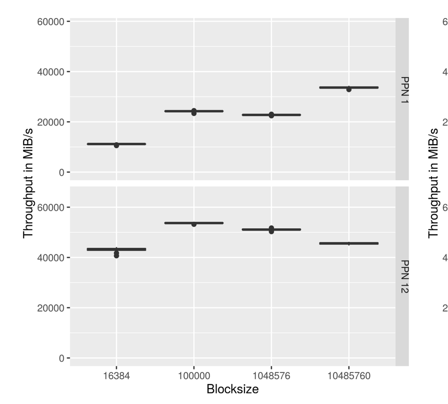
A low performance variability is important for tightly coupled applications.

Fig. 5: Density of the variability range across all conducted experiments (span across three repeats each).

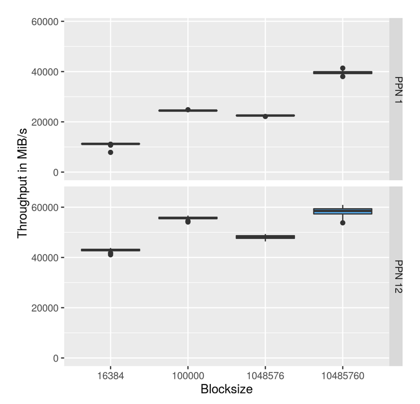


- Mean(read) = 1.23%
- Mean(write) = 1.78%
- 99% of all measurements vary < 10%
- 14 (0.6%) are > 10%

Fig. 6: Boxplots for 100 repeats on 14 nodes



(a) Read



(b) Write

Comparison to Lustre IOR

MPI-IO configuration: Collective I/O was enabled for write access, only for granularities < 512 KiB. One aggregator per node was used. The number of stripes = 2 · number of nodes.

Average speedup (in number of times) of using the XPD vs. Lustre based on random I/O of 2, 4, 8, 14 nodes and 1, 2, 3, 5, 8, 12 PPNs:

	16 KiB	100 KB	1 MiB	10 MiB
write	619	329	10	10
read	887	79	19	15

Best performance is achieved on 14 nodes, 5 PPN, 1 MiB access size:
7493 MiB/s (read), 3659 MiB/s (write)

Observations & Conclusions

- Read performance ≈ write performance
- Random I/O ≈ sequential I/O
- Highly scalable in terms of
  - client nodes
  - number of connections
- Bottlenecks are CPU and network latency
  - in particular for small blocksizes
- Low access time variability
  - read: ≤ 2.5%; write: ≤ 5%
- Insensitive to different I/O modes
  - collective I/O ≈ independent I/O
  - collective I/O ≈ ind.-chunked I/O
- Applications using NetCDF on the XPD can achieve near-optimal network bandwidth
- On GPFS and Lustre, a huge fraction of bandwidth is not utilized
- On XPD, optimizations (MPI-IO hints) can be omitted without affecting the performance
- Open/close times reduce mean performance; for larger files this shall not matter

Future work: We will work towards a full MPI-IO compatible driver to support even further workloads and deal with data migration between XPD and file system.

Acknowledgements

Thanks to Kove for their support and discussion. Thanks to our sponsor William E. Allcock for providing access and feedback. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.