"Expand: High Performance Storage System for HPC and Big Data Environments" (TED2021-131798B-I00)

High Performance Storage Sytems for HPC and Big Data (Expand)

D 3.1
Report on use cases and benchmarks

Universidad Carlos III de Madrid

June, 2025

CONTENTS

1. BENCHMARKS FOR HPC

This chapter describes the benchmarks used to evaluate Expand in HPC environments. The benchmarks used are IOR (Section 1.1), DLIO (Section 1.2), and IO500 (Section 1.3).

1.1. Interleaved or Random (IOR)

Interleaved or Random (IOR) [1] is a prominent parallel I/O benchmark developed by the Lawrence Livermore National Laboratory (LLNL) specifically for the ASC Purple supercomputer.

This benchmark is widely employed to evaluate the I/O performance of parallel and distributed file systems due to its flexibility. IOR enables the simulation of various access patterns, storage configurations, file sizes, and file types. Moreover, it provides multiple I/O interfaces for assessing file systems, including POSIX, HDF5, and Lustre, among others.

Given that IOR performs I/O operations with multiple clients, typically running on different compute nodes in a high-performance computing (HPC) environment, it utilizes MPI for process synchronization.

1.2. Deep Learning I/O (DLIO)

Deep Learning I/O (DLIO) [2] is a benchmark created by the Argonne Leadership Computing Facility (ALCF) to replicate the I/O behavior of various contemporary scientific deep learning applications.

Specifically, DLIO features four distinct deep learning models: BERT [3], a model for natural language processing; CosmoFlow [4], which employs a 3D CNN to explore the universe at scale; ResNet50 [5], designed for 3D image classification; and UNET3D [6], focused on 3D medical image segmentation.

Key features of this benchmark include its configurability through various YAML files that represent the I/O processes of different workflows, offering significant flexibility. Additionally, DLIO incorporates a data generator that facilitates the creation of synthetic datasets for subsequent I/O emulation. Notably, the I/O emulation process is entirely transparent to the user and can mimic the behavior of both sequential and parallel training.

1.3. IO500

The IO500 [7], [8] benchmark has been developed by the community to assess the user experience performance of various file systems. Additionally, it serves as a basis for the IO5000 ranking, which classifies file systems based on their overall performance.

To achieve this, the benchmark incorporates several open-source tools, including IOR [1], MDTest [9], and find [10], each executed with predefined parameters.

2. REAL APPLICATIONS

This chapter will describe the real applications used to evaluate Expand. In particular, EpiGraph (Section 2.1), Nek5000 (Section 2.2), Remote Sensing (Section 2.3), and WaComm++ (Section 2.4) have been used.

2.1. EpiGraph

EpiGraph [11] [12] [13] is a parallel simulator designed for modeling the spread of viruses such as influenza and COVID-19, initially developed by Universidad Carlos III de Madrid (UC3M). Notably, EpiGraph has played a significant role in supporting decision-making for the Spanish Ministry of Health during the COVID-19 pandemic [14].

The architecture of EpiGraph relies on agents that conduct stochastic simulations of the spread of influenza and SARS-CoV-2 across extensive geographical regions. Specifically, EpiGraph utilizes an interconnection network built on individual interactions derived from social networks and demographic data. This network accounts for various individual characteristics (such as work, school, home, leisure activities, etc.) and incorporates a transportation model that simulates viral spread at both regional and national levels. Additionally, the application features an interaction model that considers the effects of climatic factors (such as temperature, atmospheric pressure, and humidity) on the spread of influenza.

EpiGraph is developed using C and MPI, exemplifying a write-intensive application, as it periodically saves checkpoints of 884 MiB during simulations.

2.2. Nek5000

Nek5000 [15] is a prominent open-source fluid dynamics simulation application developed by Argonne National Laboratory (ANL). It has been recognized with the Gordon Bell Prize in 1999 and the R&D 100 Award in 2016.

Nek5000 is characterized by its scalability, efficiency, and accuracy, making it well-suited for solving complex geometric problems, including the Navier-Stokes equations. This application is capable of simulating the behavior of various fluids, with the turbulence use case being particularly resource-intensive during execution. For the evaluations conducted in this work, the TurbPipe use case [16] will be utilized. TurbPipe carries out a numerical simulation of turbulent flow through a pipe.

Developed in Fortran, Nek5000 exemplifies a write-intensive application, as it periodically saves checkpoints and main simulation outputs throughout its operation. This process is crucial because these outputs are later processed to derive the final simulation results, and an increase in stored outputs leads to greater accuracy in these results.

In terms of its access pattern, Nek5000 engages in frequent write operations during execution, primarily due to its practice of periodically storing both simulation outputs and checkpoints.

2.3. Remote Sensing

Remote Sensing [17] is a Deep Learning application developed by the Forschungszentrum Jülich (FZJ). This application specifically trains a multispectral ResNet convolutional neural network (CNN) using the BigEarthNet dataset, which includes RGB and infrared channels.

To achieve this, the application utilizes a remote sensing dataset and classifies a subset of it. The classification problem it addresses is multi-label, meaning that each sample can be associated with more than one label.

In terms of deployment, this Python application can be executed across multiple compute nodes. For this purpose, it employs the Horovod framework, which facilitates the distribution of application processes across several nodes while ensuring their coordination.

Notably, this application serves as a representative example of an intensive data reading application, as its training phase involves a significant number of read operations.

The access pattern of Remote Sensing primarily consists of executing read operations. This is typical for Deep Learning applications, which require access to large volumes of data during the training phase.

2.4. WaComM

WaComM [18], the Water quality COMmunity Model, is a highly scalable, high-performance Lagrangian transport and diffusion model for marine pollutants assessment. WaComM supports CUDA GPU computation, shared memory (OpenMP), distributed memory (MPI) parallelization, and computational malleability with FlexMPI.

The Water Community Model (WaComM) uses a particle-based Lagrangian approach that relies on a tridimensional marine dynamics field produced by Eulerian atmosphere and ocean models. WaComM has been developed to match the hierarchical parallelization design requirements.

WaComM is operatively used for pollutants transport and diffusion at the Center for Monitoring and Modelling Marine and Atmosphere applications (CMMMA) run by the Department of Science and Technologies (DiST) of the University of Naples “Parthenope”.

It is used to compute the transport and diffusion of pollutants for assessing the water quality for mussel farming and fish breeding.

The access pattern of WaComM primarily consists of executing read operations, with a small amount of write operations used for checkpointing. This is typical for computation applications, which require access to large volumes of data during the input of data phase.

3. BENCHMARKS FOR BIG DATA

This chapter will describe the benchmarks used to evaluate Expand in Big Data environments. The benchmarks used are WordCount (Section 3.1), and TeraSort (Section 3.2).

3.1. WordCount

Wordcount [19] is an application that consists of reading a file, counting the words in it and writing the result to another file. The main goal of this application is to evaluate the behavior of a very data intensive application with low memory workload in order to evaluate the performance offered by the file system using Spark.

3.2. TeraSort

TeraSort [20] is an application that obtains data from a file containing key-value pairs. The goal is to sort these pairs and write the result to another file. This benchmark evaluates a more realistic environment where both a high I/O demand and a high memory load coexist. This will allow to know the impact of using Expand in a Spark realistic application.

BIBLIOGRAPHY

[1] IOR, *Ior*, Accessed Feb. 22, 2025. [Online], 2025. [Online]. Available: https://ior.readthedocs.io/en/latest/.

[2] H. Devarajan, H. Zheng, A. Kougkas, X.-H. Sun, and V. Vishwanath, "Dlio: A data-centric benchmark for scientific deep learning applications," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, IEEE, 2021, pp. 81–91.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423/.

[4] A. Mathuriya *et al.*, "Cosmoflow: Using deep learning to learn the universe at scale," in *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2018, pp. 819–829. DOI: 10.1109/SC.2018.00068.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[6] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds., Cham: Springer International Publishing, 2016, pp. 424–432.

[7] J. Kunkel, J. Bent, J. Lofstead, and G. S. Markomanolis, "Establishing the io-500 benchmark," *White Paper*, 2016.

[8] IO500, *Io500*, Accessed Feb. 22, 2025. [Online], 2025. [Online]. Available: https://io500.org/.

[9] MDTest, *Mdtest*, Accessed Feb. 22, 2025. [Online], 2025. [Online]. Available: https://ior.readthedocs.io/en/latest/.

[10] VI4IO, *Parallel find*, Accessed Feb. 22, 2025. [Online], 2025. [Online]. Available: https://github.com/VI4IO/pfind.

[11] EpiGraph, *Simulating covid-19 propagation at large scale*, Accessed Mar. 11, 2025. [Online], 2025. [Online]. Available: https://epigraph.uc3m.es/.

[12] G. Martín, M. Marinescu, D. Singh, and J. Carretero, "Epigraph: A scalable simulation tool for epidemiological studies," *connections*, vol. 8, p. 9, 2011.

[13] A. Cublier Martínez, J. Carretero, and D. E. Singh, "Detailed parallel social modeling for the analysis of covid-19 spread," *The Journal of Supercomputing*, vol. 80, no. 9, pp. 12 408–12 429, 2024.

[14] SINC, *Tres modelos matemáticos ayudan a decidir quién se vacuna primero*, Accessed Mar. 11, 2025. [Online], 2025. [Online]. Available: https://www.agenciasinc.es/Opinion/Tres-modelos-matematicos-ayudan-a-decidir-quien-se-vacuna-primero.

[15] P. Fischer, J. Lottes, and H. Tufo, "Nek5000," Argonne National Lab.(ANL), Argonne, IL (United States), Tech. Rep., 2007.

[16] S. Rezaeiravesh, R. Vinuesa, and P. Schlatter, *A statistics toolbox for turbulent pipe flow in Nek5000*. KTH Royal Institute of Technology, 2019.

[17] R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and J. A. Benediktsson, "Remote sensing big data classification with high performance distributed deep learning," *Remote Sensing*, vol. 11, no. 24, p. 3056, 2019.

[18] R. Montella *et al.*, "A highly scalable high-performance lagrangian transport and diffusion model for marine pollutants assessment," in *2023 31st Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2023, pp. 17–26. DOI: 10.1109/PDP59025.2023.00012.

[19] A. Spark, *Github repository*, 2024. [Online]. Available: https://github.com/apache/spark.

[20] E. Higgs, *Github repository*, 2024. [Online]. Available: https://github.com/ehiggs/spark-terasort.