



HAL
open science

Reproducibility Report for SC25 Paper HP-MDR: High-performance and Portable Data Refactoring and Progressive Retrieval with Advanced GPUs

Philippe Swartvagher

► **To cite this version:**

Philippe Swartvagher. Reproducibility Report for SC25 Paper HP-MDR: High-performance and Portable Data Refactoring and Progressive Retrieval with Advanced GPUs. Inria, LaBRI, Bordeaux INP. 2025, pp.2314-2315. <hal-05363381>

HAL Id: hal-05363381

<https://inria.hal.science/hal-05363381v1>

Submitted on 13 Nov 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Reproducibility Report for SC25 Paper HP-MDR: High-performance and Portable Data Refactoring and Progressive Retrieval with Advanced GPUs

Philippe Swartvagher
Bordeaux INP, Inria
Talence, France
philippe.swartvagher@inria.fr

Abstract

This reproducibility report provides details about the artifact evaluation done with regards to the Artifact Description and Evaluation appendix of SC25 paper HP-MDR: High-performance and Portable Data Refactoring and Progressive Retrieval with Advanced GPUs by Podhorszki et al. The work was done as part of the Reproducibility Initiative of SC25. The author is a member of the SC25 Reproducibility Committee.

Keywords

Artifact Evaluation, Reproducibility

ACM Reference Format:

Philippe Swartvagher. 2025. Reproducibility Report for SC25 Paper HP-MDR: High-performance and Portable Data Refactoring and Progressive Retrieval with Advanced GPUs. In *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC '25)*, November 16–21, 2025, St Louis, MO, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3712285.3769449>

1 Overview of Reproduction of Artifacts

The following table provides an overview of each computational artifact's reproducibility status. Artifact IDs correspond to those in the AD/AE Appendices.

Artifact ID	Available	Functional	Reproduced
A ₁	•	•	◦
Badge awarded	yes	yes	no

2 Reproduction of Computational Artifacts

2.1 Timeline

The artifact evaluation was conducted from August 20, 2025, to September 1, 2025.

2.2 Computational Environment and Resources

The experiments conducted for artifact evaluation were performed on one server of the local cluster, equipped with two AMD EPYC 7352 processors, 32 GB of RAM and a NVIDIA Tesla T4 GPU.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SC '25, St Louis, MO, USA

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1466-5/2025/11

<https://doi.org/10.1145/3712285.3769449>

2.3 Details on Artifact Reproduction

Overall, all artifacts were functional after small adjustments. The rest of this section explains more in details the encountered shortcomings.

All programs are supposed to be executed on several datasets. For the sake of time, only the dataset NYX was used.

Artifact Setup / Building MGARD. Providing a small number of parallel jobs (8 was used) is necessary, otherwise the RAM consumption is too important.

Artifact Setup / Building Multi-Component-Compression-Framework. There is nothing to change in `CMakeLists.txt:20` despite what is requested.

To avoid build issues, precisizing to use GCC 11 was necessary in `build_script.sh:65` to build ZFP GPU:

```
-DCUDA_HOST_COMPILER=/usr/bin/gcc-11
```

```
-DCMAKE_C_COMPILER=/usr/bin/gcc-11
```

```
-DCMAKE_CXX_COMPILER=/usr/bin/g++-11.
```

To use other cloned repositories, one needs to add `-DUSE_EXTERNAL_DIR=ON` to `build_script.sh:84` to the `cmake` call for `Multi-Component-Compression-Framework`.

In `CMakeLists.txt`, the line 43 needs to be replaced by `add_library_search_paths("MGARD/install-cuda-turing")` and the line 50 by `"$EXTERNAL_DIR/MGARD/install-cuda-turing"` (just adding the suffix `install-cuda-turing`, `cuda-turing` being your GPU architecture).

In `CMakeLists.txt`, the line 107 needs to be changed to `NAMES mgard/mdr.hpp`.

Artifact Execution / C1 / Step 4. In the script `BP_Encoder_Perf.sh`, the variable `MdrXExec` needs to contain the absolute path of the binary, instead of the relative path.

One needs to also add the line `export LD_LIBRARY_PATH=/path/to/MGARD/install-cuda-turing/lib/:$LD_LIBRARY_PATH`.

Artifact Execution / C2 / Step 3. In the script `Lossless_Perf.sh`, the variable `MdrXExec` needs to contain the absolute path of the binary, instead of the relative path.

One needs to also add the line `export LD_LIBRARY_PATH=/path/to/MGARD/install-cuda-turing/lib/:$LD_LIBRARY_PATH`.

The execution does not end properly and complains about a `GPUassert: out of memory`.

Artifact Execution / C3 / first Step 1. Same problems than with Step 3 of C2 (absolute path, missing LD_LIBRARY_PATH and GPU memory error).

Artifact Execution / C3 / third Step 1. In run_script_cpu.sh, the value of the variable OMP_NUM_THREADS is hardcoded to 32.

In run_script_gpu_mgard.sh, a LD_LIBRARY_PATH is also missing.

Artifact Execution / C4 / Step 1. The script QoI_NYX.sh can be adapted to avoid downloading again the dataset.

The remaining steps requiring the JHTDB dataset were not done due to the difficulty to download and prepare the dataset. It is probably easy for someone more comfortable with this dataset and this kind of software.

Because of the numerous small adjustments necessary to have working artifacts and the GPU memory error, only the badge *Functional* is granted instead of the requested *Reproduced*.

Disclaimer: This Reproducibility Report was crafted by volunteers with the goal of enhancing reproducibility in our research domain. The time period allocated for the reproducibility analysis was constrained by paper notification deadlines and camera-ready submission dates. Furthermore, the compute hours in the shared infrastructure (e.g., Chameleon Cloud) available to the authors of this report were limited and restricted the scope and quantity of experiments in the review phase. Consequently, the inability to reproduce certain artifacts within this evaluation should not be interpreted as definitive evidence of their irreproducibility. Limitations in the time allocated to this review and the compute resources available to the reviewers may have prevented a positive outcome. Furthermore, reviewers assess the reproducibility of the artifacts provided by the authors; however, they are not accountable for verifying that the artifacts support the main claims of the paper.