Query-Driven Visualization over Compressed Scientific Data via Data–Index Co-Compression

Ning Yan[1], Sheng Di[2], Kai Zhao[3], Hanqi Guo[4], Franck Cappello[2], and Lipeng Wan[1]

[1]Georgia State University, Atlanta, GA, USA
nyan1@student.gsu.edu, lwan@gsu.edu

[2]Argonne National Laboratory, Lemont, IL, USA
sdi1@anl.gov, cappello@mcs.anl.gov

[3]Florida State University, Tallahassee, FL, USA
kai.zhao@fsu.edu

[4]The Ohio State University, Columbus, OH, USA
guo.2154@osu.edu

***Abstract*—Large-scale scientific simulations routinely produce petabyte-scale datasets, making error-bounded lossy compression indispensable for alleviating storage and I/O bottlenecks. However, most visualization pipelines still reconstruct the entire compressed dataset before analysis, even when a visualization query accesses only a small subset of the volume, resulting in substantial computational overhead and limited interactive responsiveness.**

We apply data–index co-compression to enable efficient query-driven visualization, jointly compressing block-level index with scientific data to support selective decompression while minimizing index overhead. During visualization, only the compressed blocks satisfying the query predicate are decompressed, avoiding unnecessary data access and reducing end-to-end visualization cost.

Experimental evaluation on three representative scientific applications demonstrates substantial end-to-end acceleration, achieving up to 50.1× speedup over conventional full-decompression workflows, while preserving comparable visualization quality.

***Index Terms*—Scientific data compression, Query-driven visualization, Selective decompression, Data-index co-compression, Scientific visualization**

I. Introduction

Despite the widespread adoption of error-bounded lossy compression, most visualization pipelines still require direct access to the original data and therefore reconstruct the entire compressed dataset before visualization. This mismatch between full-dataset reconstruction and the fine-grained data access requirements results in substantial computational overhead and limits the responsiveness of interactive visualization.

Many scientific visualization algorithms, however, are inherently query-driven: isosurface extraction, for instance, evaluates only regions whose values intersect a user-specified isovalue, so only a small fraction of the dataset contributes to the result. Efficient query-driven visualization therefore relies on selective decompression, whose central challenge is locating candidate blocks without scanning the entire dataset.

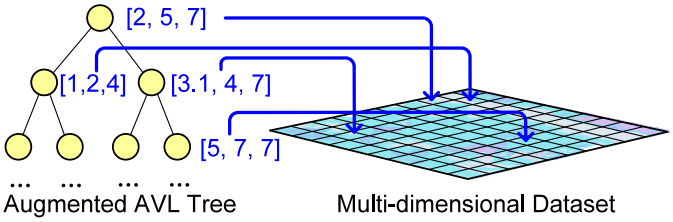


Fig. 1. Illustration of index based on augmented AVL tree.

TABLE I
Average compression ratio with and without index

Dataset	Without Index	With Index	CR Drop (%)
NYX	9.32	5.97	35.9
WarpX	36.91	11.84	67.9
Gray-Scott	9.29	6.02	35.2

Auxiliary indexing structures effectively support candidate lookup but introduce considerable storage overhead, so reducing this overhead while preserving query efficiency becomes essential. To address this challenge, we apply data–index co-compression to enable efficient query-driven visualization over compressed scientific data.

The framework combines an augmented block-level index for efficient candidate lookup with joint compression of both scientific data and index, enabling efficient selective decompression while preserving overall compression performance. Experiments show speedups of up to 50.1× over full-decompression workflows while preserving visualization quality.

II. Motivation

Query-driven visualization requires fine-grained access to only the data relevant to a query. To support this, the dataset is partitioned into independent small blocks, each recording

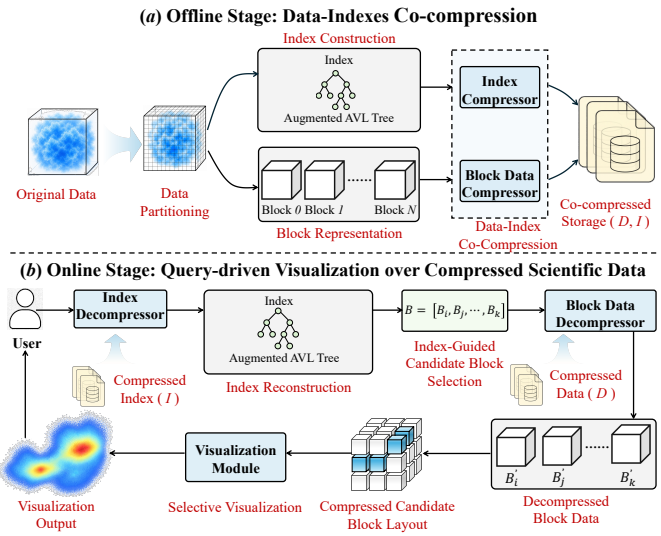


Fig. 2. Overview of the proposed data–index co-compression framework.

its value range. These ranges are organized into an augmented AVL tree mapping value ranges to spatial regions, as illustrated in Fig. 1. A query traverses the tree to identify only overlapping blocks, avoiding a full scan; only the identified blocks are then decompressed and processed, substantially reducing end-to-end cost.

Although the block-level index substantially accelerates query-driven visualization, storing the index introduces considerable storage overhead. As shown in Table I, adding the index reduces the compression ratio by 35.2%–67.9% across representative scientific datasets. Reducing this overhead while preserving efficient query processing is therefore the key challenge.

III. Methodology

The proposed framework consists of two stages: offline preprocessing and online query-driven visualization, as illustrated in Fig. 2.

During the offline stage, the scientific dataset is first partitioned into independent blocks, from which block-level value ranges are extracted to construct an augmented AVL-tree index. The block-level index is then jointly compressed with the scientific data to produce a unified compressed representation, reducing metadata redundancy while preserving efficient query processing.

During the online stage, the compressed index is first reconstructed to locate candidate blocks. Only the identified blocks are decompressed for subsequent visualization, while all other blocks remain compressed. By avoiding unnecessary decompression and visualization operations, the proposed framework significantly reduces end-to-end visualization latency.

IV. Experimental Results

We evaluate the framework on three representative scientific datasets, as summarized in Table II. Table III reports the compression ratios achieved by the co-compression framework, including data, index, and overall ratios. The overall

TABLE II
Real-world HPC datasets used in our evaluation.

Dataset	Domain	Dimensions	Fields	Size
NYX	Cosmology	2048^3	6	206.16 GB
WarpX	Plasma Physics	1024^3	6	4.69 TB
Gray-Scott	Reaction-Diffusion	1024^3	2	1.56 TB

TABLE III
Average compression ratio for data, index, and overall

Dataset	Data Avg	Index Avg	Overall Avg
NYX	9.32	4.33	8.61
WarpX	36.91	9.72	31.67
Gray-Scott	9.29	8.32	9.14

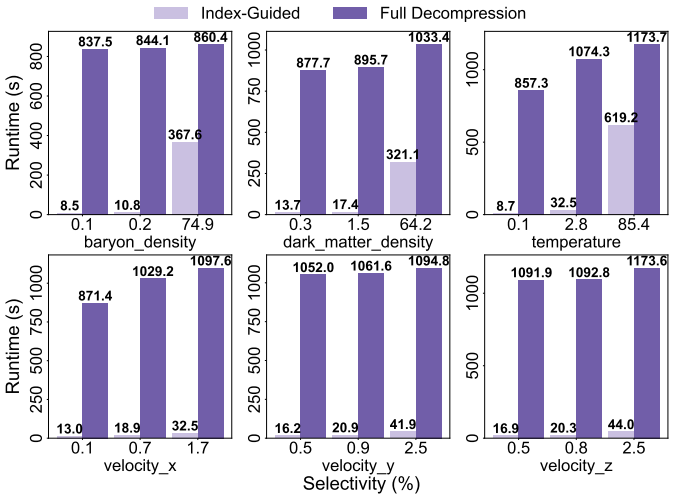


Fig. 3. Isosurface extraction runtime on NYX with and without index guidance under different isovalues.

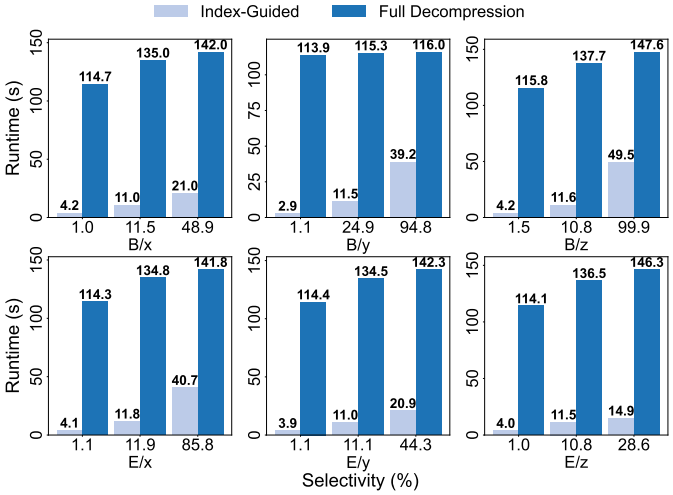


Fig. 4. Isosurface extraction runtime on WarpX with and without index guidance under different isovalues.

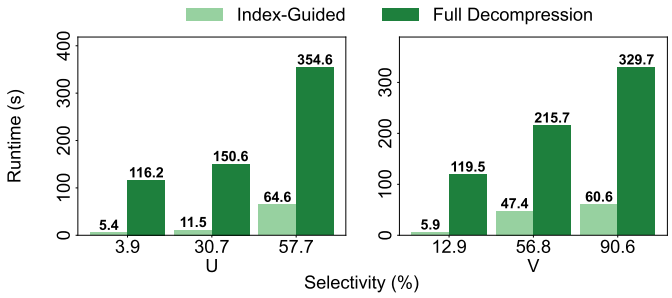


Fig. 5. Isosurface extraction runtime on Gray-Scott with and without index guidance under different isovalues.

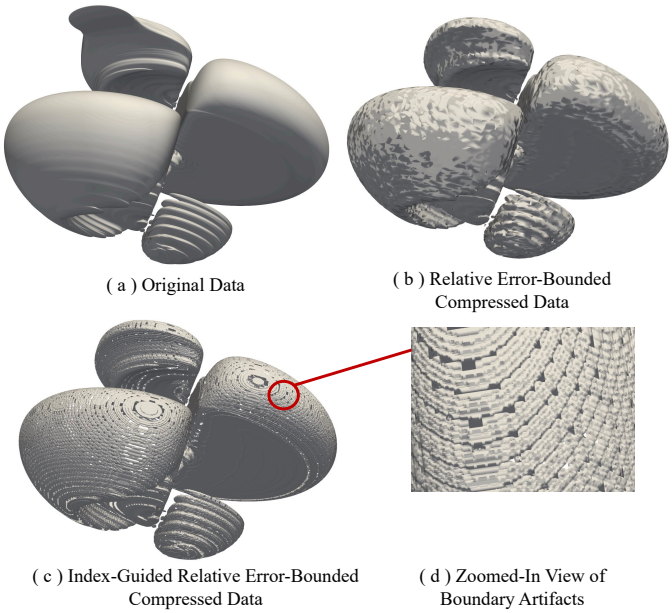


Fig. 6. Isosurface Comparison for the B/y Variable in WarpX.

compression ratio remains close to the data ratio across all three datasets, demonstrating that co-compression substantially reduces indexing overhead.

We evaluate the framework using isosurface extraction as the visualization workload on three representative datasets: NYX, WarpX, and Gray–Scott, comparing it with the conventional full-decompression workflow. The proposed method achieves average end-to-end speedups of 50.1×, 15.7×, and 11.7×, respectively, as shown in Fig. 3, Fig. 4, and Fig. 5. The speedup increases as query selectivity decreases because fewer blocks require processing.

Figure 6 compares the isosurfaces extracted from the original data, compressed data, and the index-guided compressed data. The index-guided results preserve the overall geometric structures of the original isosurfaces, with boundary artifacts visible near block boundaries.

V. Conclusion and Future Work

This work applies the data–index co-compression framework to query-driven visualization over compressed scientific data, achieving substantial end-to-end speedups while minimizing index storage overhead and preserving the overall geometric structure of extracted isosurfaces, albeit with boundary artifacts.

Future work will investigate boundary-aware compression and reconstruction techniques to improve cross-block isosurface continuity while preserving query-driven efficiency.

Acknowledgment

This research was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR), under Contract No. DE-AC02-06CH11357. Additional support was provided by the National Science Foundation under Grant Nos. 2531903, 2311875, 2344717, and 2514036.