Remote Visual Analysis of Large Turbulence Databases at Multiple Scales

Jesus Pulido[a,b,*], Daniel Livescu[b], Kalin Kanov[c], Randal Burns[c], Curtis Canada[b], James Ahrens[b], Bernd Hamann[a]

[a]*Department of Computer Science, University of California, Davis, CA 95616, USA*
[b]*Los Alamos National Laboratory, Los Alamos, NM 87544, USA*
[c]*Johns Hopkins University, Baltimore, MD 21218, USA*

Abstract

The remote analysis and visualization of raw large turbulence datasets is challenging. Current accurate direct numerical simulations (DNS) of turbulent flows generate datasets with billions of points per time-step and several thousand time-steps per simulation. Until recently, the analysis and visualization of such datasets was restricted to scientists with access to large supercomputers. The public Johns Hopkins Turbulence database simplifies access to multi-terabyte turbulence datasets and facilitates the computation of statistics and extraction of features through the use of commodity hardware. We present a framework designed around wavelet-based compression for high-speed visualization of large datasets and methods supporting multi-resolution analysis of turbulence. By integrating common technologies, this framework enables remote access to tools available on supercomputers and over 230 terabytes of DNS data over the Web. The database toolset is expanded by providing access to exploratory data analysis tools, such as wavelet decomposition capabilities and coherent feature extraction.

Keywords: Databases, Wavelets, Data Reduction, Remote Visualization, Distributed Systems, Turbulence

---

*Corresponding author

Email addresses: `jpulido@ucdavis.edu, Ph.+1 (209) 261-2674` (Jesus Pulido), `livescu@lanl.gov` (Daniel Livescu), `kalin@cs.jhu.edu` (Kalin Kanov), `randal@cs.jhu.edu` (Randal Burns), `cvc@lanl.gov` (Curtis Canada), `ahrens@lanl.gov` (James Ahrens), `hamann@cs.ucdavis.edu` (Bernd Hamann)

1. Introduction

Extremely large datasets commonly arise in science and engineering today, and it is often prohibitive to store an original massive dataset at multiple sites or transmit it over computer networks in its entirety. Regardless, such datasets represented tremendous scientific value for the broader scientific community. It is imperative to deploy effective technologies enabling the remote access to vast data archives for the purpose of having a large pool of scientists harness their value and make new discoveries. Our analysis framework presented here was driven specifically by the needs articulated by scientists from Johns Hopkins University (JHU) and Los Alamos National Laboratory. JHU hosts a large digital repository of data from several disciplines, including massive simulation data from numerical simulations of turbulent physics. The framework we present makes remote visual analysis possible via an effective protocol controlling the distribution of analysis and visualization steps to be performed on the server side (JHU site) and a remotely connected visualization client.

Furthermore, our framework incorporates the power of approximating a dataset by using cubic (bi-cubic, tri-cubic) B-spline wavelets. The utilization of wavelet approximation allows a user to generate initial previews or simply coarse approximations of a dataset quickly, making possible the efficient identification of specific regions of interest that might warrant an analysis at a more detailed level. We demonstrate our framework for datasets available in the JHU repository and for typical scientific analysis scenarios.

Most datasets encountered in applications in the physical sciences, similar to most natural images, present lower-dimensional structures whose detection, extraction, and characterization are active areas of research. The search for more efficient algorithms to detect and manipulate such structures has led to the development of a multitude of multi-resolution geometric methods supporting data analysis at multiple scales. One area that significant benefits from the application of such methods is fluid turbulence. In general, turbulent flows contain localized, highly intermittent structures as well as more stable, coherent structures. The characterization of these structures, which interact nonlinearly as they are advected by the background flow and significantly alter the local topology, is an open, fundamental question in the study of turbulence. Multi-resolution representation methods seem ideally suited for such an effort due to their localization in both the real and frequency spaces.

Recently, Pulido et al. [1], studied several multi-resolution representation methods in depth, including B-spline, Daubechies and Coiflet wavelets, curvelets and surfacelets. These methods were compared to determine their ability to capture the structure of fully developed turbulence using a truncated set of coefficients. Methods were evaluated based on their ability to approximate scalar and vector fields, including density and velocity, as well as derived data such as derivatives, spectra, and properties of constant density surfaces. The main criteria used for comparing methods were computational efficiency, numerical accuracy, and degree of data compression. While different methods performed better with respect to various metrics, B-spline wavelets consistently ranked at or near the top of the metrics considered. Except for some of the orthogonal wavelet methods, the use of multi-resolution representation techniques to study turbulence is relatively new and an emerging area of research. In particular, the B-spline wavelets have only been sparsely used for such purpose. Here, we present the addition of the B-spline wavelet representation to the JHU turbulence database. By overcoming bottlenecks in the system, we demonstrate this tool for both remote visualization and novel multi-scale analysis of turbulence data.

1.1. Contributions

This paper contains the following contributions:

- Wavelet compression is introduced at the data-level to reduce access costs, bandwidth, memory and compute footprint, therefore improving latency between database components to support many remote users.

- Remote visualization is made possible for a multi-terabyte database cluster supporting commodity hardware.

- New analysis tools are demonstrated for two datasets for these types of turbulence: a) Homogeneous Buoyancy Driven Turbulence (HBDT) [2] and b) Forced Magneto-Hydrodynamic turbulence (FMHDT) [3].

In Sections 2 and 3, we discuss related works in remote visualization and wavelet compression, describe the public JHU database cluster, and provide a brief introduction to wavelet methods. Section 4 presents a pipeline that introduces wavelet methods and visualization support to the JHU database cluster. Section 5 discusses results focused on measures assessing the performance, quality, and efficiency of using wavelet methods for analysis. Additionally, multi-scale analysis is performed, driven by domain scientists, and results are presented using the newly implemented analysis tools. Our conclusions in Section 6 summarize our results and contributions.

2. Related Work

With the continued and rapid growth of the size of datasets, movement of data is increasingly difficult. Database systems, such as the Johns Hopkins Turbulence Database (JHTDB) [4], provide public access via Web services to large datasets in a database cluster. Efficient and remote Visualization and analysis capabilities are crucially important for understanding such large-scale simulated datasets, to gain the desired value and scientific insight from the available data.

Compression for reducing bottleneck behavior in systems for data visualization purposes has been explored previously. Classic methods, such as Haar wavelets, are used in Computational Fluid Dynamics (CFD) for data transmission in Trott et al. [5]. Lippert et al. [6] applied wavelet splats, permitting lossy compression, to enable volume rendering of large datasets. Concerning large-scale and remote data visualization, Guthe et al. [7] proposed the use of hierarchical wavelets in a preprocessing step to reduce hardware requirements for volume visualization applications. While this technique reduces the cost of visualization on standard PCs, interactive data walkthrough leads to rendering times exceeding 1000ms per frame, unless wavelet coefficients are cached. Woodring et al. [8] used a commercial wavelet standard, Jpeg2000, to perform compression over the network. Lakshminarasimhan et al. [9] described a new error-bound, B-spline-fitting-based method for lossy data compression and performed a direct comparison with wavelets. When comparing the method to traditional Haar and other linear wavelet methods, it has been shown to perform rather poorly for data analysis and visualization purposes [1]. Lindstrom et al. [10] developed a fixed-rate, near-lossless compression scheme targeting floating-point gridded data. While these compression schemes are useful for data storage purposes and transfer over networks, they do not enable the analysis of data for a band-/frequency-specific analysis, which is of great interest in the context of a scale-based analysis of phenomena exhibiting behavior at multiple scales. In our framework, we use cubic B-spline wavelets that significantly improve over linear wavelets, then only stream the coefficients required for a specific scale, without decompression. This approach supports fast adaptive refinement of coefficients to improve the level of detail of an approximation when requested.

Several visualization frameworks and systems have been developed over the past years. Ahrens et al. [11] devised Paraview, which is the underlying system we used for our efforts discussed in this paper. Childs et al. [12] developed another system, called Visit. Both systems are built on the Visualization Tool Kit (VTK) [13] and provide a most of the commonly used and needed data visualization methods. Cedilnik et al. [14] presented several remote visualization schemes and implemented them in Paraview. These schemes are based on streaming compressed images and geometrical data over a network, but they do not address the problem of processing large amounts of data at the location where the data physically reside. In our framework, we compress data at the database level and augment existing features of Paraview, including decompression capability to represent reduced datasets supporting concurrent users in a single node.

Finally, [15] provides an overview of visualization of large turbulence datasets, proposing the use of a GPU-based wavelet methods for data compression. Local large data can be processed quickly on a Desktop client's GPU by using a bricking scheme that reduces workloads into smaller compute blocks. Remote visualization, however, has to contend with the existing server limitations as it is not feasible to transfer all the data locally. Additionally, the existing JHTDB framework is limited by monetary cost to the usage of headless nodes without a GPU. Our wavelet approach is CPU-based by design, and many times does not require a full wavelet computation of the domain. If a user requests a subset of a dataset, only that portion is compressed and cached rather than the entire domain as does a comparable bricking scheme. Once computed or recalled from cache, we can directly visualize the raw, compressed coefficients without the need of a full decompression (reconstruction) of the data. With this process, our approach greatly reduces processing requirements for scientific discovery as it allows multiple users to explore datasets in a single node. The bricking scheme proposed in [15] may still be useful for future larger datasets, as they become memory bound.

Several elegant techniques for the analysis of turbulent CFD datasets were recently developed. For example, Roussel et al. [16] discussed the use of biorthogonal (constant) Harten wavelets for the extraction of coherent vortices and found that biorthogonal wavelets can cause problems when modeling background noise after feature extraction. Laney et al. [17] developed an approach for analyzing Rayleigh-Taylor instabilities using the Morse complex and visualizing features based on it. Finally, Bremer et al. [18] introduced a method that employs a hierarchical segmentation method to analyze and track isosurfaces in time-dependent datasets. While computationally expensive for large datasets, the flexibility offered by our framework makes it possible the implementation for feature-sets beyond that of the JHTDB, for reduced sets, or for datasets with high temporal fidelity.

3. Background

In this section, we give an overview of a public database framework and architecture available for remote data analysis. This section also introduces the wavelet techniques used for the proposed architecture.

Accurate simulations of turbulent flows require solving all the dynamically relevant scales of motions. This technique is usually referred to as direct numerical simulation (DNS). Since turbulence is a strongly multi-scale phenomenon with a large range of dynamically relevant spatio-temporal scales, such computations are restricted to relatively simple flows and require the use of the largest supercomputers available. The pace and scale at which such simulations are performed only continues to increase; consequently, the simulations themselves are restricted to a small number of groups worldwide with access to large computational platforms. In addition, the large databases generated make the analysis, visualization and sharing of the results extremely challenging in their own rights. Even transferring the data resulting from these simulations from archive systems to the local scratch for further post processing is becoming very time consuming. Yet the petabytes of turbulence data, spanning a rapidly increasing range of flows, each with many different parameters, offer almost limitless information on many different aspects of the flow, from an infinite mathematically) hierarchy of turbulence moments and their Probability Density

Functions (PDF), spectra, and correlations, to structure-functions, geometrical properties, etc. The ability to share such datasets with other groups in the open science community can significantly reduce the time to analyze the data (currently measured in years), help the creative process and increase the pace of discovery and, ultimately, advance our knowledge and ability to model turbulent flows.

3.1. Johns Hopkins Turbulence Database

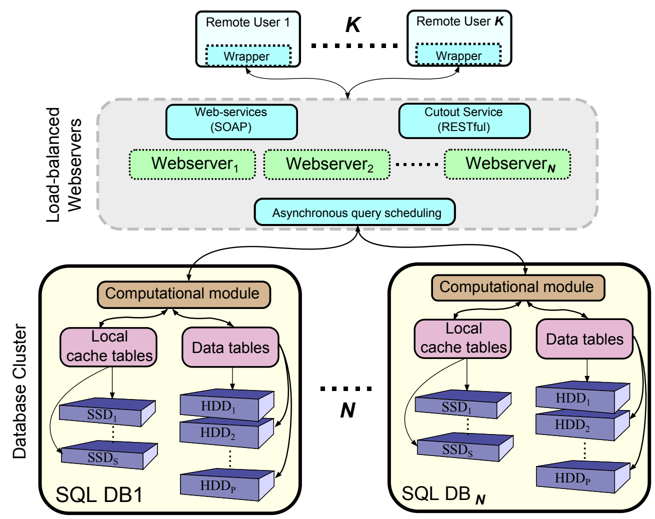

Figure 1: Architecture overview of the JHTDB. Multiple **N** server nodes create a database cluster to serve **K** users. The web services module provides an API where remote users may place requests to the database.

The Johns Hopkins Turbulence Database (JHTDB) is a public database system used for storing DNS datasets. Unlike other public databases, JHTDB also provides several remote tools that facilitate the analysis and retrieval of turbulence data [4]. The DNS datasets available are: a) forced isotropic turbulence, b) incompressible magnetohydrodynamic (MHD) turbulence [3], c) forced, fully developed turbulent channel flow [19], and d) homogeneous buoyancy driven turbulence [2]. The datasets consist of files with the values of the primary variables (e.g. velocity vector components, density, pressure etc.) specified at 1024^3 spatial points and multiple time-samples (up to 1024). Planned for the future are datasets as large as 4096^3. The complete space-time history of turbulence is currently accessible to users remotely through an interface that is based on a Web-services model. Users may write and execute analysis programs on their host computers, while the programs make calls that request desired parts of the data over the network. Users now are able to remotely calculate various statistical data by accessing the 230 Terabytes of DNS data using regular platforms such as laptops. Fig. 1 gives a brief overview of the JHTDB architecture used to support a database. Data are partitioned spatially and temporally across the cluster and accessed through a database access server hosting a Web services module. This module allows for scheduling and divides user requests to according to the partitioning of the data. Remote users interact with the database through SOAP and RESTful web protocols via wrappers in multiple languages such as Matlab, Python, C and Fortran.

The architecture of the database is explained in detail, including descriptions for locally defined functions such as differentiation and interpolation, by Li et al. [20]. In this paper, test calculations are performed to illustrate the usage of the system and to verify the accuracy of the methods in a parallel environment. The database is then used to analyze a dynamical model for small-scale intermittency in turbulence to show that these effects differ considerably among themselves and thus require different modeling strategies in Lagrangian models of velocity increments and intermittency.

Although a variety of remote analysis tools are publicly available through this cluster, there is currently no support for scalable remote visualization. In addition, the tools we have added to enable such support can also be used for novel turbulence analysis, including scale decomposition and coherent feature extraction.

3.2. B-splines and Wavelets

Wavelets are the generalization of the Fourier transform by using bases that represent both location and spatial frequency [21]. Previously, Farge [22] performed an initial analysis on the use of wavelets, for turbulence to characterize coherent and incoherent flow parts.

In a more recent effort by Pulido et al. [1], several multi-resolution representation methods were compared, including higher-order B-spline wavelets, for their ability to capture a broad range of quantities pertaining to the turbulence structure with a reduced set of coefficients. Biorthogonal B-spline wavelets have compact support and the added advantage that their bases functions can be specified analytically. The wavelets used in this analysis are mostly second-generation wavelets in implementation [23].

After a detailed analysis of multiple wavelets, Pulido et al. [1] showed that the higher-order B-spline families consistently ranked amongst the top for the metrics considered. Wavelets of lower-order (first, second) were not able to represent an original or derivative signal well and a wavelet of too high-order (sixth, seventh) introduced oscillations into the data. Based on this work, we selected cubic B-spline wavelets due to their overall very good performance for compression and analysis. By considering a sinusoidal signal instead of a typical binary, a wavelet transform is able to capture superior directionality of a dataset. Nevertheless, the wavelet framework presented in this paper has support for many other wavelet degrees and families such as Daubechies. The discrete versions of these signals are used in form of filters as described by [24].

4. Method

The ability to efficiently explore the large datasets stored in the JHTDB require that remote data exploration and visualization techniques are supported by proper multi-resolution technology. For this purpose, we decided to integrate wavelet methods as a module that is tightly integrated with the database cluster itself. Our integrated wavelet module supports the efficient decomposition and reconstruction algorithms needed for the requirements of a remote, real-time-driven data exploration framework using long-distance computer networks. Our wavelet implementation further supports various ways to effectively select and combine those coefficients of a wavelet decomposition that enable data-specific and user-specific reconstruction of data, emphasizing, for example, frequency and band-specific utilization of coefficients for data analysis. In summary, our coupling of wavelet technology and the JHTDB now make it possible to browse massive datasets much more efficiently than before, thereby greatly increasing the pool of scientific users of the JHTDB.

The following section describes the current and augmented pipeline proposed in this paper. A diagram with the high-level structure of the system can be seen in Fig. 2. There are two primary augmentations made to the existing pipeline. The first is the addition of wavelet support to the database cluster as a computational module. The second is the addition of a new type of node, a visualization node that hosts a visualization server and directly communicates with the database cluster.

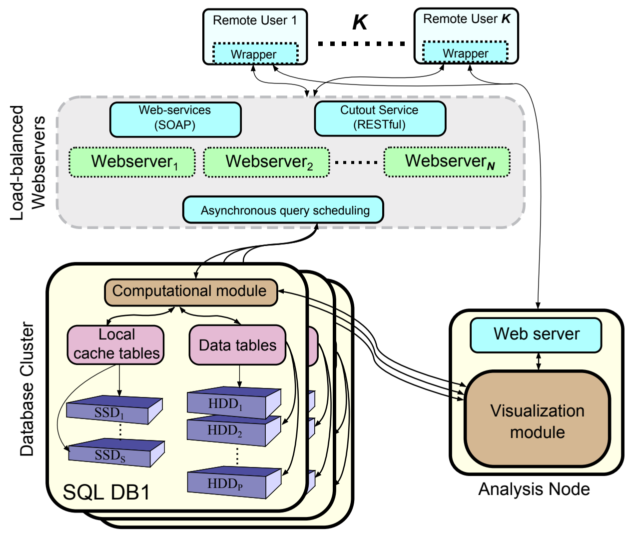

Figure 2: Updated architecture overview for the JHTDB. With wavelet support implemented on both the database cluster and the analysis node, many concurrent users are able to visualize stored datasets using the JHTDB and a visualization node with commodity hardware.

4.1. Wavelet Integration

Wavelet support is added directly to the existing compute capabilities of the JHTDB. Designed as a new database cluster's computational module, wavelet decomposition, reconstruction, and coefficient manipulation capabilities are added to the cluster itself and to the proposed visualization module.

The wavelet CPU implementation is based on the GNU Scientific Library (GSL) [25]. A large number of modifications were made to the open source library. We enabled support for 3D wavelet decomposition, higher-order B-spline basis functions, border effects, support for odd and non-power-of-two-resolution datasets, and parallel computation. Since the wavelets available in the GSL are discrete, filter-based, it is possible to consider additional discrete wavelets to be added to existing ones.

A wavelet operation (decomposition) is typically carried out when a dataset is first requested, prompting the computation of the wavelet coefficients of the entire dataset. Coefficients are cached for future requests of the same dataset. Once the coefficients are computed, there are several distinct methods for manipulating the coefficients involving the selection of specific or a series of coefficients. This selection is typically known as hard thresholding. Unnecessary coefficients do not need to be sent over the network. The thresholding process is a linear operation that can be performed efficiently. Several thresholding methods exist that allow a user to construct and explore a reconstruction, or a specific feature-set of a large dataset. Some of these methods are included as available options and described below.

A key benefit of using wavelets for data reduction rather than sampling data at lower resolutions is their ability to refine lower-resolution data representations into higher quality with low effort. The first and default method of coefficient selection involves the preservation of coefficients up to a certain scale. When supporting many concurrent users accessing data and performing analysis over a network, this method is used to reduce bandwidth to the visualization node and memory requirements. This selection results in data compressed up to several resolutions at varying quality levels, where larger-scale, dominant features are preserved at the loss of small-scale features. As an example, a 1024^3 grid dataset can be decomposed into nine resolutions/scales, where scale one contains the coefficients with the smallest (finest) features and scale nine the largest (strongest) features. In the scenario where few users are present, only the missing decomposition scales will be streamed to improve the data quality rather than having to resample the entire dataset again at a resolution closer to the original. Including all scales will result in a lossless reconstruction of the original dataset. This thresholding scheme is tested in Sections 5.1 , 5.2 and 5.3.

The second approach to coefficient selection is the isolation of specific bands, opposed to the accumulation, in order to extract structures that may exist in specific scales. This analysis-based approach that isolates structures by selecting specific scales is explored in Section 5.5. When coefficients are requested at a specific scale, all coefficients excluding the pure lowpass subset are computed and transmitted to the user through the web services or visualization nodes. The advantage of using this method for scale-based analysis is the added capability of detecting features that might not exist in other scales.

The third and more costly thresholding method sorts all wavelet coefficients by magnitude. A percentage of this ordered set of coefficients can be requested by the user for a reconstruction. Coherent and incoherent feature extraction is also possible through this method and is discussed in [1, 16]. A parallel implementation of the quick sort sorting algorithm was added to the GSL to allow this type of analysis.

A companion reconstruction framework is made available to the analysis node in order to correctly refine additional coefficients for data visualization. In practice, reconstruction is not required on the visualization node, further reducing the latency. With their spatial coherence to the data, coefficients themselves can be visualized after a simple scaling operation. Data reduction is achieved by utilizing subsets of coefficients smaller than the size of a full-resolution dataset. Bandwidth and compute are therefore reduced between the communication pipeline of the database cluster and visualization node allowing the node to exist outside of the cluster.

When a user makes a request to the database cluster, coefficients not in cache or stored already are computed through an on-the-fly data decomposition. After being cached, the coefficients are transmitted to the visualization node where they are used to perform reconstruction on-the-fly if refinement is needed or used directly for visualization.

4.2. Visualization Node

The primary visualization tool used in this node is Paraview [26]. Paraview is an open source visualization software that can be adapted to many architectures and visualization applications. The extensive feature list Paraview provides, including visualization and analysis tools, makes it an ideal companion for a large database cluster.

Modifications were made to the Paraview source to allow communication with the JHTDB to access both raw data or derived wavelet coefficients from the cluster. Natively, the tool does not support data-level wavelet compression, therefore, our heavily customized GSL was adapted to provide Paraview with full support. The purpose of data-level compression is to significantly reduce the cost of analysis for large datasets while preserving relevant features and allowing a large number of concurrent users per visualization node. As an example, a full resolution 1024^3 grid at floating-point precision can consume a minimum of 4 gigabytes (GB) of RAM for a simple full-scale analysis. By using wavelet compression at the first scale, the memory requirement can be directly reduced by 1/8th in size without significant loss of fidelity. On a commodity-based visualization node with 64GB of RAM, a 10 concurrent-user scenario can immediately be turned into one supporting 100+ concurrent users, and beyond with additional compression.

The data analysis tools provided by Paraview also expand the computational capabilities of the JHTDB. A framework for user-driven analysis removes the restriction of only conforming to the functions provided by the database cluster. Instead, a user can use a large set of mathematical operators through Paraview to manipulate data remotely.

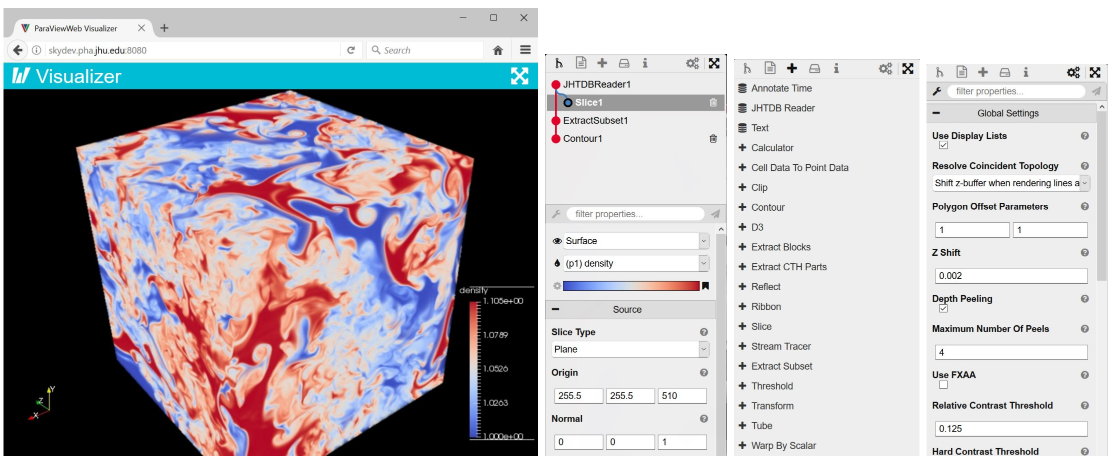

Figure 3: Remote visualization web interface (left) and sample UI elements (center,right). A sample view of the Paraview Web interface shows how many of the core paraview features are made available over the web. As an example, a large turbulence dataset (left) can be volume visualized and made viewable on any light-weight web-browser client.

Besides its analysis tools, Paraview provides several remote visualization capabilities through a direct client-server connection or a web interface. The web module (Paraview Web) allows remote users to access database datasets through a web browser in real-time. Paraview has compression capabilities already implemented at the image rendering level as well as some support for geometry processing [14]. Thus, it is possible to render data on the server itself and transmit a lower-quality image to a user in low-bandwidth remote settings. Interactive rendering framerates between a user's local web client and the remote visualization server are mainly limited by their physical distance, impacting round-trip time over the Internet. If a dataset is small enough, with wavelet compression enabled, geometry data needed to render iso-surfaces, for example, can be transmitted directly to a remote user. Geometry can be rendered locally in our web client with a modern web browser that supports WebGL for higher framerates. As seen in Fig. 3, we support an interface that is similar to that of a standard Paraview desktop interface. A subset of the most frequently used Paraview tools that can be generalized to most datasets is available over the web, reducing time and resources needed to analyze and visualize large datasets in spatial and temporal space. We view the following operators as the most commonly used: Calculator, Contour, Clip, Slice, Volume, Threshold, Extract subset, Glyph, and Stream tracer.

Our web interface significantly reduces the overall effort required to perform large-scale data analysis when no additional software must be installed by a user. This also reduces the entry compute cost for performing data analysis since light-weight mobile devices may also be used. The design trade-off for remote rendering versus sending entire datasets to a local client becomes even more significant as much larger datasets (4096^3) are planned to be made available for the JHTDB.

5. Tests and Discussion

In this section we provide results when benchmarking the architectural improvements presented in this paper with two datasets available in the database cluster. All datasets are split spatially between eight database nodes and a single visualization node is used. The first dataset is a DNS of homogeneous buoyancy driven turbulence (HBDT) [2]. The second dataset is a DNS of forced magnetohydrodynamic turbulence (FMHDT) [3]. Both datasets consist of 1024 files corresponding to different time instances. Each file stores density, three velocity components, and pressure for the first dataset and three velocity components, pressure, three magnetic field and magnetic vector potential components for the second dataset at 1024^3 grid points. Efficiency is determined for computation overhead, bandwidth savings, and total latency from a complete analysis pipeline for single and multiple time steps. Quality is examined for different scales and resolution levels. Finally, we provide an example of scale-based analysis through the use of the added wavelet framework.

5.1. Efficiency

We have also performed a series of benchmarks for a scenario where a single 1024^3 dataset is accessed and visualized by a single user at multiple scales. The visualization node hardware consists of a dual-socket, Intel Xeon E5440 at 2.83 GHz with 32GB of system RAM. The node is headless and does not have a GPU, therefore rendering is done on the CPU using Windows' Advanced Rasterization Platform included with Windows Server Datacenter edition.

Our benchmark simulates a scenario where data are decomposed by the database compute module on-the-fly, cached, and reconstructed if necessary on a visualization node. This benchmark executes a fixed analysis pipeline that generates volume and isosurface visualizations. These isosurfaces are computed at the mean of the data range. As of result, the number of possible concurrent users can be estimated by both time and memory requirements.

Fig 4 contains performance timings and memory usage metrics of the execution of this analysis pipeline on an original quality DNS dataset (HBDT) and at reduced resolution using wavelets. Compute times are based on an average of 3 runs. Visualization times are based on a single threaded versions of the framework with no GPU, where 1 thread is used per user.

Each decomposed scale is represented by the spatial size of the coarse coefficients to the power of 3. The scales considered here range from scale 1 using 512^3 coefficients (finest) to scale 3 using 128^3 coefficients (coarsest) and beyond. These sizes are indicative of the amount of data manipulation that has to be done during analysis. Data retrieval times are measured considering the initial (cache-miss) request for a dataset. A wavelet decomposition is measured by first being computed on the full scale data and only served up to the requested level. As a caching step, we store the data as its scale 2 coefficient state as seen in Fig. 4. To compute scale 3, we can decompose the data down one level incurring a small compute cost or if scale 1, we can reconstruct the coefficients up one level.

As observed, data retrieval times differ significantly due to the varying sizes of compression levels and number of coefficients. During the initial scales of a wavelet decomposition, compute times are relatively larger due to the large amount of fine coefficients computed. Additional levels are subsequently computed much quicker as data are recursively reduced. Although decomposition times are significant when computing coefficients on-the-fly, the total time is still less, nearly

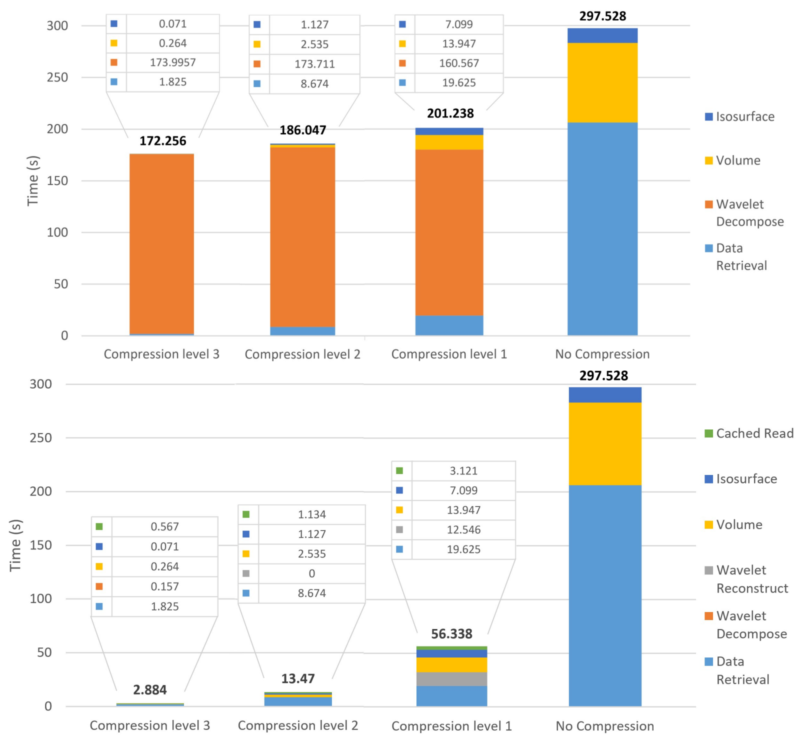

Figure 4: Benchmark. Intermediate and total time spent from request to visualization of a full resolution, 1024^3 dataset. Wavelet decomposition is performed on-the-fly before any caching is done (top) and once cached, only coefficients are served (bottom).

matching the data retrieval time of an entire dataset alone. By enabling compression, there still exists a net decrease in latency from initial query to final image. By reducing the amount of spatial data to process, the performance of downstream analysis operators significantly increases.

Unfortunately, due to storage limitations, a separate database of wavelet coefficients cannot be made available in parallel with the existing raw data. In the ideal case, a database of wavelet coefficients at all scales equal in size with the current datasets could enable minimal times for the features presented in this work. As compression still provides other benefits, we compute coefficients on-the-fly and use limited local storage as cache.

A data analysis pipeline can put significant burden on a system depending on the size of the dataset. As seen in Fig. 5, when considering the visualization pipeline, a high temporary memory

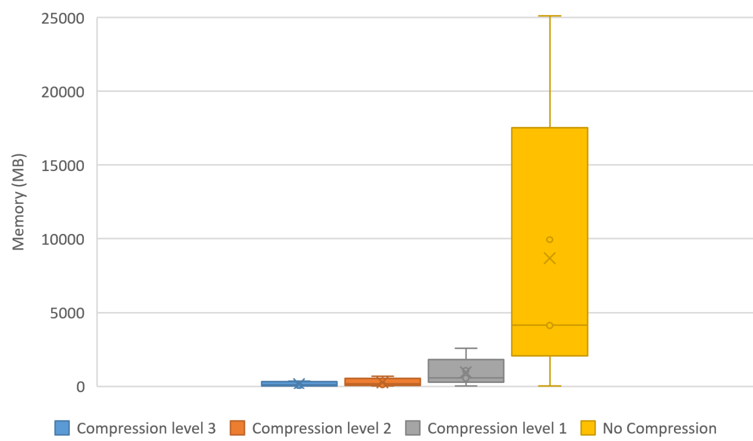

Figure 5: Memory Benchmark. Usage on the visualization node performing an analysis pipeline for a 1024^3 dataset. Analysis is done at several compression levels.

requirement was observed when computing isosurfaces and increased with the size of the dataset. During the isosurface computation for a 1024^3 data frame, the amount of RAM neared the 32GB virtual limit for the visualization node. If surpassed, this may cause the analysis node to incur a very high time penalty cost as the system would need to swap data out of RAM to the hard disk. The benefits of data analysis on spatially smaller representations of the data are both a reduction in compute cost and memory requirements, those of which significantly impact the total time to perform this analysis pipeline.

As more scales are requested, the wavelet machinery must perform more steps and decompose the data further, therefore increasing the decomposition compute cost. In addition to the reduced total compute cost, a larger amount of concurrent users can be supported as a direct result of cubic B-spline wavelet compression. Based on maximums in Fig. 5, a memory-bound node to 32GB can support many concurrent users: <200 users at level 3, <75 users at level 2, <14 users at level 1, and a single user with no compression.

Further improvements are made by storing coefficient representations of the original data, effectively nullifing the on-the-fly wavelet decomposition computation time. The result of this implementation can reduce the total request time down to 2.9 seconds if we consider a Scale 3 visualization of the data. Overall, the small initial cost of computing wavelet coefficients can be greatly offset by the time saved during the transmission and analysis phase of the dataset. By incorporating wavelets, inexpensive commodity hardware can be used to augment the turbulence database cluster.

5.2. Quality

Performing wavelet compression up to certain scales does not always incur obvious costs in quality. When a dataset is reconstructed up to a certain scale, the coefficients used for the reconstruction include those below it. As an example, reconstructing scale 3 will also include those in 4, 5,...,Nth scale where N is the largest (coarsest) scale possible. Peak signal-to-noise ratio (PSNR) is compared for various scales in Fig. 6 and Mean square error (MSE) in Fig. 7. As expected, the inclusion of more scales produces more accurate representations for each dataset. Primary quantities such as density and horizontal magnetic field are more sensitive to truncation errors over the whole range of scales, while the reconstruction quality for derived quantities, such as vorticity, stays relatively constant over the intermediate scales. This is expected since derived quantities depend more on the small scales. Nevertheless, the error levels are relatively small for all these quantities beyond scale 3. The compact support of cubic B-spline wavelets and thus, strong localization properties, allows them to represent turbulence data relatively accurately with a significantly reduced number of coefficients.

A qualitative analysis is performed on a subset of the compressed field using a volume visualization in Fig. 8. The first scale in compression shows very little changes to the HBDT scalar density field. At the initial scale, flow characteristics are preserved well at the data level and visual fidelity is hardly affected. A more pronounced effect begins to form after the second compression scale where oscillations start to form around the selected structure. This scale already represents a 1/64th resolution size of the original. While the third scale represents a very large reduction in size, overall features are still preserved but oscillations near the edges begin to intensify.

The preservation of smooth surfaces and localized features becomes important for preserving derivatives quantities. These characteristics are explored with comparisons of derived isosurfaces in Fig. 9. As with the previous analysis on the HBDT scalar density field, the first two compression scales preserve the overall smoothness and local features of turbulence data. Beyond Scale 3, the

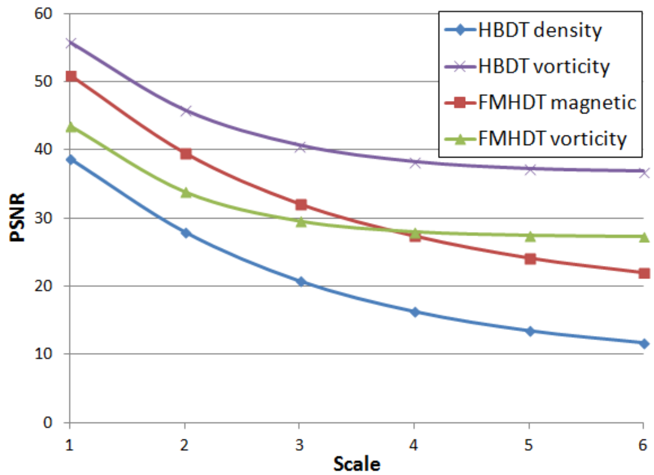

Figure 6: PSNR up-to each scale. Higher values are better.

preservation of interactions between small, localized features can no longer be guaranteed and data artifacts such as ripples begin to appear along isosurfaces. Alternatively, in some turbulence applications the interactions between large-scale features are of interest. For this type of analysis, the large-scale interactions inside the compressed flow are still well represented with even coarser decompositions than Scale 3.

5.3. *Temporal Analysis*

A direct benefit of integrating our wavelet framework with a database is the reduction in effort to perform temporal analysis. Although neither HBDT nor FMHDT datasets are spatially very large at a grid resolution of 1024^3 compared to the largest simulations performed to date, they span a time interval represented by at least 1000 time frames. A temporal benchmark has been performed on a shared resources server visualizing 60 time steps of a 512^3 HBDT density subset. As seen in Fig. 10, the summation of each time step includes the cost of requesting a new dataset from the database, wavelet decomposition and reconstruction (for Scale 2), and volume visualization. In addition, cached benchmarks for both datasets are available by using the databases native caching functionality. As observed, there is a significant improvement in latency when considering a cached version of the data, but more importantly a compressed Scale 2 equivalent of a dataset.

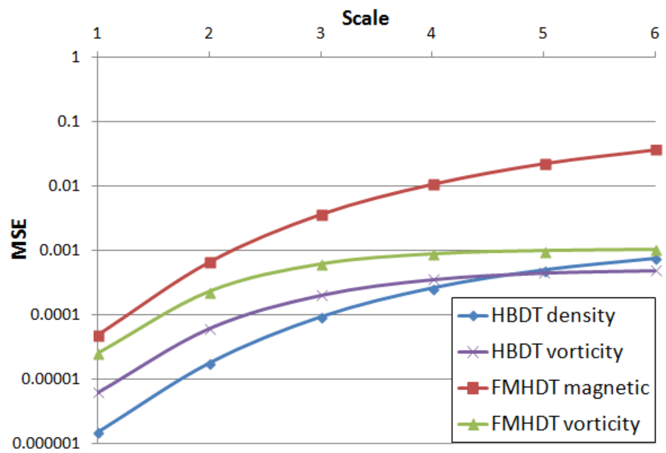

Figure 7: MSE up-to each scale. Lower values are better.

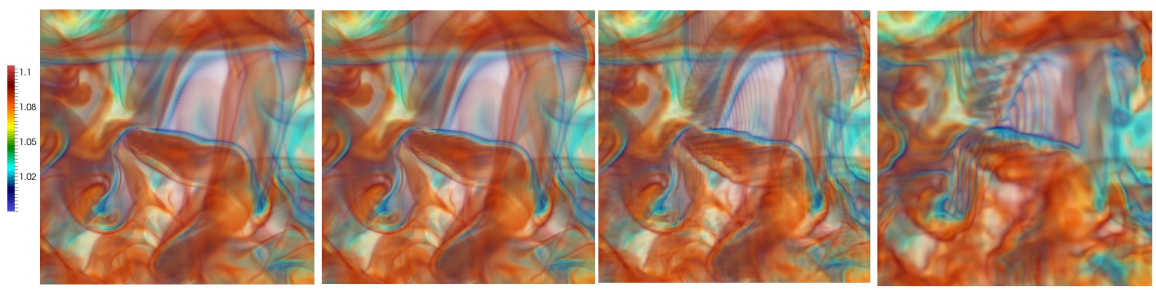

Figure 8: Volume visualization of HBDT density. A close-up of a feature to show the compression characteristics of cubic B-spline wavelets at each data level. From left to right: original, scale 1, scale 2, scale 3. Visible differences are minimal between the original and scale 1. Scale 2 begins to exhibit lossy features and by scale 3, most of the features begin to lose their finer structures around the edges.

The summation of time to visualize 60 time steps is 3024 secs (1370 secs cached) for the non-compressed data, and 1019 secs (191 secs cached) for a Scale 2 compression. The visualization results can be seen in Fig. 11. Qualitatively, HBDT volume density can be visualized at a reduced scale with no significant compromise on the final visualization.

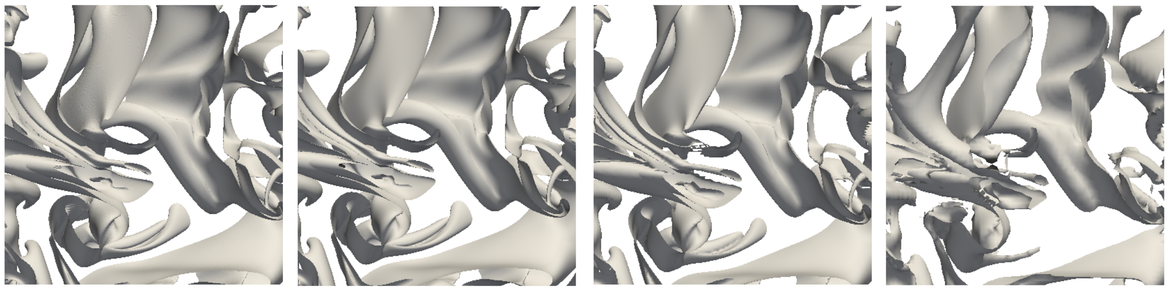

Figure 9: Isosurface visualization of HBDT density. A close-up of a feature to show the compression characteristics of cubic B-spline wavelets at the data level. From left to right: original, scale 1, scale 2, scale 3. Structures are generally well preserved up to Scale 2, where lossy features begin to appear. At scale 3, most of the features begin to lose their finer structures around the edges of isosurfaces appearing as artifacts.

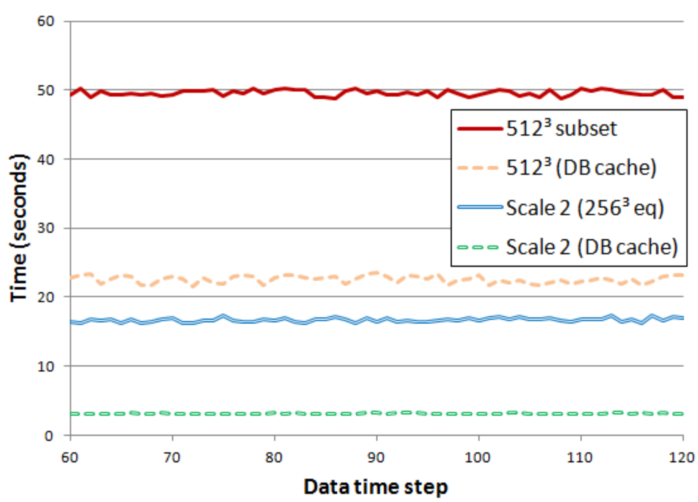

Figure 10: Temporal visualization benchmark. Volume visualization for 60 timesteps was performed for a 512^3 subset, and a Scale 2 (256^3 equivalent) wavelet. Total visualization times are 3024s (1370s cached) for non-compressed, and 1019s (191s cached) for Scale 2 compression.

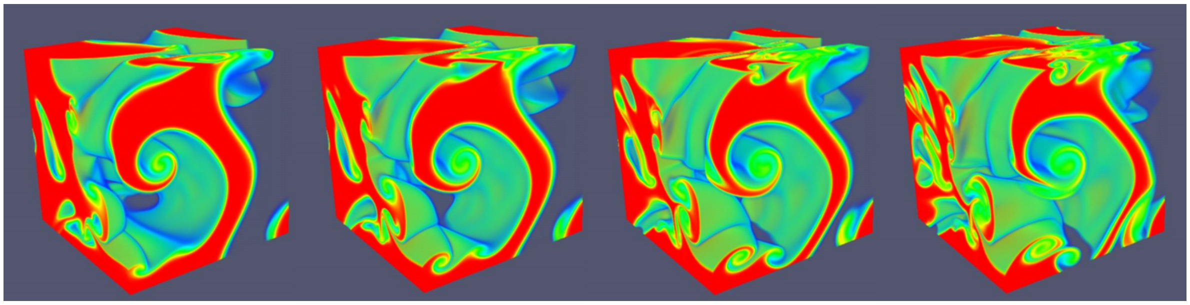

Figure 11: Temporal volume visualization of HBDT density. Wavelet integration and caching capabilities make it possible to quickly visualize multiple datasets to improve temporal understanding of a turbulence dataset. These four time steps represent frames 60,75,90,105 out of the 1024 total frames.

5.4. Scale-based Wavelet Analysis

This section provides an example of scale-based analysis through the use of the added wavelet framework. Scales are extracted from a dataset by isolating the coefficients extracted in each individual decomposition step. Compared to the previous section where we preserve coefficients up-to a specific scale, this section involves looking at each scale individually.

Fig. 12 shows the two datasets with two basic quantities specific to each dataset decomposed into several scales. Each scale contains amounts of positive (red) and negative (blue) quantity captured per scale, where the sum of all individual scales results in the original. Three-dimensional turbulence is a strongly multi-scale phenomenon, with local and non-local interactions among the various scales. When the range of scales is large enough, which is the case with most practical flows, the energy is transferred from the energetic large scales (where the energy is usually deposited) to the smallest scales (where the energy is usually dissipated).

This ”cascade” of energy paradigm involves a hierarchy of vortex sizes and structure shapes. Unfortunately, the usual decompositions in Fourier space are non-local in physical space and specific physical structures can not be easily associated with certain scales.

5.5. Significance of wavelet analysis

Multi-resolution geometric representation methods have emerged as more appropriate tools for scale decomposition and better connection with flow features. For example, using the curvelet

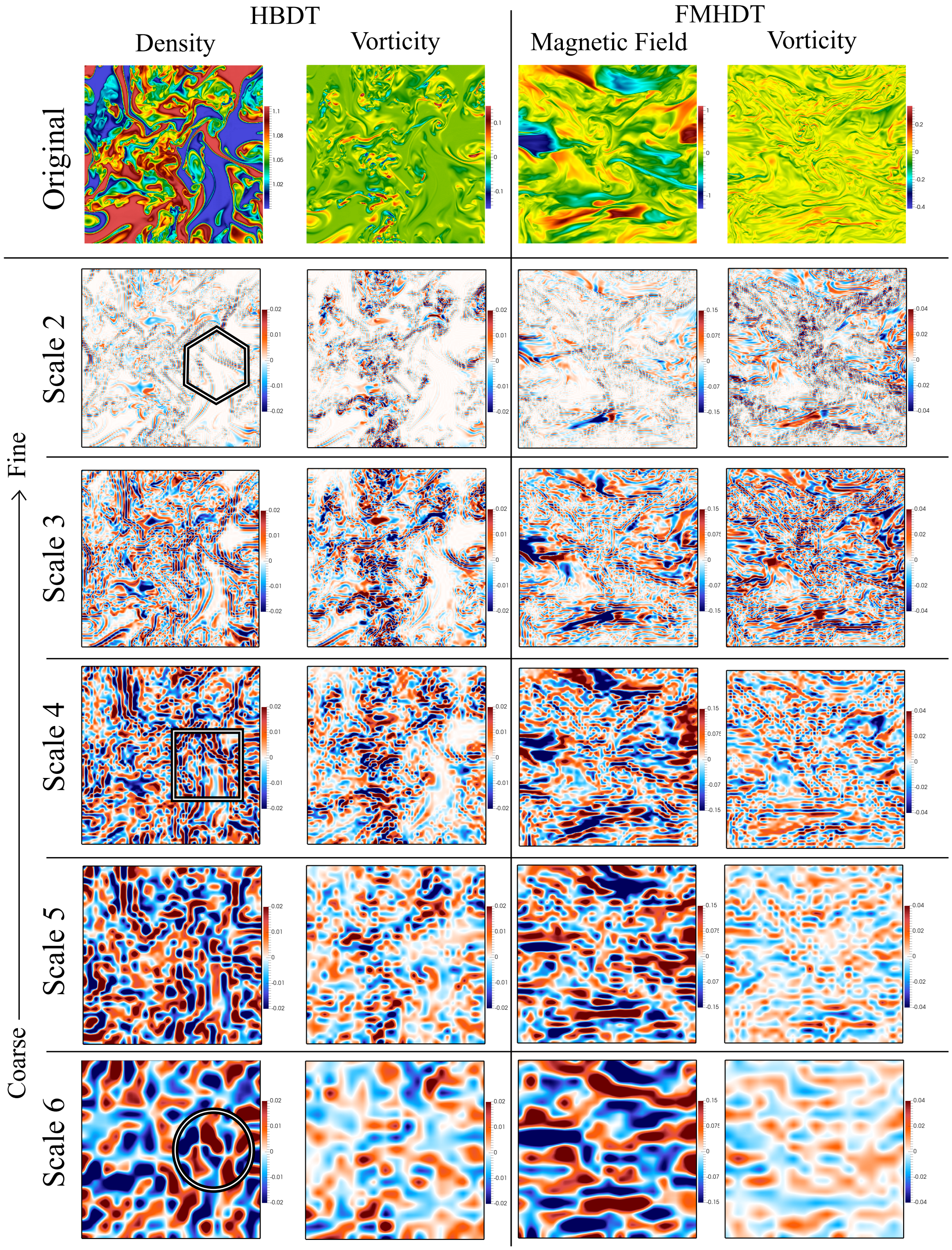

Figure 12: Turbulence visualization by scale. Homogeneous Buoyancy Driven Turbulence (Left) and Forced Magneto-Hydrodynamic turbulence (Right) are decomposed into several scales exhibiting multi-scale phenomena. Globular (blob) structures can be observed at the coarser (higher) scales (circles), while tube structures (squares) and sheet-like structures (hexagons) emerge out of finer scales. By adding all scales, the original dataset can be reconstructed losslessly.

transform, [27] [28] have shown a geometrical progression from blobs through tubes to sheet-like structures with decreasing physical scale in a simple forced turbulence flow. Such knowledge can be useful to understand and characterize the cascade process and also to inform physics-based subgrid models used in affordable coarse mesh computations.

The flows exemplified here are more complex than those used in the past for this type of analysis. In addition, to demonstrate the remote analysis of large datasets, we are using B-splines wavelets, which have been scarcely used before for such analyses, but can have distinct advantages [1].

The HBDT flow represents the mixing between initially segregated fluids with different densities as they start moving under the influence of gravity (or external acceleration). The fluids are initially at rest and organized in pure-fluid random patches. As the buoyancy force starts moving the two fluids, the velocities increase in magnitude, while hydrodynamic instabilities, such as Kelvin-Helmholtz instability, start to generate vortical motions at the interfaces between the fluid patches. Thus, unlike previous applications of multi-resolution representation methods where the scalars were passively advected by the velocity field, in this case the density represents an active scalar, which feeds back into the velocity evolution. The full field showed in Fig. 12, first column, top, is taken at the time when significant mixing and vorticity generation has already been produced, while both light (blue) and heavy (red) pure fluids are still present. Moving downwards along the first column in Fig. 12, one can look at the density field structure from small to large scales

At the level of the largest scales, the density exhibits a globular (blob) structure, with no specific orientation bias. However, at the next two finer scales, the physical structure shows the emergence of tubes preferentially oriented in the direction of gravity. As the scales become finer, the tubes also become narrower and lose the vertical orientation. Finally, at the smallest scales, the tubes are almost completely vanished and replaced with sheet-like structures. However, unlike the passive scale case studied before, these structures are more difficult to identify, as globular (blob) structures are still present and dominate the range of values shown in the legend. While the general picture encountered before in the passive scalar case of blobs to tubes and to sheets is still present approximately, we note that the presence of blobs at the finest scale is consistent with the mixing asymmetry identified in the variable density case, with the pure heavy fluid mixing slower than the pure light fluid [29, 30, 31]. Thus, these results show that the current subgrid models based on passive scalar considerations need to be revisited for variable density flows to account for the presence of unmixed (or less mixed) fluid blob structures down to the smallest scales.

A relatively different behavior is observed in the scale decomposition of the horizontal vorticity for the HBDT flow. Since the vorticity is primarily generated at the interface between the fluid blobs by fluid instabilities, there is a good correlation between density and vortical fields in the full signal. Nevertheless, the tube like structure is less apparent at intermediate scales, as well as alignment with horizontal or vertical directions. The alignment is mostly with the interfaces of the fluid patches. This, again, makes modeling of such a flow more complicated, since the subgrid models for the velocity field need to account for the structure of the active scalar (density) field.

The FMHDT flow [3], is also more complicated than previous attempts to characterize turbulence using multi-resolution representation methods. In this case, again, there is a full feed-back between the velocity and magnetic fields. Using the previous representation, we decompose the horizontal magnetic field seen as an active scalar. At large and intermediate scales, there is a good correlation between the vorticity and magnetic fields which tend to consist of structures elongated in the x-direction. These structures become narrower and sheet-like at smaller scales, where most of the correlation between the two fields is no longer present. However, even at the smallest scales, there are regions of intense vorticity and magnetic field which are relatively well correlated between the two fields, while the rest of the structures remain poorly correlated. Again, this underlies the difficulty in understanding the cascade process and constructing subgrid models for such strongly coupled flows.

6. Conclusion

We have introduced a new, improved system architecture for a public database cluster of large turbulence datasets. Until recently, the visualization and analysis of such datasets has been restricted to a few groups worldwide with access to large supercomputers. The public Johns Hopkins Turbulence database (JHTDB) simplifies the access to multi-Terabyte turbulence datasets and facilitates the computation of statistics and extraction of features through the use of commodity hardware. Visualization support has been added for this database and wavelet analysis tools have been implemented to expand the capabilities for this database cluster. Finally, wavelet compression has been introduced at the data-level to reduce access costs, reduce bandwidth and improve latency between database components. This component also reduced the memory footprint of datasets required for data analysis, effectively adding support for many concurrent users. The paper demonstrates the new tools both for enabling remote visualization and turbulence data analysis for two of the datasets hosted by JHTDB. These tools should help to extend the reach and analysis power and ultimately the goals of such public databases.

For future work, we would like to move the visualization node away from the JHTDB and into the cloud through Amazon Web Services (AWS). Currently, the user-interaction latency is limited by the physical distance between a user and the location of the JHTDB. In this paper, We've reduced the amount of data needed to visualize so it may be feasible to make transfers external to the visualization node from the database. Additionally, if we dynamically and transparently allocate nodes for a user on AWS, and select a location that is physically the closest, user-interaction latency would be significantly reduced.

Acknowledgment

The authors would like to thank LANL and Johns Hopkins University for supporting this work. Los Alamos National Laboratory is operated by Los Alamos National Security, LLC for the US Department of Energy NNSA under contract no. DE-AC52-06NA25396.

[1] J. Pulido, D. Livescu, J. Woodring, J. Ahrens, and B. Hamann. Survey and analysis of multiresolution methods for turbulence data. *Computers and Fluids*, 125:39 – 58, 2016.

[2] D. Livescu, C. Canada, K. Kanov, R. Burns, IDIES, and J. Pulido. Homogeneous buoyancy driven turbulence data set, 2014. http://turbulence.pha.jhu.edu/docs/README-HBDT.pdf.

[3] H. Aluie, G. Eyink, E. Vishniac, S. Chen, K. Kanov, R. Burns, C. Meneveau, and A. Szalay. Forced mhd turbulence data set, 2013. http://turbulence.pha.jhu.edu/docs/README-MHD.pdf.

[4] E. Perlman, R. Burns, Y. Li, and C. Meneveau. Data exploration of turbulence simulations using a database cluster. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, 2007.

[5] Aaron Trott, Robert Moorhead, and John McGinley. Wavelets applied to loseless compression and progressive

transmission of floating point data in 3-d curvilinear grids. In *Proceedings of the 7th Conference on Visualization '96*, VIS '96, pages 385–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[6] Lars Lippert, Markus H. Gross, and Christian Kurmann. Compression domain volume rendering for distributed environments. In *Computer Graphics Forum*, volume 16. Wiley Online Library, 1997.

[7] S. Guthe, M. Wand, J. Gonser, and W. Strasser. Interactive rendering of large volume data sets. In *IEEE Visualization, 2002. VIS 2002.*, pages 53–60, Nov 2002.

[8] J. Woodring, S. Mniszewski, C. Brislawn, D. DeMarle, and J. Ahrens. Revisiting wavelet compression for large-scale climate data using jpeg 2000 and ensuring data precision. In *Symposium on Large Data Analysis and Visualization (LDAV)*, pages 31–38. IEEE, 2011.

[9] Sriram Lakshminarasimhan, Neil Shah, Stephane Ethier, Scott Klasky, Rob Latham, Rob Ross, and Nagiza F. Samatova. *Compressing the Incompressible with ISABELA: In-situ Reduction of Spatio-temporal Data*, pages 366–379. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[10] P. Lindstrom. Fixed-rate compressed floating-point arrays. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2674–2683, Dec 2014.

[11] James Ahrens, Berk Geveci, and Charles Law. Paraview: An end-user tool for large-data visualization. *The Visualization Handbook*, page 717, 2005.

[12] Hank Childs, Eric S. Brugger, Kathleen S. Bonnell, Jeremy S. Meredith, Mark Miller, Brad J Whitlock, and Nelson Max. A contract-based system for large data visualization. In *Proceedings of IEEE Visualization 2005*, page 190–198, 2005.

[13] W. J. Schroeder, L. S. Avila, and W. Hoffman. Visualizing with vtk: a tutorial. *IEEE Computer Graphics and Applications*, 20(5):20–27, Sep 2000.

[14] A. Cedilnik, B. Geveci, K. Moreland, J. Ahrens, and J. Favre. Remote large data visualization in the paraview framework. In *Proceedings of the 6th Eurographics Conference on Parallel Graphics and Visualization*, EGPGV '06, pages 163–170, 2006.

[15] M. Treib, K. Burger, F. Reichl, C. Meneveau, A. Szalay, and R. Westermann. Turbulence visualization at the terascale on desktop pcs. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2169–2177, 2012.

[16] O. Roussel, K. Schneider, and M. Farge. Coherent vortex extraction in 3d homogeneous turbulence: comparison between orthogonal and biorthogonal wavelet decompositions. *Journal of Turbulence*, page N11, 2005.

[17] D. Laney, P. t. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1053–1060, Sept 2006.

[18] P. T. Bremer, G. Weber, V. Pascucci, M. Day, and J. Bell. Analyzing and tracking burning structures in lean premixed hydrogen flames. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):248–260, March 2010.

[19] J. Graham, M. Lee, N. Malaya, R.D. Moser, G. Eyink, C. Meneveau, K. Kanov, R. Burns, and A. Szalay. Turbulent channel flow data set, 2013.

[20] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, A. Burns, S. Chen, E. Szalay, and G. Eyink. A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence*, 9(31), 2008.

[21] I. Daubechies. Ten lectures on wavelets. *PA: SIAM*, 1992.

[22] M. Farge. Wavelet transforms and their applications to turbulence. *Annual Review of Fluid Mechanics*, 24(1):395–458, 1992.

[23] W. Sweldens. The lifting scheme: A construction of second generation wavelets, 1997.

[24] M. J. Shensa. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Trans. on Signal Processing*, 40(10):2464–2482, 1992.

[25] B. Gough. *GNU Scientific Library Reference Manual - Third Edition*. Network Theory Ltd., 3rd edition, 2009.

[26] U. Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware Inc., Clifton Park, NY, 2015.

[27] I. Bermejo-Moreno and D. I. Pullin. On the non-local geometry of turbulence, 2008.

[28] Y. Yang, D. I. Pullin, and I. Bermejo-Moreno. Multiscale geometric analysis of lagrangian structures in isotropic turbulence, 2010.

[29] D. Livescu and J. R. Ristorcelli. Variable-density mixing in buoyancy-driven turbulence. *J. Fluid Mech.*, 605:145–180, 2008.

[30] D. Livescu and J. R. Ristorcelli. Mixing asymmetry in variable density turbulence. In *Advances in turbulence XII*, pages 545–548. Springer, 2009.

[31] D. Livescu. Numerical simulations of two-fluid turbulent mixing at large density ratios and applications to the RayleighTaylor instability. *Phil. Trans. R. Soc. A*, 371:20120185, 2013.