# An Image-based Approach to Extreme Scale *In Situ* Visualization and Analysis

James Ahrens
Los Alamos National Laboratory
Los Alamos, NM 87545
Email: ahrens@lanl.gov

Sébastien Jourdain
Kitware Inc
Santa Fe, NM 87505
Email: sebastien.jourdain@kitware.com

Patrick O'Leary
Kitware Inc
Santa Fe, NM 87505
Email: patrick.oleary@kitware.com

John Patchett
Los Alamos National Laboratory
Los Alamos, NM 87545
Email: patchett@lanl.gov

David H. Rogers
Los Alamos National Laboratory
Los Alamos, NM 87545
Email: dhr@lanl.gov

Mark Petersen
Los Alamos National Laboratory
Los Alamos, NM 87545
Email: mpetersen@lanl.gov

***Abstract*—Extreme scale scientific simulations are leading a charge to exascale computation, and data analytics runs the risk of being a bottleneck to scientific discovery. Due to power and I/O constraints, we expect *in situ* visualization and analysis will be a critical component of these workflows. Options for extreme scale data analysis are often presented as a stark contrast: write large files to disk for interactive, exploratory analysis, or perform *in situ* analysis to save detailed data about phenomena that a scientists knows about in advance. We present a novel framework for a third option – a highly interactive, image-based approach that promotes exploration of simulation results, and is easily accessed through extensions to widely used open source tools. This *in situ* approach supports interactive exploration of a wide range of results, while still significantly reducing data movement and storage.**

## I. Introduction

The supercomputing community has embarked upon a revolutionary path towards extreme scale ($\geq 10^{15}$ FLOPS). Just as the massive computing power of these machines improve how we do simulation science, these new machines are changing how we analyze simulation results. Application scientists use visualization and analysis to understand and advance their science. At smaller scales, the data are stored or moved to another machine for post-processing. However, both the complexity and size of their scientific simulations continue to evolve as we advance en route to extreme scale computing platforms. With storage bandwidth significantly falling behind the rate needed to move data, standard post-processing techniques will not be able to effectively scale in the future. Therefore, the fundamental extreme scale visualization and analysis challenge is there is too much simulation data and too little transfer bandwidth.

*In situ* techniques that analyze the data while it still resides in simulation memory show a promising path forward [1], [2], [3], [4]. The same supercomputing resources that compute the simulation data are also used for the analysis and, as such, the data do not have to be moved. Typically, *in situ* approaches either are a predefined set of analyses or, rarely, make automatic decisions about which analyses and visualizations to create. Therefore, we see the goals of *in situ* visualization and analysis as multifaceted: 1) to preserve important elements of the simulations, 2) to significantly reduce the data needed to preserve these elements, and 3) to offer as much flexibility as possible for post-processing exploration.

Simulation results must be transformed from an extreme scale sized data space to a petascale storage system resulting in a massive compaction from sparse raw simulation data to dense visualization and analysis data. We envision the scientist using our framework to define which analyses are needed and a target data size bounds of the analysis results at the beginning of their simulation *in situ* run. Understanding the space of *in situ* visualization and analysis solutions within the context of data funneling process is a salient key to addressing the extreme scale challenge.

We present a novel framework – implemented in existing open-source tools [5] - to be used by a scientist to define a set of operations he/she finds to be most useful in exploring their data. The framework implements an image-based approach that results in a database of highly compressed data that is fundamentally different from what is currently available. Importantly, our framework effectively preserves the ability to interactively explore the same "operation space" defined at the start of the problem, so that data elements can be combined in much the same way they could in the original tool [6]. Thus, interactive exploration - so important to scientific discovery - is supported on a useful spectrum of operations.

Imagery is on the order of $10^6$ in size, whereas extreme scale simulation data is on the order of $\geq 10^{15}$ in size. As an example, suppose we have an extreme scale simulation that calculates temperature and density over 1000 of time steps. For both variables, a scientist would like to visualize 10 isosurface values and X, Y, and Z cut planes for 10 locations in each dimension. One hundred different camera positions are also selected, in a hemisphere above the dataset pointing towards the data set. We will run the *in situ* image acquisition for every time step. These parameters will produce: 2 variables $\times$ 1000 time steps $\times$ (10 isosurface values $+$ 3 $\times$ 10 cut planes) $\times$ 100 camera positions $\times$ 3 images (depth, float, and lighting) $=$ 2.4 $\times$ $10^7$ images. If we assume each image is 1MB (megapixel, four byte image), this results in approximately 24 TBs, which is a reasonable size for a large exascale simulation.

978-1-4799-5500-8/14 $31.00 © 2014 IEEE
DOI 10.1109/SC.2014.40

Thus, the image-based approach reduces the simulation output by storing a set of output images directly from the simulation into an image database. One can think of this approach as the traditional *in situ* mode, but we are sampling the visualization and analysis parameter space, such as camera positions, operations, parameters to operations, etc., to produce a set of images [1], [7], [8], [9] stored in a data-intensive database. It's important to note these images are derived from full-resolution data with high accuracy.

The framework, implementing our image-based approach as a solution for extreme data visualization and analysis challenges, makes several contributions to the "traditional *in situ*" mode.

**Interactive Exploration Database**. Our image-based approach takes traditional *in situ* visualization and analysis and enables interactive exploration using an image database. This, in turn, creates a viable solution for extreme scale visualization and analysis. Our framework:

- Enables many different interaction modes including: 1) animation and selection for objects, 2) camera and 3) time, than we imagined possible with a set of pre-generated analysis.
- Creates an incredibly responsive interactive solution, rivaling modern post-processing approaches, based on producing constant time retrieval and assembly of visualization objects from the image database.
- Encourages the use of both computationally intensive analysis and temporal exploration typically avoided in post-processing approaches.
- Demonstrates the time to create an image collection is not of great concern.

**Metadata Searching**. By leveraging an image database, our image-based approach allows the analyst to execute metadata queries or browse analysis results to produce a prioritized sequence of matching results.

**Creation of New Visualizations and Content Querying**. We've added composing of individually imaged visualization objects to our image-based approach to allow the analyst to reason about his/her simulation results from visualization space and create new content. This unique capability:

- Provides access to the underlying data to enable advanced rendering during post-processing (e.g. new lookup tables, lighting, ...).
- Makes it possible to perform queries that search on the content of the image in the database. Therefore, using image-based visual queries, the analyst can ask simple scientific questions and get the expected results. These image-based queries show promise of answering much more complicated questions.

Finally, we have exposed the framework of our image-based approach to the scientist through an advanced selection interface that allows him/her to make sophisticated (time, storage, analysis, ...) decisions for the production of *in situ* visualization and analysis output.

In the sections that follow, we illustrate how our image-based approach to extreme scale *in situ* visualization and analysis meets our goals for future post-processing exploration.

## II. Related Work

Our framework has number of contributions and, therefore, we review related work for these areas. *In situ* approaches are an important mechanism for visualization and analysis due to the cost of data movement and storage required for traditional post processing [10]. A key concern with *in situ* approaches is the need to maintain exploratory analysis capabilities despite the fact that data gathering occurs as a batch process at simulation run-time. For example, Woodring et al [11] saved a hierarchy of random samples from a particle-based simulation to create a flexible representation for later analysis.

**Interactive Exploration Database**. A key component of our solution is the creation of a large image collection from a structured sampling of camera positions, time steps and visualization operators. One option for managing these rendering is to compress them into a collection of movies. Chen et al [12] use this approach to accelerate interactive scientific visualization over the Internet. Kageyama and Yamada [13] applied this approach *in situ* to a simulation. Both create specialized "movie players" that support exploratory interactions accessing the linear movies to retrieve specific rotations, time steps and operators. Our solution extends these approaches by supporting the compositing of images to create new visualizations as well as metadata and image-based querying. Tikhonova et al in [14], [1], [15] represent the scientific data set to be visualized as a collection of proxy images. By retrieving images from this collection and applying image-based rendering techniques, interactive volume rendered results are produced. A range of view points, transfer functions, rendering, and lighting options are reproduced interactively. Our work is complementary to this image-based approach. Combining the approaches would support additional data compression, flexibility, and exploration possibilities.

**Metadata Searching**. Commercial multimedia databases such as Google's image search have brought the powerful non-traditional search/query techniques to the mainstream. Recent work by Subrahmaniam [16] identifies issues and future research directions for multimedia databases. For speed and flexibility in responding to our unique access patterns, we created our own image database. When populating an image database it is desirable to gather metadata about when and how the images are created, to enable querying of these parameters. Therefore, when we create imagery *in situ*, we save camera positions, time steps, details about the visualization operators, and statistics about the data. We highlight a connection to provenance systems, such as VisTrails, that directly store analysis results and how they are created. This is important so that results can be reproduced by others, improving scientific integrity [17]. Our approach could be evaluated as a visualization and analysis storage representation in such a system.

**Creation of New Visualizations and Content Querying**. Our approach supports the creation of new visualization by combining images with depth information. Tikhonova et al [14] also supports the creation of new visualizations from their database using interpolation. It is worth noting, when compositing images that contain opaque geometry, our results will be pixel accurate whereas with image-based interpolation, some loss is expected especially as the viewer moves away from sampling locations available in the database. Our idea for compositing visualization results evolved from the long

history of parallel compositing techniques that enable scalable interactive visualization. Moreland et al provides a recent overview of approaches [18] and apply additional optimization to this critical technique.

We are interested in supporting a variety of methods for interacting with our visualization. From an *in situ* starting point, we offer a traditional point-and-click interaction method, supporting the rotations, time steps, and visualization operations selected by the scientist in the setup of his/her simulation run. We also support interaction via an interactive database perspective. As mentioned above we support metadata searches. There are many image content search possibilities as well as search by color and search by similarly. In this paper, we focused on including image content queries that support querying about the visual weight of the objects in the generated visualization. This is a unique capability for an interactive scientific visualization framework and derives from the fact that we stored the visualization metadata, visualization objects, and their resulting 2D projections as images. There are many approaches to calculating the statistics of the 2D projection of a set of 3D objects [19]. Related to our approach, Jun Tao et al [20] computes a collection of streamline images and applies an image quality metric to select an optimal viewpoint. Our approach extends this work by virtue of being *in situ* and by our ability to change our evaluation metric dynamically with a scientist-generated query.

Complementary work would support indices on the original scientific data. These are referred to as data content-based queries and an overview is provided in [21]. An exemplar within this literature is Stockinger et al [22], which used efficient bitmap indices to support querying against the original data. Thus, we could save bitmaps with our visualization offering both image-based and data-based queries.

## III. Overview of Approach

Though we are running into a bandwidth barrier, interactive post-processing visualization and analysis is still essential at extreme scale. When creating new simulations, scientists analyze support for exploration and debugging. Also, they need the capability to share data with colleagues that do not have access to their computing resources. Finally, there needs to be an analysis transition path for existing codes from terascale/petascale ($10^{12}/10^{15}$ FLOPS) to extreme scale.

In the image-based approach, we produce a multitude of analysis outputs, such as images and plots, that will show scientists about their data through interactions with an image database. Visualization and analysis operators typically produce results that are many orders of magnitude smaller than the original data. Specifically, imagery is on the order of $10^6$ in size, whereas extreme scale simulation data is on the order of $\geq 10^{15}$ in size. Our expectation is that the memory size, with associated burst buffers [23] and storage, will be on the order of $10^{15}$ (i.e., a petabyte). We believe a petabyte is a reasonable size for an extreme scale simulation output and, therefore, this means we can store approximately $10^9$ (a billion) images of the simulation. This is on the order of the number of images uploaded to Facebook per year or the total number of photos hosted by Flickr [24]. The benefit of being able to store this many images is providing flexibility, similar to a data approach, for exploratory simulation analysis.

From our example present in the Introduction section, we would produce $2.4 \times 10^7$ images at 1MB per image for approximately 24 TBs, which is reasonable given our previous assumptions. We expect that massive computing power will be available on the supercomputer with associated burst buffer [23] and data intensive storage systems [25] to process these images.

One interesting property of this approach is that the relatively fixed size of the output imagery, due to limits of human visual acuity [26], means that as we continue to scale simulation and supercomputer sizes we will be able to store more and more imagery as machine sizes grow. While the image size might grow as the simulation size grows, the analyst is less likely to increase the sampling of the corresponding parameters, operators, and camera space. This approach supports many different potential interaction modes, and may offer different insights than interacting with simulation data with traditional analysis and visualization tools.

There are a number of useful image-based rendering approaches that may help us to sample and present generated images [1], [6]. The goal of this work has not been to produce better image-based rendering techniques. Rather, the goal is to understand how *in situ* methods are able to support flexible and accurate analysis of extreme scale datasets. In particular, we have focused on potential interaction modes with image data, but not necessarily those enabled by image-based rendering techniques.

Many scientific simulation communities produce image collections for later analysis and archival purposes. One of the best examples, the CESM (Community Earth System Model) [27], includes the diagnostics for all its component models including atmosphere, ocean, land, and sea ice. These visualizations are available on a webpage for analysis and archival purposes. They are generated in a post-processing manner after the simulations have completed. This collection represents a community consensus on a set of visualizations that are useful to the community. Other examples include astronomy [28], cosmology [29], and high-energy physics [30]. We recognize the current limitations of these collections. Specifically, most of these collections contain fully-rendered images that make it difficult to retrieve the original simulation data values. Optimizations to our approach will help to correct these shortcomings.

### A. Simulation Data to Image Database

Our image-based approach framework is built on top of ParaView, a modern visualization and analysis tool used around the world in post-processing for advanced modeling and simulation workflows. ParaView is an open-source, multi-platform data analysis application [31] developed to analyze extremely large datasets using distributed memory computing resources. Most modern post-processing applications utilize a common pipeline architecture (e.g VisiT, ...). Thus, any of these tools could be easily adapted for use with our image-based approach framework.

When starting an image-based analysis, the computational scientist will define a desired set of visualization and analysis operators using a test data set and a familiar post-processing application, as depicted in Figure 1.

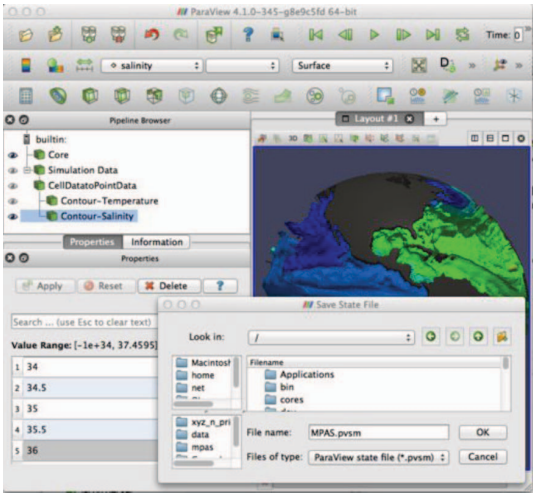


Fig. 1. Once the computational scientist has defined a reasonable set of visualization and analysis operators in a familiar post-processing tool, he/she will simply save the current state of the application to a state file.

Next, the scientist uses our advanced selection interface, shown in Figure 2, to make sophisticated prioritized decisions for the production of analysis output. By importing the state file, the advanced selection interface presents the visualization pipelines created previously using a familiar post-processing tool. Using the *Pipeline* section, the scientist determines: how often to perform *in situ* analysis, what visualization and analysis objects to create, and how to sample the visualization object parameter space.

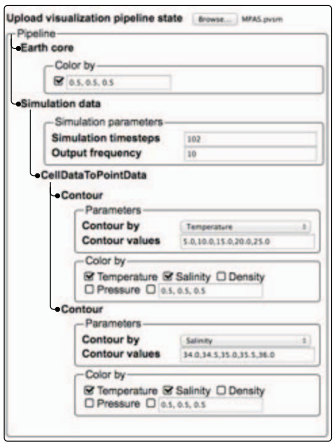


Fig. 2. The advanced selection interface enables the scientist to adjust visualization and analysis operators and how to sample the parameters space.

Then, the scientist moves to the *Camera settings* section and describes how he/she will sample the camera space by defining a camera manager and making appropriate selections for $\theta$ and $\phi$ sampling. The tool instantly updates the total number of resulting viewpoints from the sampling selections.

As simulations progress toward extreme scale, scientists must operate in a constrained storage environment. Hence, selection in the parameter space and sampling in the camera space will require prioritization to fit within a storage budget.

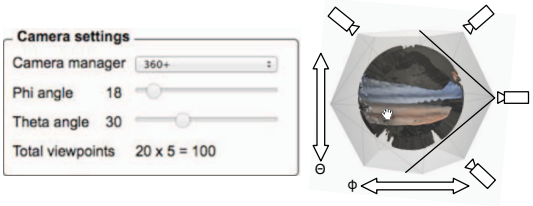


Fig. 3. By selecting camera space sampling, the scientists will receive instant viewpoint feedback in the Camera settings section.

The final decision in the *Image settings* section particularly impacts storage. The scientist would select the image sampling (resolution) and the image type. The image types include: raw data (TIFF format) and compressed lossless data (PNG format).

The results of these choices are constantly updated in the *Cost estimate* section. The costs are reported for number of images, image size, collection size, and additional computational time.

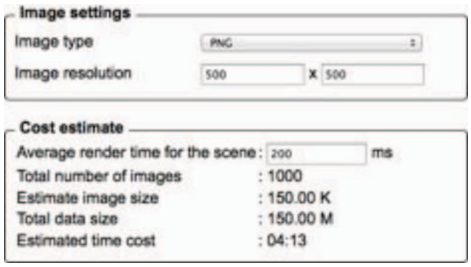


Fig. 4. The *Cost estimate* section of the advanced selection interface enables the scientist to examine "what if" scenarios for costs in a constrained storage environment.

The advanced selection interface could be very complex with many variables and operations from which to choose. The scientist's selection of different outputs from the simulation are then presented (Figure 4), depicting the managed tradeoffs between additional computation, storage space usage, and visualization and analysis outputs. This selection could be optimized with intelligent selection capabilities, such as automatic isosurface selection [32] and automatic camera selection [8], to help reduce the interface complexity.

The output of the advanced selection interface is an *in situ* analysis python script that implements the defined selections. Our framework uses ParaView Catalyst, an open-source *in situ* (and other use cases) visualization and analysis optimized C++ designed to be tightly coupled with simulation codes.

As the simulation runs, the image results are ingested by the database. By this we mean, that the metadata, image provenance (i.e. a searchable description of how the image was created – the simulation, the input deck, and which operators were applied), and the root image uniform resource locator (URL) avoiding the need to move the potentially large image data around.

Several of the contributions of our image-based approach are demonstrated using the Model for Prediction Across Scales-Ocean (MPAS-Ocean) [33] as an example. MPAS-Ocean is an unstructured-mesh ocean model capable of using

enhanced horizontal resolution in selected regions of the ocean domain. MPAS-Ocean is one component within the MPAS framework of climate models that is developed in cooperation between Los Alamos National Laboratory (LANL) and the National Center for Atmospheric Research (NCAR).

The example runs are from real-world simulations with realistic topography, wind forcing, and temperature and salinity restoring. The horizontal grids are quasi-uniform over the globe, with simulations performed at nominal grid cell widths of 120 km.

### B. Interactive Exploration Database

The interactive exploration database enables a diverse set of interactions with a set (database) of pre-generated visualization and analysis results. The interactive exploration database supports essentially three elements and two modes of interaction, depicted in Figure 5. The three elements of interaction are *time*, (visualization and analysis) *objects*, and *camera*. The modes of interaction are: *animation*, where the interaction sequence through time, objects, and camera; and *selection*, where the analyst would select time, objects, and camera.

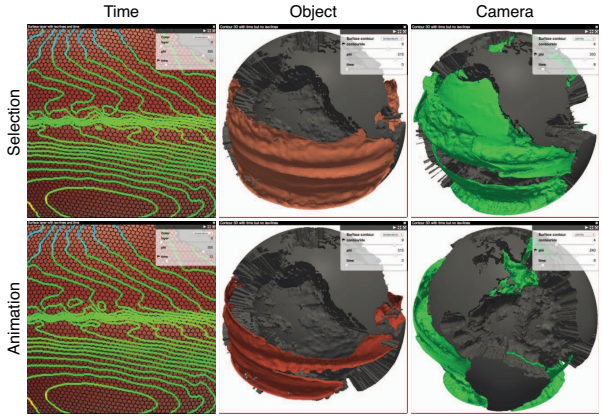


Fig. 5. The interactive space enabled by the interactive exploration database.

We have developed an interface for an interactive exploration database that supports the three elements and the two modes of interaction. From the top row of Figure 5, the analyst can select a time, an object (e.g., a three-dimension contour), and a camera, which requests the corresponding image to be fetched from the database. In fact, for each image in the top row a set describing time, object, and camera would be specified in the request. The bottom row of Figure 5 demonstrates animation of time, object, and camera, respectively, from the initial selection of the top row. We assume the sampling of the parameter space and camera space is dense enough so that interaction can be achieved on the three elements.

For the rotation images in Figure 6, we can see that the mouse-enabled rotation requires an image to be fetched from the database. Starting at the lower right image, if the analyst rotates up with the mouse, then the camera position change is queried and the upper right image is returned (likewise for rotations to the left and/or up and left). Mouse-enabled panning and zooming are simple image-based operations. Zooming displays the image by varying the pixel size to $1 \times 1$, $2 \times 2$, ...

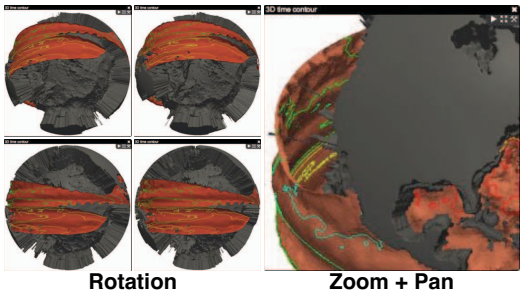


Fig. 6. Mouse enabled interactive exploration such as rotation requires an image fetch from the database, while zooming and panning are image-based operations.

$n \times n$, for zooming in, and removing necessary (row, column) pairs for zooming out. Panning simply shifts the image side to side or up and down. It's important to note that all query-based interactions are still valid and respect these image-based interactions. This is at the discretion of the scientist.

The image-based approach provides interactive (12+ fps) response from the interactive exploration database on typically available scientific network throughputs/bandwidths. In addition, adding more flexibility to the interactive exploration database only requires saving more images, which in turn more densely samples the parameter space and camera space.

One interesting advantage of this approach over the traditional interactive post-processing approach is that for the image-based approach, the time to display one image is approximately the same time for any other image because the time to compute and render complex visualization and analysis objects has been amortized *in situ* within the simulation. For a traditional post-processing approach that computes visualization and analysis objects upon request, the wait time is extremely variable, ranging from seconds (rendering) to minutes (loading, pipeline selection, and computing). This inherent time to result bias produces a corresponding bias in what visualization and analysis objects are interactively explored. Specifically, because data sets are typically stored on disk as separate files for each time step, and the time to load a dataset is typically long, very little interactive exploration in time is done. Our interactive exploration database addresses these issues, encouraging both computationally intensive visualization and analysis objects and temporal exploration typically avoided in post-processing approaches.

Although there may be a concern about the time it will take to create this image collection, we believe that this is a manageable issue since: (1) We show, in the Results and Performance section, that the creation of images is constant, $O(1)$, in time as the problem size grows; (2) we have demonstrated in the past that *in situ* data visualization and analysis weakly scales; and (3) we expect that visualization and analysis operators will be accelerated to run on next-generation hardware at tens of frames per second [34].

### C. Metadata Searching

By leveraging the interactive exploration database, our image-based approach allows the analyst to execute metadata

queries (or browse analysis objects) to produce a prioritized sequence of matching results. The metadata, produced by the *in situ* analysis python script includes data properties of the simulation data, such as histograms, as well as image properties.

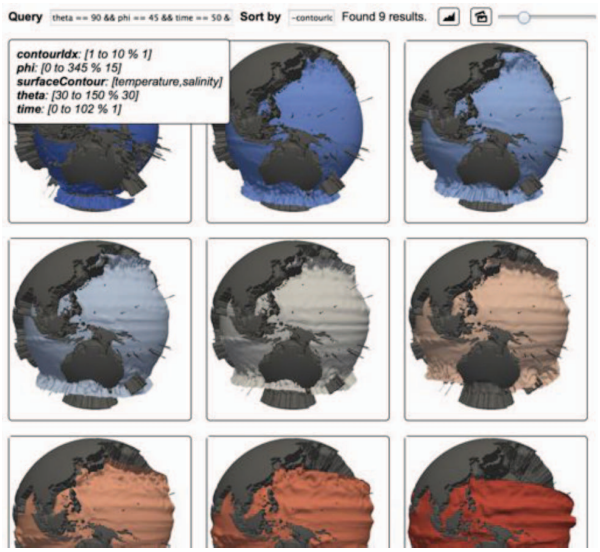


Fig. 7. A simple example of a prioritized metadata query, where the temperature contours at `time = 50` are ordered by increasing temperature isovalue for a particular camera.

In Figure 7, our desired query can be represented by leveraging the keywords in the camera metadata, `theta == 90 && phi == 45` the time metadata, `time == 50` and the visualization parameter metadata, `surfaceContour == 'temperature'`. These results would normally be presented in a sequential accessed, but unordered, manner with respect to the query. If the analyst would like to sort by increasing contour index, then the `-contourIdx` equation would present the results as desired. This is akin to the prioritized results returned from a Google search.

### D. Creation of New Visualizations and Content Querying

A core contribution of this work is the way in which new visualizations and queries are supported by the interactive exploration database. We utilize real-time compositing to create an experience similar to interactively exploring the simulation data itself, with significant additional capabilities only possible because of the image-based approach.

Adding visualization and analysis object compositing to our image-based approach framework allows the analyst to reason about his/her simulation results from visualization space as opposed to the explorations offered from image space rendering and sampling [14], [35], [20]. With the addition of visualization object compositing, the interactive exploration database retains the three elements and the two modes of interaction described in Figure 5.

For compositing, instead of a single image, the image-based approach framework creates an image sprite of the separate visualization objects to be interactively composited (see Figure 8).

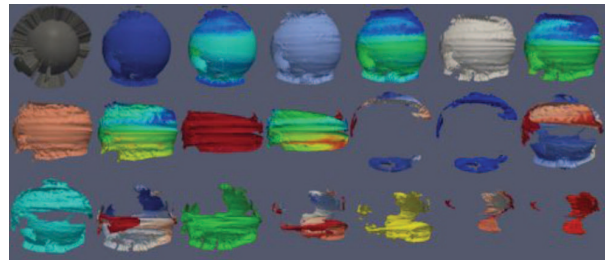

Fig. 8. For visualization object compositing, the image is replaced by an image set consisting of an *image sprite* file, a `composite` file, and a `query` file. This *image sprite* contains all the example MPAS-O images (22 images) except the background.

The visualization objects compositing provides an approximation of exploratory interaction with a raw data set. We can automatically display multiple objects from visualization space by selecting the associated image set for the (time, objects, and camera) selection from the database and composing them together. But, we do not require the analyst to do this manually through a database query. Instead, the analyst uses an interactive tool that emulates applications like VisIt and ParaView to simulate the experience of exploring simulation data.

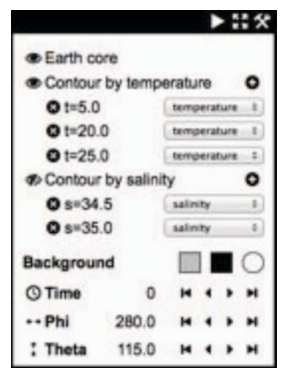


Fig. 9. The user interface of the scientist defined visualization pipeline for visualization object compositing.

In Figure 9, the *eyes* indicate visualization and analysis objects that can be interactively turned on and off, the *Background* allows for changing the background object color, and the *time* and *camera* control selection and animation for these elements.

The 22 visualization objects in the image sprite of Figure 8 can be combined, as demonstrated in Figure 10, into

$$\sum_{r=0}^{n} \frac{n!}{(n-r)!r!} = 2^n = 4,194,304$$

unique images.

While, in this example, a large number of these items occlude one another, this demonstrates the magnitude of the data space spanned by this set of elements for interactive exploration. All of the new image possibilities are a result of the compositing, which utilizes the `composite` file that is a per pixel linked list object order encoding created by comparing the z-buffers for each object.

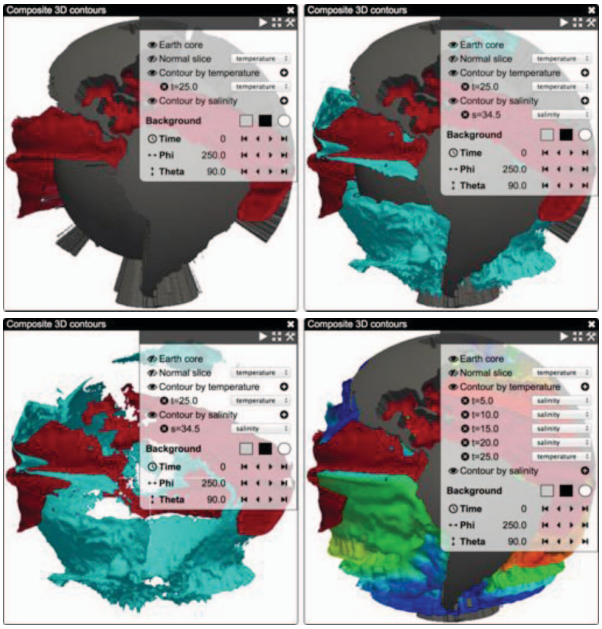


Fig. 10. A subset of the possible images the analyst can interactively create from this one viewpoint in time.

The interactive explorations database can be further extended through the output of image sets for a particular viewpoint consisting of the `composite` file, *image sprite* file, and possibly a "lighting" image and/or a raw floating point image, i.e., the simulation data values. For volume images, there has been recent work for creating images with changeable transfer functions [1]. For our opaque image sets, if we also save the simulation data associated with the visualization and analysis objects, then more capable visualization pipelines, such as the one presented in Figure 11, are possible using a number of rendering passes, including the lighting and color map passes.

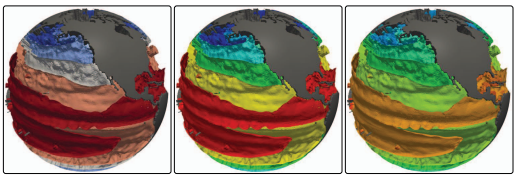

Fig. 11. Using lighting and color mapping, render passes and compositing enable more capable visualization pipelines such as changing color scale mapping for objects.

The visualization objects compositing infrastructure makes it easier to perform queries that search on the content of the image in the database [36]. For example, a query could be formulated that matches on the quality of the view of a particular isosurface value [8], [9].

The `query` file contains statistics on object mapping pixel coverage. For example, in Figure 12, if we want the pixel coverage of both object `A` and object `D` in a resulting image, then we would add up the pixel counts for all object order mappings where `A` proceeds `D`, or `D` does not exist, to get `16812` pixels with object `A`. We would also add up the pixel


```
{
    "dimensions": [500, 500],
    "counts": {
        "+" : 60868,
        "A+" : 16721,
        "AB+" : 89,
        "ABCJI+" : 1,
        "ABCJIH+" : 1,
        "DA+" : 135,
        "DJA+" : 61,
        "DJAC+" : 1,
        "DJACEH+" : 2,
        ...
    }
}
```


Fig. 12. An example `query` file.

counts for all object order mappings where `D` proceeds `A`, or `A` does not exist, to get `199` pixels with object `D`. Note that the "+" symbol is use to indicate the background object, ends each combination in the `query` file.

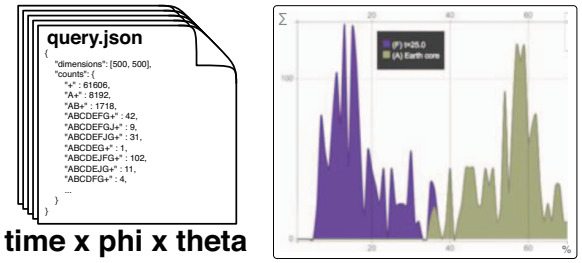


Fig. 13. The `query` files are examined to produce a histogram that depicts the percentage of coverage (x-axis) by the count (y-axis) of possible resulting images with that pixel coverage for each object independently.

The *query* histogram, shown in Figure 13, can be used to determine the reasonableness of proposed queries. In Figure 14, we want to perform a science-based visual query to determine the images (location on earth) with the largest isosurface representing a temperature of $25\,^{\circ}\mathrm{C}$ (warmest ocean) for the first four time steps. From the *query* histogram, we see that `F` represents the desired isosurface object and that the pixel coverage percentage achieves a maximum somewhere above $35.5\%$. Thus, our desired query can be represented by `(F > 35.5) && (time < 4)`. These results would normally be presented in a sequential accessed, but unordered, manner with respect to the query. If we would like to sort by the maximum pixel coverage biased by the increasing time step, then the `F - time` equation would present the results as requested.

Both Figures 14 and 15 demonstrate that we can ask simple questions and get the expected results. These image-based queries, however, show promise of answering much more complicated questions.

A second example used to test and demonstrate our image-based approach involves the xRage code, developed by LANL, which is a one-, two-,and three-dimensional, multi-material Eulerian hydrodynamics code for use in solving a variety of high deformation flow of materials problems. Examples present simulation results of the asteroid impact that created the Chicxulub crater in Mexico's Yucatan Peninsula [37].

In Figure 16, the scientist is looking for the best view of: the deep ground material threshold (`D`); the contour of the pressure wave in the ground material (`F`); slice colored by the

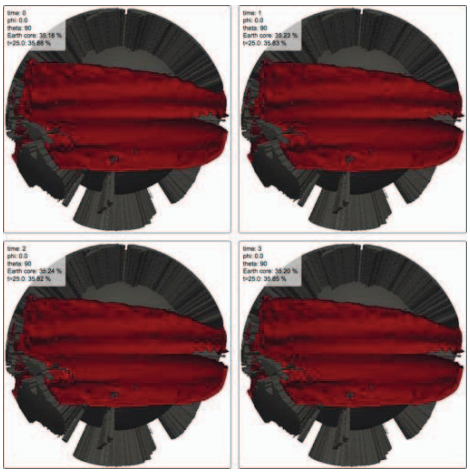

Fig. 14. Building an example science query and sorting algorithm by leveraging the *query pipeline* histogram.

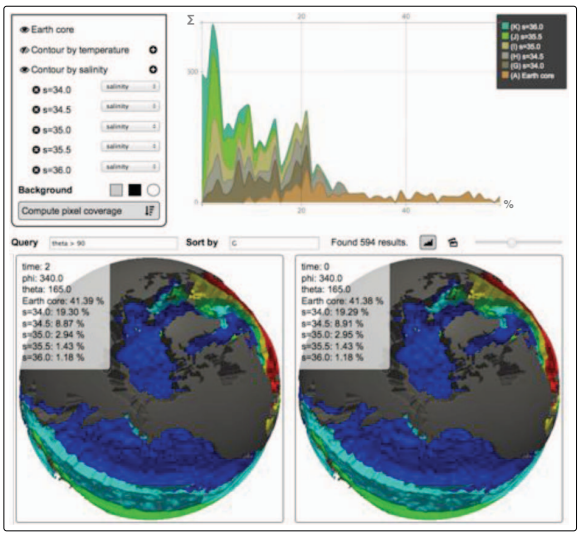


Fig. 15. Image-based query of a simple scientific question: Where is the largest visible mass of low salinity in the northern hemisphere?

velocity magnitude (H); and the threshold of asteroid material rooster tail (E). In this case, we can sort the entire data set by the equation E+F+H. The method uses visible pixel count as a metric for a 'hit', and then returns a 'possible matches' priority-sorted list. In this case, the query does return logical results showing the front and back views perpendicular to the slice at later time steps. The query results can be toggled between *query* statistics and the resulting images.

## IV. Results and Performance

The traditional GUI-based post-processing workflow suffers extremely variable wait times based on algorithm differences in loading, interactive pipeline selection, computing, and rendering. The problem quickly becomes intractable with large-scale data sets that require parallel resources. Interactive post-processing gives way to workflows requiring the scientist to write batch scripts and execute a second independent HPC

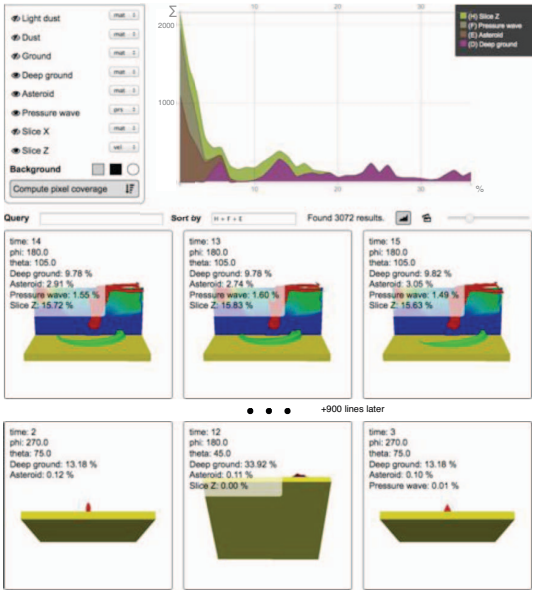


Fig. 16. Queries based on the image content can be used to search for qualitative results like "best view". The top three images show great views of the four items simultaneously. Later down in the search results are the bottom three images that obscure almost everything except the deep ground threshold.

workflow. In practice, this has an enormous impact on what visualizations and analysis methods are explored.

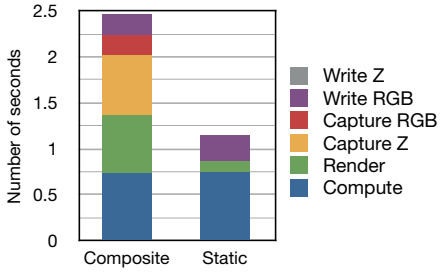


Fig. 17. The cost to produce one viewpoint of imagery for the interactive exploration database versus the production of an equal number of *in situ* images.

For our image-based approach, the time to display any image is approximately the same for any other image because the image generation time has been amortized in the single simulation workflow. The constant retrieval time (or the time to fetch an image from the database) of our image-based approach enables interactive exploration, and embedding the image capture *in situ* removes the additional HPC workflow. The remaining obstacle, for the scientist, of writing the *in situ* analysis script is greatly simplified by using our framework.

The cost of producing imagery for the interactive exploration database is roughly two times the cost of producing an equal number of *in situ* images (see Figure 17). Twice the cost is an astonishingly small price given that the interactive exploration database imagery, for example, generates 10 plus 1 background images in an image sprite to create $2^{11} = 2048$ unique images through post-processing, compared to simply

the 10 unique images.

One goal of *in situ* visualization and analysis is to reduce the time it takes a scientist to gain insight into the problem being simulated. From past studies, we see that ParaView Catalyst performs and scales well [38]. In a detailed study [39] led by the visualization group at Sandia National Laboratory, the *in situ* analysis showed weak scaling up to 64k cores on a variety of simulation codes. A second study comparing *in transit* and *in situ* analysis workflows [40], led by the same group, demonstrates the overall computational time (simulation + in situ analysis) scales for Sandia's CTH simulation code for various problem sizes and process counts. It rivals *in transit* approaches as the simulation size grows.

As the problem size increases and the number of processes increase, the benefits of using *in situ* analysis become more apparent [3].

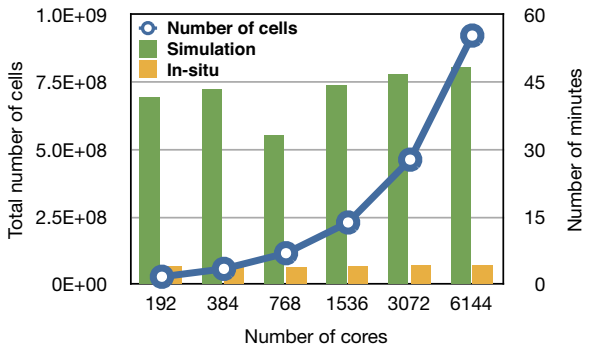


Fig. 18. The weak scaling results of compositing imagery produced every ten time steps up to 6144 cores along with the growth in problem size (number of cells).

For our performance results, we analyze the xRage code simulation results of the asteroid impact, mentioned previously. These simulations were the subject of a detailed study [41], performed by the LANL Data Science at Scale group, on *in situ* analysis image production. Their weak scaling study fixed the maximum number of cells at roughly 150K per core for AMR xRage runs. The study demonstrates that, as the problem sizes continue to grow, the image production of simple, single visualization and analysis objects remains constant.

Our weak scaling study examined the same xRage simulations from [41] using our approach to produce 10 different contour objects plus a background object at an image size of 500x500. Figure 18 shows the results of our weak scaling study (under normal cluster operation load) that demonstrates that the production of interactive exploration imagery remains constant.

Figure 19 demonstrates that significant data reduction can occur at relatively small core counts. It also demonstrates an interesting issue related to the size of the imagery as it appears on disk. Typically, extreme scale storage systems are tuned for large files. The image-based approach might benefit from a significantly different tuning of these file systems.

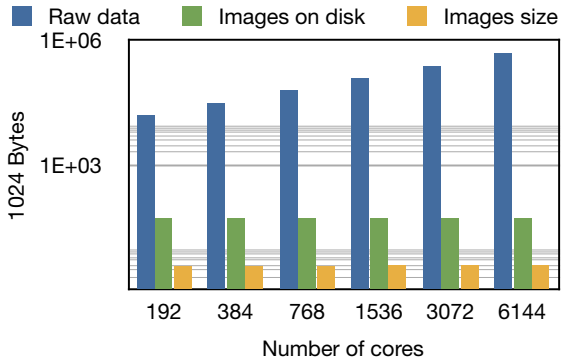


Fig. 19. Disk usage reduction comparing full xRage data files versus disk space occupied on the Panasas disk is 52MB with large block size versus the actual size of the images is approximately 3.5MB.

## V. Conclusion

We have developed a novel framework for an image-based approach to extreme scale data analysis, coupling visualization and analysis outputs with an image database query method to enable interactive exploration and metadata browsing. As implemented in this paper, the goals of the system are to 1) preserve important elements of the simulations, 2) to significantly reduce the data needed to preserve these elements, and 3) to offer as much flexibility as possible for post-processing exploration.

We have demonstrated the framework using an open-source tool and shown how a scientist can easily define a useful set of operations that will preserve important elements of the simulation. Our results demonstrate significant data reduction, especially when considering the size of data space that can be interactively explored in a post-processing workflow. The performance section demonstrates that the *in situ* production of the simulation outputs weakly scale and require constant time. Finally, we have shown the flexibility of the approach, which uses compositing to enable interactive visualization. As extreme scale computing continues to grow, we expect these methods can be tailored to effectively utilize compute resources to increase the value and effectiveness of interactive, explorable results that can be produced through *in situ* methods.

## Acknowledgment

This work was funded by Dr. Lucy Nowell, Program Manager for the Advanced Scientific Computing Research (ASCR) program office in the Department of Energy?s (DOE) Office of Science. The authors would like to thank Dr. Galen Gisler and Dr. Robert Weaver of Los Alamos National Laboratory (LANL) for the xRage input decks used in this paper. We would also like to thank Pat Fasel of LANL, and Dr. Andy Bauer of Kitware Inc., for their help and support in developing the MPAS *in situ* capability.

## References

[1] A. Tikhonova, C. D. Correa, and K.-L. Ma, "An Exploratory Technique for Coherent Visualization of Time-varying Volume Data," *Computer Graphics Forum*, vol. 29, no. 3, pp. 783–792, 2010.

[2] T. Tu, H. Yu, L. Ramirez-Guzman, J. Bielak, O. Ghattas, K.-L. Ma, and D. R. O'Hallaron, "From mesh generation to scientific visualization: an end-to-end approach to parallel supercomputing," in *SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, November 2006, p. 91.

[3] K. Moreland, N. Fabian, P. Marion, and B. Geveci, "Visualization on supercomputing platform level II ASC milestone (3537-1b) results from Sandia," Technical report SAND2010-6118, Sandia National Laboratories, Tech. Rep., 2010.

[4] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K.-L. Ma, "In situ visualization for large-scale combustion simulations," *Computer Graphics and Applications, IEEE*, vol. 30, no. 3, pp. 45–57, 2010.

[5] E. Luke and C. D. Hansen, "Semotus Visum: A Flexible Remote Visualization Framework," in *IEEE Visualization*, 2002, pp. 61–68.

[6] H. Shum and S. B. Kang, "Review of image-based rendering techniques," in *Visual Communications and Image Processing 2000*. International Society for Optics and Photonics, 2000, pp. 2–13.

[7] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita, "A Feature-Driven Approach to Locating Optimal Viewpoints for Volume Visualization," in *IEEE Visualization*, 2005, p. 63.

[8] U. Bordoloi and H.-W. Shen, "View Selection for Volume Rendering," in *IEEE Visualization*, 2005, p. 62.

[9] G. Ji and H.-W. Shen, "Dynamic View Selection for Time-Varying Volumes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1109–1116, 2006.

[10] K.-L. Ma, "In Situ Visualization at Extreme Scale: Challenges and Opportunities," *IEEE Comput. Graph. Appl.*, vol. 29, no. 6, pp. 14–19, Nov. 2009.

[11] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and K. Heitmann, "In-situ Sampling of a Large-Scale Particle Simulation for Interactive Visualization and Analysis," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1151–1160, 2011.

[12] J. Chen, I. Yoon, and W. Bethel, "Interactive, Internet Delivery of Visualization via Structured Prerendered Multiresolution Imagery," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 302–312, Mar. 2008.

[13] A. Kageyama and T. Yamada, "An approach to exascale visualization: Interactive viewing of in-situ visualization," *Computer Physics Communications*, vol. 185, no. 1, pp. 79–85, 2014.

[14] A. Tikhonova, C. D. Correa, and K.-L. Ma, "Visualization by Proxy: A Novel Framework for Deferred Interaction with Volume Data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1551–1559, 2010.

[15] A. Tikhonova, H. Yu, C. D. Correa, J. H. Chen, and K.-L. Ma, "A Preview and Exploratory Technique for Large-scale Scientific Simulations," in *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, ser. EG PGV'11. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2011, pp. 111–120.

[16] V. S. Subrahmanian and S. Jajodia, *Multimedia Database Systems: Issues and Research Directions*, 1st ed. Springer Publishing Company, Incorporated, 2012.

[17] J. Freire, D. Koop, F. Chirigati, and C. Silva, "Reproducibility using VisTrails," *Implementing Reproducible Computational Research*, 2014.

[18] K. Moreland, W. Kendall, T. Peterka, and J. Huang, "An Image Compositing Solution at Scale," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '11. New York, NY, USA: ACM, 2011, pp. 25:1–25:10.

[19] N. Elmqvist and P. Tsigas, "A taxonomy of 3d occlusion management for visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 5, pp. 1095–1109, 2008.

[20] J. Tao, J. Ma, C. Wang, and C.-K. Shene, "A Unified Approach to Streamline Selection and Viewpoint Selection for 3D Flow Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 393–406, 2013.

[21] O. Rübel, E. W. Bethel, Prabhat, and K. Wu, "Query-Driven Visualization and Analysis," in *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, ser. Chapman & Hall, CRC Computational Science, E. W. Bethel, H. Childs, and C. Hansen, Eds. Boca Raton, FL, USA: CRC Press/Francis–Taylor Group, Nov. 2012, pp. 117–144.

[22] K. Stockinger, E. W. Bethel, J. Shalf, and K. Wu, "Query-driven visualization of large data sets," in *16th IEEE Visualization 2005 (VIS 2005)*. IEEE Computer Society, 2005, p. 22.

[23] J. Bent, S. Faibish, J. Ahrens, G. Grider, J. Patchett, P. Tzelnic, and J. Woodring, "Jitter-free co-processing on a prototype exascale storage stack," in *Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on*. IEEE, 2012, pp. 1–5.

[24] Z. Sheppard. (2010, September) 5,000,000,000. [Online]. Available: http://blog.flickr.net/en/2010/09/19/5000000000/

[25] R. Hecht and S. Jablonski, "NoSQL evaluation: A use case oriented survey," in *Cloud and Service Computing (CSC), 2011 International Conference on*, December 2011, pp. 336–341.

[26] M. F. Deering, "The Limits of Human Vision," in *2nd International Immersive Projection Technology Workshop*, 1998.

[27] Community Earth System Model Working Group. (2014, March) CESM 1.0 experiments, data and diagnostics. [Online]. Available: http://www.cesm.ucar.edu/experiments/cesm1.0/

[28] A. Szalay, J. Gray, A. Thakar, B. Boroski, R. Gal, N. Li, P. Kunszt, T. Malik, W. O'Mullane, M. Nieto-Santisteban, J. Raddick, C. Stoughton, and J. vandenBerg. (2014, March) The SDSS DR1 Sky-Server: public access to a terabyte of astronomical data. [Online]. Available: http://cas.sdss.org/dr6/en/skyserver/paper/

[29] G. Lemson and VIRGO Collaboration, "Halo and Galaxy Formation Histories from the Millennium Simulation: Public release of a VO-oriented and SQL-queryable database for studying the evolution of galaxies in the Lambda-CDM cosmogony," 2006.

[30] I. Antcheva, M. Ballintijn, B. Bellenot, M. Biskup, R. Brun, N. Buncic, P. Canal, D. Casadei, O. Couet, V. Fine, L. Franco, G. Ganis, A. Gheata, D. G. Maline, M. Goto, J. Iwaszkiewicz, A. Kreshuk, D. M. Segura, R. Maunder, L. Moneta, A. Naumann, E. Offermann, V. Onuchin, S. Panacek, F. Rademakers, P. Russo, and M. Tadel, "ROOT a c++ framework for petabyte data storage, statistical analysis and visualization," *Computer Physics Communications*, vol. 180, pp. 2499–2512, December 2009.

[31] A. H. Squillacote, D. E. DeMarle, J. Ahrens, C. Law, B. Geveci, K. Moreland, and B. King, *ParaView Guide*, 1st ed. Kitware, Incorporated, 2007.

[32] C. L. Bajaj, V. Pascucci, and D. R. Schikore, "The Contour Spectrum," in *Proceedings of the 8th Conference on Visualization '97*, ser. VIS '97, 1997, pp. 167–ff.

[33] T. Ringler, M. Petersen, R. Higdon, D. Jacobsen, P. Jones, and M. Maltrud, "A multi-resolution approach to global ocean modeling," *Ocean Modelling*, vol. 69, pp. 211–232, 2013.

[34] L. Lo, C. Sewell, and J. Ahrens, "PISTON: A Portable Cross-Platform Framework for Data-Parallel Visualization Operators," in *Eurographics Symposium on Parallel Graphics and Visualization*, 2012.

[35] C. Wang and H.-W. Shen, "Information Theory in Scientific Visualization," *Entropy*, vol. 13, no. 1, pp. 254–273, 2011.

[36] K. Wu, W. Koegler, J. Chen, and A. Shoshani, "Using bitmap index for interactive exploration of large datasets," in *15th International Conference on Scientific and Statistical Database Management*, July 2003, pp. 65–74.

[37] M. Gittings, R. Weaver, M. Clover, T. Betlach, N. Byrne, R. Coker, E. Dendy, R. Hueckstaedt, K. New, W. R. Oakes, D. Ranta, and R. Stefan, "The RAGE radiation-hydrodynamic code," *Computational Science & Discovery*, vol. 1, no. 1, p. 63pp, 2008.

[38] H. Karimabadi, P. O'Leary, B. Loring, A. Majumdar, M. Tatineni, and B. Geveci, "In-situ visualization for global hybrid simulations," in *the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery (XSEDE '13)*. Association for Computing Machinery, July 2013, p. 8.

[39] N. Fabian, K. Moreland, D. Thompson, A. C. Bauer, P. Marion, B. Geveci, M. Rasquin, and K. E. Jansen, "The ParaView Coprocessing Library: A Scalable, General Purpose In Situ Visualization Library," in *IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV) 2011*. Institute of Electrical and Electronics Engineers, October 2011, pp. 89–96.

[40] D. Rogers, K. Moreland, R. Oldfield, and N. Fabian, "Data Co-Processing for Extreme Scale Analysis Level II ASC Milestone (4547) from Sandia," Technical report SAND2013-1122, Sandia National Laboratories, Tech. Rep., March 2013.

[41] J. M. Patchett, J. P. Ahrens, B. Nouanesengsy, P. K. Fasel, P. O'Leary, C. M. Sewell, J. L. Woodring, C. J. Mitchell, L.-T. Lo, K. L. Myers, J. R. Wendelberger, C. V. Canada, M. G. Daniels, H. M. Abhold, and G. M. Rockefeller, "LANL CSSE L2: Case Study of In Situ Data Analysis in ASC Integrated Codes," Technical report LA-UR-13-26599, Los Alamos National Laboratories, Tech. Rep., 2013.