2011 11th IEEE International Conference on Data Mining

S-preconditioner for Multi-fold Data Reduction with Guaranteed User-controlled Accuracy

Ye Jin*‡¶, Sriram Lakshminarasimhan*‡¶, Neil Shah*, Zhenhuan Gong*, C.S. Chang†,
Jackie Chen§, Stephane Ethier**, Hemanth Kolla§, Seung-Hoe Ku†,
Scott Klasky‡, Robert Latham‖, Robert Ross‖, Karen Schuchardt‡‡, Nagiza F. Samatova‡*††

* North Carolina State University, NC 27695, USA
† New York University, New York, NY 10012, USA
‡ Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
§ Sandia National Laboratory, Livermore, CA 94551, USA
** Princeton Plasma Physics Laboratory, Princeton, NJ 08543, USA
‖ Argonne National Laboratory, Argonne, IL 60439, USA
‡‡ Pacific Northwest National Laboratory, Richland, WA 99352, USA
¶ Authors contributed equally
†† Corresponding author: samatova@csc.ncsu.edu

***Abstract*— The growing gap between the massive amounts of data generated by petascale scientific simulation codes and the capability of system hardware and software to effectively analyze this data necessitates data reduction. Yet, the increasing data complexity challenges most, if not all, of the existing data compression methods. In fact, *lossless* compression techniques offer no more than 10% reduction on scientific data that we have experience with, which is widely regarded as effectively incompressible. To bridge this gap, in this paper, we advocate a transformative strategy that enables fast, accurate, and multi-fold reduction of double-precision floating-point scientific data. The intuition behind our method is inspired by an effective use of *preconditioners* for linear algebra *solvers* optimized for a particular class of computational "*dwarfs*" (e.g., dense or sparse matrices). Focusing on a commonly used multi-resolution wavelet compression technique as the underlying "solver" for data reduction we propose the $S-$preconditioner, which transforms scientific data into a form with high global regularity to ensure a significant decrease in the number of wavelet coefficients stored for a segment of data. Combined with the subsequent $EQ-$calibrator, our resultant method (called S-Preconditioned EQ-Calibrated Wavelets (SPEQC–WAVELETS)), robustly achieved a 4- to 5-fold data reduction—while guaranteeing user-defined accuracy of reconstructed data to be within 1% point-by-point relative error, lower than 0.01 Normalized RMSE, and higher than 0.99 Pearson Correlation. In this paper, we show the results we obtained by testing our method on six petascale simulation codes including fusion, combustion, climate, astrophysics, and subsurface groundwater in addition to 13 publicly available scientific datasets. We also demonstrate that application-driven data mining tasks performed on decompressed variables or their derived quantities produce results of comparable quality with the ones for the original data.**

***Keywords*-preconditioners for data mining; data reduction; data mining over decompressed data; in situ data analytics; extreme-scale data analytics**

I. INTRODUCTION

Data reduction is quickly becoming a top-notch priority in the field of exascale science. Scientific simulation codes like Flash, GTS, S3D, and XGC that are scalable to petascale supercomputers produce on the order of tera- to peta-byte datasets per run. As these simulations codes are allocated only a limited number of hours to perform the simulation at the scales of hundreds and thousands of cores, to allievate the time spent on writing the data out, compression becomes a necessity rather than an option. However, existing data compression methods are hardly suitable for compressing random and noisy spatio-temporal floating-point scientific data, and thus have lead to the belief that this type of data is uncompressible [1]. In fact, *lossless* compression techniques [2], [3], [4], [5] that are specifically designed for fast online compression [2], [4] are capable of reducing such data by no more than 10% of its original size.

In an attempt to alleviate this problem, Lindstrom and Isenberg [4] introduced a *lossy* compression method that functions by converting floating-point values to integers and recording the delta between a predictive estimator and the actual data points. They provide the option of discarding least significant bits of the delta and making the compresion lossy. However, to achieve the same compression rate achieved using the method proposed in this paper, we would need to discard the majority of the bits (making the resulting compression very lossy and inaccurate).

More traditional multi-resolution techniques like Wavelet-based compression have been primarily used for visualization, geometric modeling, signal denoising and filtering, feature preservation and selection, multi-level spatial data-mining, and clustering (see, for example, [6], [7], [8]). In the context of *lossy* data compression, such techniques have been targeting specific types of scientific data, such as 1-dimensional ECG signals [9], 2-dimensional piecewise smooth images [10], [11], 3-dimensional hyperspectral images [12]), and others [13], [14].

However, no effective compression techniques that function

1550-4786/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDM.2011.138

TABLE I
CHARACTERISTICS OF SIMULATION OUTPUT DATA SETS FROM SIX APPLICATIONS

Code	Application	Variable(s)	Is Double?	Reference
S3D	Combustion	Temperature, Velocity	No	[15]
XGC1	Fusion Plasma Edge	Flux, Temperature	Yes	[16]
GTS	Fusion Plasma Core	Density, Potential	Yes	[17]
GCRM	Climate	Vorticity	No	[18]
SPH	Smoothed Particle Hydrodynamics	Density	No	[19]
Flash	Astrophysics	Velocity	Yes	[20]

TABLE II
SUMMARY OF GTS AND XGC OUTPUT DATA BY DIFFERENT CATEGORIES.

Category	Write Frequency	Read Access	Size/Write	Total Size
C&R	Every 1-2 hours	Once or never	A few TBs	≈TBs
A	Every 10^{th} time step	Many times	A few GBs	≈TBs
V&V	Every 2^{nd} time step	A few times	A few MBs	≈GBs

without directly exploiting both a specific spatial and temporal correlation of data in the context of any underlying data model have been reported in literature, to the best of our knowledge. These three major requirements are crucial for any *in-situ* (i.e., in-tandem with the simulation code) data reduction scheme that is communication-free, energy-efficient, and minimally-disruptive to simulation codes. For simulation codes, meeting the first requirement would eliminate inter-process communication; meeting the second requirement would avoid memory-intensive global-context (across all time steps) processing; and finally, meeting the third requirement would offer end-users a desirable flexibility for utilizing the model optimized for their simulation codes without considering the data model required by the data reduction technique. To fill this niche, we propose a method (called SPEQC–WAVELETS) that effectively meets these needs, guarantees that user-specified accuracy constraints are respected, and demonstrates negligible or no adverse effect on subsequent data mining and analysis tasks performed over the decompressed data.

II. MOTIVATING EXAMPLES

In this paper, most of the work done in the context of in-situ data reduction is performed on results from several scientific simulation codes, including GTS [17] and XGC1 [16] for particle-based simulations of fusion plasmas for studying plasma micro-turbulence in the core and in the edge, respectively, of magnetically confined fusion plasmas of toroidal devices in nuclear reactors; S3D [15] for direct numerical simulations of turbulent combustion, and others (see Table I). For example, the Global Cloud Resolving Model (GCRM) [18] is being designed to simulate both weather and climate at cloud resolving scales in the sub 2km range. Understanding behavior of clouds is of immense importance and scientists will be interested in analyzing high temporal frequency data that matches the lifecycle of clouds, ideally, on the time scales in the order of minutes. However, tradeoffs against bandwidth and storage capacity will be required. These tradeoffs can be met through several techniques such as grid sampling, high temporal frequency I/O for regions of interest, and both lossless and lossy compression techniques.

The types of data produced by the simulation can be broadly classsified into three types: (1) Checkpoint and restart (C&R) data which is used to restart a simulation in the event of a failure (2) Analysis (A) data which is used for performing exploratory analytical processing and/ or visualization (3) Verification and validation (V&V) data which is used to affirm the smooth running of the simulation.

The analysis data is inherently lossy, as scientists resort to skipping timesteps instead of saving every timestep, in order to keep the data generated within manageable proportions. Hence, this analysis can still tolerate some form of approximation, unlike C&R data. Additionally this analysis data is repeatedly accessed by scientists to perform exploratory hypothesis testing, and visualization, and used by physics analysis codes. Large-scale data reduction would thus make the process of analysis much more efficient. Hence, this is the main focus of SPEQC-WAVELETS.

III. METHOD

The intuition behind our method is inspired by the way linear algebra problems are often being solved. Consider a somewhat simplified example to illustrate the basic idea. If one needs to find a determinant of a large matrix, then transforming the matrix to an upper- or lower-diagonal form would reduce a complex task of calculating the determinant to a simpler task of multiplying the diagonal elements of the transformed matrix. Likewise, if a matrix is known to be sparse, then transforming the matrix to a block-diagonal form would enable to decompose the same task into the product of similar tasks but over smaller size matrices. Of course, for real application matrices, an "ideal" block-diagonalization maybe hard to obtain, and an approximation to the ideal with a few non-zero elements off the diagonal blocks would require some "re-calibration" to the approximated solution.

Thus, solving a linear algebra problem, such as the one illustrated above, would require knowledge of the *data model*/computational dwarf (e.g., sparse or dense matrix), the *preconditioner* (e.g., diagonalization or block-diagonalization)

that would transform the original problem instance into the instance, for which the target *solver* (e.g., calculation of the determinant) offers the optimized performance for the given dwarf, with a possible need for running a *calibrator* to adjust the solution due to some impurities in the "ideal" problem instance.

Algorithm 1: Data Reduction's Framework

Input:
D—input data to be compressed.
S—window size.
P_{pref}—end user's preferences of Pearson Correlation, NRMSE and Relative Error per Point.
Output:
C—Compressed Data.

```
1 Window_Set ← Window Splitter(D, S).
2 forall Window in Window_Set do
      /* Window' is preprocessed window,
         C_Window is compressed window.  */
3     {Window', Index} ← S−preconditioner(Window)
4     C ← {C, Index}
5     C_Window ← Adaptive Error-Bounding
      Iterator(Window', P_pref)
6     C ← {C, C_Window}
      /* QE is quantized errors for each
         point in one window             */
7     QE ← EQ−calibrator(C_Window, Window',
      P_pref)
8     C ← {C, QE}
```

The SPEQCWavelets framework for double precision floating-point scientific data reduction consists of four main continuous procedures, which are illustrated in algorithm 1. In this paper, we present a method that offers competative performance to the traditional multi-resolution Wavelets-based solver via innovations underlying the proposed $S-$preconditioner and $EQ-$calibrator (both detailed in the sections below). For the solver, we choose to utilize Quadratic Spline (QS) W-Matrices Wavelet Transform [21], [22] for a number of reasons. First, the inverse transformation can be efficiently implemented [22]; second, this approach allows for efficient processing of arbitrary length signals [22]; third, the QS transformation has been demonstrated to performs better than the D_4 transformation [23], [24] for some types of signals; finally, the quadratic MRA (Multi-Resolution Analysis) was shown to be superior to the cubic one in both decomposition and reconstruction times of semi-orthogonal spline wavelets, while permitting greater accuracy for a corresponding number of terms in the truncated IIR (Infinite Impulse Response) filters used to perform decomposition [24], [23].

A. $S-$preconditioner

Considering a scientific data set as a linearized vector (or signal) of floating-point double-precision values, we can characterize the performance of wavelet transforms by the class of functions that best approximate this signal; we could also consider the signal in both the frequency and time domains. Theoretically, the rapid decrease of the wavelet at the zero frequency implies a rapid decrease of the wavelet coefficients at small scale. Likewise, in terms of the time representation, high global regularity of the underlying function and many vanishing moments of the wavelet implies a rapid decrease in the number of wavelet coefficients [25]. The higher number of vanishing moments leads to a higher compression rate.

Equipped with such theoretical insights, we find that our preconditioner should ideally transform the original signal to the one that is best approximated by a function of high global regularity. In this paper, we propose a preconditioner (called the $S-$preconditioner) that sorts the original input signal. Sorting will allow the data distribution in the spatial domain to be transformed from a quickly oscillating signal (Fig. 1, **A**) to a monotonously growing one (Fig. 1, **B**). The idea behind this sorting pre-conditioner for a data reduction technique is that Wavelet transforms of monotonously curved functions can provide more accurate models than the original random functions. Figure 1 presents the huge divergence in how precisely (**D**) or inaccurately (**C**) the decoded lossy compressed data approximates the original data when applying Wavelet compression with and without the $S-$preconditioner respectively.

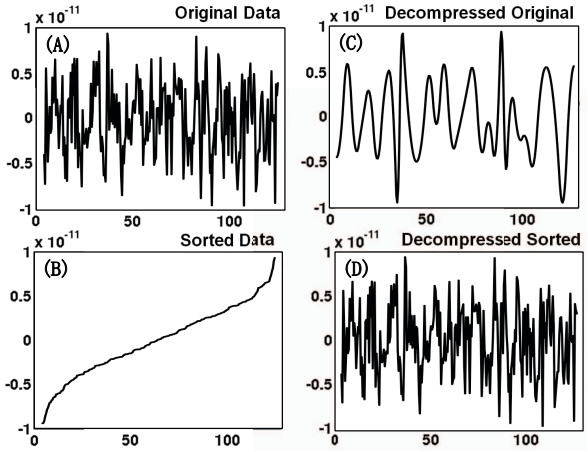

Fig. 1. A slice of GTS Potential: (A) original signal; (B) sorted signal; (C) decoded after Wavelets transformation of the original signal; and (D) decoded after Wavelets transformation of the sorted signal.

B. Window Splitter

While the $S-$preconditioner benefits us in reducing the number of Wavelet coefficients to store and enhancing the accuracy of the reconstructed data, it also forces us to pay the storage cost for saving the indices of the original data. Namely, sorting the input vector will reorder the vector elements via some permutation of its indices. Hence, we need to keep track of the permuted index (I_p) so that we can assocate the decompressed sorted vector back to the original vector by using its correct index. Note that each index value of

the vector with N elements would require $log_2 N$ bits of storage. Thus, the vector length N is the only factor that determines the storage requirements for the permuted index I_p. Such a dependence brings a couple of issues. First, the 64-bit computer architecture constrains $N \leq 2^{64}$. Otherwise, the input vector V representing the linearized data has to be split into smaller size vectors. Alternatively, one can find the window size W_0 that optimizes the overall compression ratio of the compressed vector V_c defined by Equation 1:

$$CR(V) = \frac{|V_c|}{|V|} \times 100\% \tag{1}$$

For example, splitting the entire vector V into fixed-sized windows of size $W_0 = 256 = 2^8$ would require only *one* byte to store an index of the vector element in each window, and only $S_{double}(I_p) = \frac{size(byte)}{size(double)} = 12.5\%$ of the original storage would be used for I_p index for double-precision data. Likewise, for single-precision data, 25% of the original usage would be used for the index ($S_{single}(I_p) = \frac{size(byte)}{size(float)} = 25\%$). Table III illustrates such trade-offs along with the overall storage requirement for keeping track of the index of the sorted vector elements.

TABLE III
WINDOW SIZES WITH STORAGE COST FOR INDICES

Bytes per Index	W_0	S_{double} (%)	S_{single} (%)
1	256	12.5	25
2	65,536	25	50
3	16,777,216	37.5	75

C. $EQ-$calibrator

The $S-$preconditioner ensures accurate Wavelets-based approximation only on a *per window* basis but not on a *per point* basis. As a result, in certain locations, the Wavelets-estimated data deviates from the actual by a margin exceeding the defined tolerance. To alleviate this problem, we next optimize the SPEQC–WAVELETS's performance in terms of the percentage of the relative point-by-point error (ϵ) at each index i between the original vector V and the Wavelet-decoded vector V' defined by Equation 2:

$$\epsilon_i = \frac{v_i - v'_i}{v'_i} \times 100\%. \tag{2}$$

While the number of such location points is reasonably low due to accurate fitting achieved by Wavelets on $S-$preconditioned data, SPEQC–WAVELETS guarantees that the user-specified point-by-point error is respected by calibrating/adjusting the approximated values using the *multi-level error quantization* strategy detailed next.

Via this strategy, we capture the integer representation of the relative deviation between the approximated data and the original data in an in-situ manner. While ϵ is user-defined with respect to the original data, we capture ϵ with respect to the approximated data due to the high correlation between relative errors of adjacent locations in the sorted window. This lends well to standard compression libraries, and produces a drastic reduction in size upon compression. The transformed data is reconstructed by capturing $v'_i \times (1+\epsilon_i/100)$. We then evaluate whether the reconstructed value is within user-defined ϵ. If not, then we proceed with the second-level of error encoding by capturing the relative error again, but this time, between $v'_i \times (1+\epsilon_i/100)$ and v_i. The levels of error capture are repeated until the data satisfies the user specified bounds. These extra levels of error encoding add a non-significant overhead to the storage size, since the $S-$preconditioner ensures that relatively few points ($< 2\%$) require further than a single-level of error encoding.

Quantization of these level-one errors into 32-bit integers results in a large degree of repetition, where majority of the values lie between [-2, 2]. These integer values lend themselves to high compression rates ($1\% - 15\%$) with standard lossless compression libraries, such as bzip, gzip, etc. Higher-level quantized errors induce an almost negligible storage overhead, usually less than 1%. Altogether these quantized relative errors require $S_\epsilon(EQ)$ storage (expressed in terms of the percentage of the original data size) after applying lossless compression and are stored along with the permuted index (I_p) during encoding. Upon decoding, applying these relative errors ensures decompressed values to be within a user-defined ϵ point-by-point relative error.

D. Adaptive Error-Bounding Iterator

Once applied to a vector V of the linearized double-precision data, the three major components—$S-$preconditioner, $EQ-$calibrator, and Wavelets compressor—will additively contribute to the SPEQC–WAVELETS's overall compression ratio, namely, via Equation 3:

$$CR(V) = S_{double}(I_p) + S_\epsilon(EQ) + S(Coef), \tag{3}$$

where $S(Coef)$ is the percentage of the original data storage occupied by Wavelets coefficients.

Next, we describe our adaptive and efficient strategy on optimizing $S(Coef)$. Note that $S(Coef)$ per window depends on the number of Wavelets coefficients stored for the given window. This number, in turn, is dependent on both the data and the user-defined accuracy requirements for each data window. Here, we let the end-user define accuracy in terms of the two widely used performance metrics: Pearson correlation (ρ) and Normalized Root Mean Square Error ($NRMSE$) between the original and Wavelets-based reconstructed data.

Since fixing the number of coefficients *a priori* is not an option, we propose an adaptive strategy that iteratively finds the minimal number of coefficients that satisfy user-defined accuracy constraints. Coefficients are obtained at the maximal wavelet resolution level of $log_2(W_0)$ (see algorithm 2). It ensures the efficient execution and a fast convergence of the iterative search by taking advantage of the *knowledge priors* about the optimal coefficient number from previously seen data windows and of the *dynamic updates* to calculations of the user's accuracy metrics.

A binary search of the optimal number of Wavelets coefficients for the window of size W_0 would require $log_2 W_0$

Algorithm 2: Adaptive Error-Bounding Iterator

Input:
W—preprocessed window.
PC_{pref}—end user's preferences of Pearson Correlation
N_{pref}—end user's preferences of NRMSE
NC—Number of stored wavelet coefficients of the previous window, default value is 8.
ΔNC—prior knowledge learned from previous windows of the trade-off between accuracies (Pearson Correlation and NRMSE) and number of coefficients to be stored.
Output:
C—Compressed Data (Coefficients) for the window.

```
/* W' is the wavelet transformed data
   at resolution level 8                    */
```
1 $W' \leftarrow$ Wavelet Transform(W)
2 $\tau \leftarrow$ the NC_{th} largest value in W.
3 **forall** w ***in*** W **do**
4 **if** $w < \tau$ **then**
5 $w \leftarrow 0$
6 $C \leftarrow \{C,\ w\}$
7 **while** N_{pref} *is not satisfied by* C **do**
```
   /* N is the NRMSE value of C            */
```
8 $\Delta N \leftarrow N - N_{pref}$
9 $NC \leftarrow$ Coefficients Update(ΔN, ΔNC)
10 $\tau \leftarrow$ the NC_{th} largest value in W. **forall** w ***in*** W **do**
11 **if** $w < \tau$ **then**
12 $w \leftarrow 0$
13 $C \leftarrow \{C,\ w\}$
14 **if** PC_{pref} *is not satisfied by* C **then**
15 Similar iteration of lines 9 - 16, but replace N with PC, N_{pref} with PC_{pref}.
16 Update NC and ΔNC

iterations of inverse Wavelet transformation and accuracy calculations, thus inducing a large overhead. To improve the efficiency of adaptive compression, we employ two strategies: the first is to base accuracy analysis primarily around the $NRMSE$ metric, and the second is to learn from previous windows to reduce the average number of iterations.

There are three main reasons behind applying the first strategy: (1) while checking the accuracy of the reconstructed data in the adaptive compresion process, if one of the two error thresholds is crossed, then the other doesn't need to be calculated; (2) the Pearson correlation metric is linearly related to the $NRMSE$ (low $NRMSE$ indicates high ρ); and (3) $NRMSE$ is computationally less demanding than ρ. Thus, it makes sense for $NRMSE$ to be the primary metric as opposed to the Pearson correlation.

Regarding our second strategy, to reduce the number of iterations of adaptive compression, we leverage the following two assumptions: the first is that windows in the same data set are spatially relevant, and the second is that learning from previous windows will likely identify the optimal balance with fewer attempts than the binary search. In our case, the first assumption is satisfied, since the data we are working with is spatio-temporal scientific data and the data per each time step is spatially relevant. The second assumption also holds according to our experiments (see Results Section).

IV. Results

For our performance evaluations, we have chosen two types of data to test: (a) simulation output data from six scientific application codes running on supercomputers at peta-scale (see Table I) and (b) 13 publicly available scientific datasets of one-dimensional binary sequences of double-precision floating-point numbers. The latter are commonly used for performance analysis ([2]) and include data recorded by scientific instruments, numerical simulations, and messages sent by a node in a parallel system running a computational fluid dynamics simulation. Throughout our results section, the abbreviations and contexts in which we mention various datasets will be referred to in accordance with this section.

We evaluate SPEQC–WAVELETS using compression ratio (CR) as the data reduction metric and normalized root mean standard error ($NRMSE$), Pearson correlation (ρ), and point-by-point relative error ratio (ϵ) between the original and decompressed data as the accuracy metrics. [Note that achieving $NRMSE \sim 0.0$, $\rho \sim 1.0$, $\epsilon \sim 0.0\%$, and $CR \sim 12.5\%$ for double-precision and $CR \sim 25\%$ would indicate excellent performance.] Throughout this section, unless stated others, the default values are the following: window size $W_0 = 256$, $\rho \geq 0.99$, $NRMSE \leq 0.01$, and $\epsilon \leq 1.0\%$.

A. Scientific Data Analysis and Data Mining

In this section, we show that lossy compression induces negligible or no adverse effect on subsequent data mining and analysis tasks performed over the decompressed data or the data derived from multiple decompressed variables compared to the analysis results from the corresponding original data.

1) Analysis of XGC: The XGC-1 is a full distribution gyrokinetic ion-electron particle code specifically designed for simulation of edge plasmas in nuclear reactor [26]. In this section, we analyze the fidelity of SPEQC–WAVELETS-compression on variables derived from turbulence data produced by the simulation. Figure 2 shows the variation of temperature values along poloidal and temporal resolutions over original (left), and SPEQC–WAVELETS-decompressed data (center). The rightmost image shows the absolute difference over original and approximated data on an order of magnitude lower scale. From the images created from approximate data, it is evident that SPEQC–WAVELETS-compressed data captures the same physical phenomena as the original data with high fidelity.

2) Analysis of S3D: The S3D simulation code [15] is a flow-solver that enables direct numerical simulation (DNS) based on first principles for turbulent combustion. In this section, we compare the accuracy of SPEQC–WAVELETS-compression on variables obtained from the time-dependent

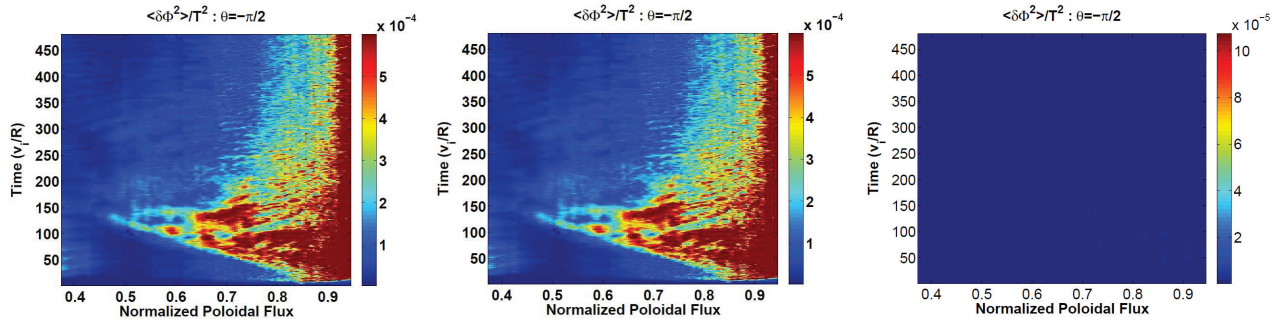

Fig. 2. Comparison of normalized turbulence intensity values in XGC field data for a fixed radial zone, across temporal and poloidal dimensions. The original (left) and SPEQC–WAVELETS (middle) decompressed data, and the absolute difference (right) are shown. Note that the figure on the right is on an order of magnitude lower scale.

turbulent flame S3D simulation of over 20 million grid points and 50 timesteps. Figure 3 shows the comparison between original and SPEQC–WAVELETS-decompressed data for the temperature variable. Our analysis revealed that the temporal average for temperature exhibited a mean relative error of only 0.04%. For variables u, v and w velocities (averaged over 50 timesteps), the relative error exceeded 1% for only 0.38% of the 20 million grid points - the average relative error was $\approx 0.4\%$.

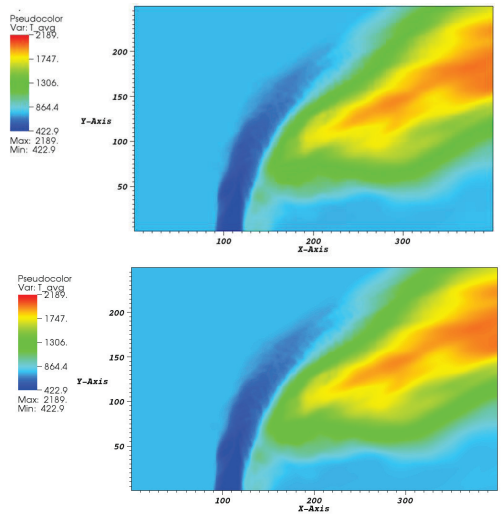

Fig. 3. Comparison of intensity plots of temperature values over 20 million points in S3D data for original (top) and decompressed (bottom) data.

3) Aggregate Errors: k-Means Clustering: To analyze the effect of $EQ-$calibrator, we performed k-means clustering on SPEQC–WAVELETS-decompressed and the original data from GTS and Flash simulations. The centroids for the k-clusters were chosen randomly and the algorithm was run for 100 iterations, for varying values of k. While increasing the number of k values increased the misclassification error as well, the resulting error rates still remained $< 1\%$ for both GTS and Flash datasets 4. In fact, for GTS data, the misclassification error rates were less than 0.1% at $\epsilon = 0.1$.

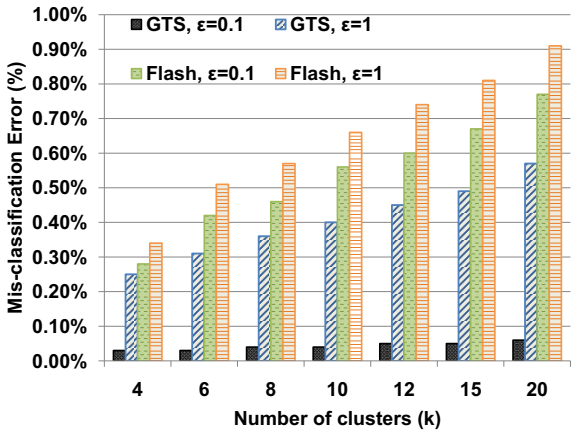

Fig. 4. Comparison of k-means clustering on 4096 point sample from Flash and GTS data over varying ϵ, and k values. k-means was performed on 100 iterations, with randomly chosen centroids.

B. *Effect of $S-$preconditioner*

Applying $S-$preconditioner alone allows SPEQC–WAVELETS for almost 85% data reduction ($CR = 15\%$), while still ensuring high quality per-window $\rho\sim$0.99 and $NRMSE\sim$0.02 using GTS Density data set over 320 time steps. In contrast, at the same CR, Daubechies Wavelets without $S-$preconditioner (**DW**) performed with $\rho<$0.75 and $0.08<NRMSE<0.15$ (see Figure 5).

Table IV shows the difference in the average $NRMSE$ and ρ correlation values between SPEQC–WAVELETS and DW for 25 datasets, for the fixed $CR \sim 20\%$ ($W_0 = 1024$), .

TABLE IV
SPEQC–WAVELETS VS. DAUBECHIES WAVELETS (DW) AT $CR = 20\%$ (†SINGLE-PRECISION AT $CR = 32\%$)

Datasets	ρ		$NRMSE$	
	DW	SPEQC–WAVELETS	DW	SPEQC–WAVELETS
msg_bt	0.729	**0.998**	0.239	**0.013**
msg_lu	0.729	**0.998**	0.244	**0.016**
msg_sp	0.706	**0.998**	0.322	**0.017**
msg_sppm	0.832	**0.988**	0.272	**0.093**
msg_sweep3d	0.924	**0.999**	0.064	**0.006**
num_comet	0.969	**0.999**	0.036	**0.002**
num_control	0.873	**0.999**	0.120	**0.003**
num_brain	0.820	**0.997**	0.081	**0.004**
num_plasma	0.703	**0.999**	0.260	**0.005**
obs_error	0.727	**0.997**	0.225	**0.009**
obs_info	0.695	**0.997**	0.247	**0.014**
obs_spitzer	**0.998**	0.997	0.002	**0.000**
obs_temp	0.789	**0.999**	0.099	**0.002**
GTS_potential	0.886	**0.999**	0.039	**0.002**
GTS_density	0.787	**0.998**	0.091	**0.011**
Flash_velx	0.886	**0.999**	0.139	**0.004**
Flash_vely	0.807	**0.998**	0.121	**0.014**
Flash_gamc	0.883	**0.998**	0.088	**0.014**
XGC1_flux	0.743	**0.997**	0.197	**0.001**
XGC1_temp	0.994	**0.999**	0.027	**0.007**
XGC1_exb_shear	0.859	**0.995**	0.071	**0.013**
S3D_temperature†	**0.999**	0.999	**0.001**	0.001
S3D_velocity†	**0.999**	0.999	**0.001**	0.011
GCRM_vorticity†	0.858	**0.983**	0.069	**0.013**
GWA_density†	**0.999**	0.999	**0.001**	0.009

We report the average correlation of 400 windows that provide poor performance on $NRMSE$ and ρ correlation in this case. While DW exhibits a < 0.75 correlation value on several of the datasets, SPEQC–WAVELETS remain consistently above 0.99. Although, the standard deviation values are not reported here, the SPEQC–WAVELETS exhibited $\approx 10^{-5}$ standard deviation that was almost two orders of magnitude better than DW. The variation in DW performance on correlation can be attributed to the noisy nature of scientific data. Hence, a larger number of coefficients is needed in these cases to provide an approximation comparable with SPEQC–WAVELETS . Note that the $NRMSE$ values for DW are consistently an order of magnitude higher than for SPEQC–WAVELETS.

C. Effect of $EQ-$calibrator

Although there are a few data sets on which both SPEQC–WAVELETS and DW perform comparably well (see Table IV), DW does not guarantee the point-by-point deviation to be small. In fact, as illustrated in Figure 7, there could be a number of points (more than 17% (4%)) with a large relative point-by-point error rate ($\epsilon \leq 2.0\%$ (10.0%)). SPEQC–WAVELETS addresses this problem by applying $EQ-$calibrator to ensure that the error remains within a small, user-defined range (e.g., $\epsilon \leq 1.0\%$).

Error quantization underlying the $EQ-$calibrator may introduce an additional storage overhead, which is typically higher for smaller values of ϵ. Figure 8 shows that increasing ϵ from 0.1% to 1.0% induces a 6% reduction in storage requirement for both linear and nonlinear stages of the GTS Potential simulation. However, the storage requirement hardly changes as ϵ increases from 1% to 5% in a 1% increment. Figure 6 shows compression ratios for both the coefficients and the quantization errors over the entire simulation run using the GTS fusion and Flash astrophysics simulation codes. For GTS Potential data, the compression ratio remains almost the same across all stages of the simulation. With Flash data most relative errors are 0's after error quantization. Compressing these values results in negligible storage overhead for the majority of timesteps.

D. Adaptive Iterator Convergence

Using prior knowledge to iteratively find the number of coefficients that ensure that all the user-defined accuracy parameters ($\rho, NRMSE, \epsilon$) are respected, SPEQC–WAVELETS typically converges within 2.3 ± 0.73 iterations, averaged across S3D, GTS, XGC (see Figure 9), GCRM, and SPH application data sets. The data set for Flash velocity exhibited much higher variabity with 5.6 ± 2.2 iterations per window to accomodate a few spikes in CR to guarantee accuracy.

E. Data Linearization

One of the key requirements for SPEQC–WAVELETS design is robust performance regardless of the data model that a scientific application utilizes. For example, XGC fusion particles are mapped onto a toroidal mesh, and thus characterized by their radial, poloidal, and toroidal dimensions.

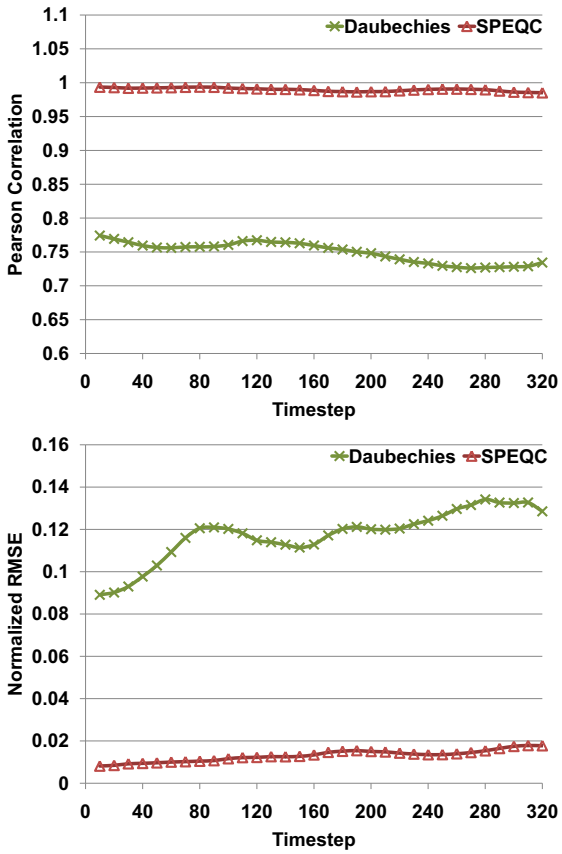

Fig. 5. Comparison of accuracy ($\rho, NRMSE$) between original and reconstructed GTS Density for SPEQC–WAVELETS and DW at fixed $CR = 15.57\%$.

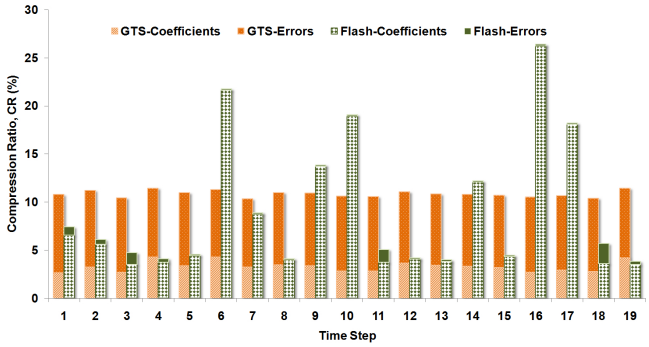

Fig. 6. Compression ratio (CR) for coefficients and quantized errors across different timesteps: $t_1 = 1,000$ and $\Delta t = 1,500$ for GTS Potential during non-linear stages; $t_1 = 3,000$ and $\Delta t = 3,500$ for Flash Velocity; $\epsilon = 1\%$; $CR = 12.5\%$ for sorted index.

To keep SPEQC–WAVELETS's applicability wide and general in regards to various application data models, SPEQC–WAVELETS accepts any stream of linearized data; the specifics of linearization are left up to the end-user. By reading linearized data as an input, we introduce a new topic of concern: is the means by which data is linearized important

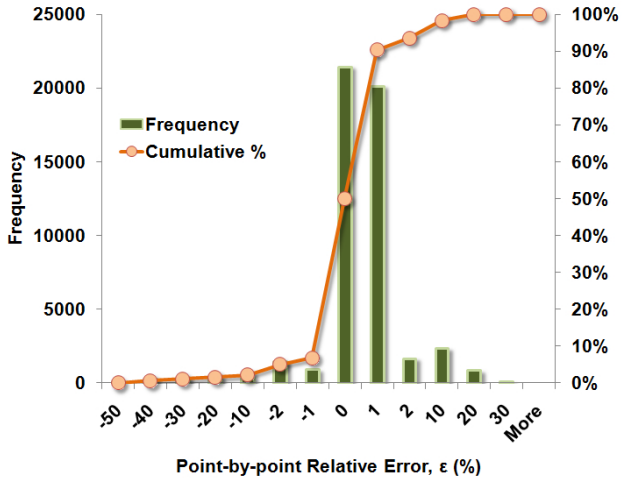

Fig. 7. Histogram of point-by-point relative errors (ϵ) for XGC Ion Temperature with DW compression.

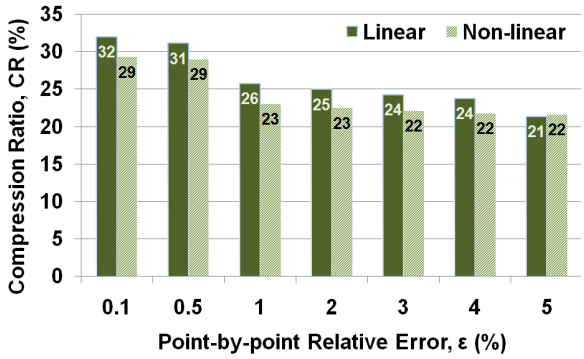

Fig. 8. Compression ratio (CR) for various point-by-point relative errors (ϵ) in GTS Potential during linear and non-linear stages of the simulation.

in determining compressibility.

We argue that our scheme does not produce significantly different results for the same data linearized in different ways. To ascertain the validity of this claim, we seek to empirically determine that various permutations of data yield resultant compression ratios that have nearly equivalent mean and median as well as a relatively low standard deviation. Our experiment revolves around calculating the aforementioned statistics and gathering results of compression of 10 different permutations of the same 172,111 element vectorof the GTS Potential during a nonlinear stage at time step 17,500. The resultant average and standard deviation of the CR are 20.0% and 0.59%, respectively.

F. Robustness and Scalability

SPEQC–WAVELETS is robust in terms of various metrics. First, the *per-window* CR and the number of iterations required to converge to the user-defined accuracy typically hardly vary (see Figure 10). Second, at a fixed CR, the *per-*

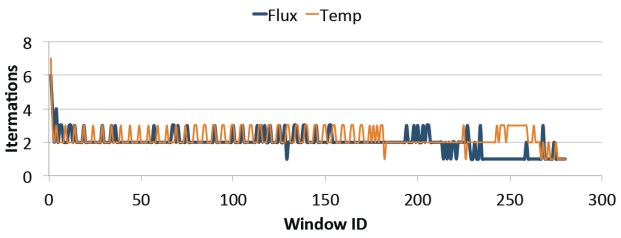

Fig. 9. Iterations of adaptive error-bounded wavelets compression for each window of two XGC data sets.

window accuracy almost stays the same for more than 25 real data sets tested. Third, the *overall* CR at $\epsilon = 1\%, \ldots, 5\%$ remains roughly the same and only increases by less than 6% as ϵ is reduced 10-fold. Finaly, the *overall* CR hardly changes when the size of the data increases. Figure 10 illustrates the latter claim when the data size is being increased in multiples of the original data fragment size. This is a typical trend for data size growth when the data is being generated along temporal dimension or being processed in blocks of spatial regions.

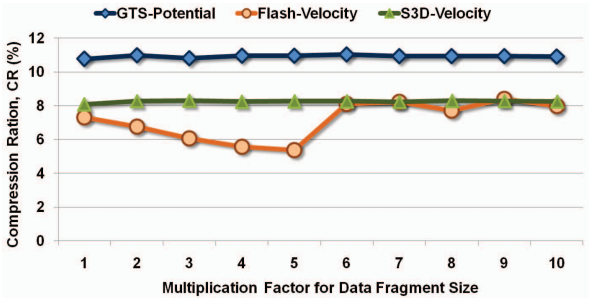

Fig. 10. Compression ratio (CR) for data size increased by a multiplicative factor of its original fragment size across different applications. $CR = 12.5\%$ ($CR = 25\%$) must be added for double- (single-) precision data.

G. Single- vs. Double-Precision Data

Scientific simulations often produce double-precision floating-point data (8-byte elements). However, for archival and community sharing purposes, these data often get converted to single-precision (4-byte elements) format. This type of storage inherently introduces a two-fold reduction in data size on a byte-level.

While we designed SPEQC–WAVELETS primarily for double-precision data reduction, in this section, we aim to evaluate its performance on single-precision data. We consider single-precision data from groundwater, climate, and combustion application domains (see Table I). Due to our $S-$preconditioner, the cost of storing the permutation index for a window size of 256 vector elements after sorting is 25% of the original single-precision data. Therefore, our overall CR can never be less than 25%; and 50% CR will imply an additional two-fold data reduction from the original double-precision data. These facts are worth keeping in mind when considering the compression ratios our method yields. In our

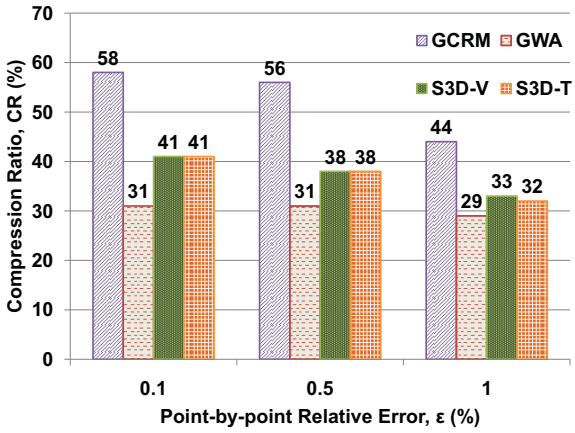

Fig. 11. Compression ratio (CR) for various point-by-point relative errors (ϵ) across different single-precision simulation data sets.

compression experiments, we took results from fixing the point-by-point relative error ϵ to values of 0.1%, 0.5%, and 1% (while keeping 256-element sized windows). Figure 11 portrays the impacts that ϵ had on compression ratio. On a macroscopic level, we see that SPEQC–WAVELETS reduces single-precision data by more than two-fold in most of the applications tested, even for small values of $\epsilon = 0.1\%$. It is worthwhile noticing that on a microscopic view (window-by-window) compression ratio does not vary significantly with standard deviation of less than 1%, and our method generally performs robustly regardless of the size of original data, as illustrated in more detail in Section IV-F.

H. Compression and Decompression Rates

When tested on a compute node containing dual hex-core 2.6GHz AMD Opteron processors with 16GB of DDR2-800 memory, SPEQC–WAVELETS demonstrated a 23MB/sec and 14 MB/sec data compression rate without and with error quantization, respectively. The corresponding decompression rates were 221MB/sec and 185MB/sec.

V. CONCLUSION

Current data reduction techniques are inadequate in keeping up with the large amounts of data generated at the tera- and peta-scale in scientific simulations. In the past, some lossless and lossy compression schemes have been explored in the hopes of providing a reasonable compression ratio on (what is now thought to be) random, noisy, and incompressible scientific data. To alleviate the lack of such compression methods, we introduce our SPEQC–WAVELETS (S-Preconditioned EQ-Calibrated Wavelets) method, a combination of techniques that allow us to reduce the size of scientific data by an amount significantly greater than current state-of-the-art techniques. Additionally, our method is robust and scalable in that overall compression ratio does not vary significantly with larger data (as well as data with various error-bounds ranging from 1%

to 5%), means of data linearization are unimportant, and the number of iterations required to converge upon user-defined accuracy parameters vary minimally.

To empirically determine these assertions, we ran experiments on 13 public data sets as well as 12 scientific simulation code datasets to ascertain that our method could be applied on datasets from various origins and disciplines. In all cases, we calculated Pearson Correlation, Normalized RMSE (Root Mean Standard Error), per point relative error, and compression ratio to numerically assess the efficacy of SPEQC–WAVELETS. We found that in almost all cases, our method yields a $0.99+$ Pearson Correlation score and an NRMSE on the order of hundredths while still in the user-specified error boundary, as well as a multi-fold reduction in original data size. Moreover, data analysis and data mining results performed over the decompressed data were of comparable quality with the ones performed over the original data.

VI. ACKNOWLEDGMENT

We would like to thank ORNLs and ANLs leadership class computing facilities, OLCF and ALCF respectively, for the use of their resources. We would also like to acknowledge the use of those scientific data sets at Flash Center for Computational Science. This work was supported in part by the U.S. Department of Energy, Office of Science (SciDAC SDM Center, DE-AC02-06CH11357, DE-SC0004935, DE-FC02-10ER26002, DEFOA-0000256, DE-FOA-0000257) and the U.S. National Science Foundation (Expeditions in Computing). Oak Ridge National Laboratory is managed by UT-Battelle for the LLC U.S. D.O.E. under contract no. DEAC05-00OR22725.

REFERENCES

[1] T. A. Welch, "A technique for high-performance data compression," *IEEE Computer Society Press*, vol. 17, pp. 8–19, June 1984.

[2] M. Burtscher and P. Ratanaworabhan, "FPC: A high-speed compressor for double-precision floating-point data," *IEEE Computer Society*, 2009.

[3] P. L. M. Isenburg and J. Snoeyink, "Lossless compression of predicted floating-point geometry," *Computer-Aided Design*, vol. 37, no. 8, pp. 869 – 877, 2005.

[4] P. Lindstrom and M. Isenburg, "Fast and efficient compression of floating-point data," *IEEE Trans. on Viz. and Comp. Graphics*, vol. 12, no. 5, pp. 1245 –1250, 2006.

[5] J. K. P. Ratanaworabhan and M. Burtscher, "Fast lossless compression of scientific floating-point data," in *Proc. of the DCC*, 2006.

[6] G. Craciun, M. Jiang, D. Thompson, and R. Machiraju, "Spatial domain wavelet design for feature preservation in computational data sets," *IEEE Trans Vis Comput Graph.*, 2005.

[7] S. G. Sheikholeslami and A. Zhang, "Wavecluster: A multi-resolution clustering approach for very large spatial databases," *Very Large Spatial Databases (VLDB)*, pp. 428–439, 1998.

[8] M. C. Shahabi, S. Chung and G. Hajj, "2D TSA-tree: A wavelet-based approach to improve the efficiency of multi-level spatial data mining," *In Statistical and Scientific Database Management*, pp. 59–68, 2001.

[9] M. Abo-zahhad, S.M. Ahmed, and A. Zakaria, "ECG Signal Compression Technique Based on Discrete Wavelet Transform and QRS-Complex Estimation," *Signal Processing: An International Journal*, 2010.

[10] J. K. Romberg, M. B. Wakin, and R. G. Baraniuk, "Wavelet-domain approximation and compression of piecewise smooth images," *IEEE Trans. Image Processing*, vol. 15, pp. 1071–1087, 2006.

[11] S. Saha, "Image compression from DCT to wavelets: a review," *ACM Crossroads*, vol. 6, pp. 12–21, March 2000.

[12] X. Tang and W. A. Pearlman, "Three-dimensional wavelet-based compression of hyperspectral images," *Hyperspectral Data Compression*, 2005.

[13] J. R. Goldschneider, "Lossy compression of scientific data via wavelets and vector quantization," *University of Washington*, 1997.

[14] A. L. Belmon, H. Benoit-Cattin and J. L. Bougeret, "Lossy compression of scientific spacecraft data using wavelets. application to the CASSINI spacecraft data compression," *A&A*, vol. 386, no. 3, pp. 1143–1152, 2002.

[15] J. H. Chen, A. Choudhary, B. D. Supinski, M. DeVries, E. R. Hawkes, S. Klasky, W. K. Liao, K. L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C. S. Yoo, "Terascale direct numerical simulations of turbulent combustion using s3d," *Computational Science and Discovery*, vol. 2, no. 1, p. 015001, 2009.

[16] S. Ku, C. S. Chang, and P. H. Diamond, "Full-f gyrokinetic particle simulation of centrally heated global itg turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry," *Nuclear Fusion*, vol. 49, no. 11, p. 115021, 2009.

[17] W. X. Wang and Z. Lin and W. M. Tang and W. W. Lee and S. Ethier and J. L. V. Lewandowski and G. Rewoldt and T. S. Hahm and J. Manickam, "Gyro-kinetic simulation of global turbulent transport properties in Tokamak experiments," *Physics of Plasmas*, vol. 13, no. 9, p. 092505, 2006.

[18] D. Randall, M. Khairoutdinov, A. Arakawa, and W. Grabowski, "Breaking the cloud parameterization deadlock." *Bulletin of the American Meteorological Society*, vol. 84, pp. 1547–1564, Nov 2003.

[19] B. Palmer, V. Gurumoorthi, A. Tartakovsky, and T. Scheibe, "A component-based framework for smoothed particle hydrodynamics simulations of reactive fluid flow in porous media," *Intl. J. HPC. Appl.*, vol. 24, pp. 228–239, May 2010.

[20] B. Fryxell, K. Olson, P. Ricker, F. X. Timmes, M. Zingale, D. Q. Lamb, P. MacNeice, R. Rosner, J. W. Truran, and H. Tufo, "FLASH: An Adaptive Mesh Hydrodynamics Code for Modeling Astrophysical Thermonuclear Flashes," *The Astrophysical Journal Supplement Series*, vol. 131, pp. 273–334, Nov. 2000.

[21] G. Redinbo and R. Manomohan, "Fault tolerance in computing, compressing, and transmitting fft data," *Communications, IEEE Transactions on*, vol. 49, no. 12, pp. 2095 –2105, dec 2001.

[22] M. K. Kwong and P. T. P. Tang, "W-matrices, nonorthogonal multiresolution analysis, and finite signals of arbitrary length," Argonne National Lab, Tech. Rep., 1994.

[23] N. F. Law and W. C. Siu, "Fast algorithm for quadratic and cubic spline wavelets," in *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, 2001, pp. 247 –250.

[24] P. C. Marais, E. H. Blake, and A. M. Kuijk, "Quadratic vs cubic spline-wavelets for image representation and compression," *CWI (Centre for Mathematics and Computer Science)*, 1997.

[25] M. Hollschneider, "More on the analysis of local regularity through wavelets," *J. Stat. Phy*, vol. 77, pp. 807–840, nov 1994.

[26] S. Ku, R. Barreto, S. Klasky, C. S. Chang, H. Strauss, L. Sugiyama, P. Snyder, D. Pearlstein, B. Ludascher, G. Bateman and A. Kritz and the CPES Team, "Plasma edge kinetic mhd modeling in tokamaks using kepler workflow for code coupling, data management and visualization," *Comm. Comput. Physics*, 2008.