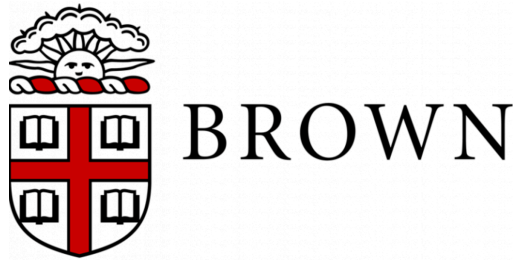
Canopus: Enabling Extreme-Scale Data Analytics on Big HPC Storage via Progressive Refactoring

Tao Lu*, Eric Suchyta, Jong Choi, Norbert Podhorszki, and Scott Klasky, Qing Liu*, Dave Pugmire and Matt Wolf, and Mark Ainsworth+

* *New Jersey Institute of Technology*
Oak Ridge National Laboratory
+ *Brown University*







Overview

- Background and related work
- Progressive data refactoring
- Conclusion

HPC systems

- Mission: illuminate phenomena that are often impossible to study in a laboratory [Oak Ridge National Lab, 2013]
  - Climate impacts of energy use
  - Fusion in a reactor not yet built
  - Galaxy formation
- Methodology: Modeling and simulation, along with data exploration
  - Data generation
  - Storage
  - Analysis
  - Visualization

HPC systems (cont'd)

- The big data management challenge [Shalf et. al., 2014]
  - Worsening I/O bottleneck for exascale systems
  - Exponentially increasing multi-scale, multi-physics data

Data compression in exascale (next-generation HPC) systems

- Goal: 10x to 100x data reduction ratio [Ian Foster et. al., 2017]
  - Reduce data by at least 90%

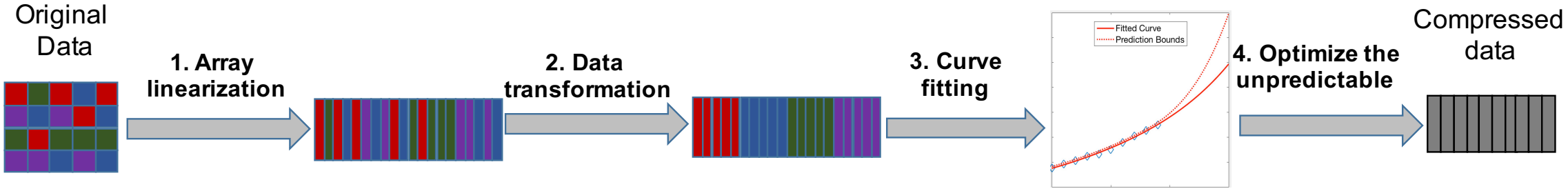
- Data features
  - Temporal and spatial
  - High-dimensional
  - **Floating-point**

Data compressors

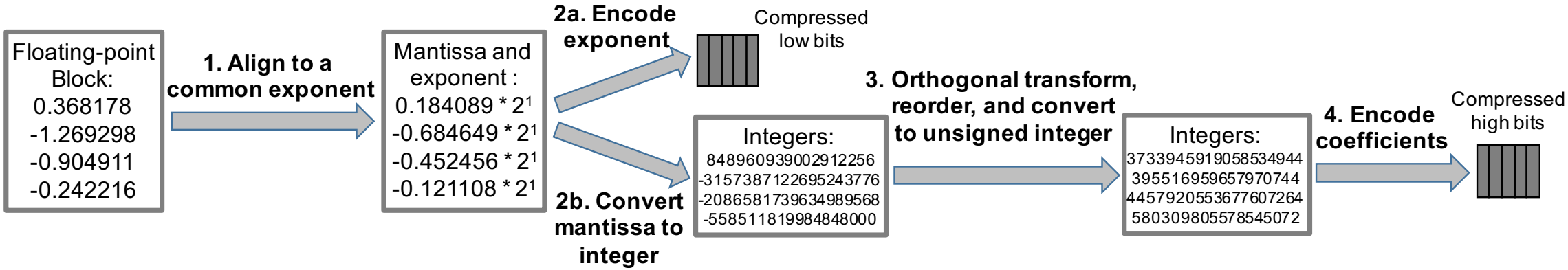
- Lossless compression
  - Deduplication
  - GZIP
  - FPC [Burtscher et. al., 2009]

- Lossy compression
  - ZFP [Lindstrom et. al., 2014]
  - ISABELA [Lakshminarasimhan et.al., 2011]
  - SZ [Shen et. al., 2016]

Lossy compression



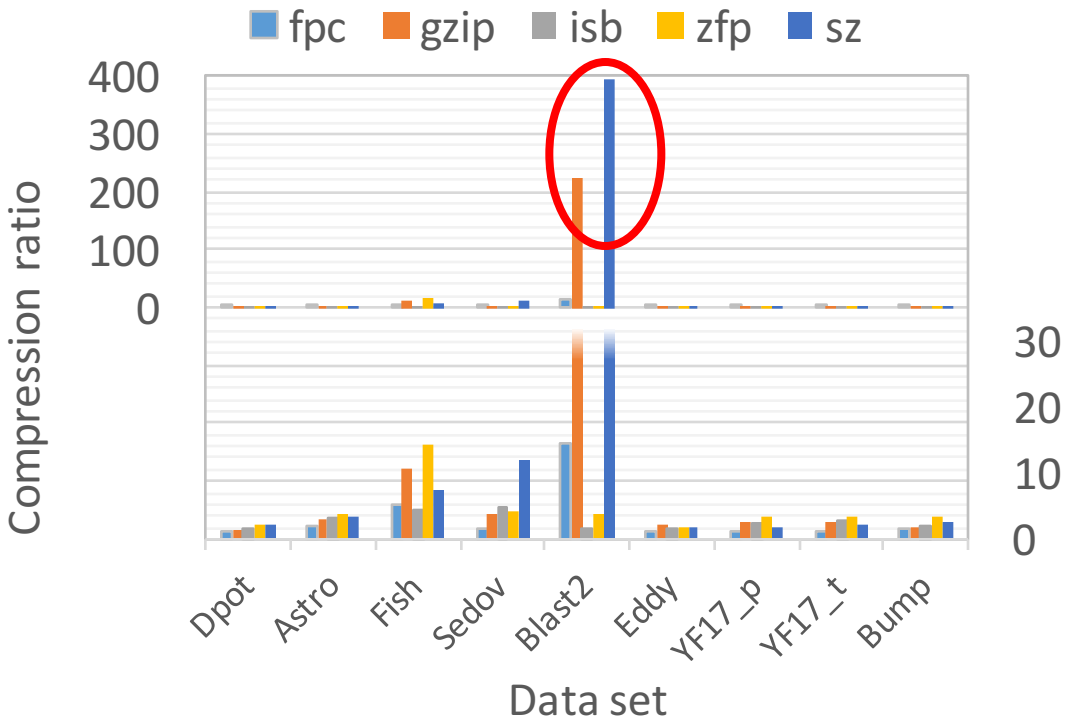
Workflow of curve fitting based compression (e.g. ISABELA and SZ)



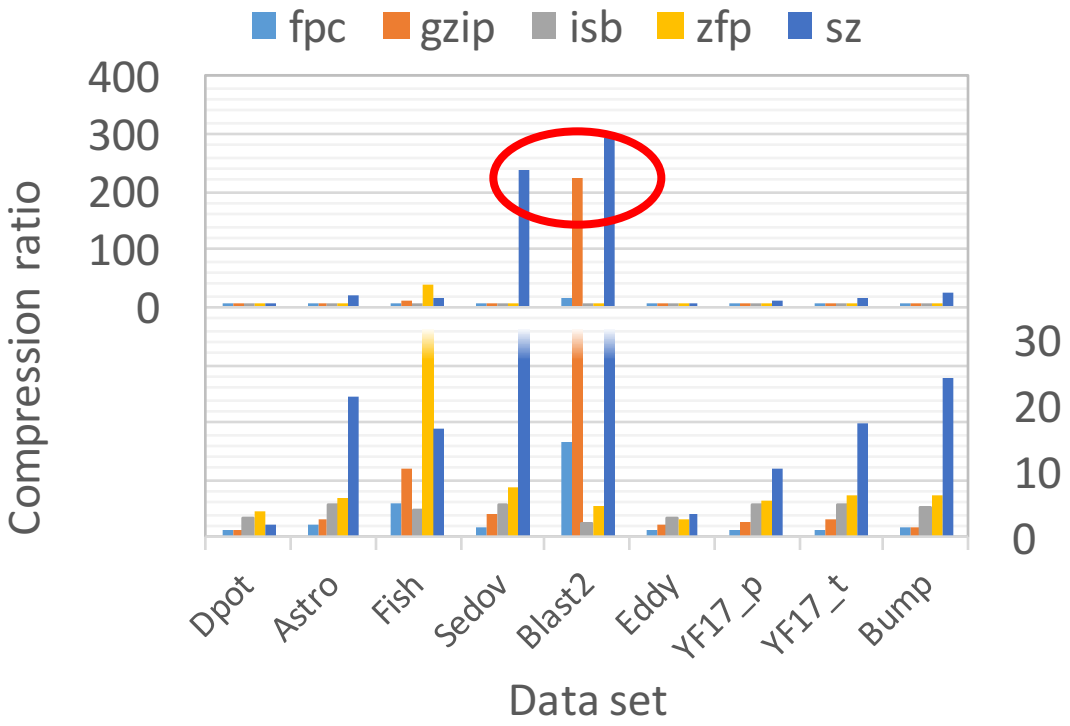
Workflow of quantization and transformation based compression (e.g. ZFP)

Can floating-point compressors achieve a near 100x compression ratio?

Performance of compressors



Relative error bound 0.000001



Relative error bound 0.001

Can floating-point compressors achieve a near 100x compression ratio?

Yes. If

Dataset contains a lot of identical values;

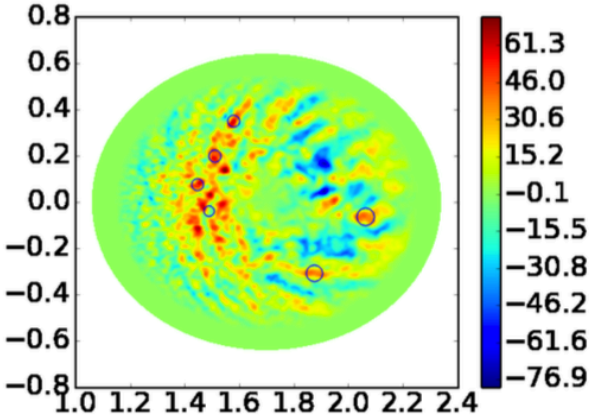
Or, data values highly skew with moderate compression error bounds.

No. For most datasets.

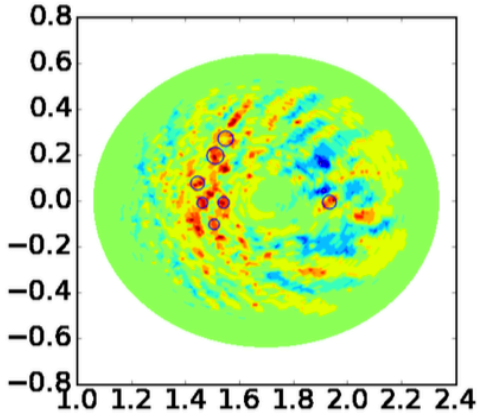
Can floating-point compressors achieve a 10x compression ratio?

Yes. For a lot of datasets with moderate compression error bounds.

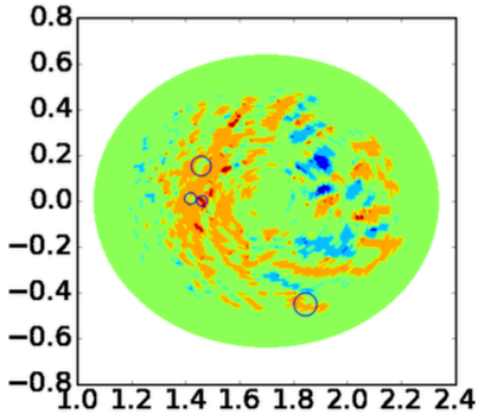
What if the compression ratio is rushed to 100x by loosening error bounds?



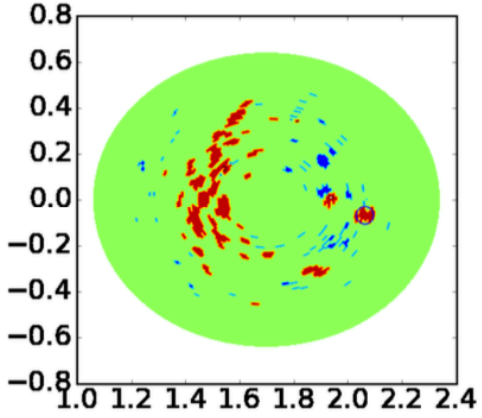
(a) ZFP 10x Compression.



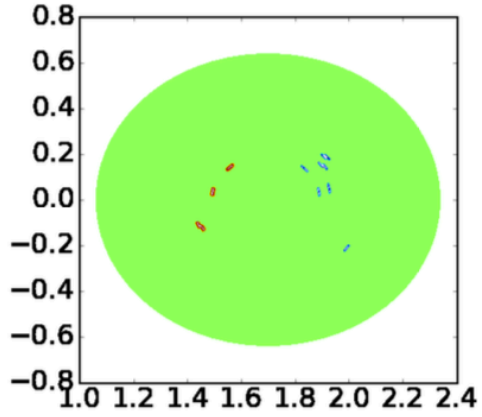
(b) ZFP 20x Compression.



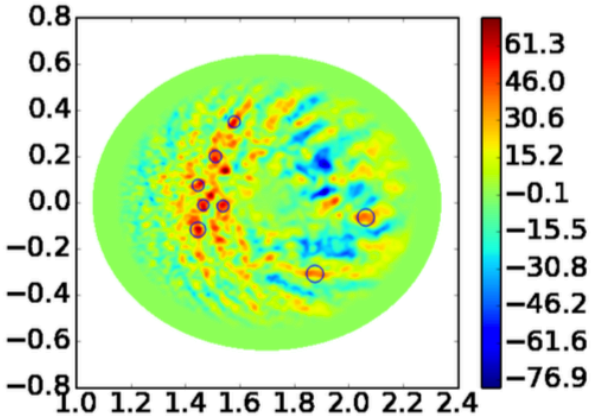
(c) ZFP 30x Compression.



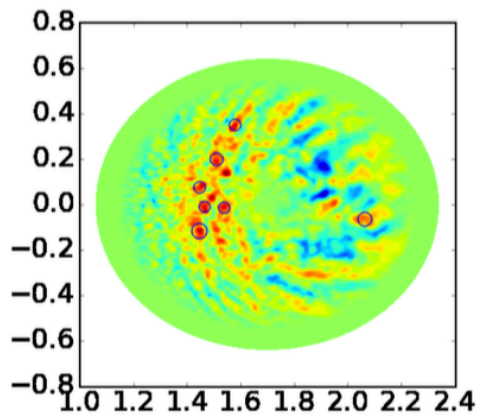
(d) ZFP 50x Compression.



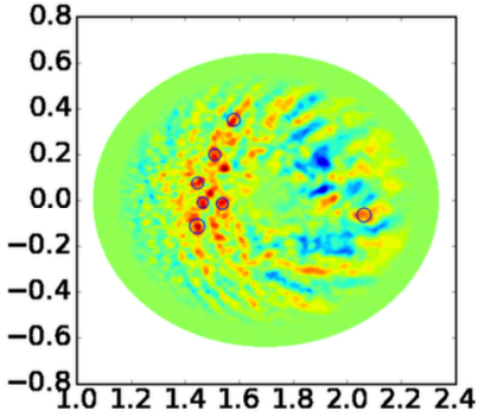
(e) ZFP 100x Compression.



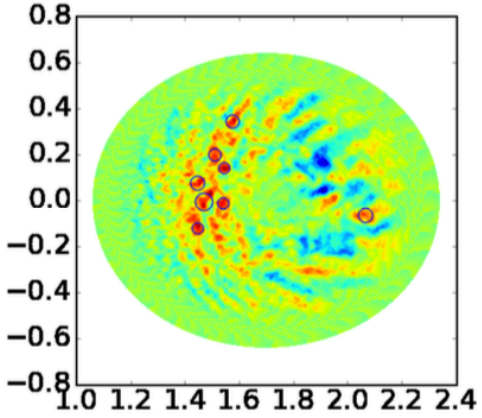
(f) SZ 10x Compression.



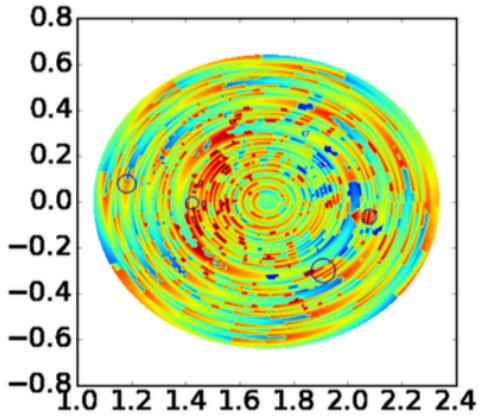
(g) SZ 20x Compression.



(h) SZ 30x Compression.



(i) SZ 50x Compression.



(j) SZ 100x Compression.

Visualization and blob detection on compressed Dpot data.

Limitations of data compression by reducing floating-point accuracy

- Near 100x compression ratio is hardly achievable
- Lost data accuracy cannot be restored

Overview

- Background and related work
- Progressive data refactoring
- Conclusion

We propose *Canopus*

- Compressing HPC data in another dimension (resolution)
- Enabling progressive data refactoring
- User transparent implementation

```
<?xml version="1.0"?>
<adios-config host-language="C">
  <adios-group name="field" stats="On">
    <var name="NX" type="integer"/>
    <var name="GX" type="integer"/>
    <var name="OX" type="integer"/>
    <var name="MX" type="integer"/>
    <var name="MY" type="integer"/>
    <global-bounds dimensions="GX" offsets="OX">
       <var name="dpot" gwrite="t" type="double" dimensions="NX" transform="none
"/>
       <var name="R" gwrite="g" type="double" dimensions="NX" transform="none"/>
       <var name="Z" gwrite="g" type="double" dimensions="NX" transform="none"/>
    </global-bounds>
       <var name="mesh" gwrite="t" type="integer" dimensions="MY,MX"
            transform="none"/>
  </adios-group>

    <method group="field" method="CANOPUS">
        decimation-ratio=4;
        save-delta=1;
        compress-delta=0;
        compression-tolerance=0.001;
        thresh_type=absolute;
        thresh=40;
        method=MPI;path=data0;parameters;
        method=MPI;path=data1;parameters;
    </method>

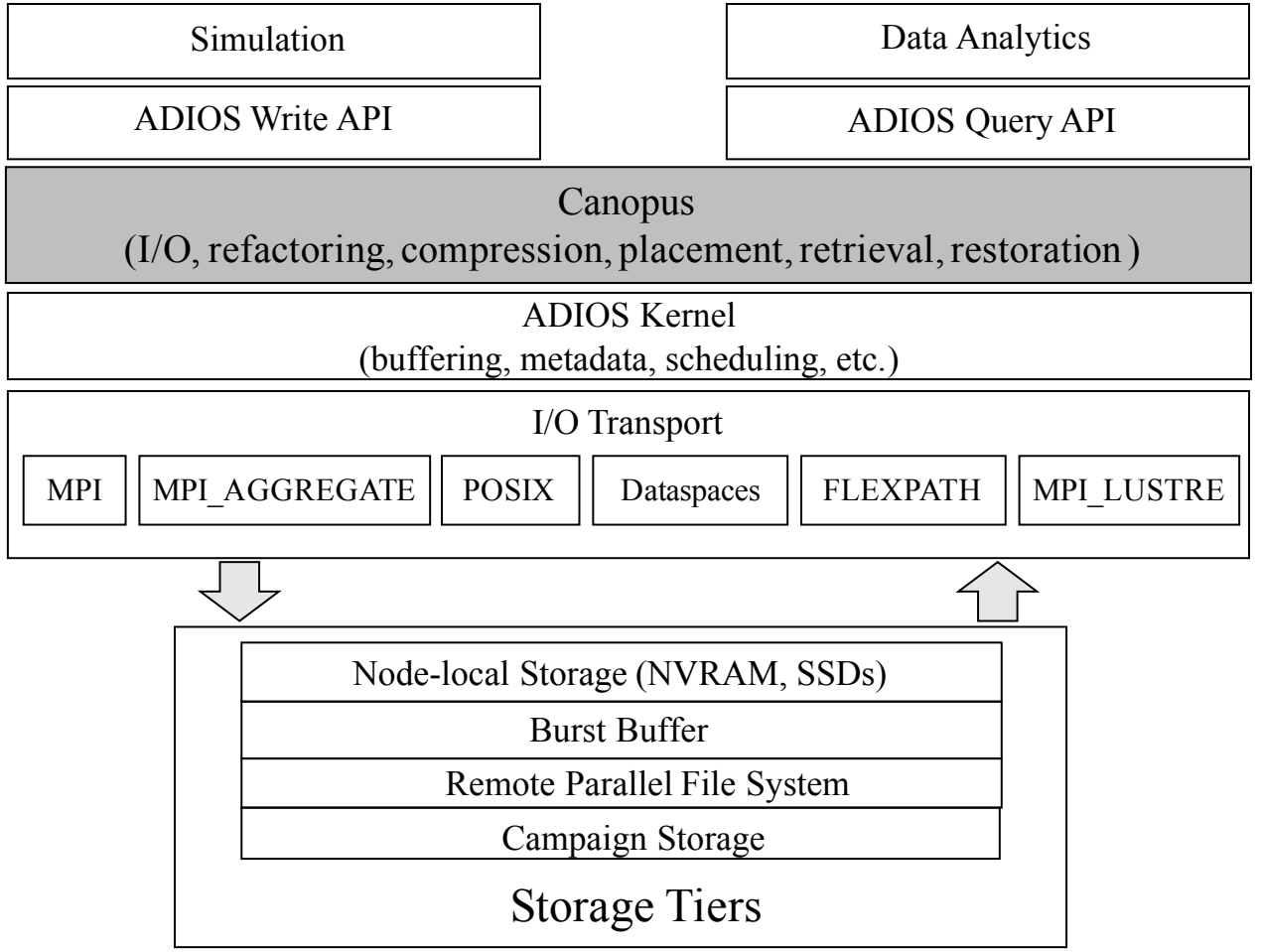
  <buffer max-size-MB="10"/>

</adios-config>
```

Canopus I/O configuration

Canopus: basic idea

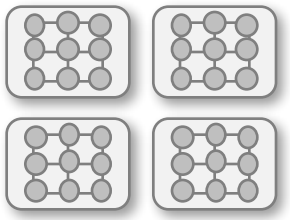
- Refactor the simulation results (via decimation) into a base dataset along with a series of deltas
- Base dataset is saved in fast devices, deltas in slow devices
- Base dataset can be used separately (at a lower resolution) for analysis
  - Selected subset of deltas to be retrieved to restore data to a target accuracy



Canopus in HPC Systems

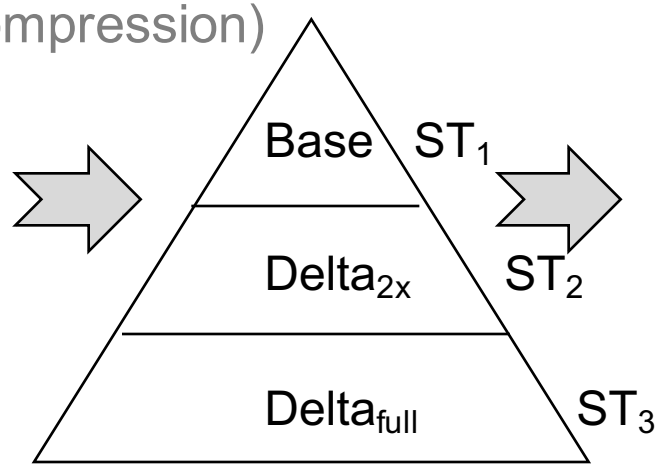
Canopus workflow

HPC Simulations (high accuracy)



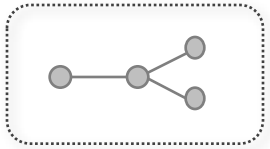
Refactoring (decimation, compression)

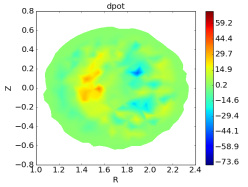
Retrieving & Reconstruction



Storage Hierarchy

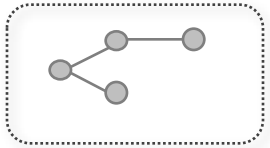
Analytics Pipeline 1 (low accuracy)

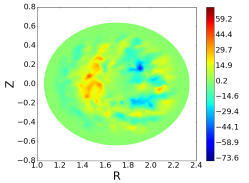




$base = L_{4x}$

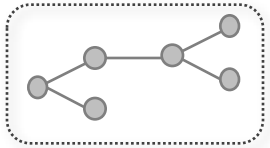
Analytics Pipeline 2 (medium accuracy)

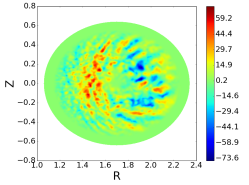




$base + delta_{2x}$

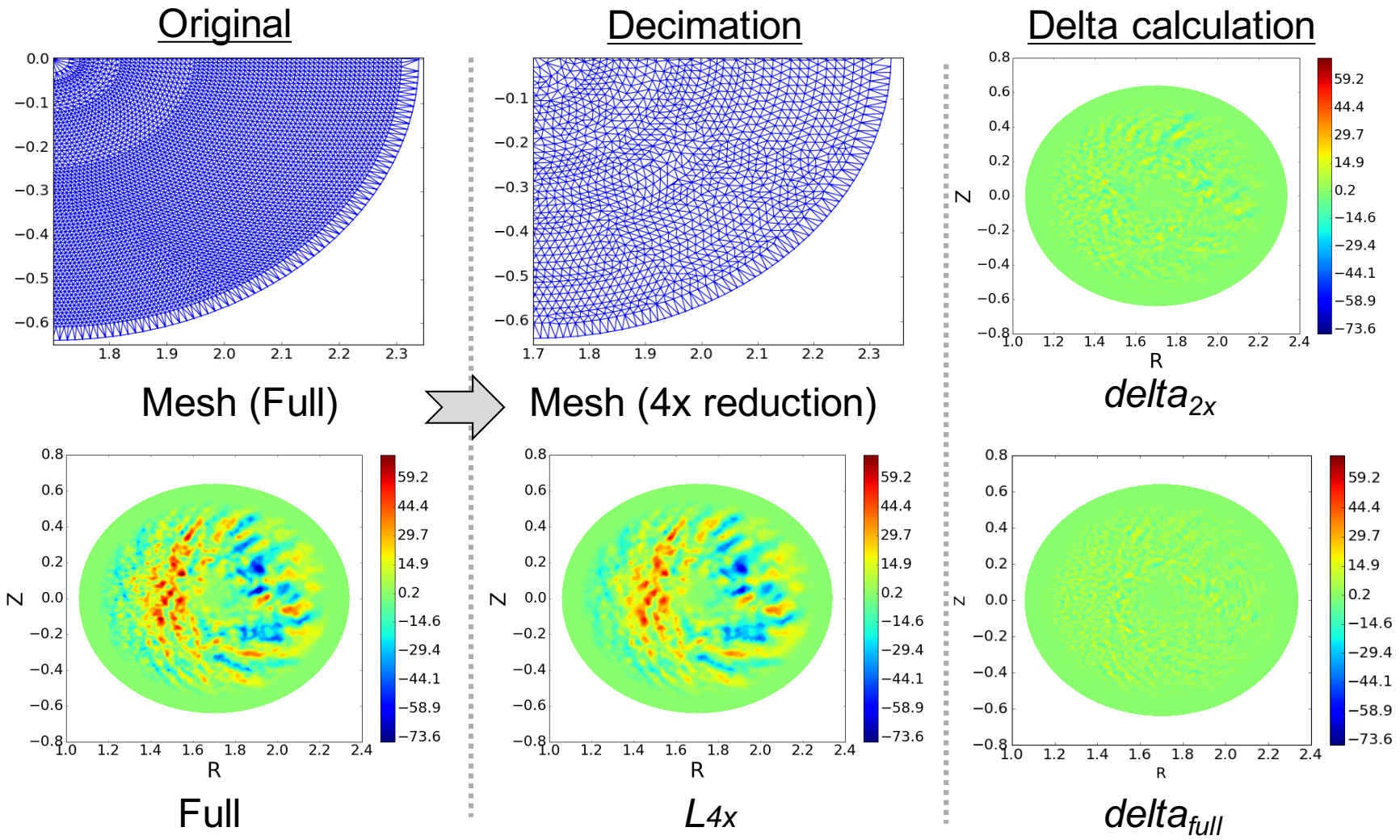
Analytics Pipeline n (high accuracy)





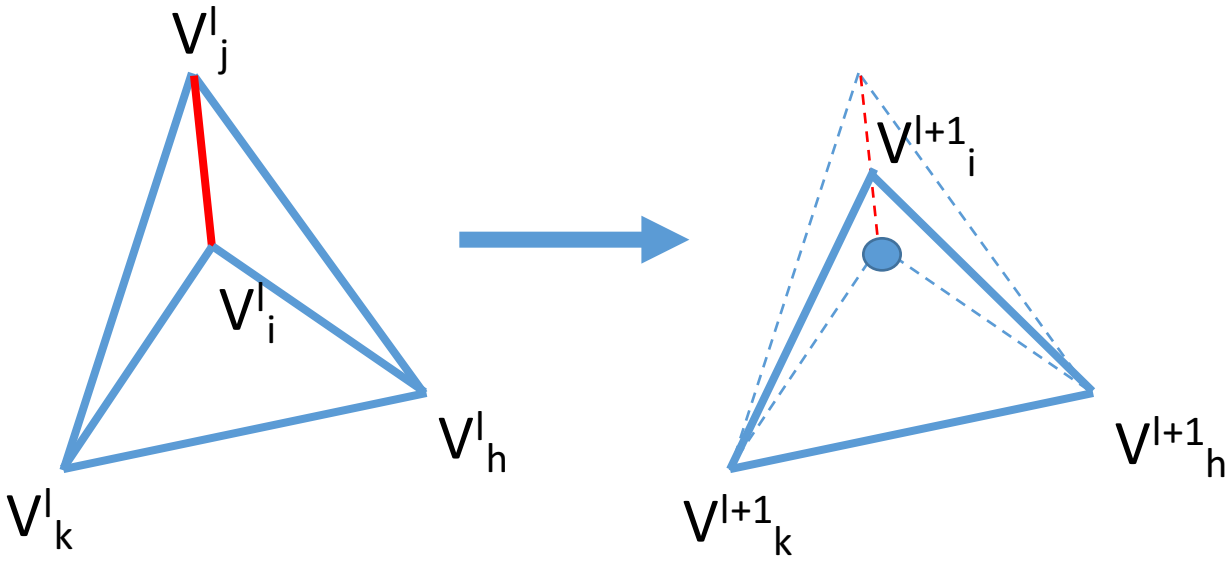
$base + delta_{2x} + delta_{full}$

Data refactoring



1. **Mesh decimation**
2. **Delta calculation**
3. **Floating-point compression**

Mesh decimation



$$V^{l+1}_i = \frac{1}{2}(V^l_i + V^l_j)$$

Delta Calculation

- For mesh data, it's common that each vertex corresponds to a value (floating-point)
- After triangular mesh decimation:

$$delta^l_n = F(V^l_n) - 1/3 * F(V^{l+1}_i) - 1/3 * F(V^{l+1}_j) - 1/3 * F(V^{l+1}_k)$$

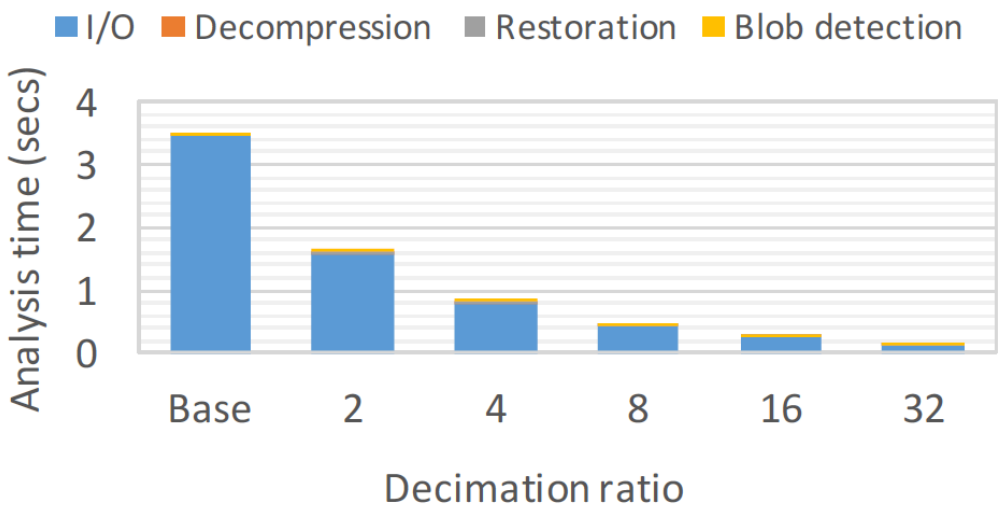
Compression

- The floating-point values corresponding to vertexes are compressed using ZFP compressor
- A potential optimization to our framework is supporting adaptive compressors based on dataset features

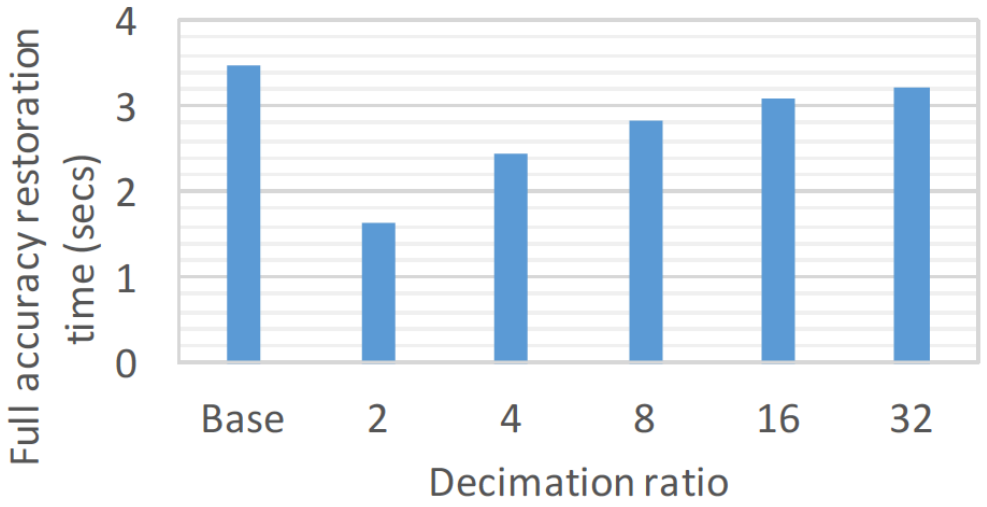
Progressive data exploration (reverse the data refactoring procedures)

- I/O (read the base dataset and deltas)
- Decompression
- Restoration

Performance gain of Canopus for data analytics

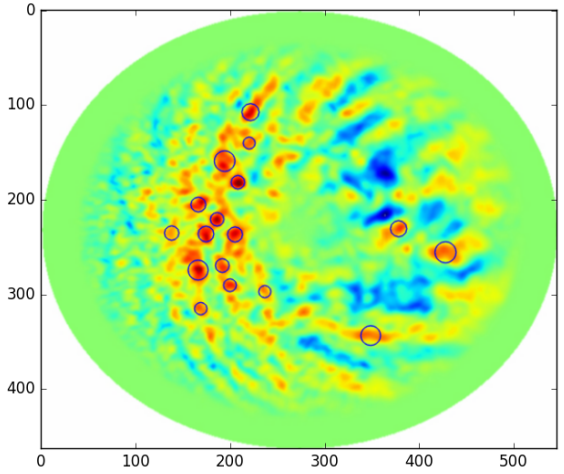


(a) End-to-end time of the analytics pipeline.

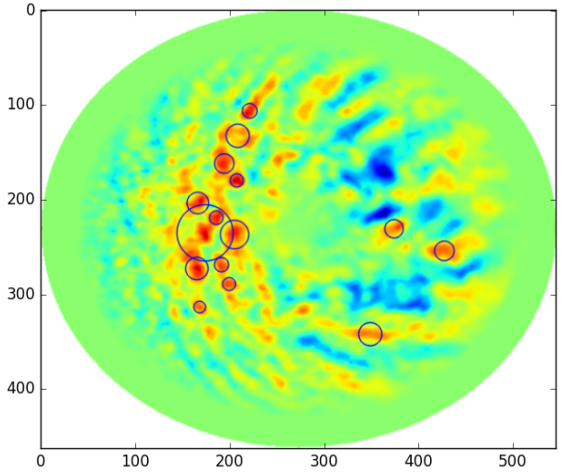


(b) Restoring full accuracy data from the base dataset and deltas.

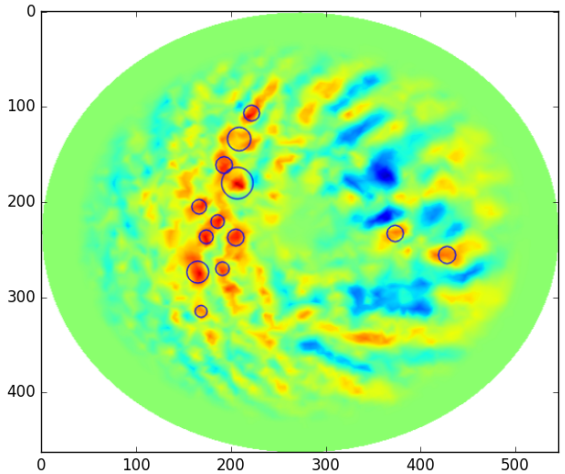
Impact on Data Analytics



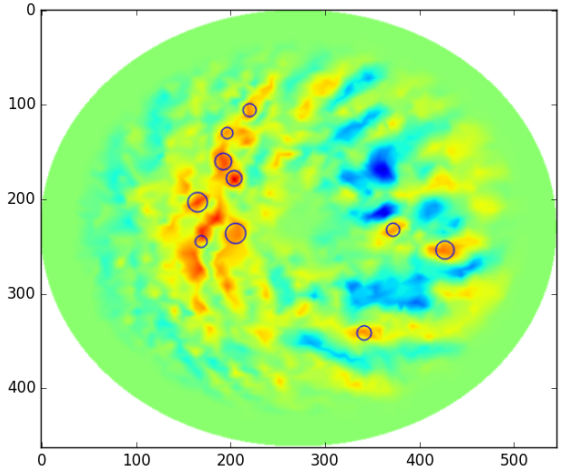
Original



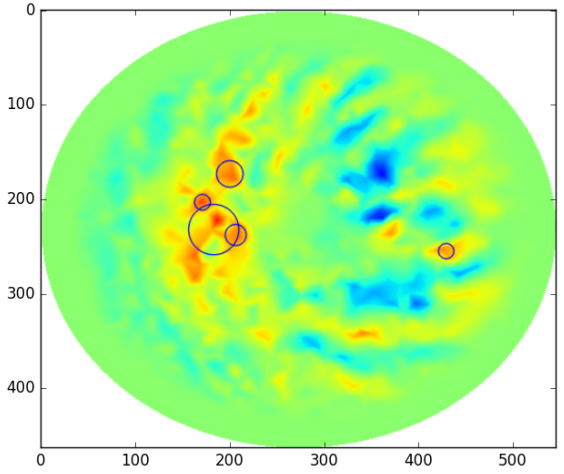
2x reduction



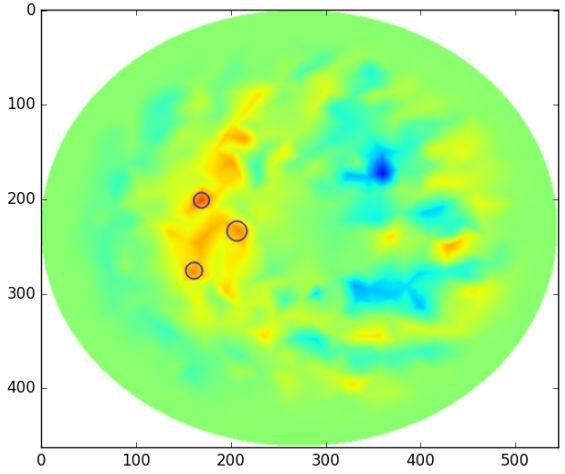
4x reduction



8x reduction

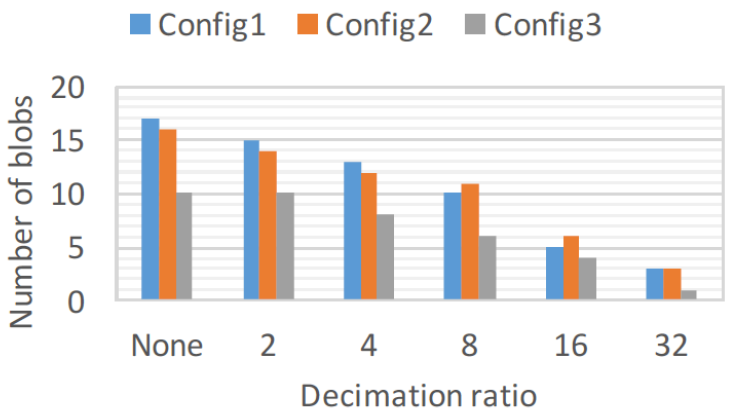


16x reduction

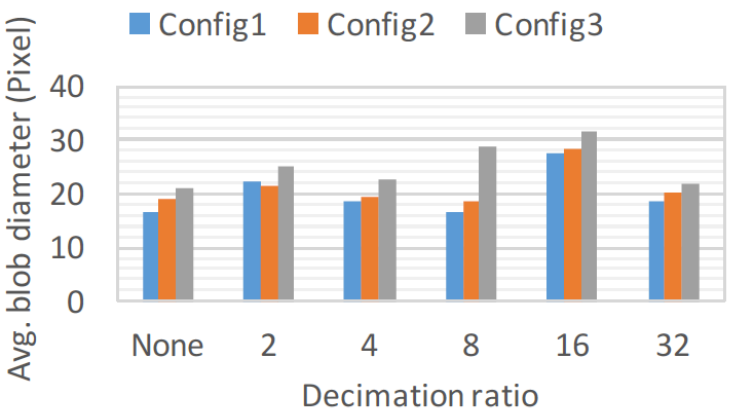


32x reduction

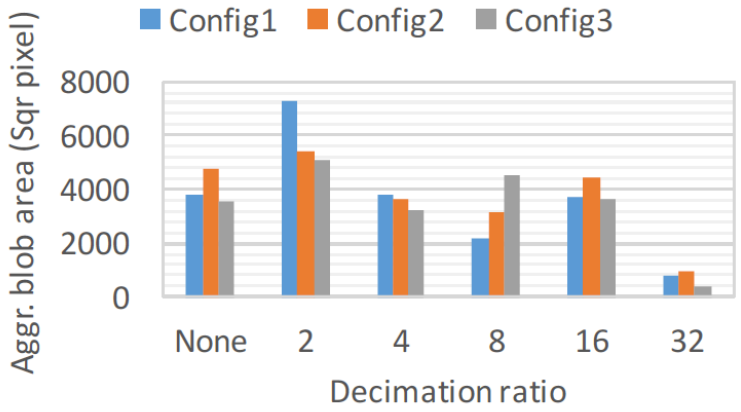
A quantitative evaluation of blob detection



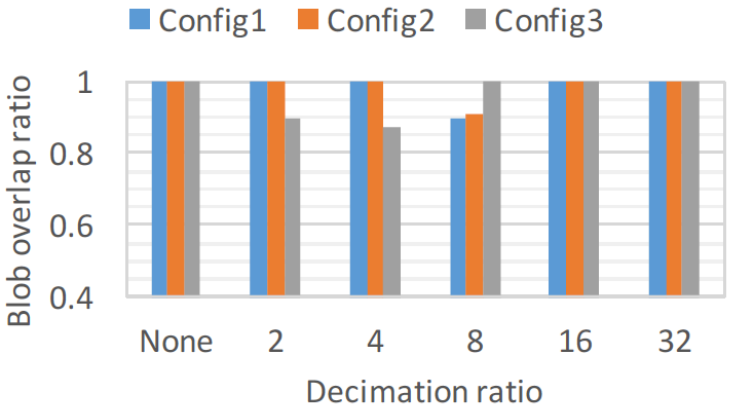
(a) Number of blobs detected



(b) Avg. blob diameters



(c) Blob area



(d) Blob overlapping ratio

Overview

- Storage stacks of HPC systems
- Progressive data refactoring
- Conclusion and future work

Conclusion

- Lossy compression may devastate the usefulness of data to achieve high compression ratio (such as 100x)
- It is critical to compress data in multiple orthogonal dimensions such as accuracy and resolution
- Canopus combines mesh compression and floating-point compression, possibly delivering a high compression ratio without devastate the usefulness of data

Future work

- Investigate the impact of lossy compression on analytical applications other than visualization
  - Original data $\mathcal{A} == \mathcal{B}$, compressed data $\mathcal{A}' == \mathcal{B}'$?
  - $\mathcal{F}(\mathcal{D}) == \mathcal{F}(\mathcal{D}')$? $\mathcal{F}$ is a function

References

- Oak Ridge National Lab, ***Solving Big Problems: Science and Technology at Oak Ridge National Laboratory***, 2013
- Foster, I., Ainsworth, M., Allen, B., Bessac, J., Cappello, F., Choi, J. Y., … Yoo, S. (2017). ***Computing Just What You Need : Online Data Analysis and Reduction at Extreme Scales***, 1–16.
- Shalf, J., Dosanjh, S., & Morrison, J. (2014). ***Top ten exascale research challenges***, 1–25. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-19328-6_1
- Burtscher, M., & Ratanaworabhan, P. (2009). ***FPC: A high-speed compressor for double-precision floating-point data***. *IEEE Transactions on Computers*, *58*(1), 18–31.
- Lindstrom, P. (2014). ***Fixed-rate compressed floating-point arrays***. *IEEE Transactions on Visualization and Computer Graphics*, *20*(12), 2674–2683.
- Lakshminarasimhan, S., Shah, N., Ethier, S., Klasky, S., Latham, R., Ross, R., & Samatova, N. F. (n.d.). ***Compressing the Incompressible with ISABELA : In-situ Reduction of Spatio-Temporal Data***, 1–14.

Thanks & Questions