Data Management and Visualization Approaches for the U.S. Exascale Computing Project

Approved for public release

James Ahrens	Data and Visualization L3 Lead
Chuck Atkins, Scott Klasky, Franck Capello, Rob Ross, Ken Moreland	FY2020 Data and Visualization L4 Leads

IEEE Cluster / Workshop First Extreme-scale Scientific Software Stack Forum (E4S Forum)

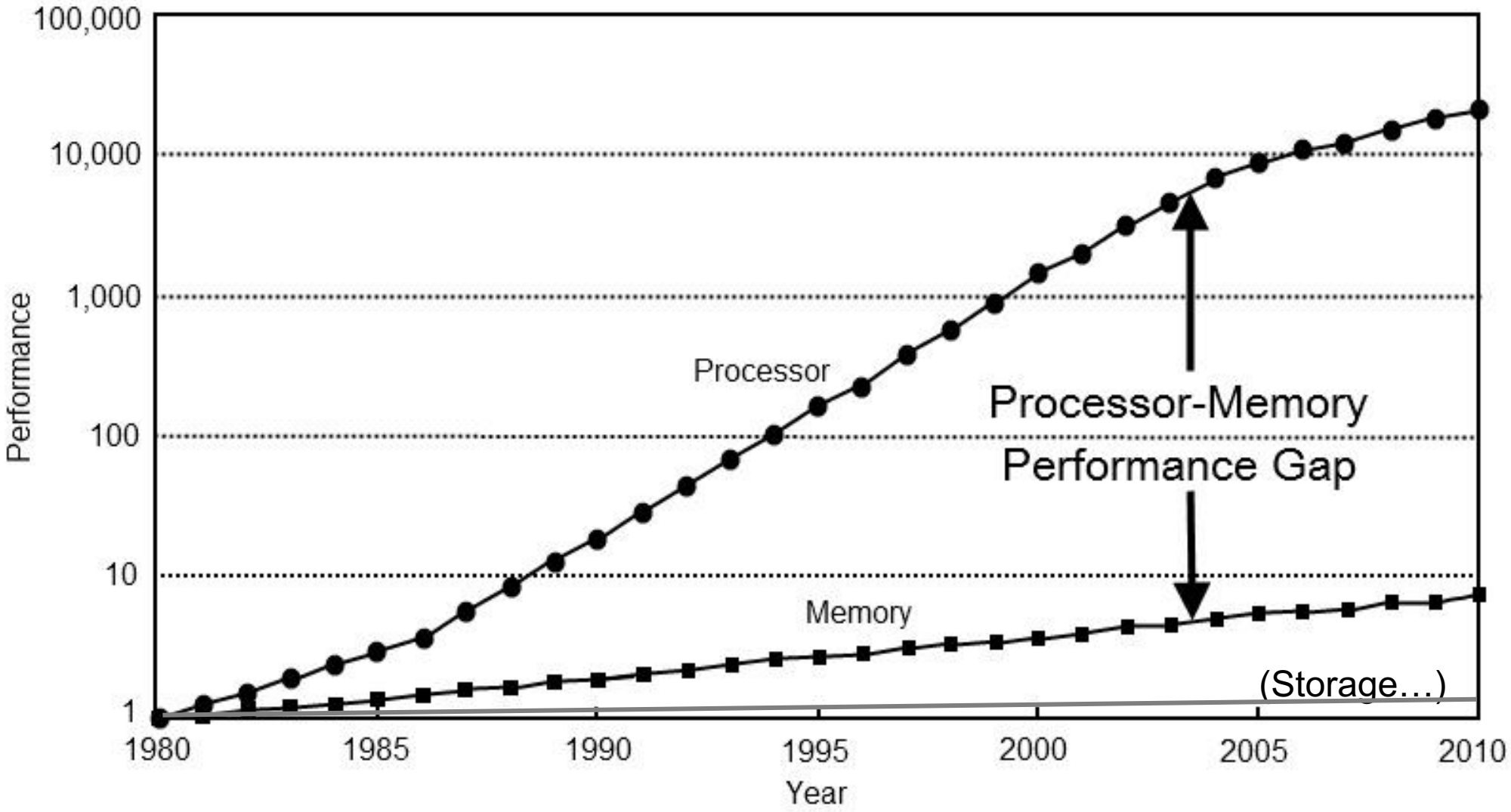
Albuquerque, New Mexico
September 25, 2019

LA-UR-19-29552

Data and visualization area summary

Vision	Supporting achieving exascale by addressing exascale data and visualization challenges	
Challenges	Exascale system concurrency is expected to grow by five or six orders of magnitude, yet system memory and I/O bandwidth/persistent capacity are only expected to grow by one and two orders of magnitude	
Mission	Deliver exascale-ready storage, data services and in situ visualization solutions for applications	
Objective	Produce data and visualization capabilities, integrate these capabilities into ECP applications, demonstrate solutions, deliver software as part of SDK	
Starting Point	Existing packages including storage tools such as ADIOS, MPI-IO, HDF5, services such as SCR for checkpoint restart, and post-processing visualization tools such as ParaView and Visit	
Portfolio Goals	**Storage**	Deliver via HDF5 API, focus on burst buffer and backends • Includes top DOE HPC storage teams - ADIOS, MPI-IO, HDF5, PnetCDF
	Services	Deliver data services (What not how) such as scientific data compression (ZFP, SZ), checkpoint restart (VeloC), storage performance tracking (Darshan)
	Visualization	Deliver new exascale-oriented in situ visualization and analysis workflow via ALPINE, Vtk-m and Cinema products

Challenge: Processor to memory/storage latency gap (on node)

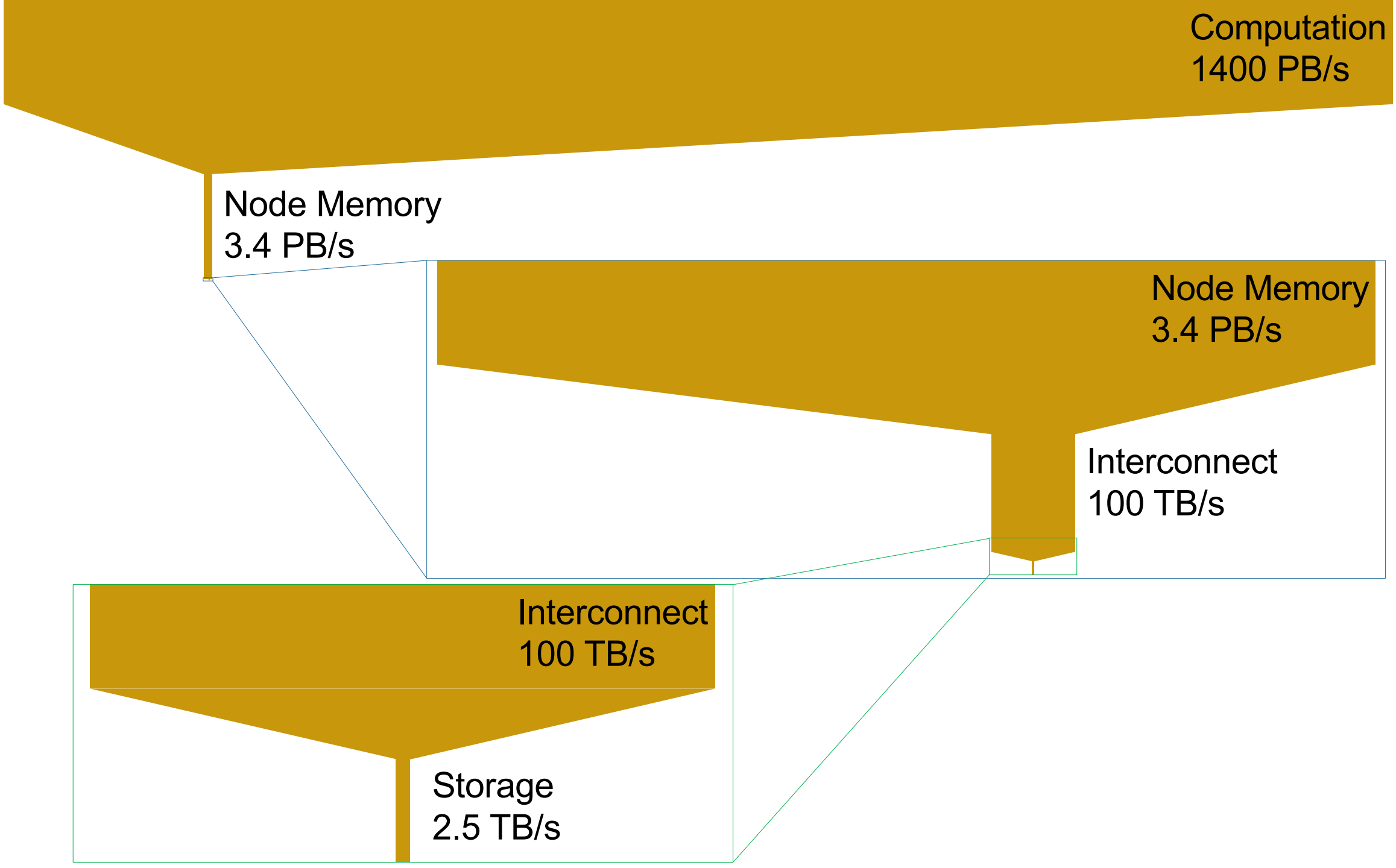


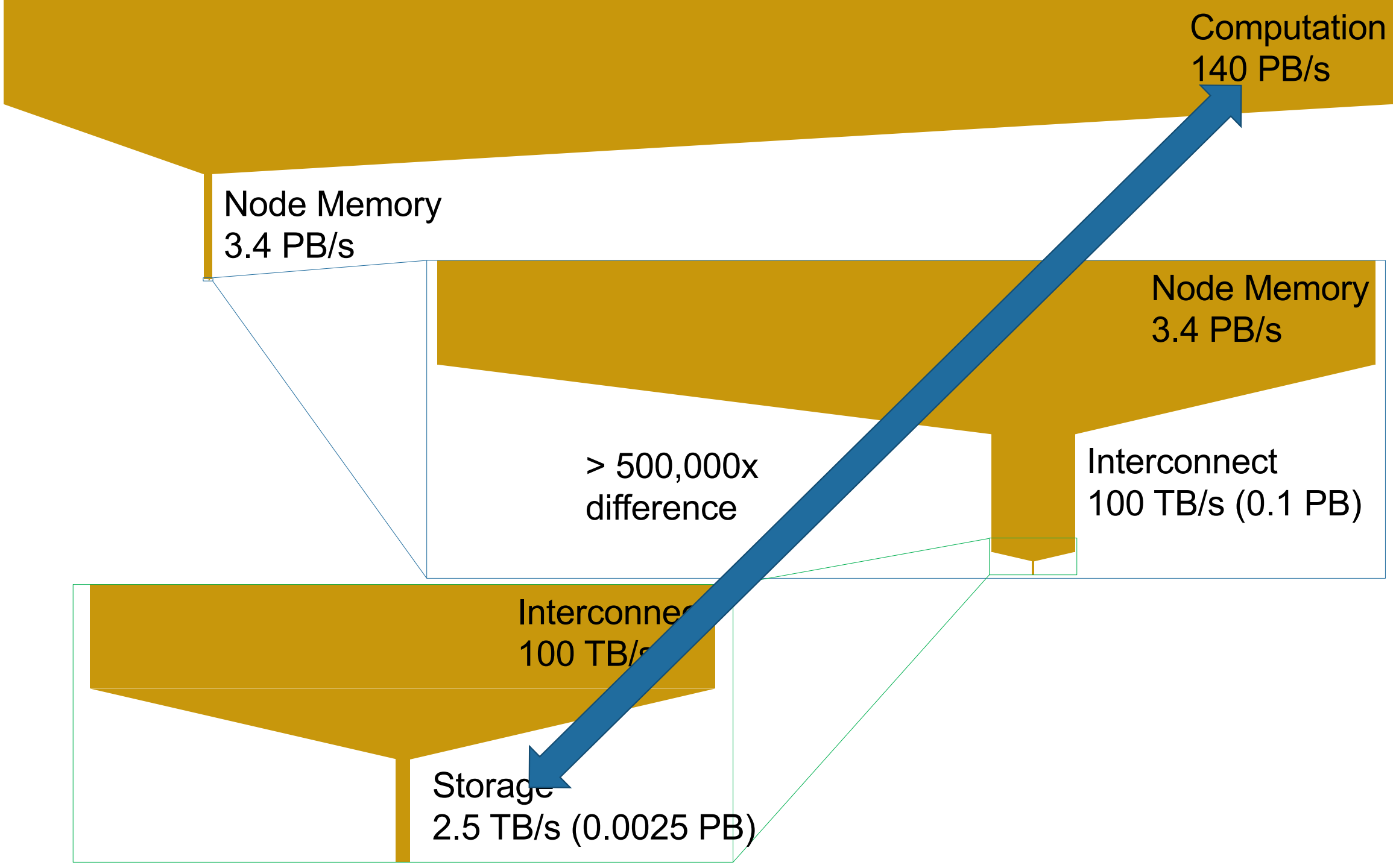
Challenge: Processor to memory/storage bandwidth gap (on supercomputer)
Example Summit supercomputer – slides courtesy of K. Moreland

Computation
1400 PB/s

Node Memory
3.4 PB/s


Computation
1400 PB/s
Node Memory
3.4 PB/s
Node Memory
3.4 PB/s
Interconnect
100 TB/s


Computation
1400 PB/s
Node Memory
3.4 PB/s
Node Memory
3.4 PB/s
Interconnect
100 TB/s
Interconnect
100 TB/s
Storage
2.5 TB/s


Computation
140 PB/s
Node Memory
3.4 PB/s
Node Memory
3.4 PB/s
> 500,000x
difference
Interconnect
100 TB/s (0.1 PB)
Interconnec
100 TB/s
Storage
2.5 TB/s (0.0025 PB)

Summary of area

- **Challenge:** Processor to Memory/Storage Gaps
  - Latency Gap
  - Bandwidth/Capability Gap
- **Advantage**: Lots of compute available
- **Services requested:**
  - Fault tolerance
  - Data storage
  - Visual analysis

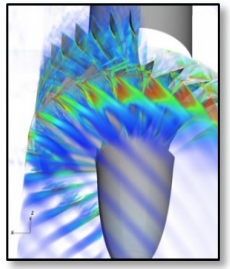
Overcome gaps via:	Specific approaches	Product highlight
Abstract/virtual interface to hide gap via messaging, caching and asynchrony	Messaging instead of communicating through filesystem	ADIOS
	Add a layer of cache – burst buffer	PnetCDF, HDF5, UnifyFS
	Asynchronous interaction between processors and storage	VeloC
Prioritize and measure functionality	Identify and evaluate I/O patterns, choose frontend API, add ADIOS, MPI-IO, pnetCDF backends, measure, compare	Darshan, HDF5, ADIOS, PnetCDF, MPI-IO
Data reduction and transformation	Compression	SZ, ZFP
	In situ analysis - Run on processors	VTK-m
	In situ analysis – Batch algorithms on processors	ALPINE
	Post-processing analysis – Exploratory	Cinema

ECP applications target national problems

National security

Stockpile stewardship

Next-generation electromagnetics simulation of hostile environment and virtual flight testing for hypersonic re-entry vehicles



Energy security

Turbine wind plant efficiency

High-efficiency, low-emission combustion engine and gas turbine design

Materials design for extreme environments of nuclear fission and fusion reactors

Design and commercialization of Small Modular Reactors

Subsurface use for carbon capture, petroleum extraction, waste disposal

Scale-up of clean fossil fuel combustion

Biofuel catalyst design

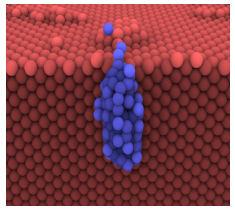
Economic security

Additive manufacturing of qualifiable metal parts

Reliable and efficient planning of the power grid

Seismic hazard risk assessment

Urban planning



Scientific discovery

Find, predict, and control materials and properties

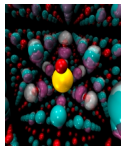
Cosmological probe of the standard model of particle physics

Validate fundamental laws of nature

Demystify origin of chemical elements

Light source-enabled analysis of protein and molecular structure and design

Whole-device model of magnetically confined fusion plasmas



Earth system

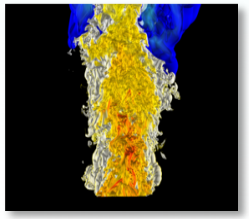
Accurate regional impact assessments in Earth system models

Stress-resistant crop analysis and catalytic conversion of biomass-derived alcohols

Metagenomics for analysis of biogeochemical cycles, climate change, environmental remediation

Health care

Accelerate and translate cancer research



Summary of area

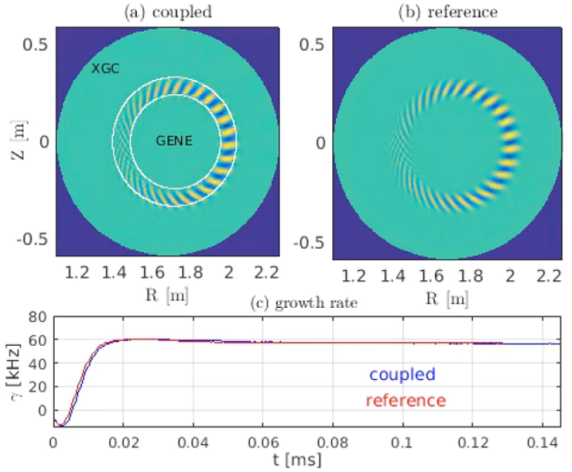
- **Challenge:** Processor to Memory/Storage Gaps
  - Latency Gap
  - Bandwidth/Capability Gap
- **Advantage**: Lots of compute available
- **Services requested:**
  - Fault tolerance
  - Data storage
  - Visual analysis

Overcome gaps via:	Specific approaches	Product highlight
Abstract/virtual interface to hide gap via messaging, caching and asynchrony	Messaging instead of communicating through filesystem	ADIOS
	Add a layer of cache – burst buffer	PnetCDF, HDF5, UnifyFS
	Asynchronous interaction between processors and storage	VeloC
Prioritize and measure functionality	Identify and evaluate I/O patterns, choose frontend API, add ADIOS, MPI-IO, pnetCDF backends, measure, compare	Darshan, HDF5, ADIOS, PnetCDF, MPI-IO
Data reduction and transformation	Compression	SZ, ZFP
	In situ analysis - Run on processors	VTK-m
	In situ analysis – Batch algorithms on processors	ALPINE
	Post-processing analysis – Exploratory	Cinema

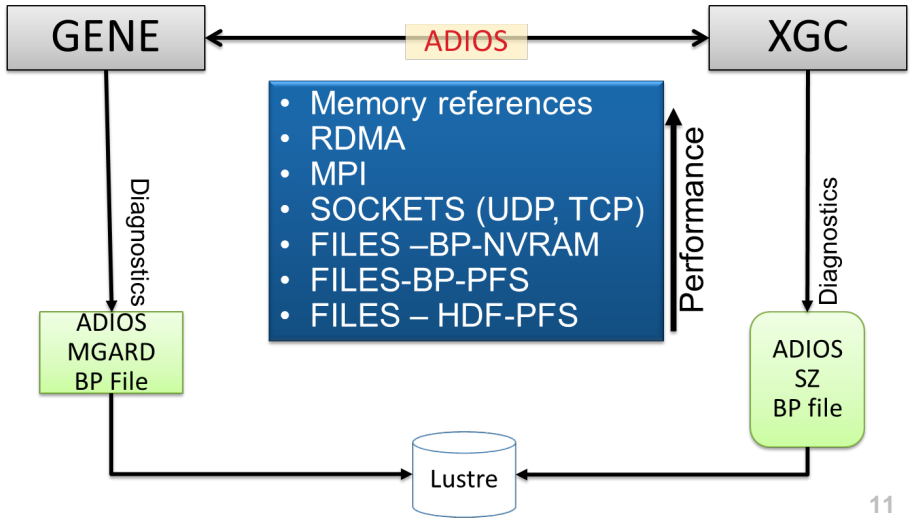
For efficient data communication: **Messaging** instead of communicating through filesystem - ADIOS

- **Problem Statement:**
  - Achieve exacale with multiple simulation codes (Core, Edge and more) fusion
- **ECP Technique solution**
  - ADIOS components allows different I/O methods (engines) to be created for code coupling, with a unified API
  - New high performance coupling engine (SST)
  - Reduced the communication time from 40s to 1s in the coupled code
  - Multiple engine type: the ADIOS coupling engine (SST), along with the MPI engine (InsituMPI), and the file based engine (BPFile) were created to target persistence, performance, and portability
- **Specific impact to ECP Application**
  - First time ever that a PIC (Edge) and Continuum (GENE) code have been coupled;
  - In situ visualization using VTK-M was integrated into the workflow, to enable near-real-time monitoring of the physics and performance (TAU) data
- **Use on ECP Architecture(s)**
  - This technology was run on Titan and has been run on Summit on 2048 nodes
  - Future plans to integrate this with the A21 machine to meet the WDM FoM goals

The figure below shows that a new tight-coupling scheme from ADIOS enabled the accurate coupling of the GENE and XGC codes

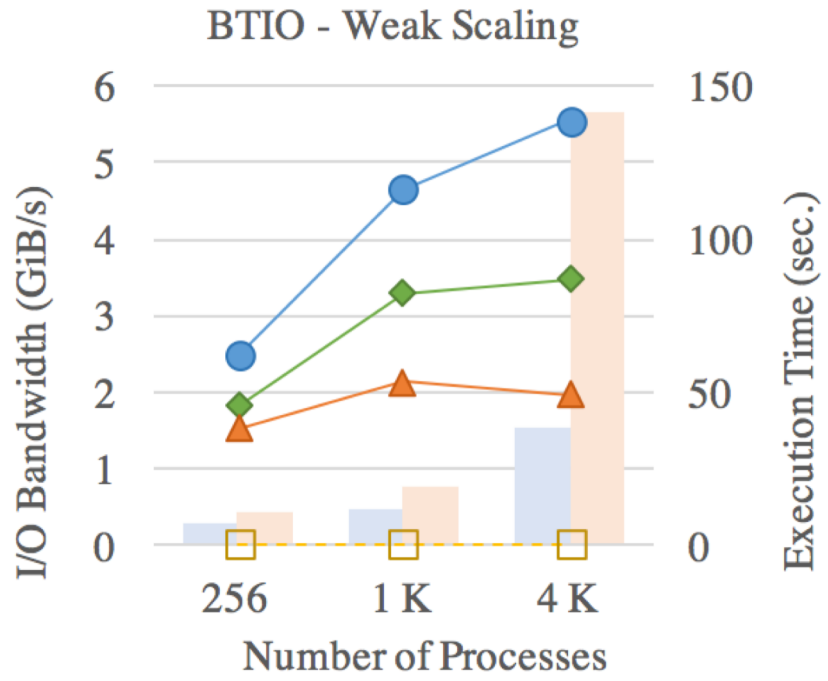


J. Dominski; S. Ku; C.-S. Chang; J. Choi; E. Suchyta; S. Parker;
S. Klasky; A. Bhattacharjee; *Physics of Plasmas* **25**, 072308
(2018) DOI: 10.1063/1.5044707



For efficient data storage: Add a layer of **cache** – Accelerated PnetCDF access using **burst buffers**

- Parallel netCDF is an important I/O library for the ECP and larger ASCR scientific computing communities.
  - Burst buffers are a partial solution to traditional I/O bottlenecks
  - The purpose of this study is to understand the efficacy of our specialized burst buffer optimizations as compared to other proposed methods of leveraging burst buffers in codes.
  - Users of PnetCDF will see performance improvements when performing I/O, with no code modifications.

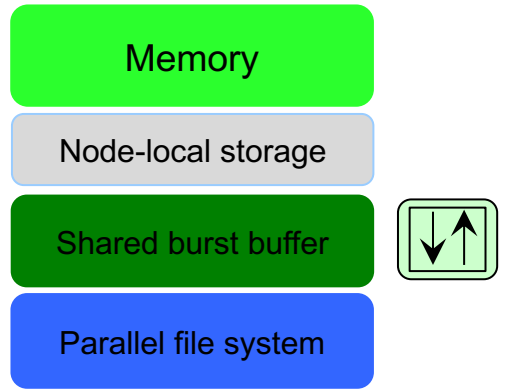


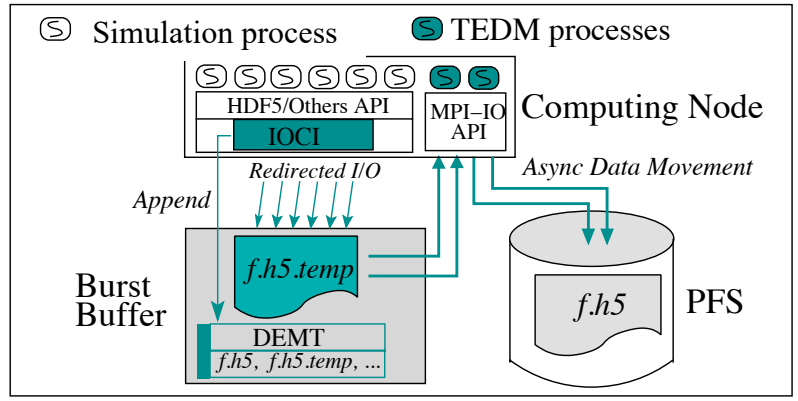
Bandwidth and execution time for BTIO benchmark on Cori.

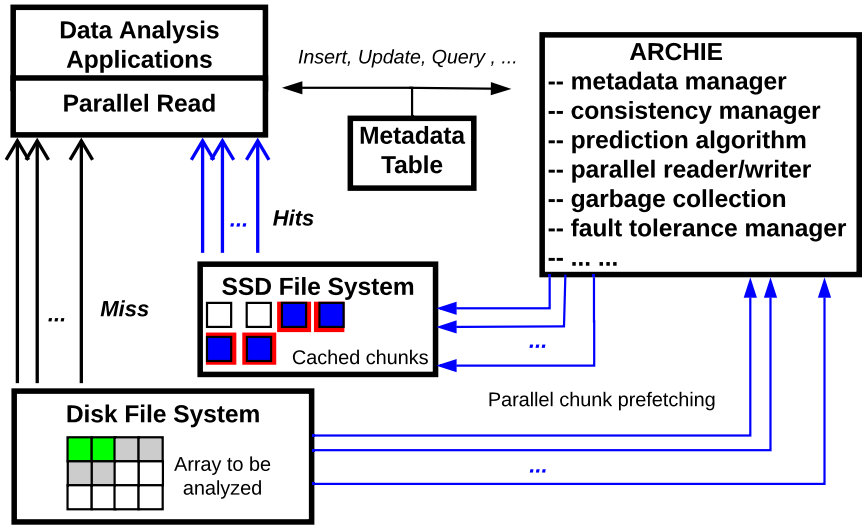
- PnetCDF burst buffer driver (blue)
- PnetCDF Direct I/O to the file system (green)
- DataWarp for stage-out (orange) which exhibits noncontiguous I/O

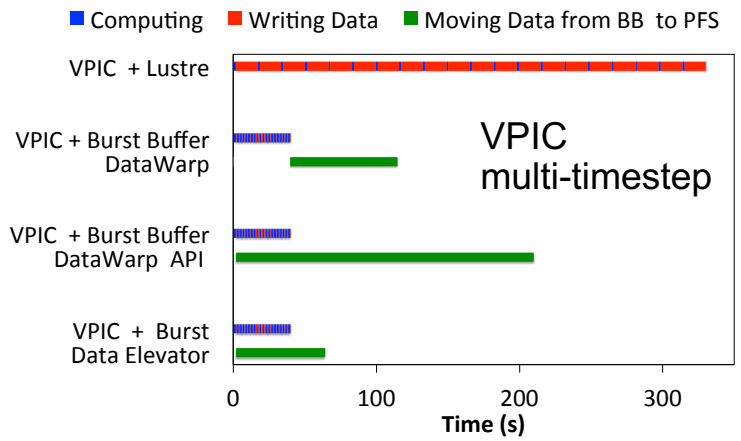
Burst buffer usage in HDF5 - Data Elevator

- Data Elevator VOL connector
  - Transparent data movement in storage hierarchy - writes and reads
  - Intercepts file opens, write, read, and close function calls and places data in burst buffers temporarily; DE moves data asynchronously
  - Prefetches predicted chunks of data to burst buffer or memory
  - In situ data analysis capability using burst buffers
  - Phase 2 plan includes extending capabilities of Data Elevator for node-local storage



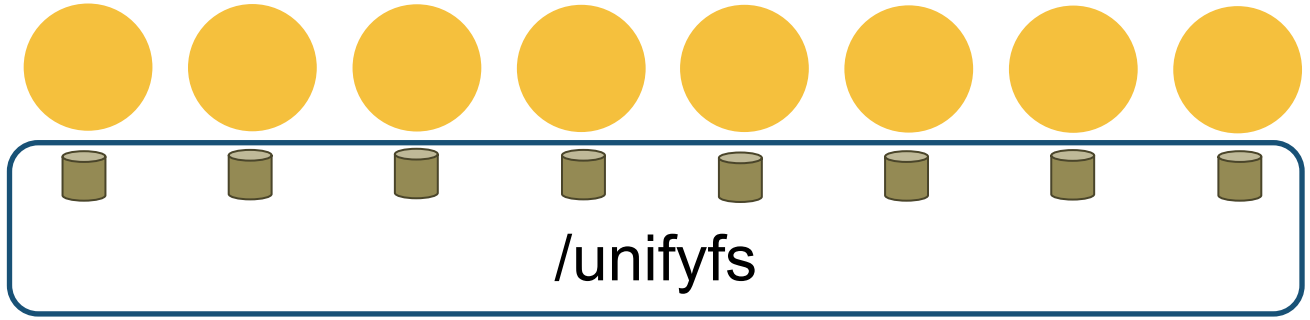






UnifyFS: A file system for burst buffers

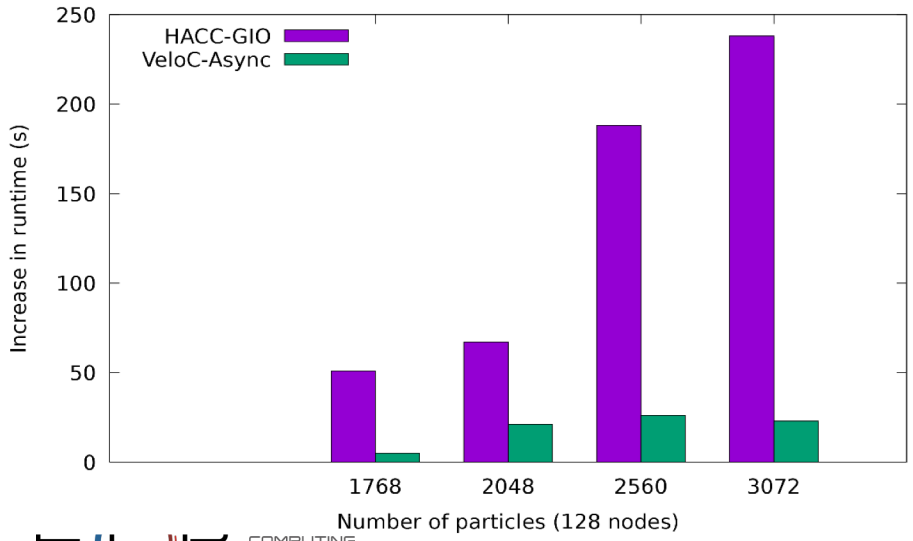
- Sharing files on node-local burst buffers is not natively supported



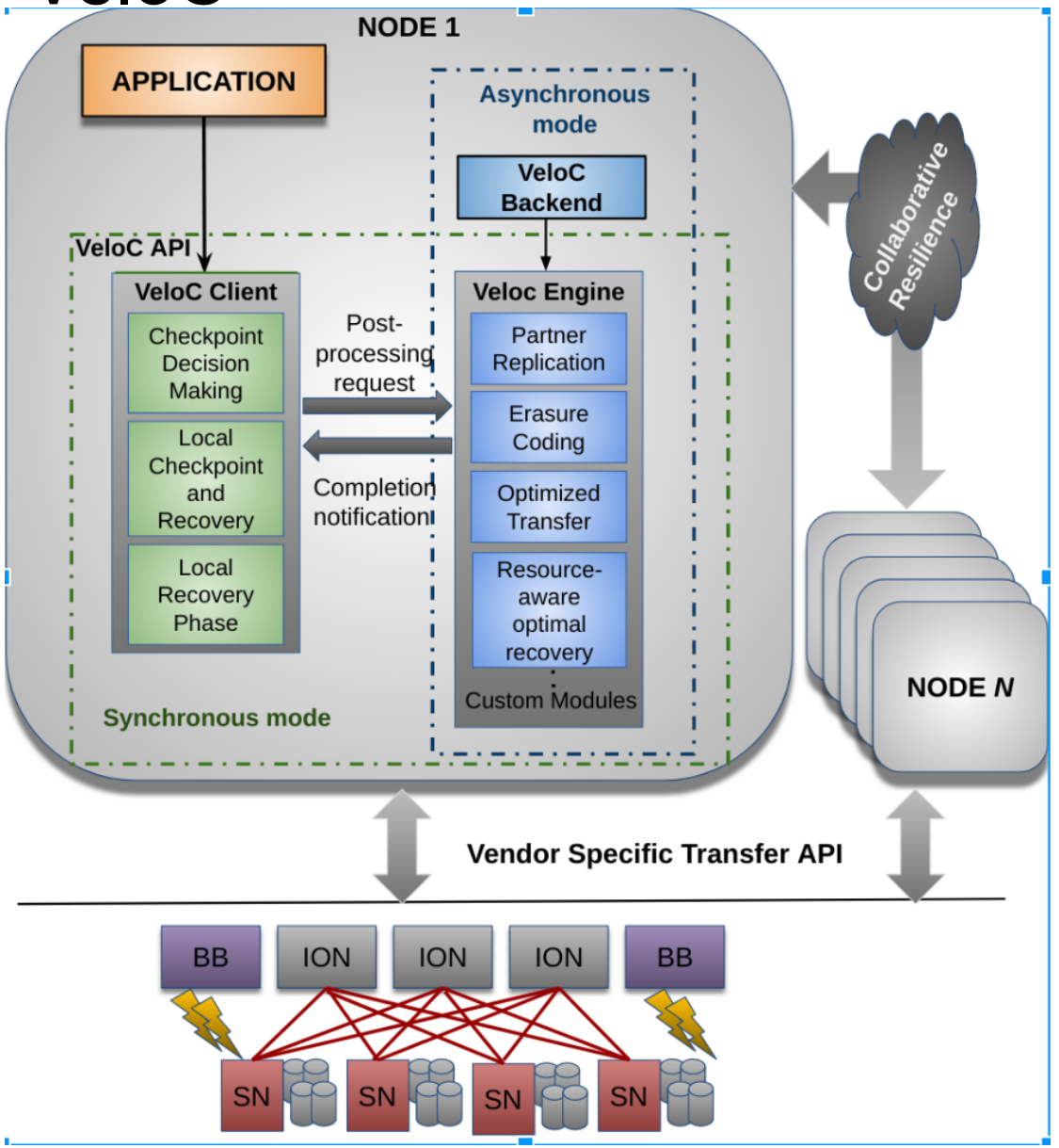
- UnifyFS makes writing/reading shared files **easy**
  - UnifyFS presents a shared namespace across distributed storage
  - Used directly by applications or indirectly via potential integration in higher level libraries like VeloC, MPI-IO, HDF5, PnetCDF, ADIOS, etc.
- UnifyFS is **fast**
  - Tailored for specific HPC workloads, e.g., checkpoint/restart, visualization output
  - Each UnifyFS instance exists only within a single job, no contention with other jobs on the system

For fault tolerance and efficient data storage: Abstract/virtual interface to hide gap via **asynchrony** - VeloC

- Checkpointing is likely to become a serious overhead for Exascale executions
- VELOC provides a low overhead and reliable checkpoint/restart framework to ECP applications leveraging all levels of the storage hierarchy
- Close collaboration with multiple ECP applications to enable integration with VELOC



VeloC in async mode is 10x faster and more scalable than original HACC ckpt strategy based on GIO



Summary of area

- **Challenge:** Processor to Memory/Storage Gaps
  - Latency Gap
  - Bandwidth/Capability Gap
- **Advantage**: Lots of compute available
- **Services requested:**
  - Fault tolerance
  - Data storage
  - Visual analysis

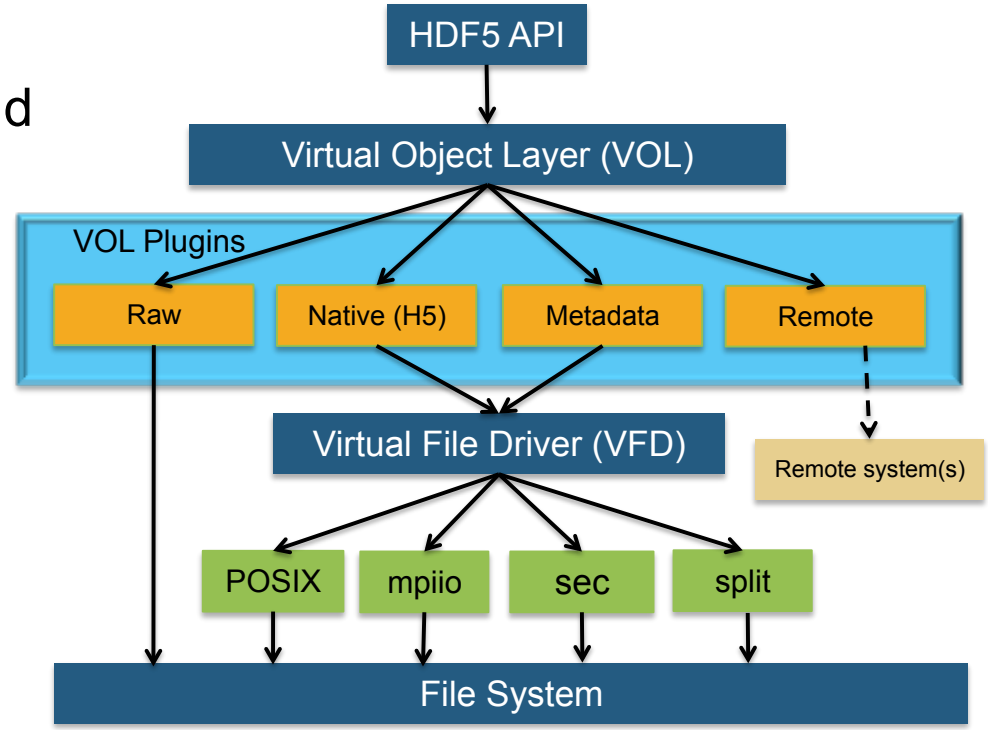
Overcome gaps via:	Specific approaches	Product highlight
Abstract/virtual interface to hide gap via messaging, caching and asynchrony	Messaging instead of communicating through filesystem	ADIOS
	Add a layer of cache – burst buffer	PnetCDF, HDF5, UnifyFS
	Asynchronous interaction between processors and storage	VeloC
Prioritize and measure functionality	Identify and evaluate I/O patterns, choose frontend API, add ADIOS, MPI-IO, pnetCDF backends, measure, compare	Darshan, HDF5, ADIOS, PnetCDF, MPI-IO
Data reduction and transformation	Compression	SZ, ZFP
	In situ analysis - Run on processors	VTK-m
	In situ analysis – Batch algorithms on processors	ALPINE
	Post-processing analysis – Exploratory	Cinema

Plan for efficient exascale data storage

Plan	Details
1. Identify & prioritize common I/O patterns	M to M, M to N, M to 1 read/writes and other patterns
2. ECP Data and Visualization supports the HDF5 API as the frontend API	
3. Use **HDF5 VOL layer** or other technology to make HDF5 backends	ADIOS, MPI-IO, PnetCDF
4. Measure and compare, repeat	Compare on supercomputing architectures - **Darshan tool**

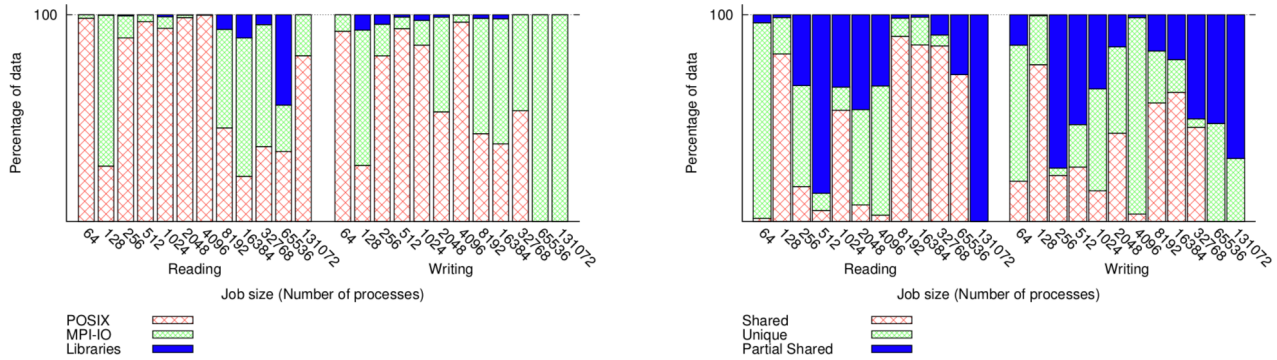
Virtual Object Layer (VOL) for opening HDF5 API

- VOL provides an application with the HDF5 data model and API, but allow different underlying storage mechanisms
- Enables developers to easily use HDF5 on novel current and future storage systems
  - VOL connectors for using burst buffer storage transparently (Data Elevator) and for accessing DAOS are available
  - VOL plugins for reading PnetCDF and ADIOS data are in development
- Integrated into the HDF5 trunk https://bitbucket.hdfgroup.org/projects/HDFFV/repos/hdf5/
- Allows ADIOS and PnetCDF file formats to use HDF5 API
- VOL Connectors repo: https://bitbucket.hdfgroup.org/projects/HDF5VOL



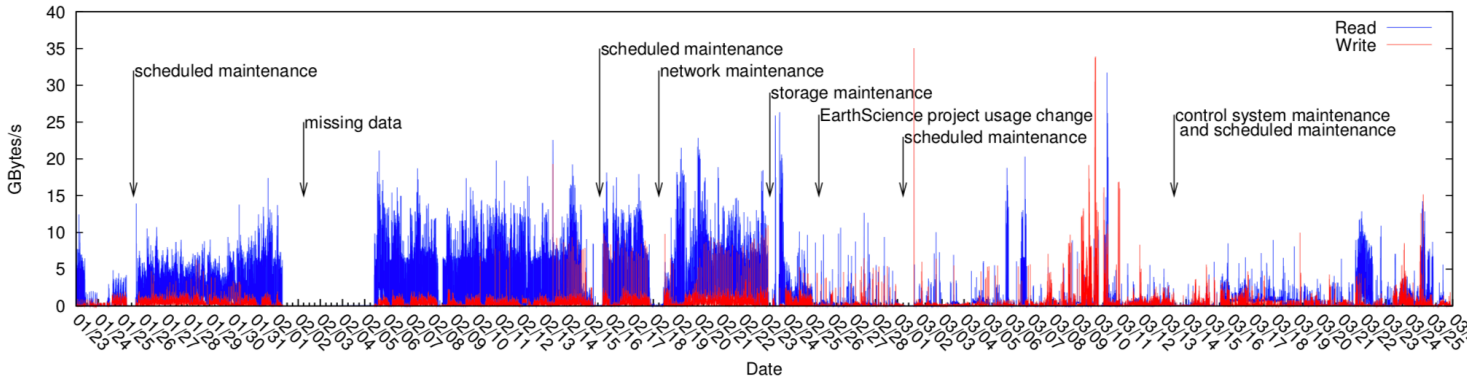
For efficient data storage – Darshan - Measure and understanding overall parallel I/O patterns on exascale supercomputers

- **Problem statement:**
  - What applications are running, what interfaces are they using, and who are the biggest I/O producers and consumers?
  - How busy is the I/O system, how many files are being created of what size, and how "bursty" is I/O?
  - What I/O interfaces and strategies are employed by the top I/O producers and consumers? How successful are they in attaining high I/O efficiency? Why?
- **Global solution:**
  - Add Darshan monitoring software to each platform
- **Benchmark solution:**
  - For specific ECP applications to compare backends



(a) Interfaces (b) File access patterns

I/O strategies used by jobs as a function of job size.



Aggregate throughput on one-minute intervals.

Summary of area

- **Challenge:** Processor to Memory/Storage Gaps
  - Latency Gap
  - Bandwidth/Capability Gap
- **Advantage**: Lots of compute available
- **Services requested:**
  - Fault tolerance
  - Data storage
  - Visual analysis

Overcome gaps via:	Specific approaches	Product highlight
Abstract/virtual interface to hide gap via messaging, caching and asynchrony	Message instead of communicate through filesystem	ADIOS
	Add a layer of cache – burst buffer	PnetCDF, HDF5, UnifyFS
	Asynchronous interaction between processors and storage	VeloC
Prioritize and measure functionality	Identify and evaluate I/O patterns, choose frontend API, add ADIOS, MPI-IO, pnetCDF backends, measure, compare	Darshan, HDF5, ADIOS, PnetCDF, MPI-IO
Data reduction and transformation	Compression	SZ, ZFP
	In situ analysis - Run on processors	VTK-m
	In situ analysis – Batch algorithms on processors	ALPINE
	Post-processing analysis – Exploratory	Cinema

Data reduction via compression – SZ compression library

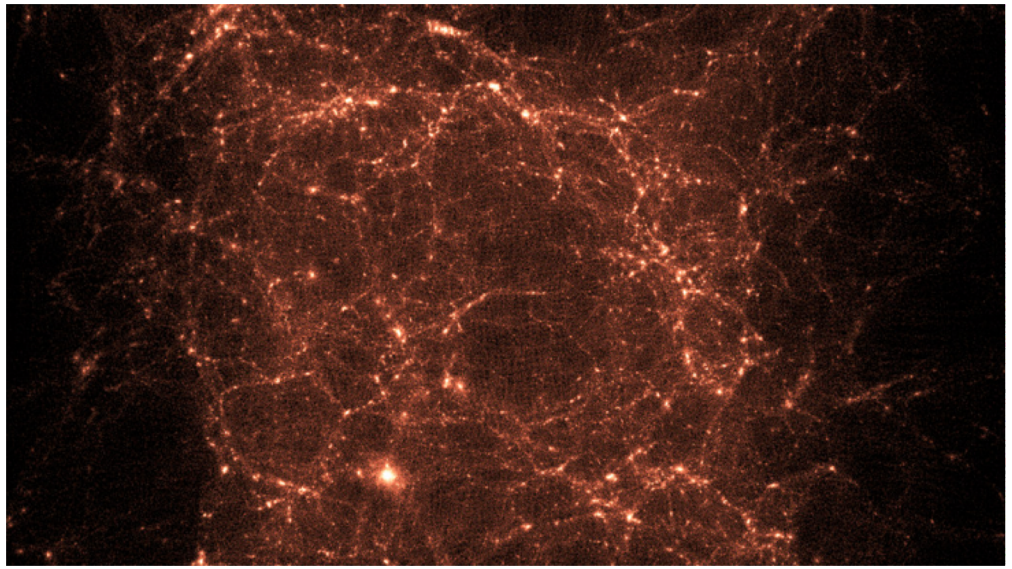
HACC Cosmology application: N-body problem with domain decomposition, medium/long-range force solver (particle-mesh method), short-range force solver (particle-particle/particle-mesh algorithm).

Particle dataset:

6 x 1D array:

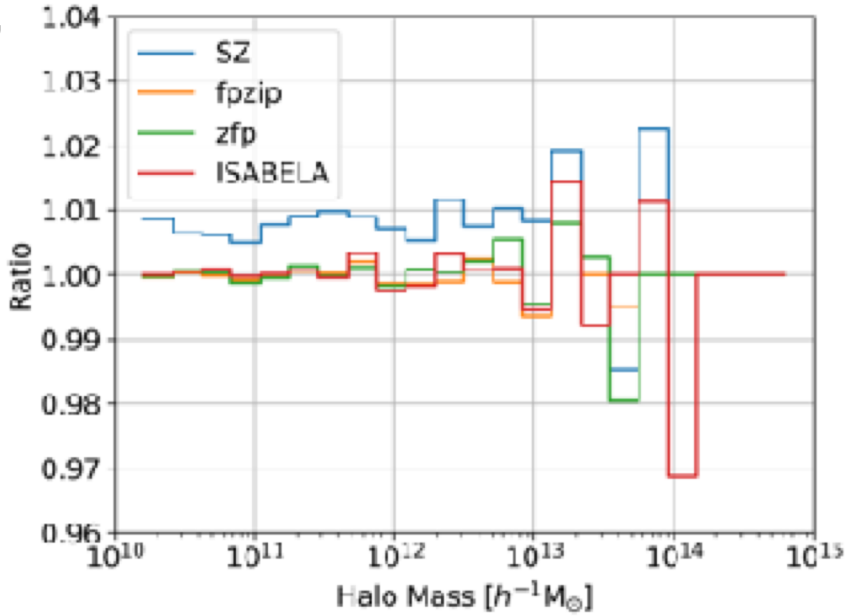
x, y, z, vx, vy ,vz

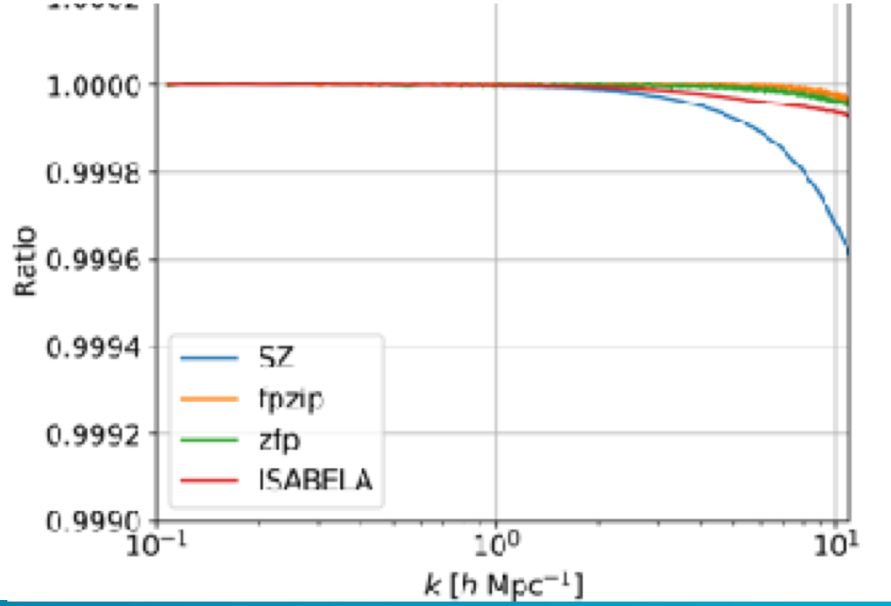
Preferred error controls:
- Point wise error bound
  - Absolute (position),
  - Relative (Velocity)



Lossless compression ratio: ~1.77
SZ lossy compression (10-3 error bound): ~4.8
Impacts : reduce data footprint on storage, accelerate I/O

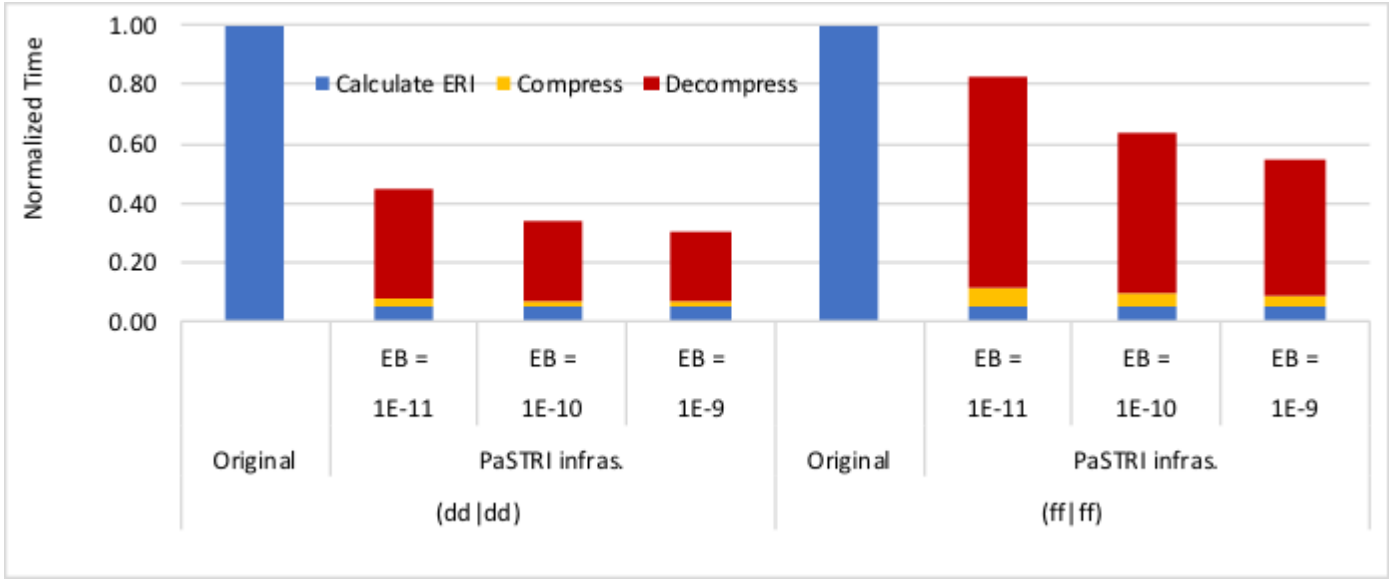
Validation of the compression results using the test harness





Data reduction via compression – SZ compression library

- Performance of two electron integral simulations is limited by the re-computation of large number of integrals at each iteration because of lack of memory space
- The SZ team developed a new algorithm for compression of two electron integral simulation with very high compression ratio and fidelity
  - Integration in SZ and test in GAMESS
  - 16.8X compression ratio, 661MB/s compression rate, 1116MB/s decompression rate
- Performance improvement of GAMESS (on the tested cases) of 1.25 to 2.5 depending on the simulations and error bound (these accelerations are highly dependent on the simulation)
- Overall best paper award at IEEE Cluster 2018



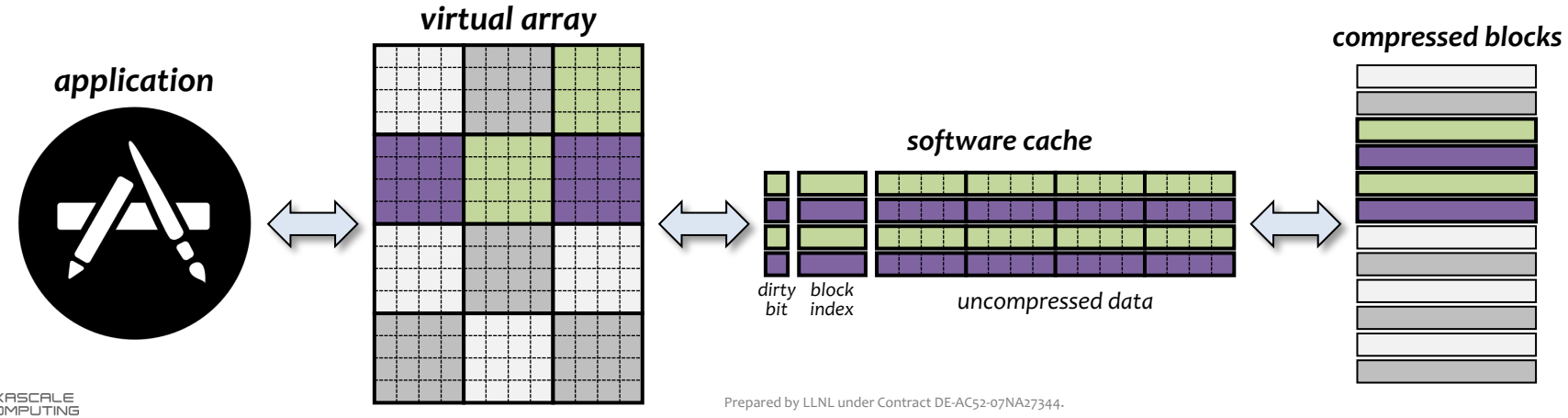
Ref: Ali Murat Gok, Sheng Di, Yuri Alexeev, and Dingwen Tao, Vladimir Mironov, Xin Liang, Franck Cappello, PaSTRI: Error-Bounded Lossy Compression for Two-Electron Integrals in Quantum Chemistry, IEEE Cluster 2018, Belfast, IEEE press

Figure: acceleration on 1 process, serial execution, compressed integrals stored in memory.

We also have the parallel performance with the compressed integrals stored on disks.

Data reduction - ZFP provides a compressed multidimensional array primitive

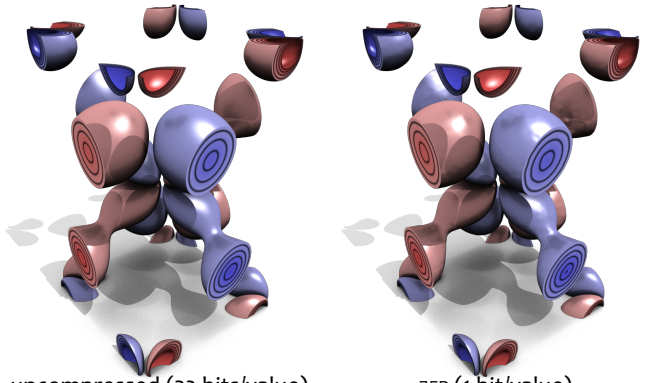
- Fixed-length compressed blocks enable fine-grained read & write **random access**
  - C++ compressed-array classes hide complexity of compression & caching from user
  - User specifies per-array storage footprint in bits/value
  - Inspired by fixed-rate texture compression methods widely adopted in graphics hardware, but has been tailored to the high dynamic range and precision demands of scientific applications
  - Based on a new, lifted, orthogonal block transform and embedded coding, allowing each per-block bit stream to be truncated at any point if desired, thus facilitating bit rate selection using a single compression scheme
- Absolute and relative **error tolerances** supported for offline storage, sequential access
- Fast, hardware friendly, and parallelizable: **150 GB/s throughput** on NVIDIA Volta
- **HPC tool support**: ADIOS, CEED, Conduit, E4S, HDF5, Intel IPP, TTK, VTK-m, …



Prepared by LLNL under Contract DE-AC52-07NA27344.

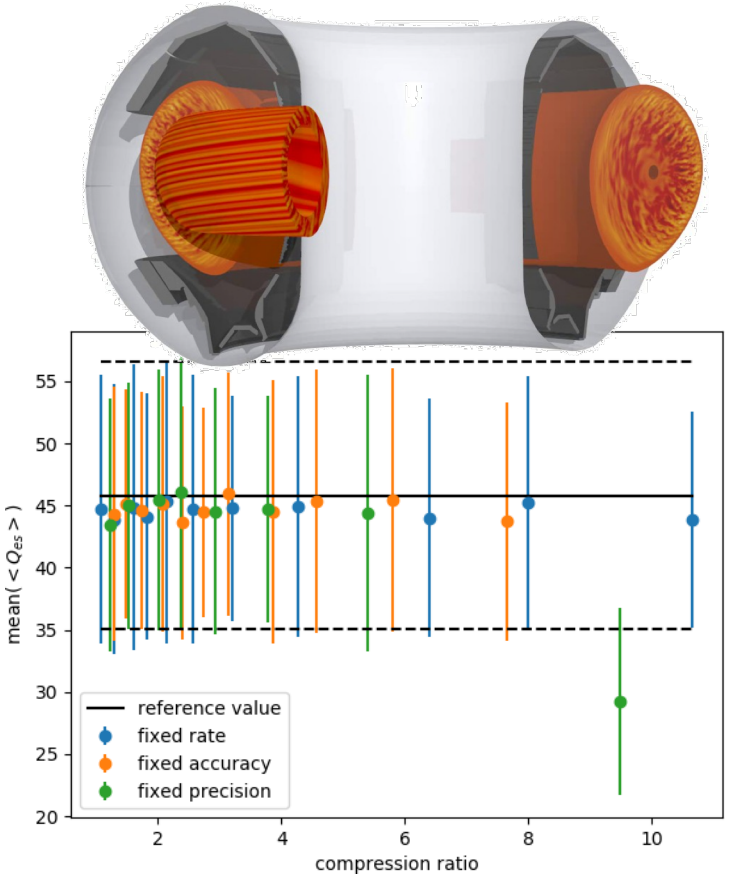
ZFP boosts available memory, I/O bandwidth, offline storage, and accuracy per stored bit in numerical computations

Compressed, pre-computed wavefunctions reduce memory footprint in QMCPACK code

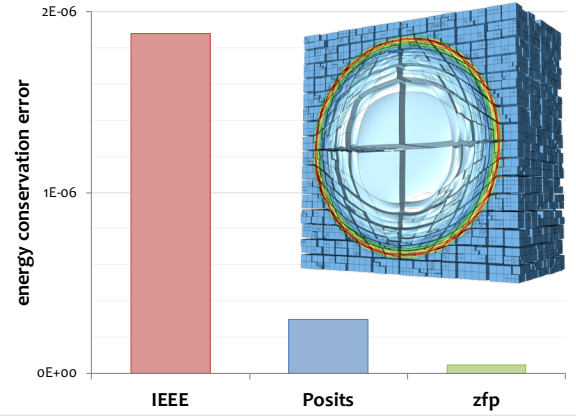


uncompressed (32 bits/value) ZFP (1 bit/value)

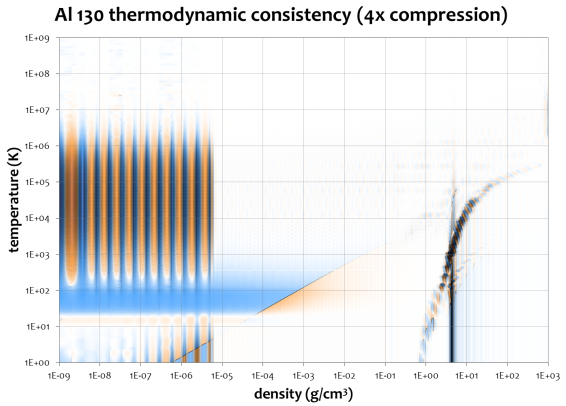
10x compression of simulation state in GENE fusion code with acceptable loss



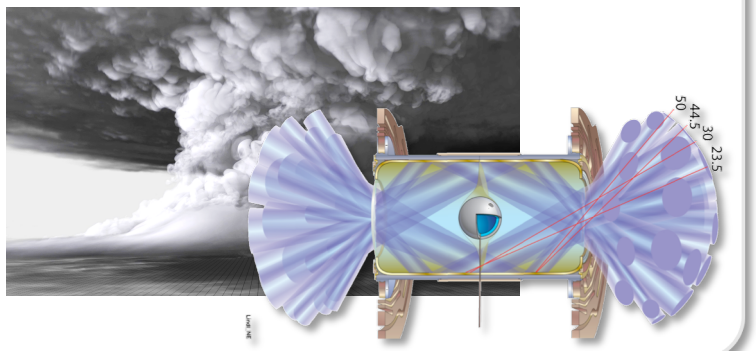
Inline compression boosts solution accuracy by 40x over IEEE in CEED code



4x in-memory reduction of EOS tables

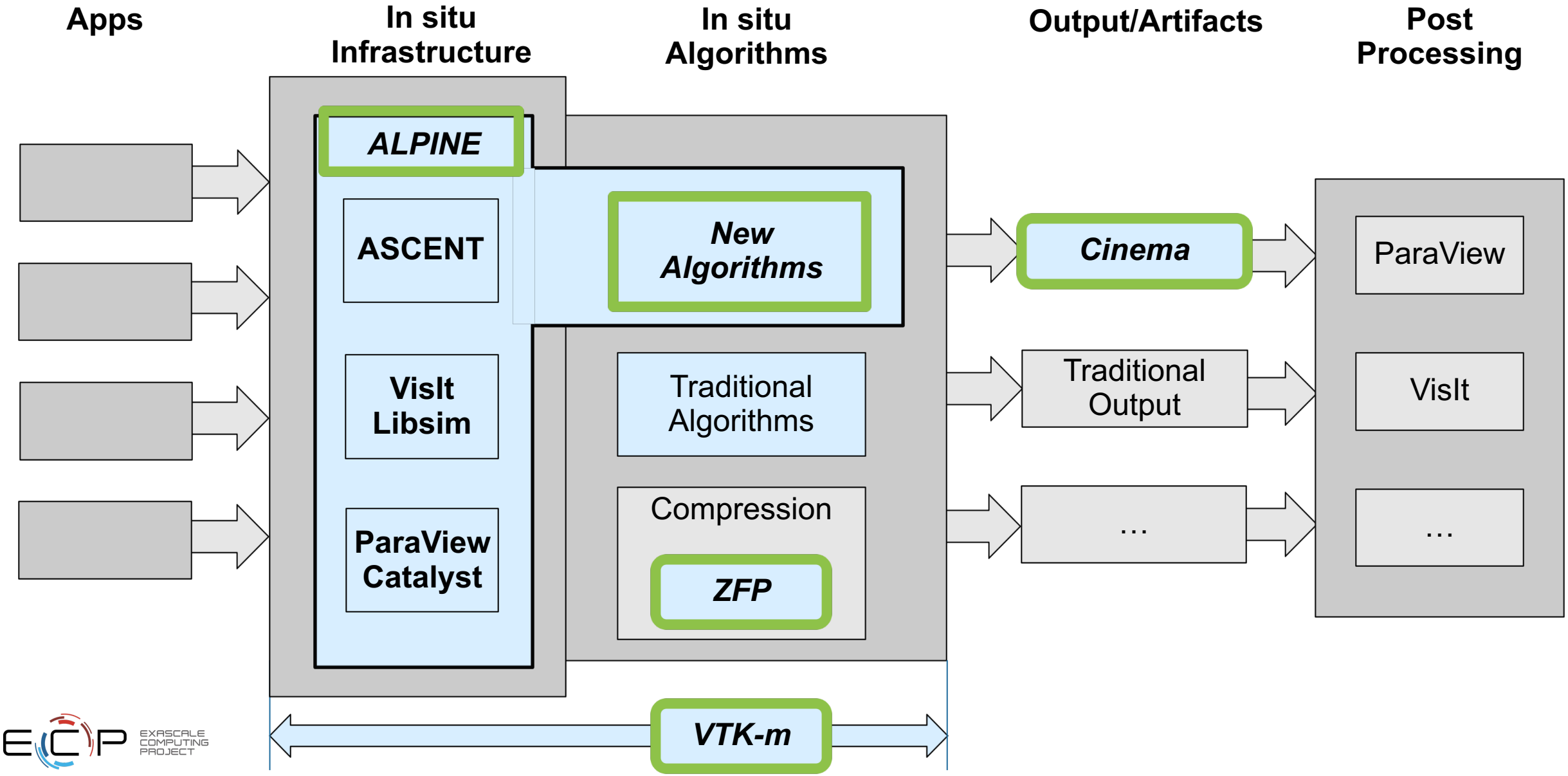


10–1,000x I/O compression enables new science through higher-fidelity analysis



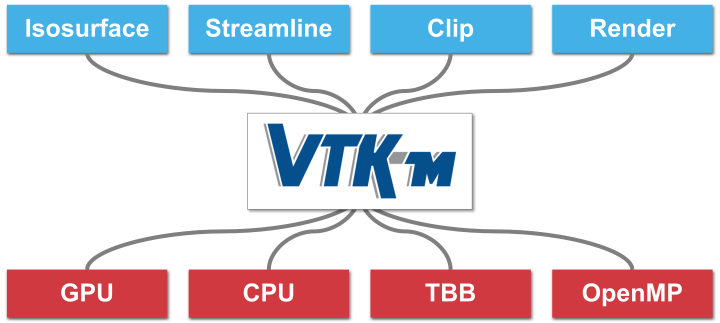
Prepared by LLNL under Contract DE-AC52-07NA27344.

ECP Software Technology Data and Visualization projects provide an integrated workflow



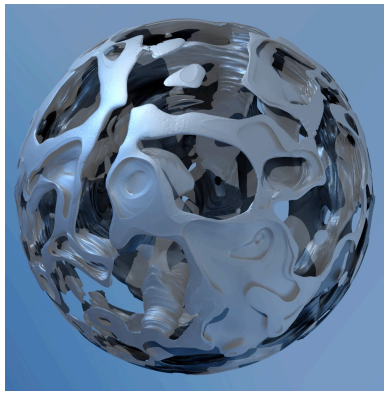
Data reduction on the exascale supercomputers - VTK-m delivers portable performance of visualization algorithms across ECP processors

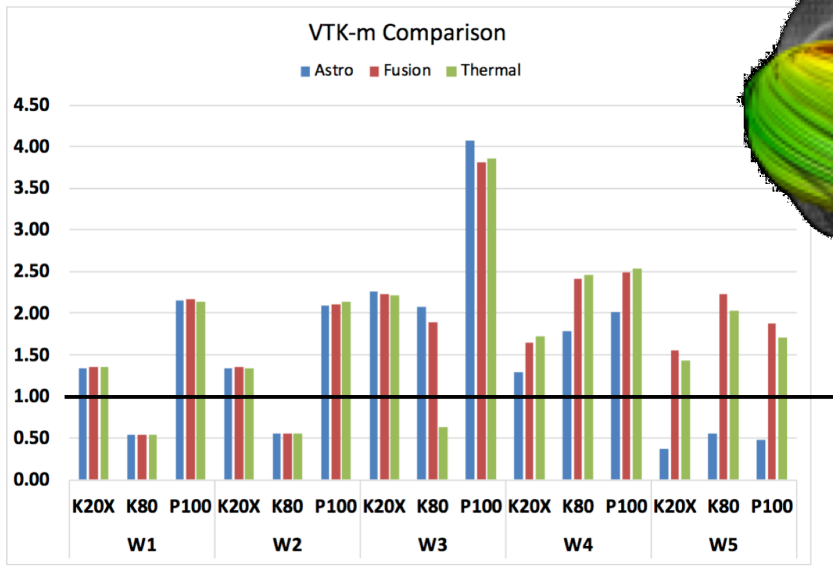
- Development is accelerated by VTK-m's code once, run everywhere
  - Data parallel programming abstraction
- Ray tracing performance in VTK-m is within a factor of 2 of highly optimized, processor-specific implementations developed by Intel and NVIDIA [Larsen et al, 2015 & Moreland, et al., 2016]
- Particle advection in VTK-m performs *better* than code specifically developed for CUDA processors [Pugmire, et al. 2018]

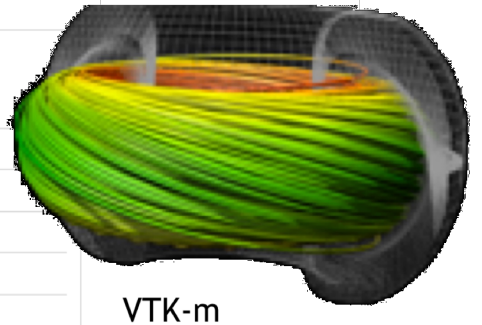


Dataset	Algorithm	Millions of rays per second
LT_350K	OptiX Prime	357.6
	EAVL	150.8
	VTK-m	164.5
LT_372K	OptiX Prime	322.4
	EAVL	124.7
	VTK-m	140.8
RM_350K	OptiX Prime	436.5
	EAVL	197.5
	VTK-m	200.8
RM_650K	OptiX Prime	420.4
	EAVL	172.9
	VTK-m	166.0
RM_970K	OptiX Prime	347.1
	EAVL	152.8
	VTK-m	163.5
RM_1.7M	OptiX Prime	266.8
	EAVL	136.6
	VTK-m	148.8
RM_3.2M	OptiX Prime	264.5
	EAVL	124.8
	VTK-m	134.5
Seismic	OptiX Prime	267.8
	EAVL	106.3
	VTK-m	119.4

Dataset	Algorithm	Millions of rays per second
LT_350K	Embree	51.9
	EAVL	27.7
	VTK-m	38.5
LT_372K	Embree	56.5
	EAVL	26.1
	VTK-m	36.0
RM_350K	Embree	64.8
	EAVL	33.3
	VTK-m	47.8
RM_650K	Embree	65.9
	EAVL	35.6
	VTK-m	49.1
RM_970K	Embree	59.1
	EAVL	29.3
	VTK-m	41.0
RM_1.7M	Embree	52.4
	EAVL	27.0
	VTK-m	37.8
RM_3.2M	Embree	48.4
	EAVL	28.3
	VTK-m	33.9
Seismic	Embree	43.2
	EAVL	25.2
	VTK-m	34.5







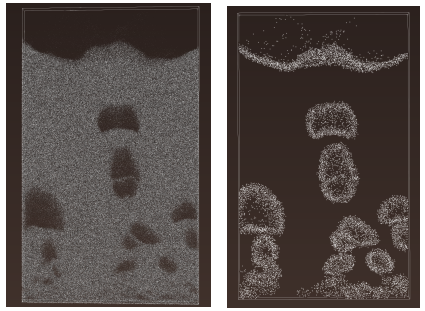
For data reduction - ALPINE in situ batch algorithms

Exascale Challenges:
- I/O limitations will require in situ approaches
- Need data-driven approaches – no human in the loop

ALPINE Algorithms:
- A suite of flexible analysis algorithms to identify important features, time steps, spatial extents & minimize saved data
- Deliver analysis capabilities to ECP applications across exascale architectures.
- Productized, scalable, highly parallel, sustainable solutions.
- These products will be available for ECP and beyond.

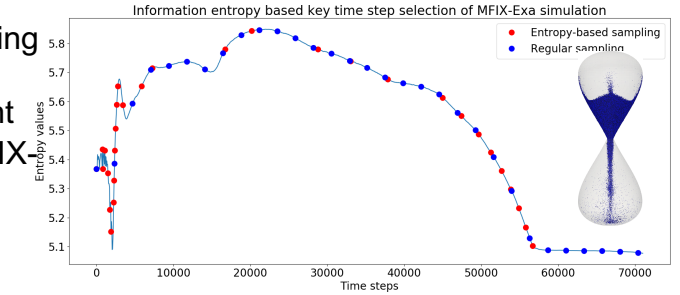
Rotation Invariant Pattern Detection

Pattern detection using moments invariance finds bubbles in a MFIX-Exa bubbling bed. Pattern can be found regardless of orientation in the data.

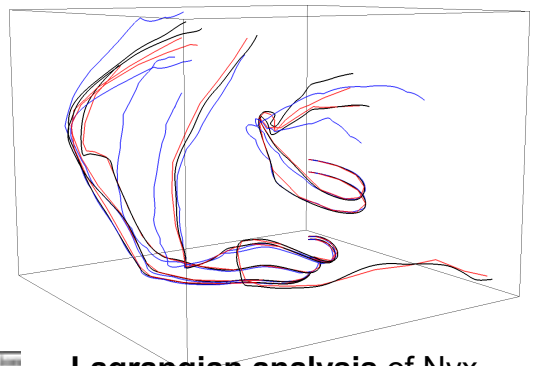


Temporal Entropy-based Sampling

Temporal sampling of entropy to identify important timesteps in MFIX-Exa hourglass simulation.

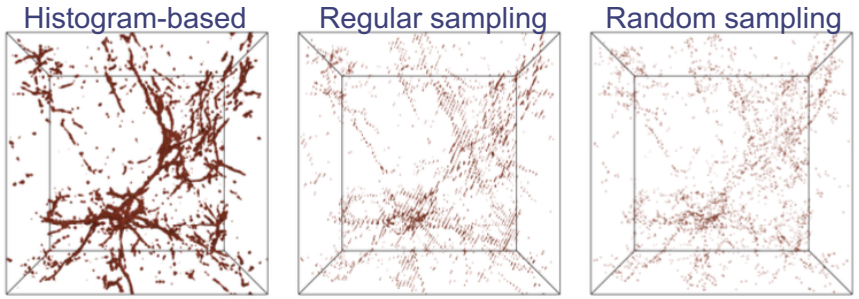


Lagrangian vector flow analysis



Lagrangian analysis of Nyx vector fields allows more efficient and complete analysis and tracking of flow. Lagrangian trajectories (red) follow ground truth (black) pathlines more closely than an Eulerian approach (blue).

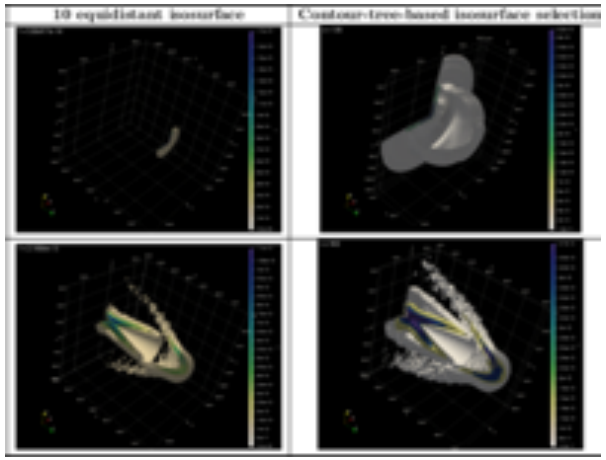
Histogram-based Sampling



Using a histogram-based sampling, ExaSky:Nyx halos are more distinct than using regular or random sampling methods.

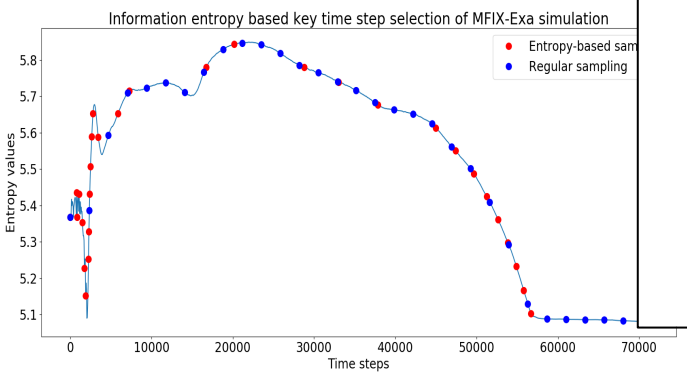
Contour Tree Algorithm

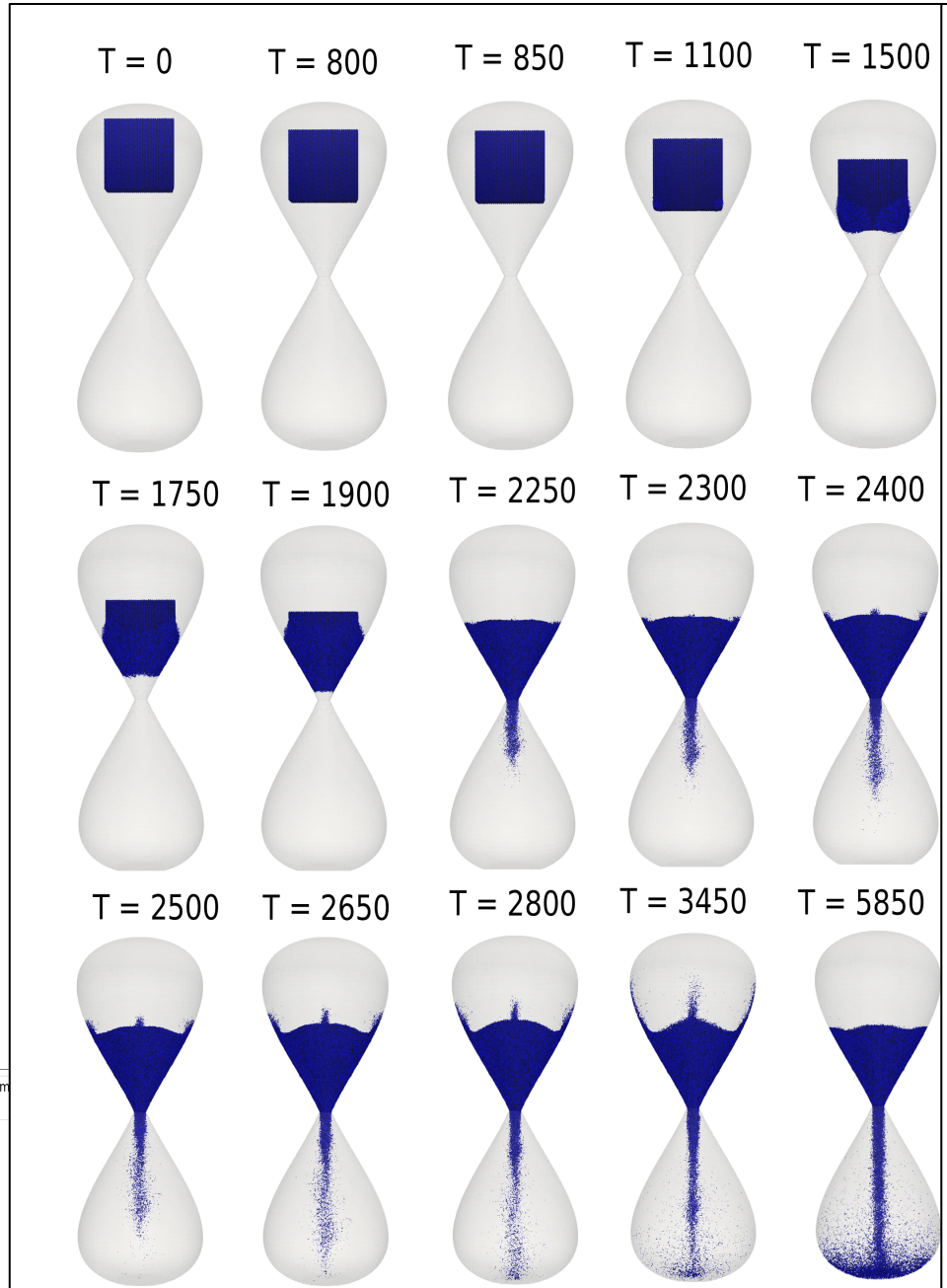
Topological analysis is used in a contour tree algorithm to adaptively compute optimal isovalues during in situ visualization of WarpX current density *J*. Left: isosurfaces selected uniformly. Right: isosurfaces adaptively selected.



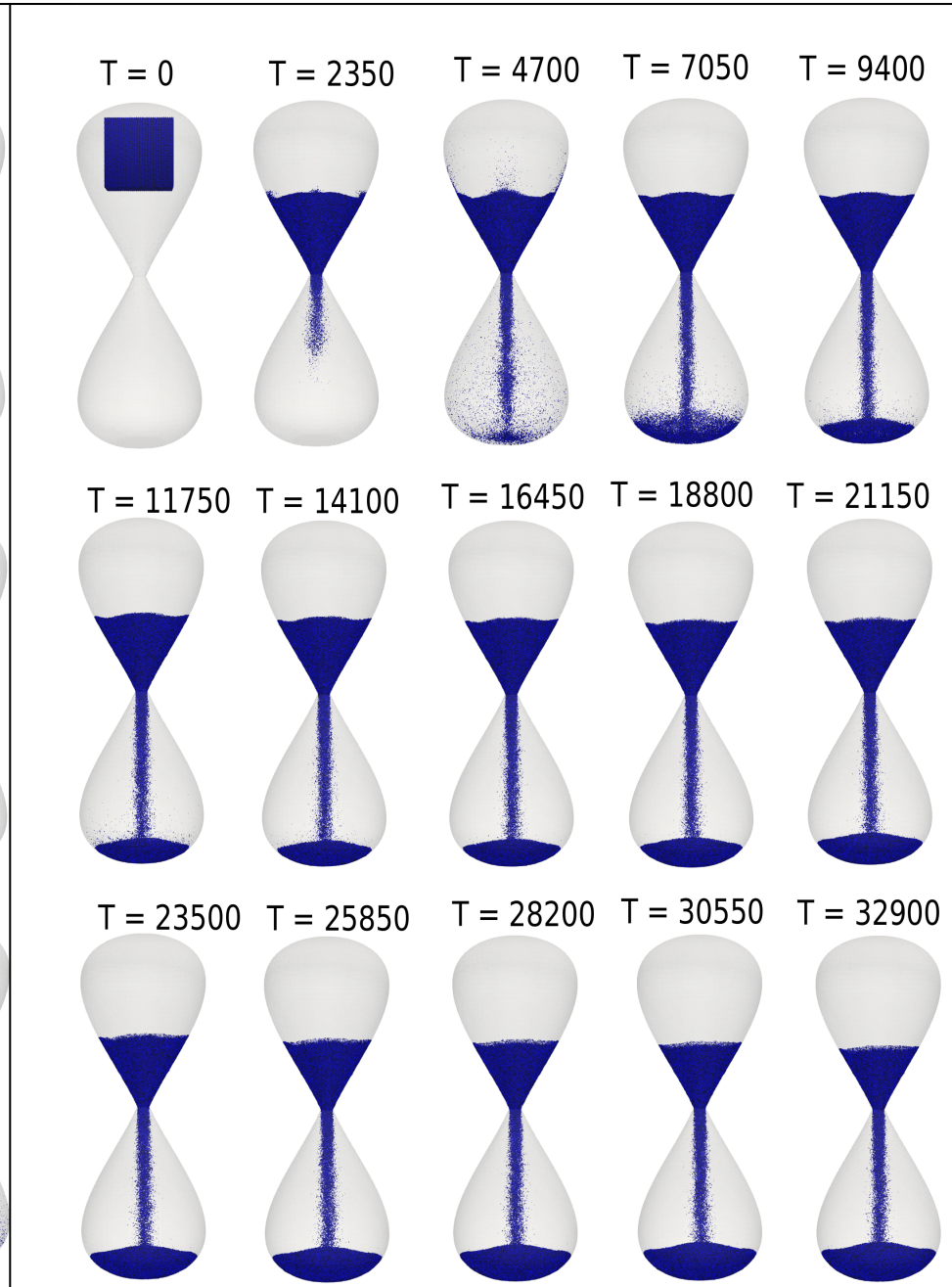
Example – Data reduction via in situ time step sampling for MFIX

- In situ entropy-based time step selection applied to Multiphase Energy simulation MFIX-Exa
- The use case simulates particles falling and interacting in an hourglass
- Result shows superiority of the proposed scheme over regular time step sampling





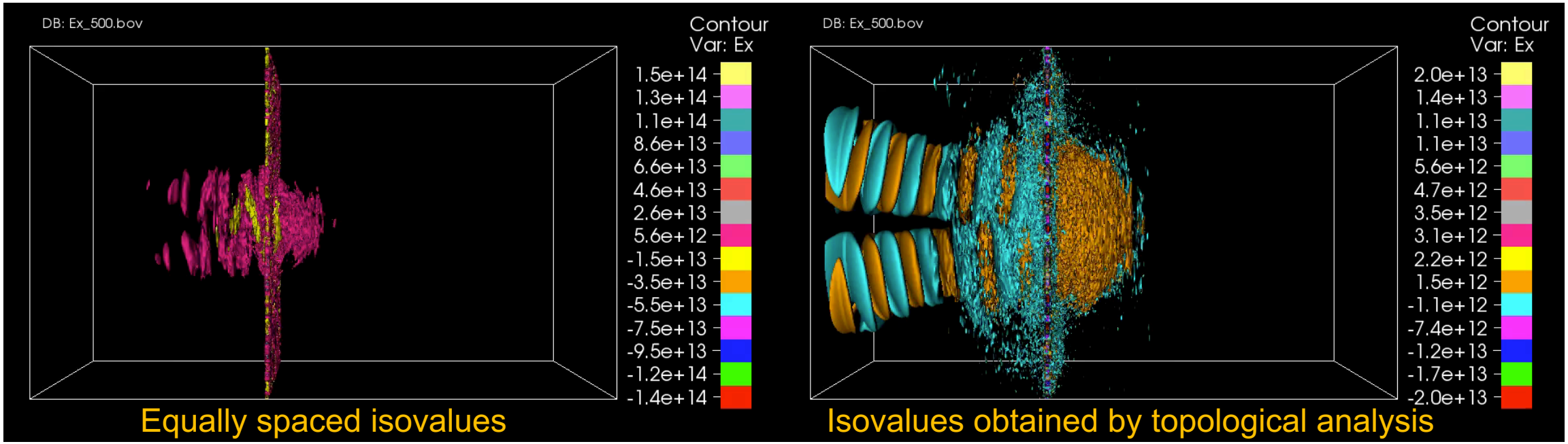
Entropy-based selection



Regular selection

ECP ALPINE

Data reduction - topological analysis identifies appropriate parameters for *in situ* visualization of simulation results



Equally spaced isovalues

Isovalues obtained by topological analysis

Problem:

- Compute power growing faster than I/O bandwidth; requires running visualization *in situ* to reduce I/O
- Need to determine visualization parameters (e.g., isovalues) without user interaction

Approach:

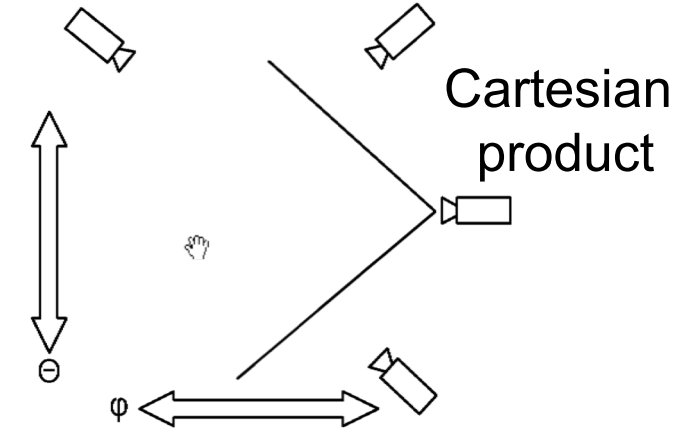
New algorithms for topologic analysis at scale; identify most relevant isosurfaces

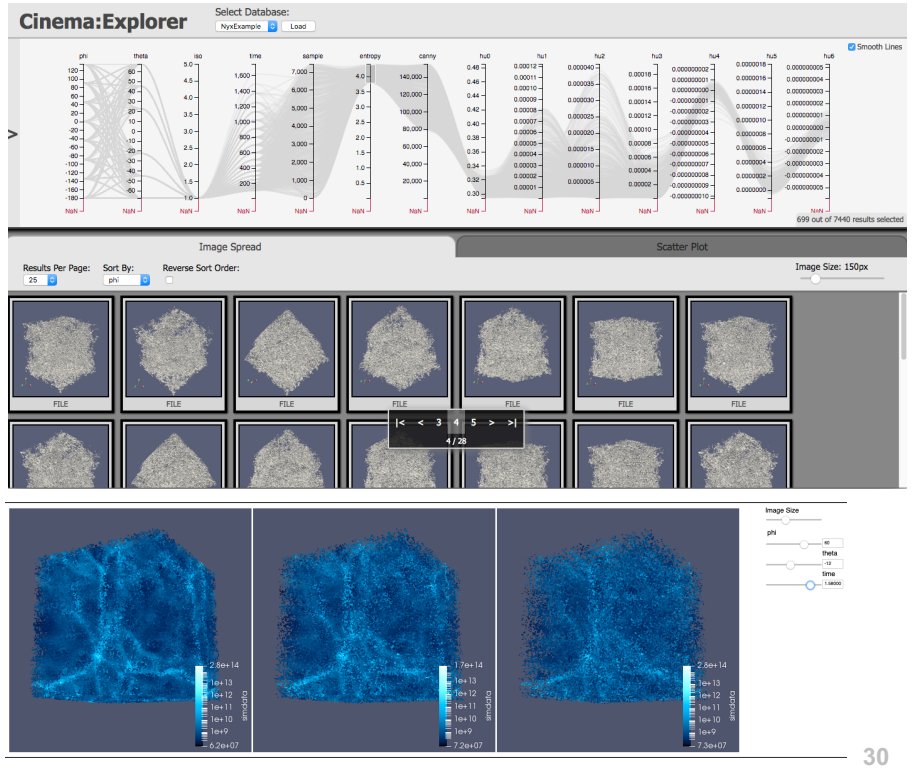
Impact:

Isosurface extraction one of the most common visualization techniques for scalar data; benefits many ECP applications

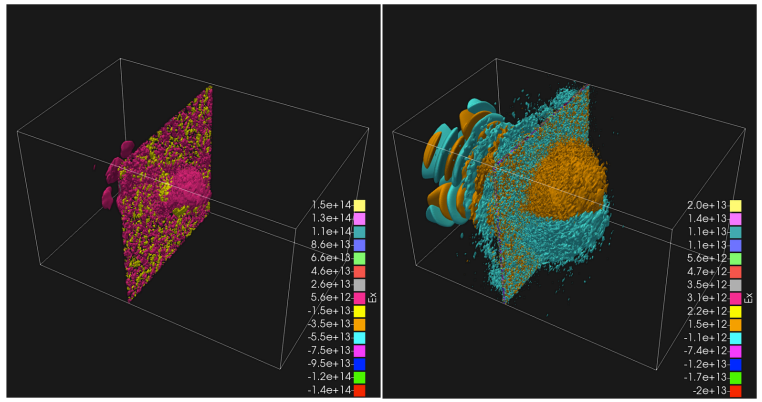
Exploratory post-processing visual analysis - Cinema

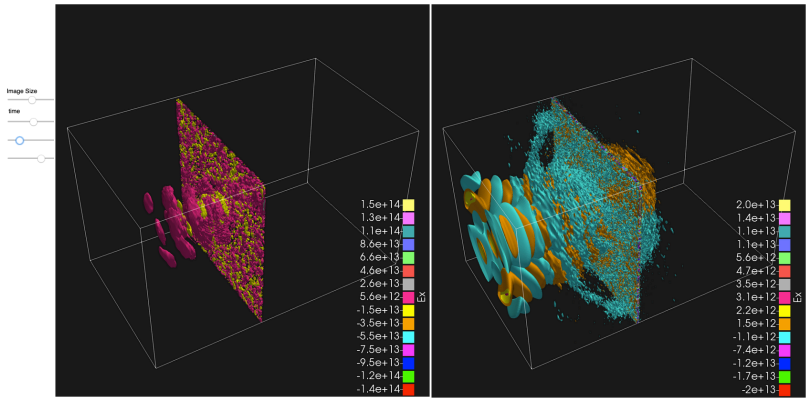
- Approach: Visualize all results needed while simulation data is in memory
  - Operators, camera positions/angles, simulation and algorithmic parameters
- Result: Database of data abstracts: images, meshes, run metadata, output variables, etc.
  - For images, pixel accurate results when compared to post-processing rendering of the same data
- Properties of this solution:
  - Sampling of visualization result image output space
  - Visualization/rendering as sampling/data reduction operator
  - Enables flexible post-processing visualization and analysis on data abstracts, e.g., image-based computer vision techniques, fast visual scans/comparisons

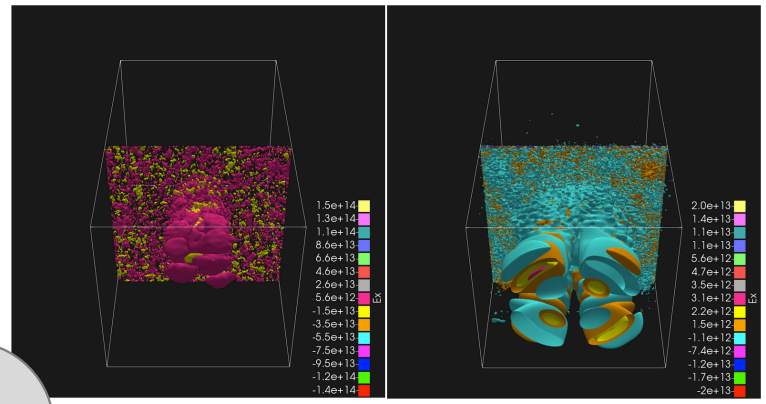


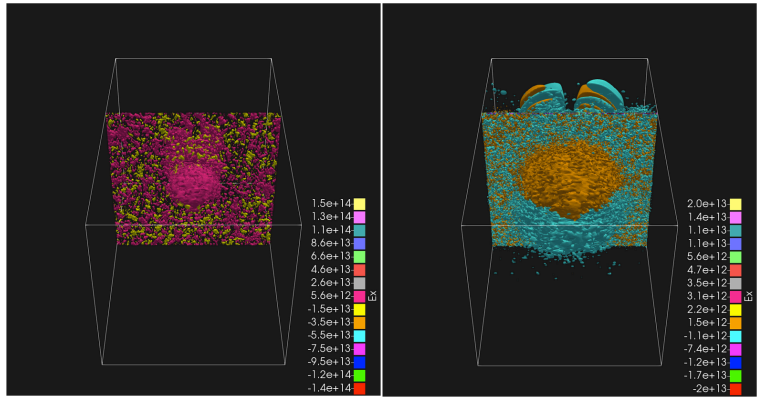


Mega	Giga	Tera	Peta	Exa
10^6	10^9	10^12	10^15	10^18
Image	Storage & network	Operations / data size	Operations / data size	Operations / data size

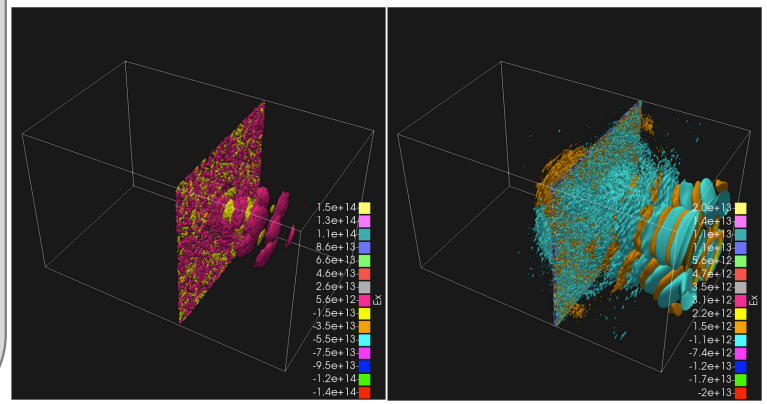


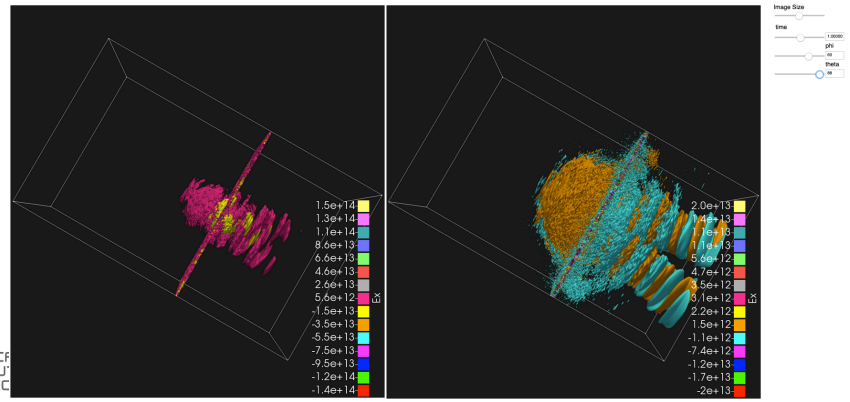


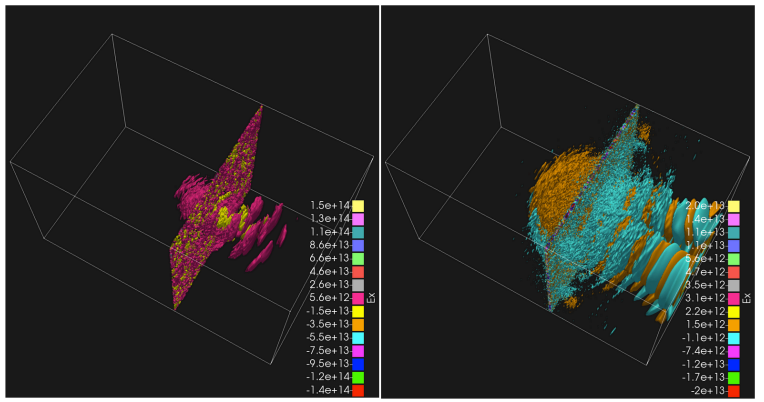




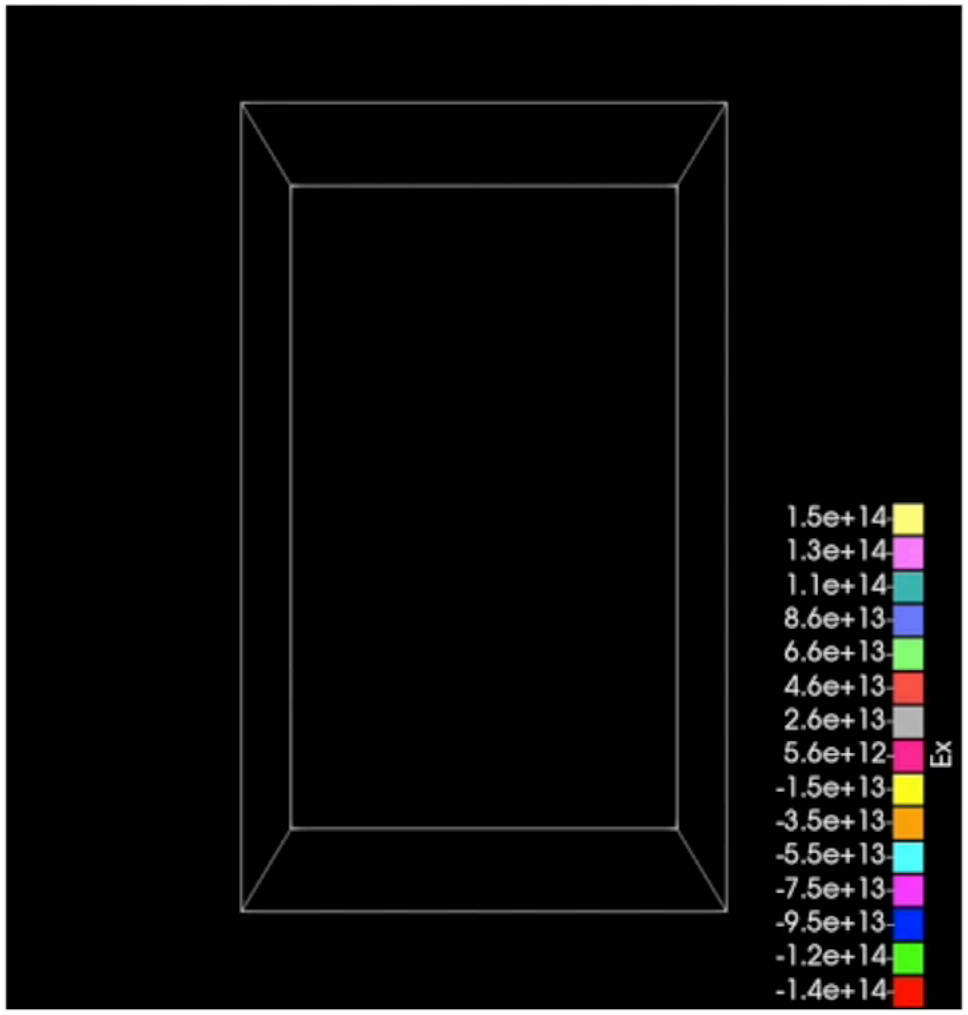
Using Cinema to visually explore spatial differences for a Warp particle accelerator simulation. Left are regularly spaced isosurfaces. Right are data-driven isosufaces selected by in situ topological analysis.







Cinema Explorer



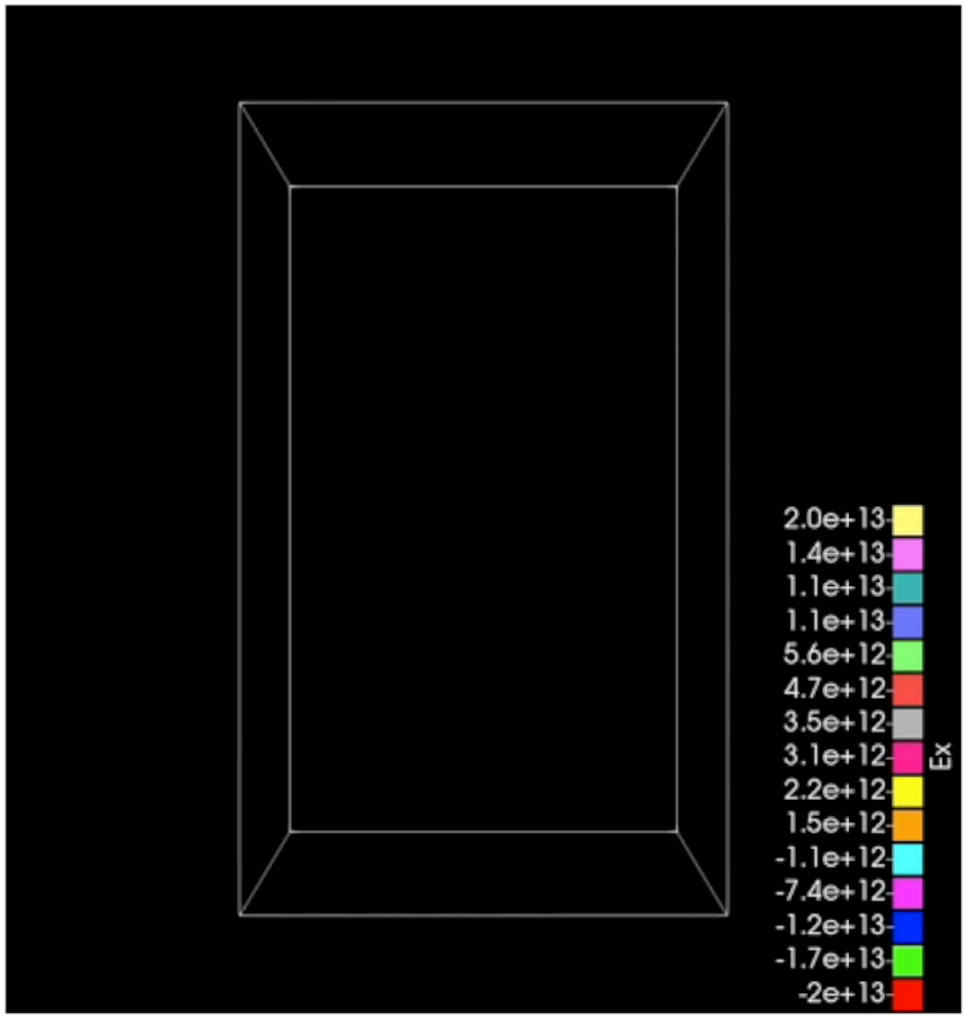


Image Size

time 0.00000

phi -180

theta -90

Conclusions

- **Challenge:** Processor to Memory/Storage Gaps
  - Latency Gap
  - Bandwidth/Capability Gap
- **Advantage**: Lots of compute available
- **Services requested:**
  - Fault tolerance
  - Data storage
  - Visual analysis

Overcome gaps via:	Specific approaches	Product highlight
Abstract/virtual interface to hide gap via messaging, caching and asynchrony	Messaging	ADIOS
	Caching	PnetCDF, HDF5, UnifyFS
	Asynchrony	VeloC
Prioritize and measure functionality	Regularize, measure, compare	Darshan, HDF5, ADIOS, PnetCDF, MPI-IO
Data reduction and transformation	Compression	SZ, ZFP
	In situ analysis - Run on processors	VTK-m
	In situ analysis – Batch algorithms on processors	ALPINE
	Post-processing analysis – Exploratory	Cinema

END