Co-design for Next-Generation Extreme Scale Data Processing: A Movement Towards ADIOS-X

International Workshop on CO-DESIGN
Peking University
Beijing, China
October 24, 2012

Dr. Scott A. Klasky, (klasky@ornl.gov)
Scientific Data Group
Oak Ridge National Laboratory +


Thanks to our funding and collaborators

Thanks to

The Data Problem (D. Hitchcock ASCR)

All of the exascale hardware trends impact data-intensive science

→ Square Kilometer Array in Australia needs 100 MW for compute infrastructure
  • Leverages investments in exascale to maximize impact on the Science missions

### Genomics
- Sequencer data volume increasing 12x over the next 3 years
- Sequencer cost decreasing by 10x over same time period

### High Energy Physics
- LHC experiments produce & distribute petabytes of data/year
- Peak data rates increase 3-5x over 5 years

### Light Sources
- Many detectors on a Moore’s Law curve
- Data volumes rendering previous operational models obsolete

### Climate
- By 2020, climate data expected to be hundreds of exabytes or more
- Significant challenges in data management, analysis, and networks

---

The Evolution of Scientific Data

2000

HDF
NetCDF

POSIX Era

I/O falls behind compute

2005

Inception of ADIOS

VisIt: Parallelism on top of VTK

VTK: Data flow networks

AVS

R

2010

VisIt-ADIOS Integration

Visualization of trillion zone dataset

I/O Pipelines

First official ADIOS release

SKEL

Asynchronous SKEL

2015

VisIt-ADIOS Staging Integration

Staged data reduction aggregation

2020

Service Oriented Architecture

Hybrid Staging

Analytics-driven Visualization

Transition to Exascale

Diversity in Scalable Analytics

Mathematics-driven Analytics Prototyping

Creation of a computational Laboratory

Creation of a Validation Laboratory

Non-interactive Visualization

Present status

I/O Pipelines

Visualization

I/O kernels

HDF
NetCDF

I/O Pipelines

Diversity in Serial Analytics

I/O Pipelines

Diversity in Scalable Analytics

service in Scalable Analytics

Diversity in Scalable Analytics

Diversity in Serial Analytics

Variability Attribution in High-Dimensional Experiments

Programming with Big Data: for Analytics Developers

Analytics-driven Visualization

Service Oriented Architecture

Hybrid Staging

Mathematics-driven Analytics Prototyping

Transition to Exascale

Diversity in Scalable Analytics

Present status
From Innovation to Impact: Research and Development in Data Intensive High Performance Computing

Over 120 Publications from 2006 - 2012

<table>
<thead>
<tr>
<th>Application</th>
<th>Code</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Astrophysics</td>
<td>Chimera</td>
<td>T. Mezzacappa</td>
</tr>
<tr>
<td>Combustion</td>
<td>S3D</td>
<td>J. Chen</td>
</tr>
<tr>
<td>AMR</td>
<td>Chombo</td>
<td>B. Van Straalen</td>
</tr>
<tr>
<td>CFD</td>
<td>Fine/Turbo</td>
<td>M. Gontier</td>
</tr>
<tr>
<td>Fusion-Edge</td>
<td>XGC1</td>
<td>C. S. Chang</td>
</tr>
<tr>
<td>Fusion-Edge</td>
<td>XGC0</td>
<td>S. H. Ku</td>
</tr>
<tr>
<td>Geoscience</td>
<td>AWP-ODC</td>
<td>Y. Cui</td>
</tr>
<tr>
<td>Materials</td>
<td>LAMMPS</td>
<td>A. Frachioni</td>
</tr>
<tr>
<td>Weather</td>
<td>GRAPES</td>
<td>W. Xue</td>
</tr>
<tr>
<td>Nuclear</td>
<td>HFODD</td>
<td>H. Nam</td>
</tr>
<tr>
<td>Relativity</td>
<td>Maya</td>
<td>P. Laguna</td>
</tr>
<tr>
<td>Sub surface</td>
<td></td>
<td>M. Wheeler</td>
</tr>
<tr>
<td>Materials</td>
<td></td>
<td>M. Parashar</td>
</tr>
<tr>
<td>Fusion</td>
<td>GTC-P</td>
<td>S. Ethier</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>Code</th>
<th>Contact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusion</td>
<td>GTS</td>
<td>W. Wang</td>
</tr>
<tr>
<td>Fusion</td>
<td>M3D-C1</td>
<td>S. Jardin</td>
</tr>
<tr>
<td>Fusion</td>
<td>M3D-K</td>
<td>G. Y. Fu</td>
</tr>
<tr>
<td>Fusion</td>
<td>GTC</td>
<td>Z. Lin</td>
</tr>
<tr>
<td>Fusion</td>
<td>GEM</td>
<td>S. Parker</td>
</tr>
<tr>
<td>Quantum</td>
<td>QLG2Q</td>
<td>M. Soe</td>
</tr>
<tr>
<td>Image Analysis</td>
<td></td>
<td>T. Kurc</td>
</tr>
<tr>
<td>AMR</td>
<td>Boxlib</td>
<td>J. Bell</td>
</tr>
<tr>
<td>Relativity</td>
<td>Cactus</td>
<td>G. Allen</td>
</tr>
<tr>
<td>Weather</td>
<td>NASA</td>
<td>T. Clune</td>
</tr>
<tr>
<td>Materials</td>
<td>QMC</td>
<td>J. Kim</td>
</tr>
<tr>
<td>Climate</td>
<td></td>
<td>J. Fu</td>
</tr>
<tr>
<td>Climate</td>
<td>CAM</td>
<td>K. Evans</td>
</tr>
</tbody>
</table>

The beginning of our research in I/O pipelines (2000)

- Focus of techniques on maintaining a maximum of 5% overhead while writing and visualizing 1 TB (GTC)/24 hours
- Exploit asynchrony wherever possible
- Enhance I/O with in situ processing for visualization
- Explore techniques with an eye on the future
- Provide solutions for real science and real applications

I/O Benchmarking with Kernels

- I/O Kernels have been used to simulate application I/O
  - Handwritten, thus time consuming to create
  - Once created, I/O Kernels tend to be static, even though applications continue to change
  - I/O Kernels vary widely
    - May require libraries or input data from application
    - May use various compilation methods (autotools, cmake, ...), often same as application
    - Output of measurements varies widely. testing process may require scripting, cutting and pasting, or even hand entry of output data
- I/O Research slow, tedious and error prone

![Diagram showing the relationship between Application and I/O Kernel]

Adaptable I/O System

- Provides portable, fast, scalable, easy-to-use, metadata rich output with a simple API
- Change I/O method by changing XML input file
- Layered software architecture:
  - Abstracts the API from the method used for I/O
- http://www.nccs.gov/user-support/center-projects/adios/

[Diagram showing components and flow of data]

Original  ADIOS

I/O performance of the Combustion S3D code (96K cores), and the S Cec PMCL3D (30K cores)

Sign.org/adios
Industrial application: Speed I/O by 100X

- **RAMGEN/Numeca**
  - CFD solver by Numeca International used by RAMGEN Power Systems at OLCF
  - Two Body test case - 500 million grid cells - 3840 processes
  - Largest run to date: 3.7 billion grid cells

Rendering of shock structures by M. Matheson, OLCF
Automatic Benchmark Generation

• **Skel** addresses various issues with I/O kernels
  • Automatic generation reduces development burden and enhances applicability
  • Skeletals are easily kept up to date with application changes
  • Skel provides a consistent user experience across all applications
  • No scientific libraries are needed
  • No input data is needed
  • Measurements output from skeletals are standard for all applications

• The correctness of Skel has been validated for several applications and three synchronous I/O methods
• July, 2012: Skel was released inside ADIOS 1.4

Skel Validation and measurements


Introduction to Staging

- Initial development as a research effort to minimize I/O overhead
- Draws from past work on threaded I/O
- Exploits network hardware support for fast data transfer to remote memory


Staging Revolution: Phase 0

- Using Staging as an I/O burst buffer
Data Service Approach

Runtime Overhead comparison for all evaluated scheduling mechanism 16 Stagers

- 64
- 512
- 1024
- 2048

Naïve scheduling (Continuous Drain), can be slower than synchronous I/O

- Output costs can be reduced
- Total data size can be managed
- Input cost to workflow can be reduced
- Meta-operations can aid eventual analysis
- Application is decoupled from storage bottlenecks

Indexing and Compression

- Extreme scale data enhancement and reduction
- Utilize in transit and in situ mechanisms
- Scientific compression schemes (ISABELA and ISOBAR)
- In situ indexing to enable fast query and data access
- Deployed as services in the pipeline

Visualization research to move to real-time analysis

Statistical Based Analysis

- Climate test cases: Extreme Value Analysis, and Peaks Over Threshold
  - Scaling studies on 100 year daily precipitation CCSM3.0 data
  - 36,500 timesteps, 8K spatial observations
  - Analysis previously took several days on researchers desktop machine
  - Now takes 4 seconds on moderately sized analysis cluster

http://www.olcf.ornl.gov/center-projects/adios/
Visit

• Next generation architectures will pose tremendous challenges
  • Supercomputer with appropriately sized, dedicated analysis cluster not practical
  • Extreme, hybrid parallelism will stress existing visualization architectures
  • Traditional file-centric, post processing will not be possible
  • Results from a “exascale sized” (multi-trillion zone) visualization experiment show that I/O is the dominant factor: up to 95% at extreme scale
Research techniques to understand and optimize reading performance

- Use Hilbert curve to place chunks on lustre file system with an Elastic Data Organization.

Y. Tian, First place
ACM student
Research Competition 2011
ADIOS-P: Mining Provenance data of I/O

- Provenance module for ADIOS
- Store and index user data access activities
- Transparently integrated with scientific and visualization applications (e.g., VisIt)
- Mining provenance data for IO prefetching, auto-tuning applications, etc.

Predictive Prefetching

**Impacts**
- Reduce overall time
- Increase throughput

- Maximum speedup with an uniform workload ratio
- High prefetching accuracy contributes more to large speedup than low prefetching accuracy
- Even small increase of prefetching accuracy is important in a balanced workload

Combustion Workflow

RHS of S3D solver at each Stage of an explicit time step

Asynchronous movement of data or share data in memory (different levels)

In situ, in transit data analysis/viz workflow via hybrid staging
Data Management in I/O Pipelines

- Perform computation in the *right* location
- Support dynamic placement
- Use data reduction techniques
- Aggregation and chunking to improve data access
- End-to-End approach to data management

---

[Diagram showing data management in I/O pipelines with nodes labeled Simulation core, Helper core, Staging core, Offline core, and PFS, with arrows indicating data flow and options for chunking and aggregation.]
Staging Revolution: Phase 1

- Monolithic staging applications
- Multiple pipelines are separate staging processes in discrete staging areas
- No scheduling of workflow
- Staging processes are individual applications using MPI
- Staging applications can also read data from files

Predata: set the stage for “next-generation” I/O pipelines.


- Use the staging nodes and create a workflow in the staging nodes.
- Allows us to explore many research aspects.
- Improve total simulation time by 2.7%
- Allow the ability to generate online insights into the 260GB data being output from 16,384 compute cores in 40 seconds.
Phase 2: Introducing Plugins

- Disruptive step
- Staging codes broken down into multiple plugins
- Staging area is launched as a framework that can launch these plugins
- Each plugin uses ADIOS api to read and write data
- Finer granularity allows better scalability for parallel plugins

- Data movement can occur between all plugins
- Scheduler decides resource allocations and which plugins are executed

Managing the Staging Area

- Key to the design is the multi-level manager
  - Global manager looks at overall metadata/workflow as well as resource allocation
  - Container manager handles the specific runtime of the replicas and measures performance
- A design feature is that it will allow replicated scalability for any stateless analytics functions
  - Automated management of routing data between independent replicas
  - Container manager can create or destroy replicas to maintain throughput

Loose Code Coupling through Staging

- Multi-physics simulations require different models to run concurrently
- Loose coupling
  - applications advance independently
  - update from other code may have some latency
- Intercommunication through direct memory-to-memory transfers
- Transparently switch from files to memory during development

File-based or mem-to-mem

- XGC Instance
  - Client Component
  - Server Component
  - Client Component

- m3d.in
  - e-qdsk
  - g-eqsk

- XGC9 Application
  - Staging nodes
  - M3D-OMP Application

I/O at exascale is expensive

S3D simulation

O(1M) cores

Synchronous I/O

1 PB/dump every 30 minutes

O(400 PB)/run

Synchronous I/O

• Storage space requirements
  • 35 disks for each dump (No RAID)
  • 1.5 KW/live dump
• Performance requirements
  • 5% overhead, ~31k disks, >1.4 MW
  • 10% overhead, ~15k disks, >0.65 MW
  • 50% overhead, ~3k disks, >0.14 MW
• Asynchronous I/O performance requirements
  (absorb output in 30 minutes)
  • ~1500 disks, 60 KW

• Analysis
  MS-Complex
• Visualization
  Volume, Surface, Particle rendering
• Downstream
  Isomap
The SOA philosophy for HPC

• The overarching design philosophy of our framework is based on the Service-Oriented Architecture
  • Used to deal with system/application complexity, rapidly changing requirements, evolving target platforms, and diverse teams
• Applications constructed by assembling services based on a universal view of their functionality using a well-defined API
• Service implementations can be changed easily
• Integrated simulation can be assembled using these services
• Manage complexity while maintaining performance/scalability
  • Complexity from the problem (complex physics)
  • Complexity from the codes and how they are
• Complexity of underlying disruptive infrastructure
• Complexity from coordination across codes and research teams
Heterogeneous Hierarchies

- Multi-level memory hierarchies with heterogeneous compute resources
- Co-design algorithms for analysis with regard to the memory hierarchy
- Exploit asynchronous nature of algorithms
1. Where do we move the data to?
2. How do we extract data from the solver?
3. What hardware features can be exploited?
4. What processing resources are allocated?
5. How do we schedule the execution of these tasks?

- How NVRAM can be used as a staging area?
- How much of each level of the memory hierarchy to use for the staging area?
- Where to move data (RAM, NVRAM, SSD, disk, network)
- When (and how frequently) to move the data over the hierarchy

Hybrid Staging

- Research new runtime to explore co-design tradeoffs
  - How to optimize for deep memory hierarchies for “hybrid-staged” analytics and visualization
  - Placement of analysis and visualization tasks in a complex system, to understand power vs. time to solution tradeoffs.
  - Understand performance/energy tradeoffs of using fast vs. slow DRAM vs. SSD vs. NVRAM.
  - Understand impact of network data movement compared to memory movement

Tradeoffs for Hybrid Staging

- Going to disk is slow even for small application sizes
- Inline approach adds more overhead to application runtime
- In transit approach gives better overall performance
  - Additional cost of data movement
- Still must explore this for future architectures with SST-macro

- Offline: Process data after writing to disk
- In line: Process data in place synchronously with the application
- Staging: Move data to staging resources for processing

Primary resources execute the main simulation and in situ computations. Secondary resources contain a staging area and a task scheduler in Dataspaces + ADIOS to manage the scheduling and execution of in transit computations.

Timing breakdown for in situ, in transit, and data movement for the simulation and various analytics algorithms using 4896 cores on Jaguar.

Simulation grid size was 1600x1372x430, measurements per simulation time step.
The Meta-Skeleton

Science space: real applications and workflow

Co-Design space: layered abstraction

Co-Design space: cross-layer performance optimization

Meta Skeleton Workflow Framework

Parameters:
- Insitu/Intransit placement,
- Communication-aware mapping,
- Scheduling of workflow components,
- Dynamic resource allocation, Managed data movement

Proxy/Skeleton Apps

Parameters:
- Parallel scalability and speedup,
- Network-based data movement,
- Communication patterns, IO patterns

Hardware Platform/Simulator

Parameters:
- Node-level concurrency (CPU, GPU),
- Node-level storage (DRAM, NVRAM, SSD), Communication latency and bandwidth, Interconnects topology,
- Storage system performance etc

Optimization Approach/Strategies

Workflow-level:
- Execution & data placement (in-situ or in-transit), scheduling, resource allocation, mapping, IO pipeline

App-level:
- Algorithms, inter-process communication, utilization of accelerators to gain speedup

Hardware-level:
- Emerging Hardware Architecture and performance improvement

Performance measurements and profiling on hardware systems to extract parameters and build performance profiles for Proxy/Skeleton kernels

Hybrid Staging Hardware Platform

Skel as a Co-Design Tool

- Accurately measure impact of asynchronous I/O
  - emulate the application’s communication and computation during I/O data movement
  - This is generally not provided by I/O Kernels, and is not easily added
- Drive co-design efforts
  - provide an early approximation of an application before the application is developed
  - Iteration of co-design process can begin earlier
- Read performance
  - provide a range of common read patterns declaratively
International Collaboration frameworks for Extreme scale Experiments (ICEE)

Add semantics to allow for better queries for a collaborative environment

- In-Transit Processing
  - In-flight/In-situ
  - Streaming data processing

- Data Indexing
  - Indexing/Query
  - FastBit

- Intelligent Workflow
  - Provenance
  - Data mining
  - Machine-guided operation
Validation Workflows

First Principles Fusion Code
XGC, "core-edge" turbulence with UQ

Use first principle calculations on HPC to generate more accurate models

Ptransp model code

Knowledge Database

Generate a DB of knowledge by codes, to use for predictive capability during/after each experiment

Visual Analytics

Analysis
Comparative Vis Mining
Experiment mock up

Automate the process of these workflows and move work/data to satisfy metrics specified by users, and track provenance


Validation workflows to
• adapt task execution to changing data sizes and available processing power at sites
• adapt to changing target metrics
  – time to solution, energy, FLOPS
• allow for annotation and modification of the analytical pipelines
• multiple users can be involved in the process at different locations
• provenance records the data creation lineage through the various validation steps
• Co-design of validation workflows to understand how to break up complex analytics to "in-situ" and "in-transit", i.e. placement of analytics, to move work left→right data
# ASCR role in Big data

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Applied Mathematics</td>
<td>45,604</td>
<td>45,604</td>
<td>49,500</td>
<td>+ 3,896</td>
</tr>
<tr>
<td>Computer Science</td>
<td>47,301</td>
<td>47,400</td>
<td>54,580</td>
<td>+ 7,180</td>
</tr>
<tr>
<td>Computational Partnerships (includes SciDAC)</td>
<td>52,813</td>
<td>44,250</td>
<td>56,776</td>
<td>+ 12,526</td>
</tr>
<tr>
<td>Next Generation Networking for Science</td>
<td>12,313</td>
<td>12,751</td>
<td>16,194</td>
<td>+ 3,443</td>
</tr>
<tr>
<td>SBIR/STTR</td>
<td>0</td>
<td>4,560</td>
<td>5,570</td>
<td>+ 1,010</td>
</tr>
<tr>
<td><strong>Total, Mathematical, Computational, and Computer Sciences Research</strong></td>
<td><strong>158,031</strong></td>
<td><strong>154,565</strong></td>
<td><strong>182,620</strong></td>
<td><strong>+28,055</strong></td>
</tr>
<tr>
<td>High Performance Production Computing (NERSC)</td>
<td>59,514</td>
<td>57,800</td>
<td>65,605</td>
<td>+ 7,805</td>
</tr>
<tr>
<td>Leadership Computing Facilities</td>
<td>158,020</td>
<td>156,000</td>
<td>145,000</td>
<td>-11,000</td>
</tr>
<tr>
<td>Research and Evaluation Prototypes</td>
<td>4,301</td>
<td>30,000</td>
<td>22,500</td>
<td>- 7,500</td>
</tr>
<tr>
<td>High Performance Network Facilities and Testbeds (ESnet)</td>
<td>30,451</td>
<td>34,500</td>
<td>32,000</td>
<td>- 2,500</td>
</tr>
<tr>
<td>SBIR/STTR</td>
<td>0</td>
<td>8,003</td>
<td>7,868</td>
<td>- 135</td>
</tr>
<tr>
<td><strong>Total, High Performance Computing and Network Facilities</strong></td>
<td><strong>252,286</strong></td>
<td><strong>286,303</strong></td>
<td><strong>272,973</strong></td>
<td><strong>-13,330</strong></td>
</tr>
<tr>
<td><strong>Total, Advanced Scientific Computing Research</strong></td>
<td><strong>410,317</strong></td>
<td><strong>440,868</strong></td>
<td><strong>455,593</strong></td>
<td><strong>+14,725</strong></td>
</tr>
</tbody>
</table>
Big Scientific Data

What part of Big Data apple should/must DOE bite off?

What is intersection of research needed for exascale compute and exabyte data?

Scott’s view of big data

- First: Data and computing power doubling every 18 months
- Complexity of most algorithms for data are $O(n \log n)$ or $O(n^2)$
- We need to understand how we can statistically reduce the amount of data goes to storage
- We need to build accurate physics-based models, from data analysis from our first principle simulations
- We need ways to mine the data from ensembles of models faster, taking into account the uncertainty of our data, and errors of our data/models