Exascale IO and in situ Data Processing on Leadership Computers
Scott Klasky (ORNL) (klasky@ornl.gov), Matthew Wolf (ORNL) OLCF meeting (May 22, 2019)

Scientific Data Management
Norbert Podhorszki –TL
Mark Ainsworth
Jong Choi
William Godoy
Tahsin Kurc
Qing Liu
Jeremy Logan
Kshitij Mehta
Eric Suchyta
Ruonan Wang
Lipeng Wan

Scientific Data Analytics
Dave Pugmire
Mark Kim
James Kress
George Ostroumov
Jieyang Chen
Nick Thompson

https://github.com/CODARcode/MGARD
https://github.com/ornladios/ADIOS2
https://gitlab.kitware.com/vtk/vtk-m
https://pbdr.org/packages.html
https://Adios.ornl.gov
Vision: Enabling High Performance pub/sub I/O

• Create a high performance I/O abstraction to allow for on-line memory/file data subscription service

• Create an abstraction for self-describing I/O for files, streams, etc.

• Work at all scales (laptops, desktops, clusters, exascale-platforms)

• Scale out to the number of nodes, sites, timesteps, variables, etc.

Co-design the next set of tools for federated computing

- The convergence of large DOE instruments with HPC centers dictates that we need to allow coupling/streaming
  - Codesign of what occurs at the edge and at HPC centers is imperative
  - Integration of ML/AI with HPC is essential to process as much data

Software we need to enable our application use-cases

- **Workflow system**
  - To coupled” applications, submit the jobs on any system, and to monitor the job

- **A Data management system**
  - To get/put data to/from one or more consumers to on or more producers
  - To track the provenance and performance of the workflow
  - To write/read data to the storage layer along with pub/sub to multiple consumers

- **Data compression/reduction**
  - Which can take user Quantiles of Interest and preserve those within an error bound

- **Analysis and Visualization services which can be incorporated into the workflow**
  - Fast, Scalable, and memory efficient

- **A Dashboard to allow scientist to collaborate during a running experiment/simulation**
  - Web based, user configurable
Software we need to enable our application use-cases

• Workflow system
  – EFFIS, or any other (e.g. EnTK, Kepler, Pegasus, Swift, ..)

• A Data management system
  – ADIOS, ...

• Data compression/reduction
  – MGARD, SZ, ZFP, blosc, ...

• Analysis and Visualization services which can be incorporated into the workflow
  – VTK-m, pbdR, Visit, Paraview, anaconda ...

• A Dashboard to allow scientist to collaborate during a running experiment/simulation
  – Kdash, ..
The new demands from applications
Simulations – Storage: Seismic Tomography Workflow (PBs of data/run)

**Scientific Achievement**

- Most detailed 3-D model of Earth’s interior showing the entire globe from the surface to the core–mantle boundary, a depth of 1,800 miles

**Significance and Impact**

- First global seismic model where no approximations were used to simulate how seismic waves travel through the Earth.
- Over 1 PB of data was generated in a 6 hour simulation (on Titan@OLCF)

**Research Details**

- To improve data movement and flexibility, the Adaptable Seismic Data Format (ASDF) was developed that leverages the Adaptable I/O System (ADIOS) parallel library
- ASDF allows for recording, reproducing, and analyzing data on large-scale supercomputers
- **1.5 PB** of data is produced in a single workflow step, which is fully processed later in another step

Simulations - Code coupling: ECP WDM - High-Fidelity Whole Device Modeling of Magnetically Confined Fusion Plasmas

- Different physics solved in different physical regions of detector (spatial coupling)

- Core simulation: GENE
  Edge simulation: XGC
  Separate teams, separate codes

- Recently demonstrated first-ever successful kinetic coupling of this kind

- Data Generated by one coupled simulation is predicted to be > 10 PB/day on Summit

J. Dominski; S. Ku; C.-S. Chang; J. Choi; E. Suchyta; S. Parker; S. Klasky; A. Bhattacharjee; Physics of Plasmas 2018, 25, DOI: 10.1063/1.5044707
Observational data: SKA

- The full SKA will use a million antennas to enable astronomers to monitor the sky in unprecedented detail and survey the entire sky much faster than any system currently in existence.
- One of SKA’s greatest challenges is in the ability to move, process, and store data, without losing information.
- The data output of the SKA is limited by the achievable I/O bandwidth.

<table>
<thead>
<tr>
<th>Telescope</th>
<th>Raw Data Rate</th>
<th>Archive Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWA</td>
<td>1.4 TB/hour</td>
<td>5 PB/year</td>
</tr>
<tr>
<td>LSST</td>
<td>1.5 TB/hour</td>
<td>6 PB/year</td>
</tr>
<tr>
<td>ASKAP</td>
<td>9 TB/hour</td>
<td>5.5 PB/year</td>
</tr>
<tr>
<td>SKA1-LOW</td>
<td>1,400 TB/hour</td>
<td>150 PB/year</td>
</tr>
</tbody>
</table>

EFFIS 2.0: an End-to-end Framework For coupling Integrated Simulations

• Why EFFIS?
  – To simplify the complexity of composing, running, and monitoring applications on HPC systems

• What is EFFIS
  – A collection of services to compose, launch, monitor, and control coupled applications

• How does EFFIS work
  – Contains new services to “easily” compose coupled HPC applications on HPC Resources using a python-like interface
    • Cori, Theta, Titan, Summit
  – Allows “easy” integration to visualization tools (Visit, Python notebooks, etc.)
  – ADIOS for data movement

I/O Framework for Data Intensive Science

- **Problem**
  - I/O is severely bottlenecked on HPC systems and experiments because of hardware limitations.
  - New I/O patterns from code coupling, AI/ML, simulations require new solutions.

- **Solution**
  - ADIOS is a DOE framework developed for sustainable I/O on Leadership Class Facilities (LCF).
  - ADIOS is a publication/subscribe I/O framework for storage and in situ processing of data.

- **Impact**
  - Over 10X I/O improvement from previous I/O methods on dozens of LCF apps.
  - Used by more than 30 LCF application areas, totaling over 1B hours on the LCFs.
  - Outside DOE: Used in Industrial Engineering, Oil Exploration, Computational Fluid Dynamics.
  - ADIOS won R&D 100 Award in 2013.
ADIOS Approach: “How”

• I/O calls are of declarative nature in ADIOS
  – which process writes what: add a local array into a global space (virtually)
  – adios_close() indicates that the user is done declaring all pieces that go into the particular dataset in that timestep

• I/O strategy is separated from the user code
  – aggregation, number of sub-files, target file-system hacks, and final file format not expressed at the code level

• This allows users to choose the best method available on a system without modifying the source code

• This allows developers
  – to create a new method that’s immediately available to applications
  – to push data to other applications, remote systems or cloud storage instead of a local filesystem

Changing from C/R data to visualization/analysis data is “easy”

- One change in the code or input file, to specify the engine

```c++
adios2::Engine writer = io.Open("analysis.bp", adios2::Mode::Write);
writer.BeginStep();
writer.Put(varT, T.data());
writer.EndStep();
writer.Close();
```

### XGC strong scaling analysis data, 6 GB on Theta using NVRAM

<table>
<thead>
<tr>
<th>PEs</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4096</td>
<td>0.05</td>
</tr>
<tr>
<td>8192</td>
<td>0.04</td>
</tr>
<tr>
<td>16384</td>
<td>0.03</td>
</tr>
<tr>
<td>32768</td>
<td>0.03</td>
</tr>
<tr>
<td>65536</td>
<td>0.03</td>
</tr>
</tbody>
</table>

You can change the engine to HDF5

### ADIOS BP file HDF5 file

Lustre GPFS
Metadata challenges in big scientific data management

- Creating “containers” to store all of the data products, code, workflow, performance, “raw data” allows scientist to better understand their campaign

- As scientist continue to save data products from their simulations/experiments, the amount of variables grow

- The cost of managing the metadata from all the data produced in the scientific process can grow to be “painful” for scientist

- There is a large overhead of generating and managing the metadata grows with number of data objects (variables, attributes), number of processes, number of simulation steps

Numerical data is challenging to compress losslessly (P. Lindstrom)

**MGARD: MultiGrid Adaptive Reduction of Data**

Decomposes data into contributions from a hierarchy of meshes,

\[ u = Q_0 u + (Q_1 - Q_0)u + \ldots + (Q_L - Q_{L-1})u \]

Adaptive reduction of data based on discarding least important contributions

\[ u = \sum_{\ell=0}^{L} \sum_{n \in N_{\ell}} \alpha_n^\ell \varphi_n^\ell \]

\[ \tilde{u} = \sum_{\ell=0}^{L} \sum_{n \in N_{\ell}} \tilde{\alpha}_n^\ell \varphi_n^\ell \]

Mathematically proven error bounds

\[ C_d \eta_s(v) \leq \|v\|_s \leq \eta_s(v) \]

where

\[ \eta_s(v)^2 = \sum_{\ell=0}^{L} 2^{2s\ell} \|\Delta_{\ell} v\|^2 \]

Applicable to structured (tensor product) grids with arbitrary spacing, integrated into ADIOS

Able to preserve quantities of interest (spectrum, averages...)


MGARD: MultiGrid Adaptive Reduction of Data

- MGARD can preserve quantities of interest specified by users $|Q(u) - Q(\tilde{u})| = |Q(u - \tilde{u})| \leq r_{L_\infty}(Q)\|u - \tilde{u}\|_{L_\infty}$

- This can be done by specifying a smoothness parameter (e.g. $s=-1/2$ for averages/blobs, -1 for streamlines),...

  $|Q(u) - Q(\tilde{u})| \leq r_s\|u - \tilde{u}\|_s$

- E.g. a user can supply a routine to compute a functional Maximum resolved wave number in energy spectra can be supplied as $|\tilde{k}| \leq (r/\varepsilon)^{-1/s}$

Effects of MGARD compression on turbulent energy spectra.

Compression of S3D combustion data
Puncture plot

• Can we reduce data from a fusion code of the magnetic field vector (65x257x161) and still compute accurate puncture plots?

• Trajectories depend on

\[ Q_k(v) = \int_{t_k}^{t_{k+1}} v(\tilde{x}(t)) \, dt \]

• Continuous linear functional

\[ |Q(u - \tilde{u})| \leq r_s(Q)\|u - \tilde{u}\|_s \]

• For \( s = (d-1)/2 \) in \( d \)-dimensions

• Here \( d = 3 \), so take \( s = 1 \)

• Hence, we apply reduction where control loss \( \|u - \tilde{u}\|_s \)
Why In situ

• Timeliness (Resources)
• Reduction... save information, not all the data

Rich Design Space for Staging

- Location of the compute resources
  - Same cores as the simulation (in situ)
  - Some (dedicated) cores on the same nodes
  - Some dedicated nodes on the same machine
  - Dedicated nodes on an external resource

- Data access, placement, and persistence
  - Direct access to simulation data structures
  - Shared memory access via hand-off / copy
  - Shared memory access via non-volatile near node storage (NVRAM, BB)
  - Data transfer to dedicated nodes or external resources (decoupled in space)

- Synchronization and scheduling
  - Execute synchronously with simulation every $n^{th}$ simulation time step
  - Execute asynchronously (decoupled in time)
  - Dynamic execution

Data Staging in ADIOS

- **Sustainable Staging Transport (SST)**
  - In situ infrastructure for staging in a streaming-like fashion using RDMA, SOCKETS with “active” connect/disconnect

- **InSituMPI**
  - One way staging for MPMD applications, for strong coupling

- **DataMan**
  - WAN transfers using sockets and ZeroMQ for EO data

- **SST-SM**
  - Staging infrastructure providing a shared memory abstraction for coupling on shared nodes

- **InSitu-sync**
  - Synchronous in situ, direct pass through of data structures to analytics

---

In-line vs. In-transit Visualization Techniques at Scale

- Scientific workflows are complex
  - Numerous analysis and visualization tasks need to be coordinated
  - Inefficiency = less scientific insight, longer simulation times, larger cost

- Choosing the in situ strategy can be complex
  - Overheads of in situ visualizations are not well understood

- Study some examples from a Cloverleaf3D MiniApp
  - What configuration will impact the simulation time the least?
  - What configuration will be most cost effective, at scale?

- Study two visualization algorithms: Isocontour, and rendering

Placement for Visualization Services

- Placement of services has a dramatic impact on performance
- Algorithms with communication (e.g. parallel rendering) exhibit poor performance at scale
- Move data to staging resource improves scalability of visualization services
- Visualizing every “Nth” step provides improves scalability of smaller resources

XGC Analysis – moving from synchronous to staged in situ

- InSituMPI utilize zero-copy method
- Share data to minimize data copies
- Calculations (1-4) are embarrassingly parallel routines

Maximize Data Locality

Node

Core

Adios Push

IMPI SST
New Developments in Scalable Analytics with R

**pbdR is R plus...**

- More numerical methods
- Large parallel computing
- GPU infrastructure
- I/O tools, advanced profilers, remote computing

**New CPU Capabilities**

- New ML methods optimized for Summit GPUs
- Addresses the CORAL 2 benchmarks
- Fast random generators from 6 common distributions
- Bindings for NVIDIA Management Library (NVML)
- Infrastructure for low-level "roll your own" CUDA primitives
Visualization Services with VTKm

- Analysis and visualization (1D, 2D, 3D) are needed to feed a scientific dashboard
- Services are lightweight, configurable components within a scientific workflow
- VTKm is a visualization toolkit aimed at the heterogenous architectures of supercomputers
  - Architecture supports “write once run anywhere” required by visualization services
- Support for large number of key algorithms:
  - Isocontour, slice, histogram, streamline, rendering, ...

Portable Performance of Particle Advection: The VTKm implementation was shown to be comparable in performance to custom CPU and GPU implementations

Performance-Portable Particle Advection with VTK-m, D Pugmire, A Yenpure, M Kim, J Kress, R Maynard, H Childs, B Hentschel. EGPGV, 2018
Ingest (Python)

Fetch images and metadata from simulation via HTTP

Girder (CherryPy)

HTML5 client

MongoDB

XGC streamers $T=259$

XGC B-extent $T=259$

XGC $dpot.core$ $T=259$

XGC $dpot.edge$ $T=259$

XGC $C_{dpot}$ $T=259$

XGC $C_{dpot}$ $T=202$

kdash
Summary: Co-design the next set of tools for federated computing

• The convergence of large DOE instruments with HPC centers dictates that we need to allow coupling/streaming
  – Codesign of what occurs at the edge and at HPC centers is imperative
  – Integration of ML/AI with HPC is essential to process more data

• [https://github.com/CODARcode/MGARD](https://github.com/CODARcode/MGARD)
• [https://github.com/ornladios/ADIOS2](https://github.com/ornladios/ADIOS2)
• [https://gitlab.kitware.com/vtk/vtk-m](https://gitlab.kitware.com/vtk/vtk-m)
• [https://pbdr.org/packages.html](https://pbdr.org/packages.html)
• [https://Adios.ornl.gov](https://Adios.ornl.gov)