SAND2019-0050C

Sandia National Laboratories

WBS 2.3.4.04
SNL ATDM Data and Visualization Projects

SNL ATDM: I/O and Data Management

Gregory Sjaardema (PI), Ron Oldfield (PM)
Shyamali Mukherjee, Craig Ulmer, Lee Ward

Overview

Problem: Application developers do not want to modify their source code for each new platform, for each new hardware capability (e.g., burst buffer), or for each new I/O library option. In other words, all I/O options should work portably, performantly, and seamlessly.

Similarly, production workflows are becoming increasingly more sophisticated and require a more fluid way to exchange data that potentially bypasses the filesystem. For production Exascale Computing, we need a new generation of data management services that can manage the platform's memory, nonvolatile memory, and persistent resources, while providing familiar APIs to users.

Approach: We address this problem in this project through two related efforts.

First we are updating Sandia's production meshing library, **IOSS**, to leverage **Burst Buffers**, to serve as a front end for testing new I/O research, and to isolate hardware and library differences.

Second, we are developing a new set of data management libraries named **FAODEL** that are capable of moving data objects within and between applications, as well as managing distributed memory, nonvolatile memory, and storage resources in a system.

Use Cases: Workflows, application coupling, checkpoint/restart, in-memory handoff of application data to analysis tools,

External I/O Libraries

An efficient and performant production I/O and Data Management system requires efficient and performant production-quality external libraries. We collaborate with external "third-party" library developers to improve their parallel capabilities, performance, scalability, maintainability, and software quality.

NetCDF: have helped rewrite internal data model, improve parallel capabilities, provided testing at scale, general code improvements. Developed very good working relationship with developers.

PnetCDF: a few PR and interactions with developers. Good collaboration among IOSS, Exodus, PnetCDF, and NetCDF developers. Supports burst-buffer output option.

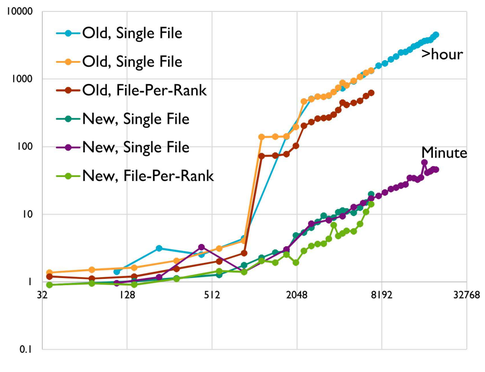
CGNS: have implemented some missing parallel API functions, provided testing at scale, worked to improve open/metadata read performance at scale (multiple orders of magnitude improvement), and other improvements. Several PRs submitted and accepted.

HDF5: Tri-Lab (LLNL, LANL, SNL) contract with HDF5 Group. Concentrating on improving HDF5 library on HPC systems at the laboratories. Includes scalability, maintainability, testing, build systems.

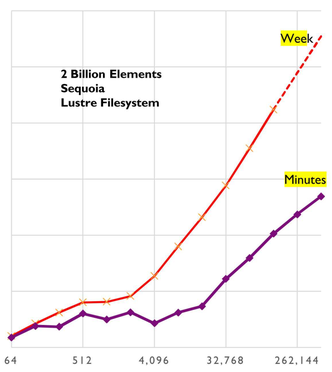
ADIOS: ECP-funded effort with KitWare to add ADIOS database option to IOSS.

Catalyst: Embedded Visualization Options.

CGNS open and metadata read timing



Exodus/NetCDF/HDF-5 Metadata



IOSS Library Supporting Sandia's Mesh Datasets

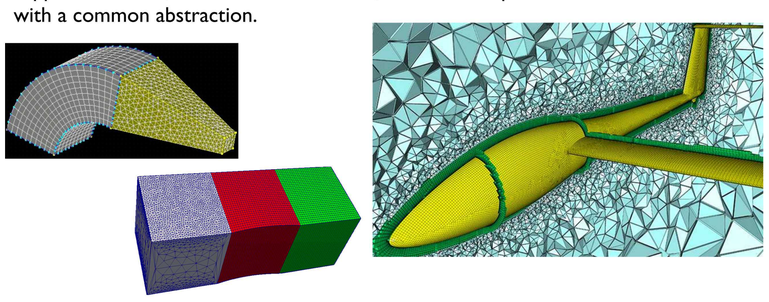
Hybrid Mesh

The choice of whether to use a structured or unstructured mesh is problem specific and involves tradeoffs and engineering judgement:

- **Generation**: unstructured grids are faster to generate than structured grids
  Unstructured: ≈ hours to days; Structured: ≈ weeks to months
- **Accuracy**: structured generally more accurate per unknown than unstructured
- **Convergence CPU time**: structured calculations usually take less time than unstructured

A hybrid mesh representation in which the mesh can use both structured and unstructured regions to represent portions of the geometry can, in theory, use the representation best suited to portions of the geometry..

Sandia is developing a hybrid mesh capability for the IOSS library. It will support structured, unstructured, and hybrid meshes in CGNS and Exodus formats and also support FAODEL, Embedded Visualization, and DataWarp/Burst Buffer interfaces with a common abstraction.



Burst Buffer

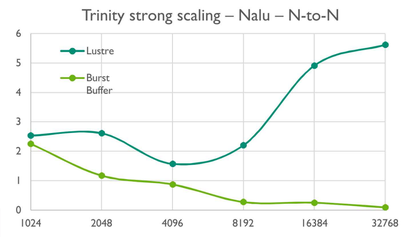
We have incorporated Cray DataWarp calls inside the IOSS library to provide Burst Buffer (BB) capability to all file-based output options (currently CGNS and Exodus). The IOSS cyclic output option is used for asynchronous migration of intermediate, per-timestep, output from the BB to the parallel filesystem (Lustre). After each timestep is complete, the file is closed which initiates an automatic stage-out of the file from the burst buffer to the file system. This can execute concurrently with output of subsequent timestep data.

The IOSS BB capability has been demonstrated on Mutrino and Trinity using **Nalu**--a generalized unstructured massively parallel low-Mach flow code (https://github.com/NaluCFD/Nalu). The test problem was strong-scaling a medium-sized 'jet' model with ~30 Million elements on 1,024 to 32,768 ranks.

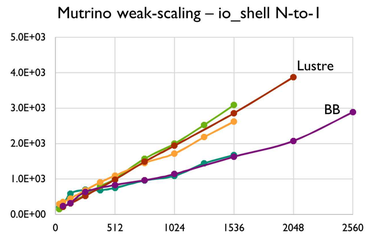
Preliminary results show improving benefits as the processor count increases. Additional testing with larger models, strong and weak scaling with N-to-N and N-to-1 output, is planned for this CY.

This capability will be released soon and available to all clients using IOSS library.

Trinity strong scaling – Nalu – N-to-N



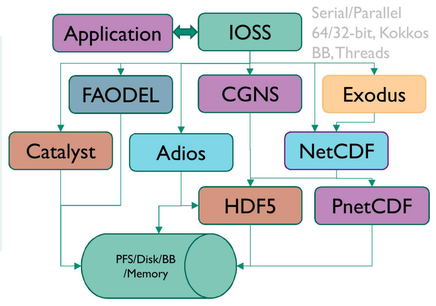
Mutrino weak-scaling – io_shell N-to-1



SEACAS: Packaging, Workflow

SEACAS is a a suite of preprocessing, postprocessing, translation, decomposition, and utility applications supporting finite element analysis software using the Exodus database file format. The applications and libraries read, write, modify, and/or query an exodus or cgns mesh database. Primarily command-line driven with a consistent user-interface. They provide analyst workflow support; regression test utilities; and database debugging capabilities.

aprepro, algebra, blot, conjoin, ejoin, epu, exo2mat, exo_format, exodiff, exomatlab, fastq,gen3d, genshell, gjoin, grepos, grope, io_shell, mapvar, mat2exo, nem_slice, nem_spread, numbers, exotxt, txtexo



Optimization:

Workflow..App..IOSS..Exodus..TPL

Original	Final	Complexity
13 Hours	3 Minutes	Lots of Timesteps
1 Week (est.)	8 Minutes	Lots of Processors
3 Hours	11 Minutes (ser) 3 Minutes (parallel)	Lots of Files
25 Minutes	1 Minute	Lots of Blocks
10.5 Minutes	9.7 Seconds	Lots of Variables
2 hours	~1 Minute	Library Usage
36 Minutes	6 Minutes	TPL overhead

Integration

Multiple options including Burst Buffer, FAODEL, Catalyst, CGNS, Exodus, NetCDF-4, NetCDF-5, Structured, Unstructured, 64-bit / 32-bit integers, Lustre, GPFS, Kokkos, … make it difficult for application code and analysts to manage, test, and implement all features. The IOSS library can package all options and present them to the application code in a configurable package. At application level, interface is consistent; at hardware level, IOSS configures parameters necessary for efficient I/O.

Download / Installation

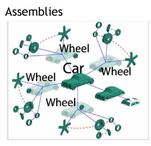
CMake-based build system with TriBITs. Automatic download and installation of required and optional Third-Party libraries. Serial and parallel build options. Includes IOSS and all SEACAS applications which provide manipulation and query capabilities for Exodus files.

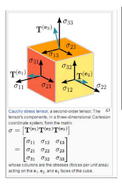
- Github -- https://github.com/gsjaardema/seacas
- Trilinos -- https://github.com/trilinos/Trilinos
- SPACK – *spack install seacas*

Enhancements

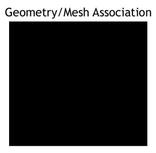
- Anticipate Needs
- Explore Options
- Evolve Capabilities
- Long-term sustainability

Assemblies





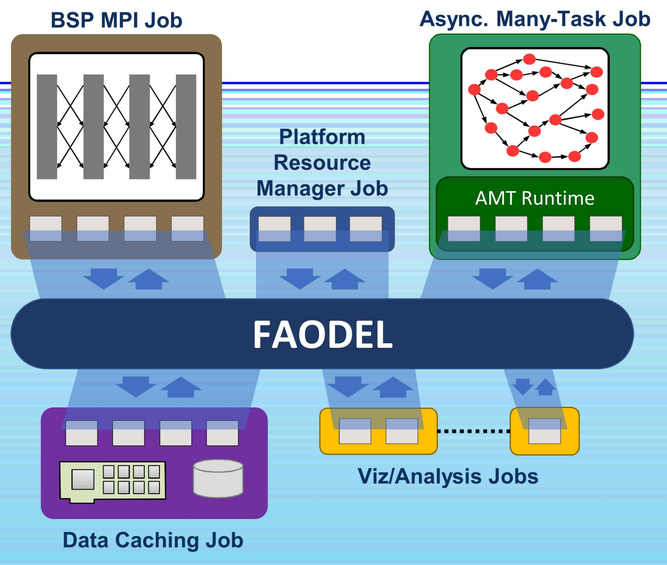
Geometry/Mesh Association



Vector/Tensor Fields
Integer Fields
Units (m/sec^2) / Dimensions (L/T^2)

- Robustness
- Capability
- Capacity
- Scalability

FAODEL: Flexible, Asynchronous, Object Data-Exchange Libraries for Data Management Services



FAODEL Support in EMPIRE

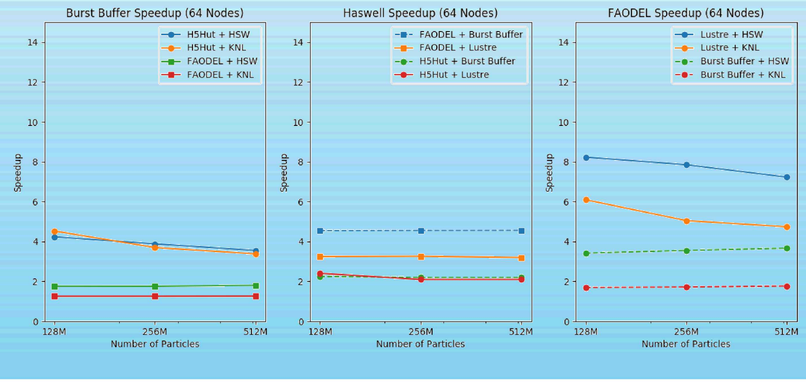
Exascale workflows require an efficient mechanism for moving massive datasets from one parallel application to another. Implementing these handoffs by passing files through the filesystem is expensive. As an alternative, Sandia is developing new data management services that allow concurrent applications to share in-memory data structures more easily. These services are based on FAODEL, an open source collection of communication libraries that enables developers to decompose their datasets into objects that can be migrated between distributed memory and nonvolatile memory resources in a platform.

ATDM's **EMPIRE** application was modified to provide a FAODEL interface for exchanging mesh and particle data. This interface currently serves as a checkpoint/restart mechanism for the application, and is being extended to serve as a conduit for allowing external analysis applications to inspect live EMPIRE simulations.

Additional FAODEL work scheduled for this year will port the mesh interface into IOSS, thereby enabling a large number of Sandia applications to take advantage of FAODEL.

EMPIRE with FAODEL I/O on Trinity

A large number of EMPIRE I/O experiments were conducted on the Cray XC40 platforms to validate the FAODEL implementation and explore the impact of different architecture components on performance. The figure below summarizes the impact of different technologies on EMPIRE's I/O performance for checkpointing data. While the DataWarp Burst Buffers delivered an expected speedup over Lustre, KNL nodes were noticeably slower than Haswell for I/O due to the serial nature of the I/O libraries. FAODEL provides speedups over traditional file I/O libraries (H5Hut and Exodus) because it streamlines I/O operations.



FAODEL Release Info:
Version: 1.1811.1 (DIO)
Date: Nov. 2018
License: MIT

https://github.com/faodel

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND No. _____

EXASCALE COMPUTING PROJECT

Sandia National Laboratories