Sandia National Laboratories

WBS 2.3.4.04
SNL ATDM Data and Visualization Projects

SNL ATDM: I/O and Data Management

Craig Ulmer (PI), Ron Oldfield (PM)
Todd Kordenbrock, Margaret Lawson, Scott Levy, Jay Lofstead, Shyamali Mukherjee, Greg Sjaardema, Gary Templet, Lee Ward, Patrick Widener

Overview

Problem: While Burst Buffers offer significant performance advantages over traditional storage, they are often underutilized by application teams because existing I/O libraries lack a means of transparently leveraging the hardware. Similarly, production workflows on these systems are becoming increasingly more sophisticated and require a more fluid way to exchange data that 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 and serve as a front end for testing new I/O research. 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

FY18 Schedule:

Description	Schedule
SPARC: Checkpoint/Restart	Q1,Q2
SPARC: Production IOSS-Based partitioning for structured/unstructured grids at scale	Q2
EMPIRE Checkpoint/Restart (Fluid, PIC)	Q2,Q3
SPARC: Demonstrate hybrid-based mesh	Q4
Demonstrate HDF/DataElevator Burst Buffer Capability	Q4
Transition I/O Components to ATS-2	Q4

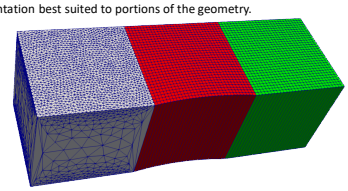
IOSS Library: Supporting Sandia's Mesh Datasets

IOSS Library: Adding Hybrid Mesh Capabilities

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

- **Generation**: unstructured grids are much faster to generate than structured grids
  - Unstructured: ≈ hours to days
  - Structured: ≈ weeks to months
- **Accuracy**: structured grids are 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.

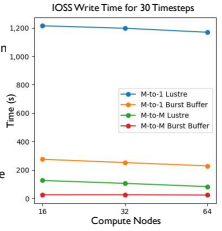


Sandia is developing a hybrid mesh capability for the IOSS library. It will support structured, unstructured, and hybrid meshes in CGNS and Exodus formats and also support FAODEL, Embedded Visualization, and DataWarp/Burst Buffer interfaces with a common abstraction.

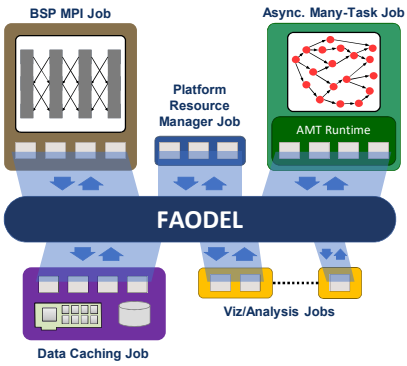
IOSS Library: Adding Support for Burst Buffers

Given that Burst Buffers can dramatically reduce application I/O overheads, Sandia is investigating different techniques for incorporating Burst Buffers into its production IOSS stack. While the most direct approach would be to use DataElevator at IOSS's HDF5 layer, there are incompatibilities between the current version of DataElevator and IOSS that prevent integration today. As an alternative, we constructed a custom HDF VOL to intercept I/O and route writes through the Burst Buffer using the DataWarp C API. This HDF VOL initiates asynchronous migrations as files are closed.

Performance experiments were conducted on Trinitite to observe how different storage configurations affect an application that produces datasets through IOSS. For a fixed problem size and rank size, we varied the number of compute nodes, M-to-1 vs. M-to-M file generation, and whether the Burst Buffers were enabled. Composed I/O (i.e., M-to-1) remains expensive, even with Burst Buffer support. Spreading the ranks across more nodes only offers a slight improvement for this ratio of Compute-to-Burst Buffer nodes.



Advanced Data Management: Flexible, Asynchronous, Object Data-Exchange Libraries (FAODEL)



Exascale workflows will require the dataflow between applications to be efficiently mapped to resources in the platform's memory/storage hierarchy. Sandia is developing FAODEL, a set of tools for creating new data management services, to provide applications with two distinct but related advantages. First, FAODEL provides general data description and manipulation mechanisms which allow developers to avoid lock-in to a particular problem space. Second, FAODEL decouples data management from I/O implementation technologies (e.g., file formats, databases, or filesystem APIs) for greater platform portability. The examples in this poster demonstrate how FAODEL can be adapted to different scenarios, freeing developers to focus on application needs rather than platform or I/O stack details.

Goal: Develop a collection of next-generation communication libraries that can be reused to implement new services

*Formerly the SNL **Data Warehouse***

Key Requirements:

- **Job-to-Job Communication**:
- **Asynchronous** and **Event Driven**
- **Modern C++ primitives**
- **Portability** (Network, NVM, Storage)

Release Info:
Current Version: 1801.1 (belfry)
New Home: https://github.com/faodel

Currently going through review for open source release.

Kelpie: Distributed, In-Memory Key/Blob Service
Kelpie provides a way for applications to decompose a dataset into a series of contiguous objects that can be distributed across a collection of nodes. Users typically construct one or more distributed hash tables on top of a set of nodes to spread the dataset, and use asynchronous publish/retrieve operations to manipulate objects.

The belfry release adds an updated API for transferring objects between memory and POSIX storage devices, and includes improvements to asynchronous message handling.

OpBox: Asynchronous, Communication Engine
Data management services such as Kelpie often need to execute a complex sequence of network operations in order to orchestrate application data transfers. OpBox is a communication library that enables service developers to describe their operations as event-driven state machines. State machines allow progress to take place as events occur without user intervention.

The belfry release includes an experimental unit for processing independent Op state machines concurrently in separate threads.

Lunasa: Network Memory Managemnt
Data management services typically manage memory in an explicit manner for both performance reasons (e.g., NIC memory registration overheads) and practicality (e.g., allocation tracking for local and remote memory). The Lunasa component provides a flexible registered memory management unit that is used throughout FAODEL. Lunasa allocates large blocks of contiguous memory that are then suballocated to users as reference-counted data objects.

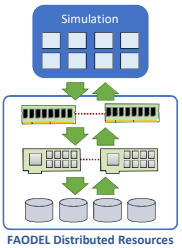
RDMA: Nessie NNTI or Libfabric
FAODEL uses low-level RDMA operations to implement job-to-job communication in a platform. This communication coexists with an application's MPI and *does not* require an MPI com-splitter.

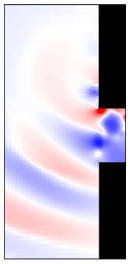
Additional RESTful Interface
FAODEL provides a separate, socket-based communication plane for RESTful operations. Users may add application-specific hooks (e.g., for debugging) by simply registering C++ lambdas.

Data Services for ATDM's SPARC and EMPIRE Applications

ATDM's SPARC and EMPIRE applications need portable I/O mechanisms for reading/writing datasets, handling checkpoint/restarts, and interfacing with visualization/analysis tools. The long-term goal of this project is to converge on a single I/O library (IOSS with FAODEL) that can support all of these requirements in a scalable and portable manner. This year we are focusing on adding checkpoint/restart capabilities to SPARC and EMPIRE.

An initial checkpoint/restart capability for SPARC was developed in Q1. This work explored two options: a baseline implementation that uses libhio to manage checkpoints through the Burst Buffer and an experimental implementation that uses Kelpie. Performance testing will be performed in Q2. A similar combination of checkpoint/restart units is under development for the EMPIRE application, and will be reported on in Q4.

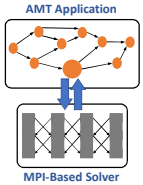




Code Coupling for Asynchronous, Many Task

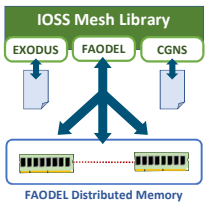
In order to gain wider adoption, Asynchronous, Many-Task (AMT) programming models need mechanisms for allowing application developers to couple an AMT code with existing MPI-based solvers that are already debugged and performance tuned.

In support of ATDM's AMT effort, we investigated whether FAODEL could be used to connect DARMA/Charm++ with external applications. We confirmed that Charm++'s underlying Converse communication layer could coexist with FAODEL's, and that a Charm++ task (or chare) could use FAODEL to exchange data with a separate application.



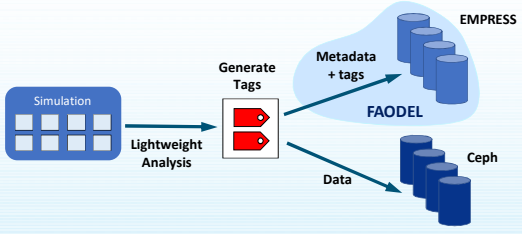
FAODEL Backend for IOSS

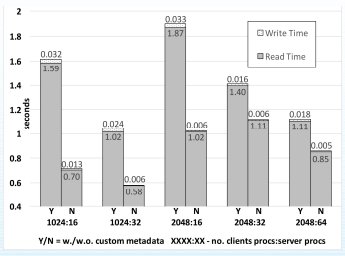
The IOSS library is the standard interface by which many Sandia applications access file-based, mesh databases such as Exodus and CGNS. We are developing a new backend for IOSS that will allow FAODEL to be used as an alternate mesh database housed in distributed memory. This backend converts IOSS API calls into Kelpie operations that use Lunasa data objects to hold mesh data. When completed this work will allow applications to offload common I/O operations (e.g., M-to-1 merging, committing checkpoints, and handling file formatting) to other nodes, and provide an alternate path for connecting applications to viz./analysis tools.



Accelerating Insight through Better Metadata (SIRIUS/ASCR)

EMPRESS offers a new level of metadata to reduce time to insight. Existing I/O libraries, such as ADIOS, HDF5, and NetCDF, offer rudimentary support for attributes, but the attributes are either applied to the entire file or attached to a single variable. EMPRESS adds a new capability for attaching fully customizable metadata to part of a variable to highlight a data feature in addition to the functionality supported by the I/O libraries. The granularity of this metadata is on a per-process or per-node basis and is maintained external to the data. With the separation of metadata and data, all of the metadata for an application run can be stored in a single location and managed separately. This offers a way to explore data features identified either at simulation time or during analysis (that are persisted) without having to load all of the associated data. Initial performance evaluations demonstrate that this approach is scalable.





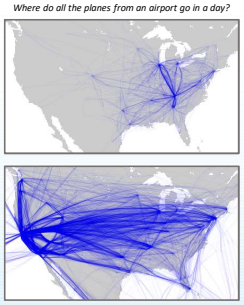
Other Communities: Data-Intensive Applications

Beyond ATDM, FAODEL's data management services have relevance to communities that are focused on data-intensive work. FAODEL provides a straightforward way to load a large dataset into distributed memory that other tools can inspect. This caching is particularly important in analytic scenarios where a user interactively steps through a large dataset.

As a prototype demonstration, we retrieved a large amount of airplane position data from a public website and then ingested it into FAODEL for simple analysis. Airplane tracks were extracted from the daily data files (8GB compressed) and then decomposed into 64MB track-data objects that point tools could more easily inspect.

Note: *This work is simply a strawman example. No ATDM or ECP funds were used to implement this example.*

Analyzing Airplane Track Data
Where do all the planes from an airport go in a day?



Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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. SAND2018-1096 PE

EXASCALE COMPUTING PROJECT

Sandia National Laboratories