Data Management, I/O Libraries and Databases at NERSC

Quincey Koziol
NERSC New User Training
February 24, 2017
koziol@lbl.gov
Outline

• Data Management Best Practices and Guidelines
• I/O Libraries
• Databases
Outline

- Data Management Best Practices and Guidelines
- I/O Libraries
- Databases
Why Manage Your Data?

• “Data management is the development, execution and supervision of plans, policies, programs and practices that control, protect, deliver and enhance the value of data and information assets.”*

*DAMA-DMBOK Guide (Data Management Body of Knowledge) Introduction & Project Status
Data @ NERSC

NERSC offers a variety of services to support data-centric workloads. We provide tools in the areas of:

- Data Analytics (statistics, machine learning, imaging)
- Data Management (storage, representation)
- Data Transfer
- Workflows
- Science Gateways
- Visualization

http://www.nersc.gov/users/data-analytics/
NERSC offers a variety of services to support data-centric workloads. We provide tools in the areas of:

• Data Analytics (statistics, machine learning, imaging)
• Data Management (storage, representation)
• Data Transfer
• Workflows
• Science Gateways
• Visualization

http://www.nersc.gov/users/data-analytics/
General Recommendations

• NERSC recommends the use of modern, scientific I/O libraries (HDF5, netCDF, ROOT) to represent and store scientific data.

• We provide database technologies (MongoDB, SciDB, MySQL, PostGreSQL) for our users as a complementary mechanism for storing and accessing data.

• Low-level, POSIX I/O from applications to NERSC file systems, if necessary. Details here:
  
  http://www.nersc.gov/users/storage-and-file-systems/
Notes on NERSC File I/O

• Use the local scratch file system on Edison and Cori for best I/O rates.
• For some types of I/O you can further optimize I/O rates using a technique called file striping.
• Keep in mind that data in the local scratch directories are purged, so you should always backup important files to HPSS* or project space.
• You can share data with your collaborators using project directories. These are directories that are shared by all members of a NERSC repository.

*HPSS: http://www.nersc.gov/users/storage-and-file-systems/hpss/getting-started/
Lustre

- Scalable, POSIX-compliant parallel file system designed for large, distributed-memory systems
- Uses a client-server model with separate servers for file metadata and file content
Scientific I/O

I/O is commonly used by scientific applications to achieve goals like:

- Storing numerical output from simulations for later analysis or workflow stages
- Implementing 'out-of-core' techniques for algorithms that process more data than can fit in system memory and must page in data from disk
- Checkpointing application state to files, in case of application or system failure.
Types of Application I/O to Parallel File Systems

File-per-processor

Shared file (independent)

Shared file (collective buffering)
MPI Collective I/O

• *Collective I/O* refers to a set of optimizations available in many implementations of MPI-IO that improve the performance of large-scale IO to shared files.

• To enable these optimizations, you must use the *collective* calls in the MPI-IO library that end in *_all*.
  – For instance: `MPI_File_write_at_all()`.

• And, all MPI tasks in the given MPI communicator must participate in the collective call, even if they are not performing any IO operations.

• The MPI-IO library has a heuristic to determine whether to enable *collective buffering*, the primary optimization used in collective mode.
Outline

- Data Management Best Practices and Guidelines
- I/O Libraries
- Databases
Why I/O Middleware?

• The complexity of I/O systems poses significant challenges in investigating the root cause of performance loss.

• Use of I/O middleware for writing parallel applications can greatly enhance application developer productivity.
  – Such an approach hides many of the complexities associated with performing parallel I/O, rather than relying purely on programming language aids and parallel library support, such as MPI.
• HDF5
  – A data model and set of libraries & tools for storing and managing large scientific datasets.

• netCDF
  – A set of libraries and machine-independent data formats for creation, access, and sharing of array-oriented scientific data.

• ROOT
  – A self-describing, column-based binary file format that allows serialization of a large collection of C++ objects and efficient subsequent analysis.

• Others
HDF5

• The Hierarchical Data Format v5 (HDF5) library is a portable I/O library used for storing scientific data.

• The HDF5 technology suite includes:
  – A versatile data model that can represent very complex data objects and a wide variety of metadata.
  – A completely portable file format with no limit on the number or size of data objects in the collection.
  – A software library that runs on a range of computational platforms, from laptops to massively parallel systems, and implements a high-level API with C, C++, Fortran 90, and Java interfaces.
  – A rich set of integrated performance features that allow for access time and storage space optimizations.
  – Tools and applications for managing, manipulating, viewing, and analyzing the data in the collection.

• HDF5's 'object database' data model enables users to focus on high-level concepts of relationships between data objects rather than descending into the details of the specific layout of every byte in the data file.
netCDF

- netCDF (“Network Common Data Form”) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.

- netCDF is:
  - Typically used in the climate field
  - More constrained than HDF5
  - At a higher level of abstraction

- More netCDF information here:
  http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/
ROOT

• A set of object oriented frameworks with the functionality needed to handle and analyze large amounts of data in an efficient way.
  – Heavily used in experimental HEP/NP

• ROOT is written in C++ and creates self-describing files, with a flexible object serialization and fast column-oriented access.

• Originally designed for particle physics, its usage has extended to other data-intensive fields like astrophysics and neuroscience.
  – Integrated histogramming / querying/ machine learning and in most HEP experiment frameworks.
  – ROOT is mainly used for data analysis at NERSC.

• ROOT Docs: https://root.cern.ch/drupal/
Outline

• Data Management Best Practices and Guidelines
• I/O Libraries
• Databases
Databases @ NERSC

- NERSC supports the provisioning of databases to hold large scientific datasets, as part of the science gateways effort.
- Data-centric science often benefits from database solutions to store scientific data or metadata about data stored in more traditional file formats like HDF5, netCDF or ROOT.
- Our database offerings are targeted toward large data sets and high performance. Currently we support:
  - MySQL
  - PostgreSQL
  - MongoDB
  - SciDB
- If you would like to request a database at NERSC please fill out this form and you'll be contacted by NERSC staff: [http://www.nersc.gov/users/data-analytics/data-management/databases/science-database-request-form/](http://www.nersc.gov/users/data-analytics/data-management/databases/science-database-request-form/)
PostgreSQL

• PostgreSQL is an object-relational database. It is known for having powerful and advanced features and extensions as well as supporting SQL standards.

• NERSC provides a set of database nodes for users that wish to use PostgreSQL with their scientific applications.

• PostgreSQL documentation here: http://www.postgresql.org/docs/
MySQL

- **MySQL is a very popular and powerful open-source relational database.**
- **Many features:**
  - Pluggable Storage Engine Architecture, with multiple storage engines:
    - InnoDB
    - MyISAM
    - NDB (MySQL Cluster)
    - Memory
    - Merge
    - Archive
    - CSV
    - and more
  - Replication to improve application performance and scalability
  - Partitioning to improve performance and management of large database applications
  - Stored Procedures to improve developer productivity
  - Views to ensure sensitive information is not compromised
  - ...
- **MySQL user documentation:**
SciDB

- SciDB is a parallel database for array-structured data, good for TBs of time series, spectra, imaging, etc.
- A full ACID database management system that stores data in multidimensional arrays with strongly-typed attributes (aka fields) within each cell.
- SciDB User Documentation:
  
  https://paradigm4.atlassian.net/wiki/display/ESD/SciDB+Documentation

- To request access to NERSC SciDB instances, please email consult@nersc.gov
MongoDB

• A cross-platform document-oriented database.
• Classified as a *NoSQL* database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas, making the integration of data in certain types of applications easier and faster.
• MongoDB user documentation:
  https://docs.mongodb.com/v2.6/
Questions, Comments, Feedback?