Accelerating Discovery From High Performance Computing Applications Using Descriptive Metadata Management

Margaret Lawson

ILLINOIS
Computer Science
GRAINGER COLLEGE OF ENGINEERING

Insights

- Analysis often requires loading in data from short- or long-term storage
  - Not all analysis can be done in-situ or in-transit
  - Analysis is often iterative

- Loading in this data uses a lot of core hours
  - Applications produce large amounts of data (e.g., 100 PB for 24 hours run of XGC1)
  - I/O bottleneck
    - Expected to get worse with exascale

- Scientists often do not need all of the data
  - Many analyses only require datasets that contain a feature of interest
    - Astronomy: supernovae, halos
    - Climate: tropical cyclones

Insights (continued)

- Scientists have many ways to identify these features of interest
  - Gradients
  - Thresholds
  - Entropy
  - Variance and range
  - Machine learning
  - Visualization

Opportunity

- Main idea: help scientists load only the data containing features of interest

- Overall approach:
  - Store descriptive metadata (high-level information) about features of interest that have been identified in-situ, in-transit, or during post-processing using existing feature identification techniques
  - Provide efficient ways for scientists to search this metadata to identify the relevant features for a given type of analysis
  - Allow scientists to efficiently determine what data is associated with a particular feature (that is described using descriptive metadata)

Importance

- Time spent performing I/O is time not spent making discoveries
  - Many discoveries will be delayed or never made at all

- HPC is used for many critical tasks:
  - Improved medical treatments
  - Developing safe and efficient energy sources
  - Improved prediction and planning for natural disasters and climate change

- Exascale
  - The problem is going to continue to get worse

If we can reduce the time scientists spend loading data for analysis scientists will be able to make discoveries that would otherwise be impossible

State of the Practice

- Existing descriptive metadata solutions are limited to only:
  - Storing features that are associated with entire files
  - Efficiently supporting a single kind of scientific analysis
  - Supporting a single application or a small group of applications

- Key takeaway: there is no existing general descriptive metadata solution
  - No existing system can help scientists working with a **wide range of applications, analyses, and mesh types** to restrict their reads to data subsets containing features of interest
  - With a general solution, more scientists can benefit and more discoveries will be made

Challenges in Creating a General Solution

- The system must be able to efficiently map from metadata describing features of interest to the associated data
  - Particularly challenging for applications with complex meshes
- Substantial flexibility and/or extensibility are required to support the wide range of applications, analyses, and mesh types used in HPC
- HPC systems must be high performing, have good scalability, and require minimal storage overhead

Problem: Supporting Complex Meshes

- Q: can we perform metadata-data mapping for complex meshes in a way that is high performing, scalable and has minimal storage overheads?

- Our novel solution: store spatial bounding boxes for features of interest. Then, perform metadata-data mapping by doing an (indexed) range query on the mesh
  - Structures for efficient range queries have been studied for decades
  - Can use an in-memory mesh index to obtain O(log(n)) search complexity with 0 storage overheads
    - This solution is portable and not affected by data layout changes
  - We can parallelize mesh index creation and searching

Problem: Implementing the Mesh Index

- Q: do any existing range queries indices provide the level of performance, scalability, and memory overheads needed in HPC?
- Approach: analyze 20 state of the art spatial index implementations
  - Free, open-source, C/C++
- Findings: Boost's R-tree and CGAL's R-tree are well-suited to an HPC environment

Problem: Flexible and Efficient Metadata Storage and Retrieval

- Q: can we enable efficient storage and retrieval of descriptive metadata while providing sufficient flexibility to support the wide range of applications and analyses used in HPC?

- Our solution:

High-level
  - A domain-independent metadata model
  - Support for extensible, user-defined metadata

Low-level
  - A flat namespace
  - Separate values that do and do not need to be searched
  - Support storing values of flexible data type

Problem: Implementing Our Metadata Storage Solution

- Q: do any existing storage backends provide the level of performance, scalability, and storage overheads needed for HPC descriptive metadata workloads?

- Approach:
  - Develop a set of evaluation criteria, use to determine which databases are best suited to use in HPC
  - Evaluate the most promising databases using a wide range of descriptive metadata workloads

- Findings: Overall, SQLite is well-suited to HPC
  - Can achieve good multi-threading using a separate database per core

EMPRESSA – the First General Descriptive Metadata Solution

- Uses our novel metadata-data mapping and metadata storage solutions
- A production-ready solution that scientists can integrate into their workflows due to its free, permissive open-source license
- Is implemented as header-only C++ library
- Has two components: metadata management engine and mesh index

Metadata Management

- Embedded (shared-nothing) servers

- For each piece of metadata users can store:
  - The type of feature
  - What application run, timestep, and variable the feature is associated with
  - The spatial location of the feature
  - Information about the feature

- Supports analysis:
  - That is global, temporal, spatial, multi-variate, value-based
  - Can be perform exact matches, range queries, substring matches

Evaluation Setup

- Machines:
  - Eclipse:
    - Each node has 128 GB RAM, 36 cores
    - All-HDD Lustre parallel file system (PFS)
  - Stria
    - Each node has 128 GB DRAM, 56 cores
    - All-SSD Lustre parallel file system (PFS)

- For all tests we use hybrid MPI+OpenMP programming with one thread per core

Evaluation Datasets

- 3 different datasets, and associated metadatasets and meshes
  - CFD, climate, Dark Sky

- Highly varied in terms of the number of application runs, timesteps, variables, metadata types, total metadata count, number of files per application run

- Represents some of the variability found in HPC, tests the most complex mesh type (unstructured)

Evaluation Process

1. Writing
   - All metadata is written
2. Reading
   - The mesh indexes are created
   - The metadata databases are opened
   - The metadata is queried to find the features needed for various analysis routines
   - For each query
     - The mesh indexes are queried to find what data is associated with these features of interest
     - The associated data is read in for analysis

Results – Writing

- Total time
  - Ranges from 0.6 – 47.2 seconds
- Scalability
  - Can achieve weak and strong scaling of ~80%+

Results – Reading

- Total time
  - 10.2–29.2 seconds for all datasets

- Scalability
  - CFD and Dark Sky: ~70-100%
  - Climate: 50% or less
    - Scalability suffers when the dataset has a large number of temporally decomposed files -> contention

Results – Accelerating Analysis (continued)

- Can speed up analysis by orders of magnitude in most cases

- Only cases when we fail to produce any speedups
  - We have to read in all data
  - The total amount of data is very small (e.g., 1 MB)

Results – Metadata-Data Mapping

- Mesh index build throughput:
  - 3-7.5 million mesh vertices inserted per second per core
- Mesh index search throughput:
  - ~5-225 million mesh nodes searched per core per second
    - Better throughput for larger meshes
- Takeaways
  - We achieve high build and search performance, excellent parallelization

Results – Storage Overheads

Metadataset	Raw Metadata Size (GB)	Database Size (GB)	Storage Overhead	
			Metadataset	Dataset
CFD	1.08	2.12	96%	0.003%
Climate	9.05	19.9	120%	0.03%
Dark Sky	10.1	13.5	34%	0.7%

- Takeaways:
  - While the overheads are large compared to the metadataset size, they are very small compared to the total dataset size

Contribution

- Overall, we have shown that ***a general descriptive metadata management solution will be an essential component for any exascale system*** since it enables analyses that would otherwise be impossible by allowing scientists to only load in the data that is needed for analysis

Future Work

- Explore additional classes of applications
  - AMR applications
- Improved metadata-data mapping for arbitrarily shaped mesh elements and features of interest
- Integrate EMPRESSA with existing application workflows

Thank you!