DAMASC

Adding Data Management Services to Parallel File Systems

Scott Brandt (lead PI), Carlos Maltzahn (co-PI), Kleoni Ioannidou, Joe Buck, Noah Watkins
UCSC – Systems Research Lab

Neoklis Polyzotis (co-PI), Wang-Chiew Tan (co-PI), Jeff LeFevre
UCSC – Database Systems Group

Maya Gokhale (PI, LLNL), Celeste Matarazzo
LLNL

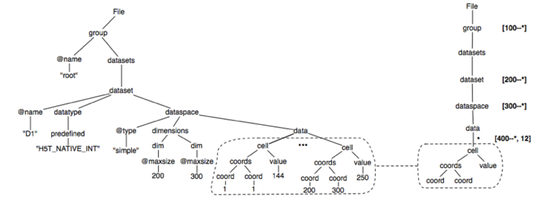
Jim Ahrens, John Bent, Gary Grider, James Nunez, Meghan Wingate
LANL

Russ Rew
UCAR/Unidata

Objectives of DAMASC:

Coalesce data management with parallel file system management to present a declarative interface to scientists for managing, querying, and analyzing extremely large data sets efficiently and predictably.

Scientific Data Model



Example representation of HDF5 file in scientific data model:
• Contents represented as a node-labeled graph
• Information encoded in the node labels and their nesting.

Sample schema generated by parser:
• Omits contents of data space
• Points to byte extents.

Data Management Services

Incremental Parsing

• Format-specific parsers expose common scientific data model
• Parse trees form structural indices into byte stream data
• Lazy, demand-driven parsing

Query Execution

• Optimized queries executed over parse trees and unmodified data
• Scientific file-formats can be modified to use DAMASC transparently from the application view-point.
• System feedback is used to tune system indices and caching behavior

Automatic Indexing

• Automatic indexing of content is critical for optimized query evaluation
• Content managed by DAMASC and query patterns are continuously monitored and re-organized

Provenance Tracking

• Understanding the lineage of extreme scale data is crucial as complex workflows manipulate and copy raw data
• Automatic provenance capture monitors the flow of queries in DAMASC.

File Views

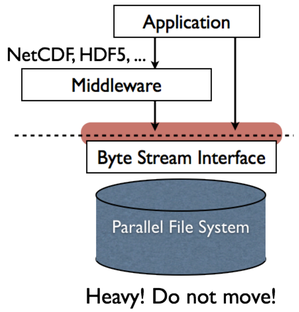
• Views are expressed in the flexible, semi-structured data model used by DAMASC.
• A view is a query result, and can provide a unified view of content stored in multiple formats simultaneously.
• A view is the result of a query over the DAMASC data model
• Non-materialized views are constructed on the fly without having to create new copies of data.
• Maximum flexibility is available by allowing views to be defined on top of other views.

The POSIX IO Bottleneck

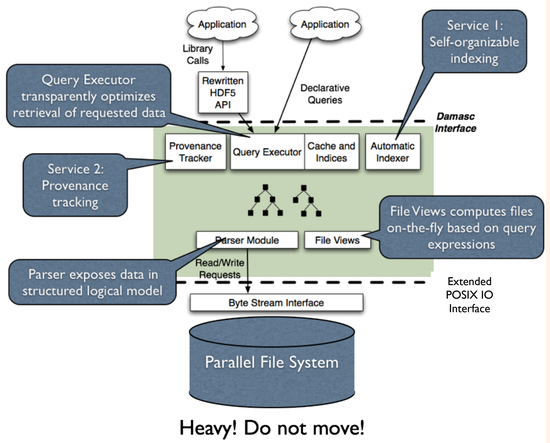
POSIX IO dominates File system interface
• POSIX IO does not scale
• **~50 years ago**: 100MB of data at high end
• Now: up to **1 billion times** more data

Performance price of POSIX IO is high
• Workload-specific interposition layers (e.g. PLFS):
• almost **1,000x speed-up**

Common Workaround
• **Middleware** tries to make up for limitations



DAMASC Components:

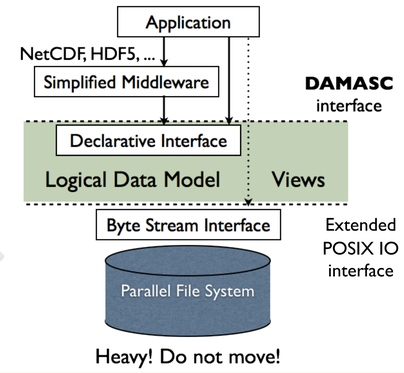


DAMASC:

DAta MAnagement in Scientific Computing

Enhance parallel file system with data services
• Declarative **querying**
• **Views**
• Automatic content **indexing**
• **Provenance** tracking

***In situ* processing** on storage nodes



Current Work

SciHadoop:
• Intelligent input partitioning based on logical/physical mapping
• Query optimizations enabled by semantic awareness of mappers
• Use cases: HDFS and Ceph

SciZorba:
• Declarative query processing
• Integration of NetCDF and HDF5 operators in XML queries

SciZorba

SciZorba is a system for executing scientific data queries expressed in the flexible XQuery language over data stored in the netCDF file format.

• Based on the Zorba XQuery processing engine

• Stores data in semi-structured XDM data model

• Incremental parsing loads data on-demand as required by an executing query

• Currently SciZorba runs on a single-node, and we plan to extend this to multiple cooperating nodes each running SciZorba instances that communicate to process a single query.

SciHadoop

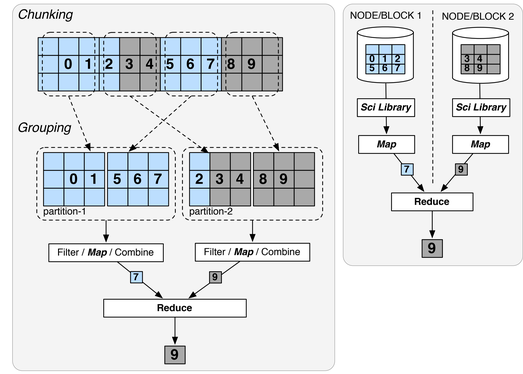
SciHadoop is a MapReduce-based framework for large-scale processing of scientific, array-based data.

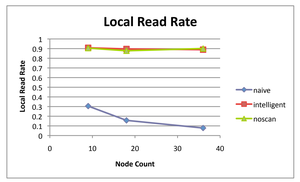
• MapReduce partitions an input file's byte stream into blocks that are processed in parallel

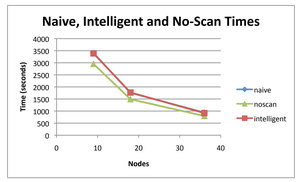
• Reading scientific data requires passing through middleware that hides physical layout information

• These middleware libraries make it difficult to predict data locations

SciHadoop: Logical Plan *vs* Physical Accesses







Preliminary Results

Graphs on the left show SciHadoop enabling better locality and more efficient data access for queries over NetCDF data stored in HDFS

Future Work

• Extending support to other file formats such as HDF5 and FITS.

• Supporting write workloads by replacing HDFS with a file system such as Ceph.

Acknowledgements

This work is funded by DOE grant DE-SC0005428, partially funded by NSF grant #1018914, and the UCSC/LANL Institute for Scalable Scientific Data Management.

Further Information

All information relating to the DAMASC project can be found at http://srl.ucsc.edu/projects/damasc