Damasc
Adding Data Management Services to Parallel File Systems

Scott A. Brandt (lead-PI – UCSC),
Maya B. Gokhale (PI – LLNL),
Carlos Maltzahn, Neoklis Polyzotis, Wang-Chiew Tan (co-PIs – UCSC),
Kleoni Ioannidou (post-doc – UCSC)

Objective: 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.

One of the greatest challenges of exa-scale computing is the management of extremely large data sets. The energy and cost of moving such data sets mandates designs where computation is close to where the data is stored. Current architectures consist of compute and analysis clusters that access data at a physically separate parallel file system and leave it largely to the scientist to worry about reducing the overhead of data movement.

An often considered approach is to allow the execution of arbitrary *in situ* data processing code within a parallel file system. A drawback of this approach is that it makes performance of such "active" file systems highly unpredictable: parallel file systems are highly optimized for high throughput at low latency requiring resource schedulers finely tuned to a limited and well-known set of functions.

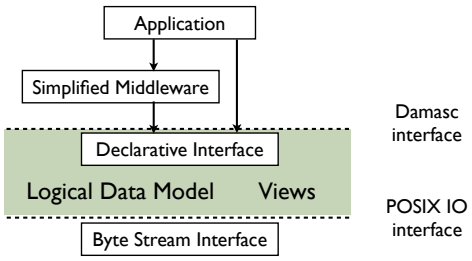


Figure 1: Damasc provides a declarative interface based on a logical data model of highly structured scientific data stored as byte stream files.

Over the past decades the high-end computing community has adopted middleware with multiple layers of abstractions and specialized file formats such as NetCDF-4 and HDF5. These abstractions provide a limited set of high-level data processing functions. However these functions have inherent functionality and performance limitations: middleware that provides access to the highly structured contents of scientific data files stored in the (unstructured) file systems can only optimize to the extent that the file system interfaces permit, and the highly structured formats of these files often impedes native file system performance optimizations.

We are developing Damasc, an enhanced high-performance file system with native rich data management services. Damasc will enable efficient queries and updates over files stored in their native byte-stream format while retaining the inherent performance of file system data storage via *declarative* queries and updates over *views* of underlying files.

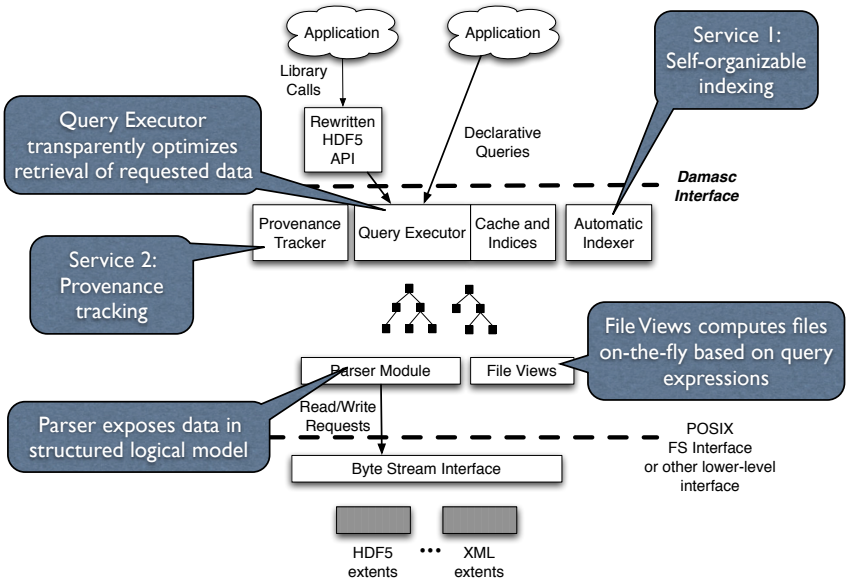


Figure 2: Damasc has four key benefits for the development of data-intensive scientific code: (1) applications can use high-level services, such as declarative queries, views, and provenance tracking, that are currently available only within a database system; (2) the use of these services becomes easier, as they are provided within a familiar file-based ecosystem; (3) common optimizations, e.g., indexing and caching, are readily supported across several file formats, avoiding effort duplication; and, (4) significant performance improvements as data processing is integrated more tightly with data storage.

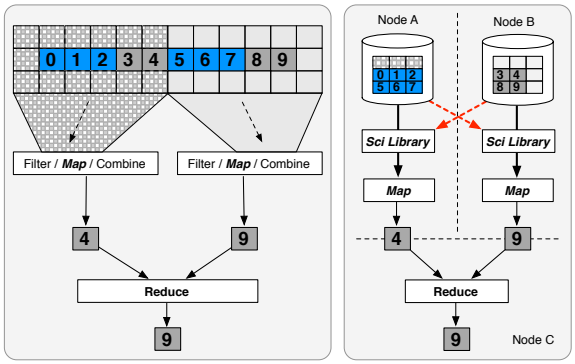
SciHadoop: Array-based Query Processing in Hadoop

A Damasc Project

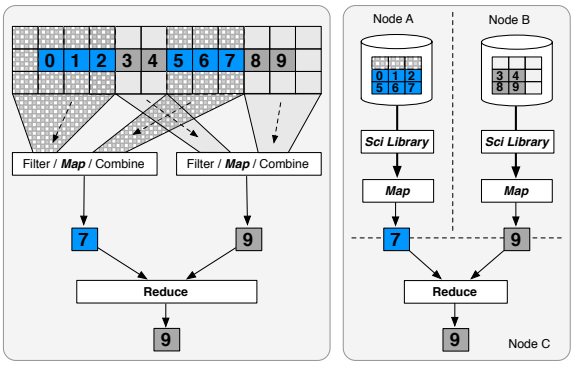
Hadoop has become the de-facto standard platform for large-scale analysis in commercial applications and increasingly also in scientific applications. However, applying Hadoop's byte stream data model to scientific data that is commonly stored according to highly-structured, abstract data models causes a number of inefficiencies that significantly limits the scalability of Hadoop applications in science. In this paper we introduce SciHadoop, a modification of Hadoop which allows scientists to specify abstract queries using a logical, array-based data model and which executes these queries as map/reduce programs defined on the logical data model. We describe the implementation of a SciHadoop prototype and use it to quantify the performance of three levels of accumulative optimizations over the Hadoop baseline where a NetCDF data set is managed in a default Hadoop configuration: the first optimization avoids remote reads by intelligently subdividing the input space of mappers on the logical level and instantiates mapper tasks with subqueries against the logical data model, the second optimization avoids full file scans by taking advantage of metadata available in the scientific data, and the third optimization further minimizes data transfers by pulling holistic functions (i.e. functions that cannot compute partial results) to mappers whenever possible.

A Technical Report about SciHadoop is available at: http://www.soe.ucsc.edu/research/report?ID=1603

Logical Plan *vs* Physical Accesses: Naive Partitioning



Remote Reads Minimization



SRL SYSTEMS RESEARCH LAB — Baskin Engineering UC SANTA CRUZ

The **Systems Research Lab** (SRL) is part of the Jack Baskin School of Engineering at the University of California, Santa Cruz. The SRL is interested in a broad range of topics including real-time systems, performance management, and large-scale storage systems. We are particularly interested in the intersection of these topics, and in their application to problems requiring inter-disciplinary collaboration. The SRL is composed of a broad range of students and collaborators from industry, government research labs, and other universities.

An important mission of the SRL is to come up with practical and efficient solutions for challenging problems in large-scale systems and pair them with theoretical insights that enable guarantees. We draw upon a large range of systems expertise from our lab members and collaborators from industry, governmental labs, and universities.

The SRL is closely associated with the Institute for Scalable Scientific Data Management (ISSDM), a collaboration between Los Alamos National Lab (LANL) and the University of California Santa Cruz (UCSC).

Sponsors and Collaborators

Los Alamos NATIONAL LABORATORY EST. 1943
Lawrence Livermore National Laboratory
unidata

Funded by DOE grant DE-SC0005428, partially funded by NSF grant #1018914, and by the ISSDM.

Project Web Site

http://srl.ucsc.edu/projects/damasc