Abstract—Scientific datasets and libraries, such as HDF5, ADIOS, and NetCDF, have been used widely in many data
intensive applications. These libraries have their special file
formats and I/O functions to provide efficient access to large
datasets. When the data size keeps increasing, these high level I/O
libraries face new challenges. Recent studies have started to
utilize database techniques such as indexing and subsetting, and
data reorganization to manage the increasing datasets. In this
work, we present a new approach to boost the data analysis
performance, namely Fast Analysis with Statistical Metadata
(FASM), via data subsetting and integrating a small amount of
statistics into the original datasets. The added statistical
information illustrates the data shape and provides knowledge of
the data distribution; therefore the original I/O libraries can
utilize these statistical metadata to perform fast queries and
analyses. The proposed FASM approach is currently evaluated
with the PnetCDF on Lustre file systems, but can also be
implemented with other scientific libraries. The FASM can
potentially lead to a new dataset design and can have an impact
on big data analysis.

Keywords: high performance computing; data intensive
computing; big data; storage systems; statistical techniques; FASM

I. INTRODUCTION

The volume of data collected from instruments and
simulations for scientific discovery and innovations is
increasing rapidly. For example, the Global Cloud Resolving
Model (GCRM) project [1], part of DOE's Scientific Discovery
through Advanced Computing (SciDAC) program, is built on a
geodesic grid that consists of more than 100 million hexagonal
columns with 128 levels per column. These 128 levels will
cover a layer of 50 kilometers of atmosphere up from the
surface of the earth. For each of these grid cells, scientists need
to store, analyze, and predict parameters like the wind speed,
temperature, pressure, etc. Most of these global atmospheric
models process data in a 100-kilometer scale (the distance on
the ground); however, scientists desire higher resolution and
finer granularity, which can lead to significantly larger sizes of
datasets. The data volume processed online by many
applications has exceeded TBs or even tens of TBs; the off-line
data is near PBs of scale [6]. How to efficiently analyze and
retrieve the datasets has become a key challenge in processing
the rapidly growing volume of data.

Scientific datasets and libraries, including HDF5, NetCDF,
PnetCDF, and ADIOS, have been well developed and widely
used in recent years for scientific data management [2, 3]. With
these libraries, datasets are stored in a specific format.

Scientific applications can make use of these libraries’ high
performance I/O to accelerate the processing performance
instead of using traditional databases. Those libraries, however,
are still not efficient enough for scientists, e.g., they lack non-
procedural query and sufficient indexing. Thus researchers
have begun employing existing useful techniques of traditional
databases into scientific datasets such as indexing and
subsetting [4, 5].

In this study, we argue that the raw datasets and current
formats are not sufficient for achieving the best performance.
Significant performance improvements can be achieved with
redesigning the datasets and enhancing relevant libraries. Our
main idea is to perform data subsetting on raw datasets, and
then add a small amount of statistical metadata into raw
datasets to guide data accesses. In this paper, we present our
initial investigation results and a prototyping system named
Fast Analysis with Statistical Metadata (FASM).

II. BACKGROUND AND MOTIVATIONS

Scientists are interested in understanding the phenomenon
behind the data. The data in many data-intensive scientific
applications are often write once and read many, which is
known as WORM access pattern. A typical data analysis
conducted by atmosphere scientists for climate modeling study
is shown in Fig. 1. Scientists often need to perform such
queries to select data points of interests for in understanding the
phenomenon behind the data.

```
Select data points
From datasets
Where pressure>80 And 12<temperature<25;
```

Fig. 1. Sample Query Request

Traditionally, without any prior knowledge of the datasets,
one would need to look at all data points and compare the
values of pressure and temperature within a specified range to
carry out this data analysis, which is a costly process. Instead,
we could potentially utilize the prior knowledge of the
underlying datasets, such as the data distribution, variable
ranges, and other statistical information, to facilitate such data
analysis. For this specific example, if the maximum pressure
within a subset is less than 80 (kPa), then there is no need to
look into the data points in such subset, and therefore the data
analysis performance can be significantly improved. This
thought motivates our idea of integrating statistical metadata
into scientific datasets and the proposed FASM system.
III. FAST ANALYSIS WITH STATISTICAL METADATA

In this section, we introduce the design of the proposed Fast Analysis with Statistical Metadata (FASM) system and discuss each component.

A. System Architecture

As shown in Fig. 2, the FASM system is designed to have four major components: Subsetting, Statistics Generating, Metadata Rich Datasets and FASM Runtime. When datasets are stored in the file system, two components, subsetting and statistics-generating, will process the data first. The subsetting component will partition the datasets into subsets and return the partition information. The statistics-generating component will calculate the statistics of each subset and store them together with the raw data as the Metadata Rich Datasets. The enhanced datasets will be used in subsequent analyses, with statistics guiding data access and analyses. These steps are supported by the enhanced I/O library and optimized access codes.

B. Subsetting

Many scientific datasets are high dimensional data arrays, such as three-dimensional array of double-precision floating-point data. Data subsetting is essentially to partition such a dataset into a group of smaller subsets, and then the statistics can be calculated for each subset and used to direct the data analysis. In Fig. 3, we demonstrate a fixed-size subsetting of a 3-D dataset. In the following subsections, we present two different subsetting schemes we have developed, i.e., locality-driven subsetting and concurrency-driven subsetting.

1) Locality Driven Subsetting

It is often desired to prune the search space with integrated statistics, while accessing as much contiguous data as possible at the same time. We discuss the consideration of locality in subsetting schemes in this subsection.

As shown in Fig. 4, if the subsetting happens along the (lon, level) plane, accessing a 2-D plane in these two dimensions will generate many non-contiguous I/Os. The elements on (lon, level:0) and (lon, level:1) are logically contiguous, however, the physical distance between them is lon×(lat−1). The same situation happens in subsetting (lat, level) plane. When it comes to (lat, time), the locality tends to be worse, as the physical distance between (lat, 0) and (lat, 1) is (level×lon−1)×lat, which is the size of a single 3-D subset. Table 1 summarizes subsetting schemes and impact on locality (the distance between contiguous subsets).

![Fig. 4. Temperature Datasets](image)

**TABLE I. SUBSETTING SCHEMES AND LOCALITY**

<table>
<thead>
<tr>
<th>Type</th>
<th>Dimension</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub1</td>
<td>(lat, lon)</td>
<td>0</td>
</tr>
<tr>
<td>sub2</td>
<td>(lon, level)</td>
<td>(lat−1)×lon</td>
</tr>
<tr>
<td>sub3</td>
<td>(lat, level)</td>
<td>lat×(lon−1)</td>
</tr>
<tr>
<td>sub4</td>
<td>(lon, time)</td>
<td>(level×lat−1)×lon</td>
</tr>
<tr>
<td>sub5</td>
<td>(lat, time)</td>
<td>(level×lon−1)×lat</td>
</tr>
<tr>
<td>sub6</td>
<td>(level, time)</td>
<td>level×(lon×lat−1)</td>
</tr>
</tbody>
</table>

The FASM is designed to take the locality into consideration. The FASM will compute the distance of different subsetting schemes to direct the subsetting scheme selection with consideration of the impact of data locality.

2) Concurrency Driven Subsetting

Concurrency plays a critical role in exploring parallelism in the access and analysis of scientific datasets. It is contributed by data distribution among a set of nodes managed by parallel file systems.

Suppose the strip size is denoted as stripe size and the number of storage nodes is n in a parallel file system. The size of each 3-D(level×lon×lat) subset is m. In PnetCDF, we need to specify the access length (MPI_Offset count[]), for instance, when the access is conducted along (lat, lon) plane, we can specify the number of level steps, denoted as x. Then on each node, there are \( \frac{m}{\text{stripe size}} \) subsets. The total number of
levels in one node is $\frac{\text{stripe size}}{m} \times \text{level}$. If we access $x$ levels, the number of targeted storage nodes should be $x / \left(\frac{\text{stripe size}}{m} \times \text{level}\right)$, and the maximum value is the number of the total nodes $n$. The concurrency of other subsetting schemes can be calculated similarly, and are shown in Table 2. This table is used to guide the subsetting scheme in order to achieve the best concurrency in parallel data accesses.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Concurrency</th>
</tr>
</thead>
<tbody>
<tr>
<td>sub1</td>
<td>$\min\left(\frac{x \times m}{\text{stripe size} \times \text{level}}, n\right)$</td>
</tr>
<tr>
<td>sub2</td>
<td>$\min\left(\frac{x \times m}{\text{stripe size} \times \text{lat}}, n\right)$</td>
</tr>
<tr>
<td>sub3</td>
<td>$\min\left(\frac{x \times m}{\text{stripe size} \times \text{lon}}, n\right)$</td>
</tr>
<tr>
<td>sub4, sub5, sub6</td>
<td>$\min\left(\frac{x \times m}{\text{stripe size}}, n\right)$</td>
</tr>
</tbody>
</table>

C. Statistics Generating and Enhanced Datasets

There are different statistics we can utilize. A typical case is the minimum and maximum values extracted from the subset. With those statistics, we can skip searching certain amounts of subsets and improve the access performance.

The raw dataset enhanced with subsetting and generated statistics is called a metadata rich dataset in this study. A sample enhanced dataset generated from the previous dataset in section II is illustrated in Fig. 5. We introduce a statistical metadata portion to describe integrated statistics.

```c
netcdf temperature {
    dimensions :
        time = UNLIMITED ; // (37658 currently)
        lat = 120 ;
        lon = 20 ;
        level = 30 ;
    variables :
        int temperature(time, level, lat, lon) ;
        statistics:
            min:
                (time:0,level:0,lon:0,lat:0):10
                (time:0,level:0,lon:1,lat:0):11
            max:
                (time:0,level:0,lon:0,lat:0):37
                (time:0,level:0,lon:0,lat:1):47
        data :
            data =
                30, 21, 32, 33, 33, 32, 26, 37, 28, 29, 10, 11, 12,
                12, 11, 15, 16, 17, 16, 19, 20, 21, 23, 24, 23, 26,
                27, 27, 29, 30, 33, 32, 35, 34, 25, 46, 47, 38, 40,
                40, ...
}
```

Fig. 5. FASM Dataset Format

D. FASM Runtime

The FASM runtime component leverages integrated statistical metadata to facilitate data analysis and queries. For instance, the `ncmpi_get_vara_type` function can read an array of various types from a PnetCDF dataset. The access pattern is defined by giving starting position and a vector of edge lengths.

With the support of FASM, the library is enhanced to take advantage of integrated statistics for better analysis performance. As shown in Fig. 6, the read function gets a modified access pattern with skipping subsets that do not need access.

```c
Input: query request and statistics_metadata;
Fasm read operation:
Step 1: In each access, get Statistics_metadata from previous fasm analysis;
Step 2: Filter useless subsets;
Step 3: Modify accessing pattern:
    new_start[] = fasm.start;
    new_count[] = fasm.count;
Step 4: fasm-guided read:
    ncmpi_get_vara_float(ncid, varid, new_start[], new_count[],*fp);
Return: Query Result
```

A. Experimental Platform

We have conducted experimental tests with both a single node and a 640-node cluster. We have generated a group of synthetic datasets. The data size is ranged from 300 KB to 100 GB. The data analysis and query statements are also generated randomly. The generated query statements have a normal distribution of different ranges.

B. Performance Comparison of Statistics

As reported in Fig. 7, the FASM approach with integrated statistics does not affect the performance obviously when the dataset size is small. As the dataset size increases, the FASM approach demonstrates clear performance advantages.

C. Overhead Analysis

![Fig. 8. Storage Overhead of Additional Metadata](image-url)
The integrated additional statistical metadata can cause storage overhead. We have calculated the size of additional metadata for each subsetting scheme and plotted in Fig. 8. From Fig. 8, we can conclude that the storage overhead is less than one percent for three subsetting schemes. The overhead is near nine percent for only one subsetting scheme. Overall, the FASM system can utilize this small amount of metadata to improve the performance without dramatically increasing the storage overhead, which is normally a concern for other techniques like indexing.

D. Locality Analysis

We have carried out evaluations of the impact of locality on different subsetting schemes. As shown in Fig. 9, the result matches with our theoretical analysis. Sub1, sub2 and sub3 are better than sub4 and sub5. This observation confirms that the subsetting schemes of sub4 and sub5 decrease the locality, whereas sub1, sub2, and sub3 schemes are significantly better. The sub3 scheme is slightly better than sub1 and sub2 schemes. This is because, even though the distance between two consistent data is the same as in the sub2 scheme and larger than that in the sub1 scheme, the data access pattern has an impact. The sub3 scheme accesses a bigger chunk each time and then moves certain distances to access another chunk; while in sub1 and sub2 schemes, the access of the same size of chunk as in the sub3 scheme results in noncontiguous accesses, which means multiple small noncontiguous chunks are read to construct a big chunk. Thus, the access latency is worse than that in the sub3 scheme. With considering the locality, the FASM system can achieve desired performance.

E. Concurrency Analysis

To verify the impact of concurrency on subsetting schemes, datasets were stored on multiple nodes using a round-robin distribution on the Lustre file system. We set the stripping units to 40 (that is, data blocks are striped over 40 OSTs). We varied the strip size as 1MBs, 2MBs, 5MBs, 10MBs and 50MBs separately to observe the impact. The result is shown in Fig. 10. The sub3 scheme achieved the best performance when the strip size is 5 MBs. This observation matches with our theoretical analysis. The performance under such a setting is 1.67 times faster than the worst case.

F. Real Application Test

We have performed tests with real application datasets from the BCCR model as well (Fig. 11). It can be observed that the FASM approach clearly reduced the response time in all cases. The FASM approach was able to achieve up to 3.5 times speedup across all tests.

V. CONCLUSION

Scientific datasets are used in many data intensive computing fields. In this study, we have shown how the integrated statistical metadata improves the query and analysis performance. We have also carried out initial analysis of the impact of various subsetting schemes. The evaluation results have indicated that the FASM approach is promising and can be potentially beneficial for many data-intensive scientific applications. We are continuing developing the FASM system. In the near future, we will investigate further to support resubsetting and regeneration of statistics at runtime.

REFERENCES