Leveraging STARE for Co-aligned Data Locality with netCDF and Python MPI

Kwo-Sen Kuo[1,2,3], Hongfeng Yu[4], Yu Pan[4], and Michael L Rilee[1,5]

[1]NASA Goddard Space Flight Center, Greenbelt, Maryland, USA
[2]ESSIC, University of Maryland, College Park, Maryland, USA
[3]Bayesics LLC, Bowie, Maryland, USA
[4]University of Nebraska, Lincoln, Nebraska, USA
[5] Rilee Systems Technologies LLC, Derwood, Maryland, USA

ABSTRACT

We have leveraged STARE indexing to package partitioned data chunks from diverse datasets into netCDF files, distributed them on a cluster of 16 lightweight nodes with their placements spatiotemporally co-aligned, and demonstrated a few integrative analyses using netCDF parallel I/O and Python MPI, with single-user performance and scalability comparable to, or even better than, that of a parallel array database management system (ADBMS) such as SciDB. However, records of the node location and STARE index ranges for each data chunk, similar to the chunk maps of SciDB, must be maintained and consulted by the I/O and analysis code for coordinating the analytic operations in parallel, in order to achieve the good performance and scalability.

Index Terms— Big Data, interoperability, parallel processing, scalability, data-intensive analysis

1. INTRODUCTION

Relative to the Moore's law [1], the productivity of geo-spatiotemporal data analysis has been comparably stagnant for decades, primarily due to the enormous volume and diversity of the data involved. The existing practice of packaging these data in files (albeit in only a few standard file formats) for dissemination causes tremendous waste in communication bandwidth utilization and storage-compute resource duplication. Moreover, while almost all of these (geo-spatiotemporal) data are expressed using the *array* data structure, the process of packaging them into files inevitably decouples the array indexing from corresponding geo-spatiotemporal coordinates (e.g. longitude-latitude and date-time), due to the differences in data resolutions as well as data models used. Analyses of these diverse datasets thus demand case-by-case considerations, seriously impacting interoperability and limiting scalability.

Our innovation, the SpatioTemporal Adaptive-Resolution Encoding (STARE; see section 2 for a description), is developed as a universal indexing scheme for all geo-spatiotemporal data, thereby establishing a one-to-one coupling between STARE index pair and geolocation & time (up to a precision of better than 10 cm in geolocation and millisecond in time). Since all geo-spatiotemporal data can be indexed consistently and uniformly with STARE indexing, STARE enables a consistent interpretation and achieves unparalleled interoperability among the vast varieties of these data.

In addition, since STARE also encodes data resolution, for both geolocation and time, in its indices, it not only enables efficient geo-spatiotemporal set operations (e.g. union, intersect, and difference) but also supports spatiotemporal data placement alignment when partitioning diverse datasets onto a distributed cluster environment. Such data placement alignment engenders the optimal data locality (aka data-compute affinity) for integrative analysis requiring spatiotemporal coincidence; that is, analyzing data of the same area and time from related but different datasets.

W have presented an integration of STARE, as an outcome of our previous effort [1], with the use of a parallel array database management system (ADBMS), SciDB, to further multiply its power. For scaling large volume, the obvious and only solution is parallel processing. Without variety scaling, however, the scaling achieved by parallel processing is at best piecemeal, i.e. one variety at a time. While the variety-scaling power of STARE through data placement alignment can minimize data movement and guarantee pleasingly parallel problems remain pleasingly parallel, integrating STARE with a parallel ADBMS enables high scalability of a more advanced class of parallel processing, i.e. distributed memory parallelization (DMP). The consistent partitioning of all geo-spatiotemporal data serves as a reusable domain decomposition, which the parallel ADBMS can take advantage repeatedly in a predictable manner for coordinating the communication needed for DMP.

In this paper, we introduce another integration of STARE with conventional array-oriented scientific data management and processing tools, i.e. netCDF and Python, instead of an ADBMS. Our design of STATE makes the deployment to netCDF (for data chunk packaging) straightforward and requires only marginal effort. For processing with Python MPI, records of chunk locations and STARE index ranges must be maintained and consulted during analysis operations. We demonstrate that, using netCDF parallel I/O and Python MPI, STARE again enables the optimal scaling of variety and, a

more thorough scaling of volume through DMP for geo-spatiotemporal data analysis.

In the following sections, we first describe STARE briefly in section 2. The advantageous features of STARE are outlined next in section 3. The integration of STARE and netCDF is described in section 4. We then introduce the experiments and their purposes in section 5. Section 6 concludes.

2. STARE

The SpatioTemporal Adaptive-Resolution Encoding, STARE, consists of two parts, a spatial and a temporal component, as summarily described below. More detailed description can be found in [3].

The spatial part of STARE uses a 64-bit integer and is based on the hierarchical triangular mesh (HTM) [4][5], which is a way to address the 2D angular space (i.e. the solid angle) of the spherical coordinate system using a hierarchy of spherical triangles. The mesh is generated following the procedure below:

1. Start with an octahedron inscribing a sphere.
2. Bisect each edge of its eight triangular facets.
3. Project the bisecting points to inscribe the sphere from its center to form 4 smaller spherical triangles.
4. Repeat from step 2, until a desired resolution (precision) is reached.

After the initial octahedron, each iteration from step 2 is termed a *quadfurcation*, i.e. division/branching into 4 parts.

The spatial index of STARE is a customized variant of the HTM with two distinctions. 1) While right-justified encoding is used for the original HTM indexing, we choose a left-justified encoding to facilitate spatial data placement alignment. 2) In addition, geolocation uncertainty (commensurate with data resolution) is added to the encoding using a few least-significant bits to facilitate set operations among diverse datasets [6].

Essentially, STARE's spatial index is a one-dimensional equivalent way (to the use of latitude-longitude) of specifying geolocation to a given uncertainty. For example, with 23 quadfurcations (i.e. at the 23rd depth level), a latitude-longitude coordinate is concisely and uniquely mapped to an integer with ~1-m uncertainty.

The temporal index of STARE also uses a 64-bit integer. It is also hierarchical but, to avoid unnecessary translations between temporal frameworks, it uses calendrical date/time units, such as year, month, week, day, hour, etc., to build the hierarchy. It is thus called hierarchical calendrical encoding (HCE). The least significant few bits of the index are also used to denote the approximate temporal resolution of the data.

3. ADVANTAGEOUS FEATURES OF STARE

Since STARE indexing is hierarchical and carries with it (approximate) data resolution information, it embodies many advantageous characteristics. We list some of the most important ones below.

First, STARE affords sophisticated and yet highly efficient set operations, including conditional subsetting, which samples a second dataset based on properties (usually filtered) of a first, and likely dissimilar, dataset. For example, one may wish to correlate cloud-top infrared brightness temperature with the presence and intensity of precipitation. With STARE, this sort of set operations is turned into fast operations on STARE-index integer intervals, as opposed to much more complicated and slower operations on floating-point latitude-longitude pairs.

Moreover, because each edge of a spherical triangle in HTM is a segment of a great circle, it is more straightforward to ascertain which hemisphere (delineated by the great circle) a given geolocation belongs to. This property can thus be utilized to quickly determine the set of STARE indices (even with varying quadfurcation levels) corresponding to a user specified region of interest (ROI).

The hierarchical nature of STARE also supports progressive visualization. That is, we may use coarser resolution (lower-level quadfurcation) and thus smaller data volume to rapidly render initial visualization and use progressively higher resolutions to refine the visualization until a desired quality is reached. Bandwidths in a data traffic chain, especially when low-bandwidth connections (e.g. Internet) are involved, can therefore be better utilized to provide a more pleasant user experience.

A disadvantage of STARE, however, is the lack of a straightforward way in specifying an overlap (aka halo) for neighboring partitions of data, which is important to performance for operations that are not pleasingly parallel. Determining neighboring STARE cells is straightforward, but requires a (small) tree traversal, as opposed to a simple index increment, that must then be convolved with the parallel computing platform's data distribution scheme.

4. INTEGRATION OF STARE AND NETCDF

In a conventional netCDF file, data elements are typically arranged according to certain attributes or indices. For example, one common practice is to use regular gridding in their spatial and/or temporal coordinates. In a distributed environment (e.g. a computing cluster), data elements often are partitioned into chunks and distributed among the computing nodes according to the order of the indices. However, it is hard to guarantee that data chunks for the same space-time are placed on the same node. We leverage the universality of STARE indexing with netCDF, to spatiotemporally co-align data chunk placements on the computing nodes.

Our design of STARE makes it easy to integrate with netCDF, as shown in Fig. 1. First, given an input netCDF file of a dataset, we compute a STARE index for each data element according its spatial and temporal coordinates, and add the STARE index as a new attribute into the netCDF file. Second, the file is partitioned into a number of data chunks according to the order of STARE indices and prescribed inter-

Table 1 Main properties of the datasets used in our experiments

	Spatial		Temporal		Remarks
	Resolution	Coverage	Resolution	Duration	
MERRA-2 (PRECTOT)	0.625°×0.5°	Global	1 hr	3 mon	Grid
NMQ	0.01°×0.01°	CONUS	5 min	3 mon	Grid
TRMM	4~5 km	Tropics	~	3 mon	Swath

vals in geolocation and time. The size of each chunk is specified as a parameter, which is set to contain 4096 data elements in each chunk in this work. Finally, the data chunks of a dataset are distributed among the computing nodes in a round-robin fashion.

This STARE-based partitioning and distributing approach has several advantages: First, it ensures that a dataset can be equally partitioned and distributed into data chucks, leading to a balanced workload on each computing node. Second, for input netCDF files of different varieties, their partitions on each computing node are co-aligned spatiotemporally; that is, data elements with the same STARE index are

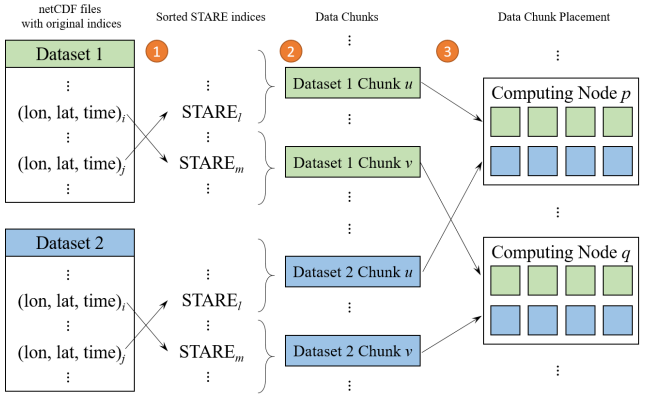

Fig. 1 STARE-based partitioning and distribution of netCDF files.

placed on the same node. This facilitate high performance for those geophysical data analyses (e.g., join queries) requiring spatiotemporal coincidence by minimizing cross-node data communications.

5. EXPERIMENTS AND RESULTS

We conduct our experiments using a cluster of 16 nodes. Each node has identical features: 32GB of main memory, an 8-core CPU and 9TB of local disk storage. They all run Centos 7 Linux operating system. The nodes are interconnected with 10 Gigabit Ethernet. We use the enterprise edition of SciDB release 16.9 and netCDF 4.3.3.1.

Two regular gridded datasets and one swath dataset for the period of Winter 2010 (i.e., from December 1st, 2009 to February 28th, 2010) are used to conduct the experiments. The first regular gridded dataset is extracted from an hourly dataset of the NASA Modern Era Retrospective-analysis for Research and Applications (MERRA-2) [7] data collection, while the second dataset is extracted from a reprocessed 5-minute National Mosaic and Multi-sensor QPE (NMQ, where QPE stands for quantitative precipitation estimate) [8]. The swath dataset, from NASA's Tropical Rainfall Measuring Mission (TRMM), derives vertical hydrometeor profiles using data from Precipitation Radar (PR) and TRMM Microwave Imager (TMI). Table 1 summarizes the main properties of the datasets. These datasets are placed on our cluster nodes using both SciDB and netCDF.

One of the performance comparisons we have conducted is the join query of the three datasets with STARE and regular gridding using both SciDB and netCDF stores. As shown in **Fig**. 2, the query time of our STARE-based approach increases marginally with the number of days, and is significantly lower than regular gridding. The performance of netCDF is slightly better than SciDB. This is possibly because of the overhead incurred by sophisticated data management functionalities of SciDB.

6. CONCLUSIONS

We have demonstrated that adapting STARE to existing parallel computing approaches is straightforward, requiring moderate implementation effort. Our work clearly shows that, when combined with STARE, both parallel ADBMS (e.g. SciDB) and existing data management/processing tools (e.g., netCDF+Python) can scale well and achieve their optimal efficiency in geo-data-intensive analysis to deliver the best value. However, only single-user mode has been tested with the netCDF+Python approach. The multiuser scalability of SciDB is expected to be better than that of netCDF+Python, because it is constructed as a multiuser system.

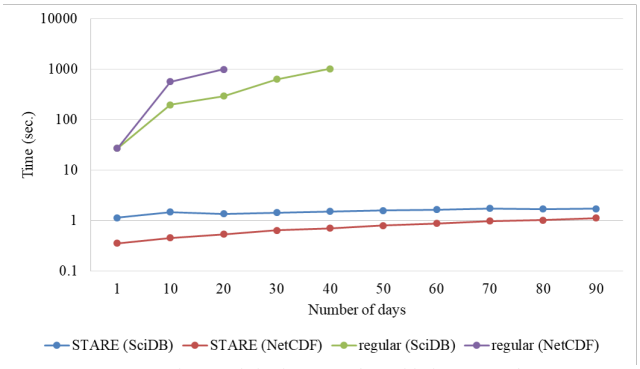

Fig. 2 Comparison of timing results of join query between STARE and regular gridding using three datasets with different number of days

Acknowledgement – We are grateful for the support provided by NASA Advancing Collaborative Connections for Earth System Science (ACCESS) program, NSF EarthCube program, and NASA Advanced Information Systems Technology (AIST) program that made this research and development possible.

REFERENCES

[1] https://en.wikipedia.org/wiki/Moore%27s_law

[2] Yu, Lina, Michael L. Rilee, Yu Pan, Feiyu Zhu, Kwo-Sen Kuo, and Hongfeng Yu. "Visual analytics with unparalleled variety scaling for big earth data." In Big Data (Big Data), 2017 IEEE International Conference on, pp. 514-521. IEEE, 2017.

[3] Kuo, K-S, and ML Rilee, "STARE – Toward unprecedented geo-data interoperability," *2017 Conference on Big Data from Space*. Toulouse, France. 28-30 November 2017.

[4] P.Z. Kunszt, A.S. Szalay, and A.R. Thakar, "The Hierarchical Triangular Mesh. In Mining the Sky" Proceedings of the MPA/ESO/MPE Workshop, Garching, Berlin/Heidelberg, Ch. 83, p631, 2001.

[5] A.S. Szalay, J. Gray, G. Fekete, P.Z. Kunszt, P. Kukol, and A. Thakar, "Indexing the Sphere with the Hierarchical Triangular Mesh," Micr. Res. Tech. Rpt., MSR-TR-2005-123, 2005.

[6] Rilee, M. L., K-S Kuo, T. Clune, A. Oloso, P. G. Brown, and H. Yu, "Addressing the big-earth-data variety challenge with the hierarchical triangular mesh," *2016 IEEE International Conference on Big Data* (Big Data, IEEE), 1006–1011, 2016.

[7] Bosilovich, M. G., R. Lucchesi, and M. Suarez. "MERRA-2: File specification GMAO Office Note No. 9 (Version 1.1).", 2016.

[8] Zhang, J., K. Howard, C. Langston, S. Vasiloff, B. Kaney, A. Arthur, S. Van Cooten, K. Kelleher, D. Kitzmiller, F. Ding et al., "National Mosaic and Multi-Sensor QPE (NMQ) system: Description, results, and future plans," Bulletin of the American Meteorological Society, vol. 92, no. 10, pp. 1321–1338, 2011.