A Review of Options for Storage and Access of Point Cloud Data in the Cloud

A White Paper prepared by the NASA ESDIS Standards Coordination Office

Table of Contents

Status of this Report

This report is published for information purposes only. This report does not specify an ESDS standard of any kind. This report has not undergone community review.
Distribution of this report is unlimited.

Change Explanation

This is the first version.

Copyright Notice

Suggested Citation

S.J.S. Khalsa, E.M. Armstrong, J. Hewson, J.F. Koch, S. Leslie, S.W. Olding, A. Doyle, "*A Review of Options for Storage and Access of Point Cloud Data in the Cloud",* NASA ESDIS Standards Coordination Office, February 2022. https://doi.org/10.5067/DOC/ESO/ESCO-PUB-003VERSION1

1. Abstract

Point cloud data, representing an extremely dense set of data points with known positions, have traditionally been the purview of commercial applications like laser scanners and photogrammetry software. However, the application of point cloud data in Earth science is growing, with applications using current space-based lidars being among the best examples. More point cloud data requirements are expected to arise for future NASA missions as well. Thus, understanding the implications of point cloud data and related services for storage, dissemination, and computation in cloud-based environments has been identified as a key emerging need by NASA's Earth Science Data and Information System (ESDIS) Project. This document is intended to supply basic information about the options available to providers, managers, and consumers of geospatial point cloud data.

For this review, the members of the ESDIS Standards Coordination Office (ESCO) assessed the current landscape of both existing and emerging point cloud data formats, content organization schemes[1], and services. We performed a qualitative assessment of the current best practices and use of point cloud data. This review did not attempt a quantitative assessment or trade study as we were not in a position to derive any type of performance metrics (though results were noted where they are available in the literature). As cloud-based data storage (with Amazon Web Services, Google Cloud, Microsoft Azure as leading examples) is becoming more prevalent for ongoing and future satellite missions, we also looked at the ramifications of point cloud data formats in a cloud environment and the types of services that will impact their access and manipulation. In conclusion, this review provides an understanding of the current suite of point cloud data formats, and guidance on the factors that data producers should take into account when bringing their point cloud data into the cloud. It is also meant to serve as a foundation for further investigations and assessments of point cloud formats.

2. Background

A point cloud is commonly defined as a 3D representation of the external surfaces of objects within some field of view, with each point having a set of X, Y and Z coordinates. Points are zero dimensional data, i.e. they refer to a specific position in space. This contrasts with 2D raster data acquired by imaging sensors which have a finite ground sample distance associated with each pixel.

A traditional and still common method of generating point cloud data is the use of lidar scanners mounted on tripods or aircraft (including drones and UAVs). These data, used for surveying surface elevation, forest canopies, buildings and other infrastructure, produce large numbers of points that are irregularly distributed in space, i.e. "clouds" of points.

A laser pulse is emitted by an instrument at a certain pulse repetition rate, and the time it takes for the pulse to reach a surface, be reflected and travel back to the instrument is used to evaluate

[1] We use the term *content organization scheme* to refer to any means for enhancing the addressing and access of elements contained in a digital object in the cloud (see Glossary)

the distance from the instrument to the bounce point. Each laser pulse has a finite duration, and the energy that returns to the instrument may come from different surfaces within the area illuminated by the laser. There are three basic modes of operation of laser scanners. The first mode of operation records the earliest return (shortest travel time) of each laser pulse. Energy received after the first return is assumed to be the result of multiple reflections or coming from other surfaces that are not of interest. The second type of sensor records the entire duration of the returned pulse; this is full waveform lidar. Full waveform lidar can distinguish the heights of different surfaces within the area illuminated by the pulse, such as the branches of a tree and the surface of the ground beneath. The third type of lidar instrument records the travel times of the individual photons contained in a laser pulse. These “photon counting” lidar systems can operate at extremely high pulse-repetition rates and are thus capable of creating highly detailed 3D maps. These three modes of operation yield point cloud data that can be analyzed using desktop 3D modeling software. Such software allows a user to inspect the data from different viewpoints, filter the data, and create polygonal meshes representing the viewed surfaces.

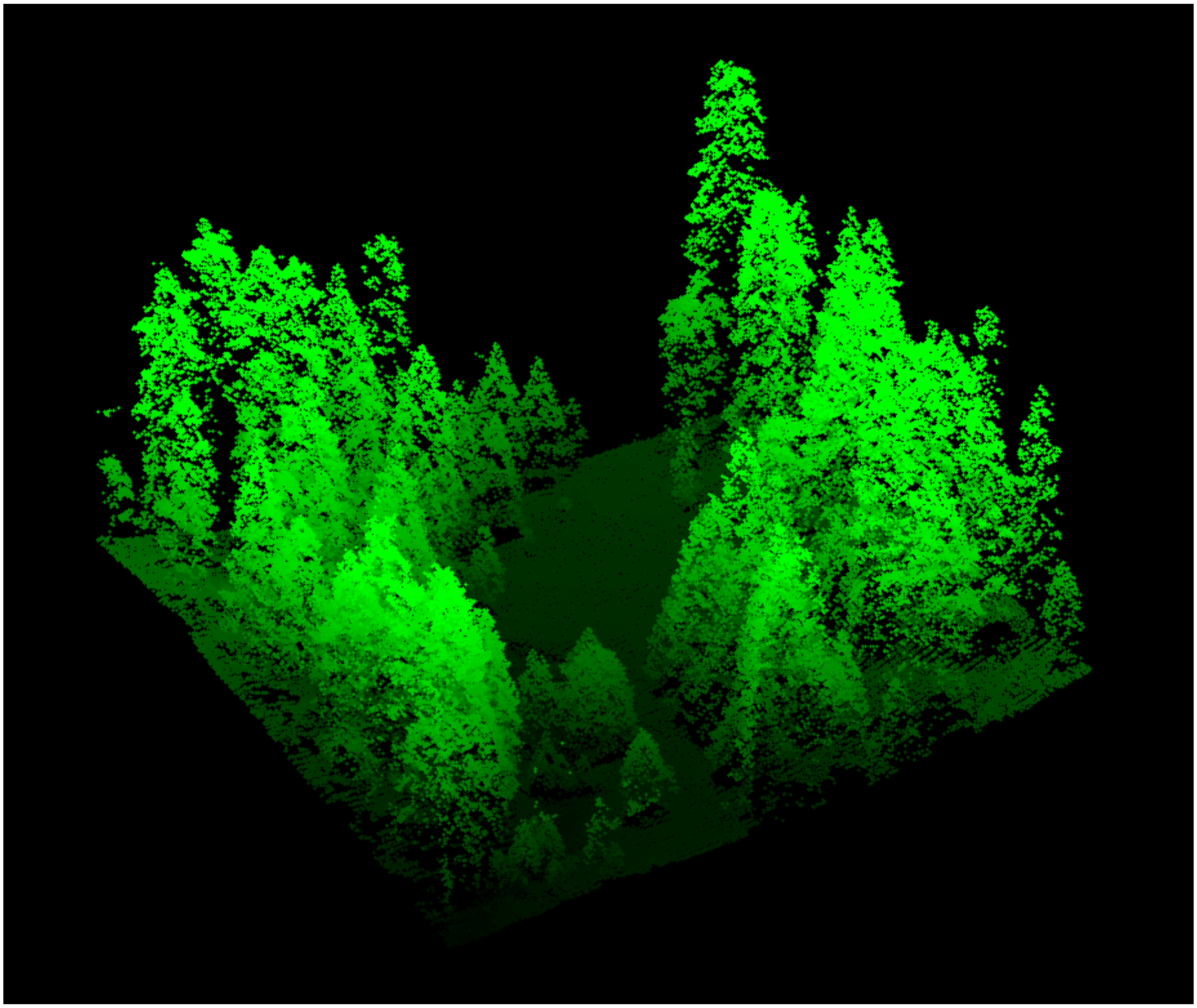

Figure 1. View of a point cloud acquired over a forest using airborne lidar.

Multiple considerations exist when analyzing point cloud data. For example, point cloud data can be more challenging to work with than raster data since the data are usually written serially in files, making it difficult to associate a location in a file with a location in space. The massive number of points typically stored in a file makes efficient indexing and random access to the data difficult. In fact, much of the work to-date in establishing and evaluating conventions and best practices for cloud optimization, as they pertain to Earth remote sensing data, has focused on raster data. Less work has been done with point cloud data. As new and existing NASA Earth Observing System Data and Information System (EOSDIS) data products are established in cloud-based environments where in-place analysis is the goal, it also becomes necessary to examine what cloud-optimized point cloud data should look like.

The goals of this document are to: provide a review of existing point cloud data formats, focusing on those most frequently used with data accessible through the NASA EOSDIS Distributed Active Archive Centers (DAACs) and other repositories; evaluate the suitability of different formats and content organization schemes for use in cloud-based environments; and, review emerging technologies for handling point cloud data in the cloud. By addressing each of these elements, we aim to provide the context for the type of performance testing that is needed to identify the most suitable analysis-ready, cloud-optimized data format(s) for point cloud data that satisfy storage, processing, and streaming demands in cloud-based environments.

This document provides a general view, based on both research and experience, to initiate a discussion about data formats and access services with an eye towards interoperability between and among future (near-term and long-term) cloud-based environments. It is intended for multiple audiences including DAAC data management professionals, science data producers, and EOSDIS developers. Within this context, it is paramount to consider the range of practitioners who may use point cloud data as well as the types of analyses they need to perform. We provide a list of potential practitioners and use cases in Table 2 of the Appendix.

3. Established Formats for Point Cloud Data

Traditionally, a *data format* refers to the way that information is encoded for storage in a digital storage medium. The specification for a data format will, at a minimum, describe the structure and organization of information in the file. It may also specify a filename extension and the convention used for storing metadata in the file, which may be as simple as a few bytes of header information, or a comprehensive set of metadata that makes the file fully self-describing.

Multiple data format considerations exist for point cloud data. For example, the raw output from a laser scanner must first be cleaned and edited before further processing and/or classification can be done. A data format should accommodate the point data as well as all the metadata necessary to interpret the data. In commercial laser scanning systems, the raw data is typically stored in native, proprietary formats. After processing by the vendor, these data are generally delivered in an ASCII format which is readable by virtually all software designed to work with point cloud data. However, storing point cloud data in ASCII format tends to produce very large files. Data formats based on binary storage, some employing compression, have been developed to reduce size and promote interoperability. Those most commonly employed by NASA Earth Science data products are described in later sections. Another consideration is that the format used for data storage may differ from the format used in transferring data to a user. This can become an important consideration for point cloud data, both in the context of cloud-based storage and processing, and in data streaming.

In other applications, point cloud data storage presents less of a consideration. For example, point clouds are increasingly used in machine vision and autonomous vehicle navigation, where processing of the data and subsequent information extraction is done in real time and, therefore, storage of the data in these environments is not as much of a concern.

A primary area of consideration in our review is point cloud data in cloud-based environments. In fact, the storage of point cloud data in the cloud represents a rapidly emerging field, especially as NASA migrates data and data operations to the cloud. For example, by the end of 2022 NASA's Earthdata cloud migration project, representing the largest such project in NASA to-date, will have already migrated Petabyte scale amounts of data to the cloud. As more remote sensing data are moved into the cloud, the implementation of data access, data analysis, and data archiving no longer fits the model of traditional storage. This presents novel considerations that require different approaches. Our review discusses several considerations relevant to point cloud data in the cloud, and provides further considerations in Section 5.

The complexities of cloud-based storage are generally abstracted from the user with the added benefit of providing scalable computation where the data source resides, thus reducing data movement or download and providing increased efficiencies. Conversely, for data providers, moving point cloud data to the cloud may require new approaches that consider packaging, availability of format specific software libraries, and the web services available for data discovery, manipulation, and transformation. Given point cloud data in the cloud represents a rapidly emerging field, considerations around technology and formats, etc., are also rapidly evolving.

Since point cloud data have been generated for different purposes using various technologies, no single format exists for its storage or transmission. Lidar scanners represented the first sensors to generate point cloud data and are still the most common source of point cloud data. Two branches of laser scanning remote sensing initially evolved - terrestrial (or tripod) laser scanning (TLS) and airborne laser scanning (ALS). TLS is typically used for mapping the exteriors and interiors of buildings or other localized applications, whereas ALS is used for wide-area infrastructure mapping, biomass assessment and other geophysical applications. More recently, space-based lidar instruments have been launched including NASA's Ice, Cloud, and land Elevation Satellite (ICESat) Geoscience Laser Altimeter System (GLAS) (2003-2010), its successor, Advanced Topographic Laser Altimeter System (ATLAS) on ICESat-2 (launched in 2018), and the Global Ecosystem Dynamics Investigation (GEDI) (installed on the International Space Station in 2018). While these instruments do not scan in the same manner as a TLS or ALS, the data they produce are still considered point cloud data.

Multiple point cloud data formats are used with data accessible through the NASA EOSDIS DAACs. A list of these formats is included below, and additional context is provided in Appendix Table 1. The formats in bold are described in subsequent sections of this document.

- **ASCII**
- **LAS/LAZ**
- **HDF5/NetCDF-4**

The following are some additional point cloud data formats commonly used by other repositories including USGS, NOAA, NGA, and OpenTopography:

- **LAZ (USGS 3DEP)**
- PTX/PTS (Leica)
- RDB (RIEGL)
- **SIPC/MIPC**
- **E57**

4. Review of Common Point Cloud Formats

This section reviews the following formats: ASCII, LAS/LAZ, E57, HDF5 and NetCDF-4, and SIPC/MIPC. For each format, we present the main characteristics followed by some considerations for cloud optimization. Additional considerations for cloud optimization are presented in the next section.

ASCII

Characteristics

Plain text (ASCII) files are a common format for storage and exchange of point cloud data where XYZ coordinate values are stored as a table with columns separated by a character such as pipe, comma, space or tab. There may be a table header containing metadata and there may be additional columns with per-shot information such as intensity. Typical file extensions are TXT, XYZ, PTS and PTX. The main drawbacks of text files are the required storage space and the compression limitations; only normal file compression can be applied [1].

Considerations

Several factors mitigate against the use of ASCII format for large point cloud datasets:

1. ASCII files are on the order of 2x bigger than the equivalent LAS files. This can be seen in a representative sample of 10 data files stored in both ASCII (.csv) and LAS (.las) formats (see the table at [2]).
2. Ingesting ASCII point cloud data is significantly slower than ingesting LAS format data as shown in the OpenTopography data ETL (Extract, Transform, Load) process (see section 4.1 [3]).

3. Data within ASCII files are typically ordered in increasing time, reflecting their origin as a real-time recording of data. Measurements are sometimes taken at regular intervals which would allow software to compute the data range within a file to find data from a specific temporal extent.

For this review, we searched for sources of NASA-originated and NASA-hosted point cloud data. [c.f., Appendix, Table 1]. The relatively small volume of ASCII formatted sources found, in comparison to the significantly larger volume of binary formatted data, suggests most cloud-based processing pipelines will be developed to handle large binary datasets vs ASCII formatted.

LAS/LAZ

Characteristics

A very common exchange format for lidar point cloud data is LAS (LASer file format), which was developed and is maintained by the American Society for Photogrammetry and Remote Sensing (ASPRS). LAS was developed as a public interchange file format for users of 3D point cloud data and the latest version, version 1.4, was approved in 2011, with a latest revision (R14) in 2019 [4] . The format is binary (little endian) and consists of a public header block, any number of Variable Length Records (often containing proprietary vendor data), the Point Data Records, which are usually time-ordered, and any number of (optional) Extended Variable Length Records. Different sets of attributes are specified in different configurations for the Point Data Records but all include attributes such as X, Y, Z coordinates, intensity, return number, and classification. The ability to store full waveform data was added in version 1.4.

Considerations

Reading a LAS file to extract only certain attributes is inefficient since all the attributes in a row must be read. Additionally, LAS files can be large. This has resulted in compressed versions of LAS including LAZ.

HDF5 and NetCDF-4

Characteristics

The HDF5 data model/container, and the closely aligned NetCDF-4 format, have been employed to store point cloud and similar data for several NASA missions including ICESat/ICESat-2 [5], IceBridge [6], GEDI [7], and the Ocean Melting Greenland (OMG) [8] project. The ICESat-2 mission contains the ATLAS (Advanced Topographic Laser Altimeter System) instrument that measures surface reflected laser pulses in the visible green spectrum to monitor earth elevations focusing on the cryosphere. Similarly, many of the instruments flown during IceBridge utilize lidar, radar and other airborne instruments in support of cryosphere science to monitor changes in polar ice. OMG contains many diverse instruments but includes an ocean vessel deployment of sonar instrumentation to measure seafloor bathymetry in the coastal regions of Greenland. The storage format for ICESat-2 is HDF5. For IceBridge, the instrument data (including lidar) are packaged into various formats including HDF5. OMG data are packaged exclusively into NetCDF-4.

Considerations

Essentially, the data model employed for these elevation measuring instruments allow the relevant “altitude” or “height” measurement to be encoded as single, one dimensional arrays as a function of observation time. The data model at times can be quite complex with hundreds of variables stored in a single file often organized to take advantage of HDF5 group structures. Additional examples of HDF5 packaging include the Sorted Pulse Data (SPD) format, a data and metadata storage and vocabulary community best practice [9]. However, no current NASA datasets appear to adhere to the SPD convention.

An advantage of the HDF5/NetCDF-4 container is the vast array of third party software tools and libraries that are available to the user community for data manipulation and visualization. Much of this usability is derived from the Climate Forecast (CF) metadata conventions that are typically employed in self describing HDF5/NetCDF-4 datasets especially for identifying variables with standard names, and documenting variable units, coordinate variables, projection information and data “type” and packaging. For example, the popular Panoply viewer is capable of reading variable and geo-referencing information to properly visualize the data from each of the missions listed above. However, a side effect is that large array sizes of over 100K points significantly slows the plotting, at times making even the simplest plots unwieldy.

OPeNDAP’s Hyrax server can efficiently access specific segments of NetCDF-4/HDF5 files, retrieving just the bytes associated with the desired variables. See the **OPeNDAP with DMR++** section for further discussion. However, the server typically needs the segments to be expressed as array indices, implying that something has translated the geospatial coordinates desired into indices. The server does have a geogrid() function that can subset gridded (L3/L4) datasets by simply providing geolocations of the subset corners but this capability is not available for point cloud data structures.

SIPC/MIPC

Characteristics

The Sensor Independent Point Cloud (SIPC) format was identified by the Harris Corporation as part of a trade study for the DoD that evaluated 20 point cloud storage formats. The study concluded that HDF5 was the optimal container format to be used by their SIPC Standard for analysis-ready point cloud data.

The Modality Independent Point Cloud (MIPC) format is similar to SIPC but is designed to accommodate any sensing modality (lidar, radar, Electro-Optical (EO) Imagery, etc.). Like SIPC, MIPC uses HDF5 as the container data format. It allows multiple modalities to be contained in a single file and multiple clouds per modality.

Considerations

SIPC can be spatially tiled (octree, quadtree, etc.) and the metadata content and data structures of MIPC provide efficient discovery and random access of data across multiple point clouds within a single file. Although the US National Geospatial Intelligence Agency has developed conversion tools and data viewers, we could find no evidence of adoption of SIPC and MIPC outside of US Defense agencies and we are aware of no serious independent comparisons of it relative to the other point cloud formats mentioned here.

E57

Characteristics

The American Society for Testing and Materials (ASTM) Committee E57 created the specification ASTM E2807 - 11(2019) Standard Specification for 3D Imaging Data Exchange for the exchange and storage of 3D point cloud data [10]. The ASTM E57 3D file format (aka E57) can store data produced by laser scanners as well as other systems including flash lidar systems, structured light scanners, stereo vision systems, and others. E57 also provides support for storing 2D images associated with a scan and all metadata associated with the 2D images and 3D data. In addition, the format supports organized "gridded" datasets which can be very valuable for long range spherical scans and can also support data in multiple coordinate systems. The format uses a combination of binary and eXtensible Markup Language (XML) formats and is considered vendor neutral, currently supported by multiple vendors including Leica, Optech, RIEGL, Trimble, and Z+F. There is an open source software implementation, libE57, in C++ that provides support for the format through two APIs (Foundation API and Simple API) [11]

Considerations

While E57 has a considerable user base, it is not a common format for airborne or spaceborne point cloud data. It is listed on the Drone Standards Info Portal, along with LAS, indicating that it is used for some applications. The specification must be purchased.

5. Considerations for Point Cloud Data in the Cloud

There has been a vast increase in the utilization of satellite imagery since Landsat data were made free and open. Further scientific discoveries and applications emerged as these data were made readily available in the cloud, processed to analysis-ready status and provided through services that permitted easy access to desired subsets. This expanding availability and uptake of Landsat data in the cloud has been followed by the availability of other types of imagery such as radar backscatter from the Sentinel-1 satellites.

Efforts to treat point cloud data in a similar fashion in terms of being readily available in the cloud, processed to analysis-ready status, and made available through services that permit efficient access to desired subsets, have only recently been undertaken. A “cloud-optimized” data source has several desirable characteristics that include:

- Being user accessible and readily downloadable
- Supporting incremental partial reads over HTTP
- Incorporating good compression
- Allowing dimension-selective reads
- Providing sufficient metadata for both human and machine consumption

- Supporting quadtree or octree, or similar, organization schemes

In this section we review multiple point cloud content organization schemes and services for the cloud including: EPT, COPC, Zarr, TileDB, 3D Tiles, Parquet, OPeNDAP with DMR++, STAC Point Cloud Extension, and STARE.

Entwine Point Tiles (EPT)

Entwine is an open-source content organization scheme and library for point cloud data [12]. Entwine Point Tiles (EPT) can work with any data format that is readable with PDAL (Point Data Abstraction Library [13]), and can read/write to a variety of cloud services such as Amazon S3 or Dropbox. It is completely lossless, so no points, metadata, or precision is discarded, even for terabyte-scale datasets. Entwine allows large lidar collections to be organized and served via HTTP clients over the internet. EPT uses an octree-based storage format and contains metadata in JSON. Direct access to the data is possible using the EPT reader that is available in PDAL.

Under a NASA Advancing Collaborative Connections for Earth System Science (ACCESS) project, Element84 performed a proof-of-concept study of placing ICESat-2 point cloud data (ATL06) into a cloud-native format [14]. Data elements of the ATL06 product (segment x, y, z locations and select metadata), which are provided in HDF5 format, were converted to LAZ and then indexed to EPT and uploaded to Amazon S3. From the 1,208 source HDF files, totaling 88GB, the intermediate LAZ files totaled 4.5 GB and the final 13,127 EPT files totaled 6.4 GB or about 7.25% of the original volume (albeit with many fewer variables). While this study showed major reductions in volume, it did not evaluate performance relative to other options for storing and accessing point cloud data.

Cloud-Optimized Point Cloud

Cloud-Optimized Point Cloud (COPC), a newly developed organization scheme, combines the characteristics of EPT with LAZ [15]. While EPT supports LAZ, by storing each chunk of data at each octree level as an individual LAZ file, this results in large EPT trees which can mean collections of millions of files. LAZ allows concatenation of individual LAZ files into a single, large LAZ file, providing essentially a dynamically-sized chunk table. This chunk table provides the lookups needed for an HTTP-based client to compute where to directly access and incrementally read data. Therefore, COPC takes advantage of specific characteristics of EPT and LAZ as it is similar to EPT but it clusters the octree as chunked LAZ. This has two distinct advantages. First, it can be read by any reader that reads LAZ and, second, it facilitates access to a spatial subset. Version 1 was released in October 2021.

Zarr

Zarr is an open source, emerging format for storing multidimensional array data in the cloud [16]. Zarr takes advantage of cloud object storage to implement a content organization scheme that stores chunks of individual files. The ability of Zarr to store chunks of datasets as separate objects in cloud object storage makes it efficient for parallel CPU access [17]. Zarr also allows all the metadata of a file to be in a single location, which then requires only one read of the metadata to determine the location of the data chunks. While storing data directly in Zarr format and accessing it via Zarr-enabled software is one option, it is also possible to store, for example, HDF5 chunk locations in the Zarr metadata and to use the Zarr library to access multiple chunks of an HDF5 file via byte-range requests in parallel. In fact, in adhoc testing, reading Zarr data with the Zarr library in the cloud was shown to be significantly faster than reading, for example, NetCDF-4/HDF5 data with NetCDF-4/HDF5 libraries [17]. However, if Zarr is used to access HDF chunks (chunked using Xarray) and HDF chunk locations are stored in Zarr metadata, such data can be accessed as efficiently as Zarr data with a Zarr library. It is worth noting that the adhoc testing involved slight modifications to both the Zarr and Xarray libraries.

Zarr would become suitable to store point cloud data if it used a recursive hierarchy of nested arrays, i.e., index the point cloud data with an R-Tree like index where the leaf nodes contain the data as a Zarr array, group, and chunk. Storing chunks contiguously would enable byte-range reads. Metadata could be consolidated and stored in a database or index which would allow planning of reads/writes in advance. Ideally, the metadata would be stored as an R-tree style index as well to support hierarchical sparse structures more efficiently.

Chunk size is known to impact performance. The Pangeo project, a project designed to facilitate *Big Data geoscience research*, reported chunk sizes ranging from 10-200MB, when reading Zarr data stored in the cloud using Dask [17]. Amazon’s Best Practices for S3 recommends making concurrent requests for byte ranges of an object at the granularity of 8–16 MB [18]. It should be noted that NetCDF-4/HDF5 datasets may not perform well if the chunk sizes are too small (less than a few MB) and therefore chunks should be created or rechunked with cloud-appropriate chunk sizes.

Recognizing that the storage format offered by NetCDF-4 may not be efficient for cloud-based environments, Unidata has recently added Zarr as a storage format for the NetCDF C library. This will allow any NetCDF-based code to read and write Zarr.

TileDB

TileDB is marketed as “the industry’s only universal database” [19]. As its name suggests, it is a database for managing any type of data (tabular, genomics, seismic, video, point cloud). TileDB stores all data, plus metadata, in multi-dimensional dense or sparse arrays, and unifies all spatial and temporal information in a way that allows efficient access to subsets. TileDB offers cloud-optimized writing/reading performance. TileDB Embedded is the open-source option from TileDB. It is a universal storage engine based on an embeddable library written in C / C++ that eliminates the need for an additional catalog service. TileDB Embedded models and efficiently stores all data as dense or sparse multi-dimensional arrays. It provides a common API and integrates with multiple APIs and tools. It supports updates natively, and employs fast spatial indexing (such as R-Trees for its tiles). With TileDB Embedded it is also possible to ingest a LAS point cloud file as a 3D TileDB sparse array with PDAL and run SQL queries on the data directly from TileDB.

TileDB supports sparse matrix data with space-filling curve organization and time-based checkpointing, which could be considered a more substantive exploded-storage approach than Zarr.

3D Tiles

3D Tiles was first introduced in 2015 by Cesium, which creates open APIs and open source software, and was accepted as an OGC Community Standard in 2019 [20]. The design of 3D Tiles supports streaming massive heterogeneous 3D geospatial datasets primarily for visualization applications. It consists of a hierarchical data structure and a set of tile formats, including point cloud, which deliver renderable content. A Feature Table, listing the contents, and a Batch Table capturing additional information, are also included with a tile. A 3D Tiles dataset may contain any combination of tile formats organized into a spatial data structure. 3D Tiles are declarative, extendable, and applicable to various types of 3D data. Cesium also developed Time-Dynamic Point Clouds for playback of time-dynamic point cloud data such as from autonomous vehicle lidar [21].

Indexed 3D Scene (I3S) 3D Tiles

ESRI developed the I3S format to support streaming of large 3D datasets (termed I3S datasets or ‘scene layers’) and is designed to be cloud, web and mobile friendly. It is based on JSON, REST and modern web standards, making it easy to handle, parse, and render by web and mobile clients. I3S is an open specification maintained by ESRI [22] and not dependent on ESRI software; it was adopted by OGC in 2017, I3S supports three types of large, heterogeneous datasets including mesh, 3D object, and 3D point.

Parquet

Parquet, part of the Hadoop ecosystem, is an open-source columnar storage format [23]. It can be read natively by Amazon S3 Select queries and Apache Spark [24]. Parquet is also able to capture rich metadata on a per-column basis and supports a wide array of data types and structures (including point cloud). Parquet is a column-based data storage format vs a row-based format (e.g. CSV) and, as such, it is best suited to situations where data can be expressed as a fixed-size tuple such as point cloud data written as x, y, z. This allows rapid database-style queries using both Hadoop and S3 Select, however the architecture to perform complex spatial queries beyond simple geographical bounding boxes is lacking. Additionally, Parquet’s column-based format facilitates applications where specific fields will be accessed. As an example application, Parquet has been used to store lidar point cloud data used in autonomous vehicle navigation.

In an EOSDIS study, Parquet received the lowest score for data integrity, as it neither supports internal file integrity checks nor fine-grained integrity checking in any particular way [24]. It also is not well supported by existing community tools, but was ranked as a “moderate” overall format choice for point cloud data.

Apache Arrow

Apache Arrow is a performance-optimized binary columnar memory layout specification for encoding vectors and table-like containers of flat and nested data [25]. The Arrow specification is designed to eliminate memory copies, align columnar data in memory to minimize cache misses, and take advantage of the latest SIMD (Single instruction, multiple data) and CPU operations on modern processors. It has been applied to represent spatial geometries but its capabilities to represent discrete spatial coordinate systems is less known [26]. Some of the many open source projects taking advantage of, or supporting, Apache Arrow capabilities include Apache Spark, pandas, Parquet and Graphistry [27].

OPeNDAP with DMR++

The OPeNDAP Hyrax server can subset data stored in HDF5/NetCDF-4 directly from a S3 or Google Cloud Store, using information in an ancillary DMR++ file [28]. DMR encodes the location of data directly, allowing direct access without the need for an API [29]. OPeNDAP allows reading of chunks from these files in parallel without having to retrieve and process the entire file locally. If the granule file contains multiple variables and only a subset of them are needed, DMR++ enabled software can retrieve just the bytes associated with the desired variables values. Likewise, if the source file contains metadata that maps array indices to geospatial coordinates, this can be used in conjunction with the DMR++ file to access only the chunks of the source file corresponding to a specified spatial region. In an experiment performed at the San Diego Supercomputer Center it was shown that a Hyrax server could access requested subsets of ICESat-2 point cloud data in HDF5 with associated DMR++ files from the cloud with access speeds comparable to reading the data from an optimized PostgreSQL database.

STAC Point Cloud Extension Specification

The SpatioTemporal Asset Catalog (STAC) family of specifications aim to standardize the way geospatial asset metadata is structured and queried [30]. A "spatiotemporal asset" is any file that represents information about the Earth at a certain place and time. The original focus was on scenes of satellite imagery, but the specifications now cover a broad variety of uses, including point cloud data.

The STAC point cloud extension adds fields to a STAC Item to enable STAC to more fully describe point cloud datasets [31]. This extension, developed in October, 2018, has been put forward for community feedback to continue to refine the specification.

SpatioTemporal Adaptive-Resolution Encoding (STARE)

STARE is a spatiotemporal hierarchical indexing scheme in which data elements are assigned 64-bit integers locating them in space and time [32]. The indices also specify the hierarchy level associated with the native spatial resolution of the data. STARE indices enable the co-location of diverse datasets allowing for efficient organization and analysis of data in their native format. STARE has a C++ library with a python API.

While STARE-based packaging enables spatiotemporal alignment of data from diverse sources for optimized parallel analytic operations, there is a cost to creating and storing the STARE indices, as well as a cost in converting latitude/longitude pairs into STARE indices when accessing data. To be considered a viable option for accessing point cloud data in a cloud environment, further studies are needed.

6. Summary and Next Steps

Given the multiple point cloud formats that exist, and the relatively nascent development of cloud-optimized point cloud data formats and content organization schemes, identifying and advocating for a specific point cloud solution requires further analysis. In addition to the point cloud data formats in use by NASA EOSDIS (see Appendix, Table 1), there are over 20 additional formats in use for storing point cloud data. Within NASA, the most typical formats used include HDF5, NetCDF-4, LAS, and LAZ. However, these formats, including LAS and LAZ, were created prior to widespread utilization of cloud environments, and designed primarily as data archive and distribution formats (via CD, FTP, HTTPS, etc.). Their key features include self-contained metadata, compactness, and maintenance of file integrity. Such key features are not necessarily beneficial or supported in cloud environments where compute-time considerations, object data stores and other factors are emphasized. To specifically support the use and optimization of point cloud data in cloud environments, newer formats/organization schemes and related data services have emerged including EPT and COPC. In addition, Zarr, TileDB, 3D Tiles, OPeNDAP with DMR++, STARE and STAC are all being experimented with to bring point cloud data into the cloud.

While several ongoing efforts exist to inform the discussion around the most appropriate solution for point cloud data in cloud environments, no single solution stands out as the most reasonable. Additionally, while some performance testing has been undertaken, these studies have been small in scale (Appendix, Table 3).

This review aimed to initiate a discussion about point cloud storage formats and data services, and to provide the context for the type of performance testing needed to identify the most suitable analysis-ready, cloud-optimized data format(s) that satisfy storage, processing, and streaming demands in cloud-based environments. As a result, several potential next steps have emerged. First, it will be important to continue to monitor community efforts and developments that are underway. For example, it will be crucial to survey community experts with the aim of achieving consensus on recommendations as well as determining the advantages/disadvantages of storing data as point cloud. This could be pursued through the development of a working group by the ESCO, and by actively engaging other repository managers, such as collaborators involved in the USGS 3D Elevation Program that recently announced the release of lidar data as EPT in Amazon S3 [33].

Further areas of study are needed for determining the feasibility of utilizing point cloud storage solutions for other types of non-regularly distributed data such as that coming from spaceborne conically scanning instruments (e.g. AMSR and SMAP), profiling instruments (e.g. AIRS, GPM) and sensors deployed on in situ sampling aircraft (e.g., aerosol and gas capture along a flight line). Lessons learned from existing and future NASA point cloud datasets should also be examined (e.g, for SWOT pixel cloud product, L2_HR_PIXC). Additionally, with the goal of more clearly quantifying the performance characteristics of point cloud data services, EOSDIS should identify opportunities to fund broader-scale performance and trade studies such as that recently performed by the EED2 contractors for a study on various cloud data format rapid extractions and analytics [24]. Facets to consider in such a study include tradeoffs on relative file size, internal/external compression and I/O performance. An additional consideration is adoption of the format by open source software libraries. Some suggested criteria for evaluation of cloud-optimized point cloud formats and data organization schemes are included in the Appendix. Finally, opportunities may emerge to organize point cloud workshops in coordination with future ESIP meetings to engage stakeholders, explore these concerns, and build community interest.

References

[1] Pirotti, F., 2019, “Open software and standards in the realm of laser scanning technology,” Open Geospatial Data, Software and Standards, 4:14, pp. 1-13. Available: https://doi.org/10.1186/s40965-019-0073-z [Accessed 13 October 2021].

[2] “ABoVE: Terrestrial Lidar Scanning Forest-Tundra Ecotone, Brooks Range, Alaska, 2016”. Available: https://daac.ornl.gov/cgi-bin/dsviewer.pl?ds_id=1782 [Accessed 13 October 2021].

[3] Krishnan, S., Crosby, C., Nangidam, V., Phan, M., Cowart, C., Baru, C., Arrowsmith, R., 2011, “OpenTopograhy: a services oriented architecture for community access to LIDAR topography,” Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications, Article No.: 7, pp. 1–8. Available: https://dl.acm.org/doi/pdf/10.1145/1999320.1999327 [Accessed 21 October 2021].

[4] The American Society for Photogrammetry & Remote Sensing, 2019, “LAS Specification 1.4 - R14”, Available: http://www.asprs.org/wp-content/uploads/2019/03/LAS_1_4_r14.pdf [Accessed 21 October 2021].

[5] Ice, Cloud, and land Elevation Satellite (ICESat). Available: https://icesat.gsfc.nasa.gov [Accessed 27 October 2021].

[6] IceBridge. Available: https://nsidc.org/data/icebridge [Accessed 27 October 2021].

[7] Global Ecosystem Dynamics Investigation (GEDI). Available: https://gedi.umd.edu/ [Accessed 27 October 2021].

[8] Ocean Melting Greenland (OMG) project. Available: https://podaac.jpl.nasa.gov/OMG [Accessed 27 October 2021].

[9] Bunting, P., Armston, J., Lucas, R., Clewley, D., 2011, “Sorted Pulse Data (SPD) Format: a new file structure for storing and processing LiDAR data”. Available: https://www.cabdirect.org/cabdirect/abstract/20123180616 [Accessed 27 October 2021].

[10] ASTM E2807 - 11(2019) Standard Specification for 3D Imaging Data Exchange, Version 1.0. Available: https://www.astm.org/Standards/E2807.htm [Accessed 27 October 2021].

[11] libE57. Available: libe57.org [Accessed 12 January 2022].

[12] Entwine Point Tile (EPT). Available: https://entwine.io/entwine-point-tile.html [Accessed 21 October 2021].

[13] Butler, H., Chambers, B., Hartzell, P., Glennie, C., 2020, “PDAL: An open source library for the processing and analysis of point clouds” Computers & Geosciences. 148. 104680. 10.1016/j.cageo.2020.104680.

[14] Skaggs, T.L., Pilone, D., Hanson, M., 2020, “Exploring the advantages of cloud native, easily consumable, scalable formats for downstream scientific exploitation of point cloud data”

Presented at the AGU Fall Meeting Abstracts, pp. IN031-0003. Available: https://agu.confex.com/agu/fm20/meetingapp.cgi/Paper/759095 [Accessed 13 October 2021].

[15] Cloud Optimized Point Cloud, 2021, "DRAFT Cloud Optimized Point Cloud Specification – 1.0 DRAFT* | COPC – Cloud Optimized Point Cloud". Available: https://copc.io/ [Accessed 28 October 2021].

[16] Zarr. Available https://zarr.readthedocs.io/en/stable/index.html [Accessed 21 October 2021].

[17] Signell, R., Jelenak, A., Readey, J., 2020, "Cloud-Performant NetCDF4/HDF5 Reading with the Zarr". Available: https://medium.com/pangeo/cloud-performant-reading-of-netcdf4-hdf5-data-using-the-zarr-library-1a95c5c92314 [Accessed 21 October 2021].

[18] AWS, 2019, "Best Practices Design Patterns: Optimizing Amazon S3 Performance: AWS Whitepaper". Available: https://d1.awsstatic.com/whitepapers/AmazonS3BestPractices.pdf [Accessed 21 October 2021].

[19] TileDB - Data management made universal. Available: https://tiledb.com/ [Accessed 27 October 2021].

[20] 3D Tiles | OGC. Available: https://www.ogc.org/standards/3DTiles [Accessed 27 October 2021].

[21] TimeDynamicPointCloud. Available: https://cesium.com/learn/cesiumjs/ref-doc/TimeDynamicPointCloud.html [Accessed 27 October 2021].

[22] I3S https://www.esri.com/arcgis-blog/products/arcgis-pro/3d-gis/3d-new-layer-types-and-capability-with-i3s-1-6/ [Accessed 27 January 2022].

[23] Parquet. Available: https://parquet.apache.org/ [Accessed 27 October 2021].

[24] Durbin, C., Quinn, P., Shum, D., 2020, "Task 51 - Cloud-Optimized Format Study EED2-TP-125, Revision 01 Technical Paper January 2020". Available: https://ntrs.nasa.gov/api/citations/20200001178/downloads/20200001178.pdf [Accessed 21 October 2021].

[25] Apache Arrow. Available: https://arrow.apache.org/ [Accessed 21 Jan 2022].

[26] Dunnington, Dewey, 2021 "Prototyping an Apache Arrow representation of geometry". Available: https://fishandwhistle.net/post/2021/prototyping-an-apache-arrow-representation-of-geometry/ [Accessed 21 Jan 2022].

[27] Projects and Product Names using Apache Arrow. Available: https://arrow.apache.org/powered_by/ [Accessed 21 Jan 2022].

[28] OPenDAP Hyrax data service. Available: https://www.opendap.org/software/hyrax-data-server [Accessed 29 October 2021].

[29] DMR++. Available: https://docs.opendap.org/index.php?title=DMR%2B%2B [Accessed 27 October 2021].

[30] SpatioTemporal Asset Catalog. Available: https://stacspec.org/ [Accessed 27 October 2021].

[31] Point Cloud Extension Specification. Available: https://github.com/stac-extensions/pointcloud#point-cloud-extension-specification [Accessed 27 October 2021].

[32] Rilee, M.L., Kuo, KS., Frew, J. *et al.*, 2020, "STARE into the future of GeoData integrative analysis." *Earth Sci Inform.* Available: https://doi.org/10.1007/s12145-021-00568-8 [Accessed 27 October 2021].

[33] USGS 3D Elevation Program. Available: https://www.usgs.gov/core-science-systems/ngp/3dep [Accessed 21 October 2021].

Authors

Edward M. Armstrong (NASA JPL PO.DAAC / Caltech)
Allan Doyle (Retired)
Jennifer Hewson (NASA ESDIS / SSAI)
Siri Jodha S. Khalsa (Univ. Colorado, Boulder, CIRES / NSIDC)
Joseph Koch (NASA ASDC DAAC / BAH)
Shannon Leslie (NASA NSIDC DAAC / Univ. Colorado, Boulder, CIRES)
Stephen Olding (NASA ESDIS / SSAI)

Document prepared by ESDIS Standards Coordination Office staff
eso-staff@lists.nasa.gov

Glossary

ACCESS – Advancing Collaborative Connections for Earth System Science
AIRS – Atmospheric Infrared Sounder
ALS – Airborne Laser Scanning
API – Application Programming Interface
ARD – Analysis-ready data (data that are prepared for a user to analyze without having to perform further processing of the data)
ASCII – American Standard Code for Information Interchange
ASPRS – American Society for Photogrammetry and Remote Sensing
ASTM – American Society for Testing and Materials
ATLAS – Advanced Topographic Laser Altimeter System
CF – Climate and Forecast (metadata conventions for describing Earth Science data)
Cloud Optimized Format – a file format having an internal organization that enables more efficient workflows in the cloud environment
Cloud Native - something designed and built to exploit the scale, elasticity, resiliency, and flexibility that the cloud provides
COG – Cloud Optimized GeoTIFF
Content organization scheme – (any means that enhances the addressing and access of elements contained in a digital object stored in the cloud)
COPC – Cloud-Optimized Point Cloud
CSV – Comma-Separated Values
DAAC – Distributed Active Archive System
Data Access Service – (network service for performing data access and remote transactions typically invoked via an API)
Data Format – the manner in which digital data is encoded for storage in a digital medium
DMR++ – Dataset Metadata Response (plus(++) additional information about where elements are located in the file)
E57 – File format for storing 3D data, such as point cloud, from the ASTM E57 Committee
EOSDIS – Earth Observing System Data and Information System
EPT – Entwine Point Tiles
ESDIS – Earth Science Data and Information System project
ESIP – Earth Science Information Partners
ESCO – ESDIS Standards Coordination Office
GEDI – Global Ecosystem Dynamics Investigation
GHRC DAAC – Global Hydrology Resource Center Distributed Active Archive Center
GLAS – Geoscience Laser Altimeter System
GPM – Global Precipitation Measurement
GPU – Graphics Processing Unit
HDF4 – Hierarchical Data Format Version 4
HDF5 – Hierarchical Data Format Version 5
ICESat – Ice, Cloud, and land Elevation Satellite
LAS – LASer file format
LAZ – LAS compressed file format
MIPC – Modality Independent Point Cloud

NetCDF – Network Common Data Form (classic)
NetCDF-4 – Network Common Data Form (version 4)
OMG – Ocean Melting Greenland project
OPeNDAP – Open-source Project for a Network Data Access Protocol
Parquet – columnar storage format supporting nested data structures; part of Hadoop open-source ecosystem
PDAL – Point Data Abstraction Library
PTS – Proprietary Leica file format
PTX – Proprietary Leica file format including registration information
RDB – REIGL Data Base file format
SIMD – Single Instruction, Multiple Data
SIPC – Sensor Independent Point Cloud file format
SPD – Sorted Pulse Data file format
STAC – SpatioTemporal Asset Catalog
STARE – SpatioTemporal Adaptive-Resolution Encoding
SWOT – Surface Water and Ocean Topography
TLS – Terrestrial Laser Scanning
XML – eXtensible Markup Language
Zarr – Format and library for storing multidimensional array data

Appendix

Table 1. Examples of NASA EOSDIS point cloud data products.

Data Product	Data format	Platform	Instrument Type	Tools/services for working w/ data
ATLAS/ICESat-2 L3A Land Ice Height	HDF5	Space Based	Lidar	ArcGIS, ENVI
GLAS/ICESat L1B Global Elevation Data	HDF5	Space Based	Lidar	
ABoVE LVIS L1B Geolocated Return Energy Waveforms	HDF5	Airborne	Lidar	
IceBridge LVIS L2 Geolocated Surface Elevation	ASCII	Airborne	Lidar	PDAL/ilvis2 reader
IceBridge Photon Counting Lidar L1B Subset Geolocated Photon Elevations	HDF5	Airborne	Lidar	
SnowEx20 Boise State University Terrestrial Laser Scanner (TLS) Point Cloud	LAZ	Tripod/Terrestrial	Lidar	
GEDI Level 2B Canopy Cover and Vertical Profile Metrics	HDF5	Space Based	Lidar	
ABoVE: Terrestrial Lidar Scanning Forest-Tundra Ecotone, Brooks Range, Alaska, 2016	LAS and ASCII	Tripod/Terrestrial	Lidar	PDAL/text reader
Pre-Delta-X: Channel Bathymetry of the Atchafalaya Basin, LA, USA, 2016	ASCII	Shipborne	Sonar	PDAL/text reader
OMG Conductivity Temperature Depth (CTD) Profiles	NetCDF-4	Shipborne	CTD	OPeNDAP
OMG Swath Gridded Multibeam Echo Sounding (MBES) Bathymetry	NetCDF-4	Shipborne	Sonar	OPeNDAP
OMG Airborne eXpendable Conductivity Temperature Depth (AXCTD) Profiles	NetCDF-4	Airborne	CTD	OPeNDAP
G-LiHT: Goddard's LiDAR,	LAS	Airborne	Lidar	LAS Tools

Hyperspectral & Thermal Imager				
CALIPSO Lidar Level 1B Profile, Validated Stage 1 V3-41	HDF4	Space Based	Lidar	
FIREX-AQ ER-2 Cloud Physics Lidar Remotely Sensed Data	HDF5	Airborne	Lidar	

Table 2. Practitioner type and use case of point cloud data.

Practitioner	Use Case
Civil engineer	I want to extract all ICESat-2 and GEDI surface elevation data from 2015 to present for all hillsides with a slope greater than 10 degrees along a highway corridor between two cities in California so that I can perform slope stability analysis for the highway maintenance department.
Forester	I want to use GEDI data in Adirondack Park to periodically monitor Hemlock trees for signs of subcanopy defoliation resulting from Hemlock Wooly Adelgid infestation so I can take preventative measures to protect nearby trees.
Forester	I want to use lidar for species identification so I can track invasive species distribution
Forester	I want to estimate above-ground biomass for fuel mapping so I can make informed decisions regarding fire hazard assessment and mitigation
Cryosphere researcher	I want to study and monitor sea ice thickness using satellite lidar data
USDA water management decision-maker	I want to combine ICESat-2 observations over inland water with in-situ hydrologic monitoring networks and other altimetry data to monitor reservoir heights and better understand drought and hydrological status that may affect food availability globally
Forest restoration specialist	I want to annually map vegetation density and tree mortality to support the UN Decade on Ecosystem Restoration

Table 3. Examples of small-scale performance tests of access services on point cloud data for cloud optimization.

Access Service	Performance test results
OPeNDAP with DMR++	In an experiment performed at the San Diego Supercomputer Center it was shown that a Hyrax server could access requested subsets of ICESat-2 point cloud data in HDF5 with associated DMR++ files from the cloud with access speeds comparable to reading the data from an optimized PostgreSQL database.
Entwine Point Tiles (EPT)	Under a NASA ACCESS grant, Element84 performed a proof-of-concept study of placing ICESat-2 point cloud data (ATL06) into a cloud-native format [14]. Data elements of the ATL06 product (segment x, y, z locations and select metadata), which are provided in HDF5 format, were converted to LAZ and then indexed to EPT and uploaded to Amazon S3. From the 1,208 source HDF files, totaling 88GB, the intermediate LAZ files totaled 4.5 GB and the final 13,127 EPT files totaled 6.4 GB or about 7.25% of the original volume (albeit with many fewer variables).
Parquet	In an EOSDIS study, Parquet received the lowest score for data integrity, as it neither supports internal file integrity checks nor fine-grained integrity checking in any particular way [24]. It also scored very low for supporting a variety of data types from the perspective of the spatiotemporal nature of the data, and is not well supported by existing community tools. But overall ranked “moderate” for point cloud data.

Proposed Evaluation Criteria to be Applied in Future Investigations

Methodology

This section describes proposed criteria for evaluation of different cloud-optimized point cloud formats, or data organization schemes, for use in cloud computing environments. The evaluation is primarily based on qualitative rather than quantitative criteria.

To truly assess the viability of different point cloud data formats for use in cloud computing (cloud-optimized point cloud formats), one needs to consider use cases, the cloud environment itself, and the overall system architecture. An evaluation should consider tool support of point cloud data, current community usage of different formats, and the potential of a given format to be cloud-optimized. These represent important factors in driving further adoption of a format or service. Additionally, it is important to highlight that there is a difference between *capability* and *performance*. Namely, whether a format or service has a particular feature versus how well that feature works when measured against other formats or services

Tool Support

- What tools provide support for the point cloud data format?
  - Tools available for writing data
  - Tools available for reading data
  - Tools available for visualizing data, esp. spatially
  - Tools available for performing data analysis and interrogation
- Is the format supported by popular third-party applications and software tools?
- Does the format support or enable programming environments that will support a wide user base and promote development of relevant tools and services?

Several geospatial analysis software packages (such as Esri ArcGIS, ENVI, QGIS, ERDAS IMAGINE) already provide some tool support for point cloud data, though the level of support differs (i.e., the capability to read point cloud data vs. the capability to visualize point cloud data vs. the capability to analyze and interrogate point cloud data).

Community Usage

- Does the point cloud data format provide for, and enable, the widest possible use of the data product, including potentially unforeseen new applications and research?
- Is the format widely used for similar data or similar data analysis workflows?
- Is the format compatible with formats used for past, long-term observations or models, and therefore provides for, or enables, more efficient data processing and interoperability?

- Does the format enable efficient data analysis workflows for climate research on a global scale over long time periods as well as enable potential use in weather, hydrology, agriculture, pollution, hazard events and other socio-economic applications?
- Does the format support local (as opposed to global) applications, given such applications often require frequent subsetting, reprojection and reformatting of the data for combination with in-situ point observations and physical models?
- Does the format support near real time data analysis operations, and at what scales?
- Does the format increase interoperability and is it compatible with data from other agencies, e.g., NOAA or USGS?

Other Considerations

- Is the format an open specification or not?
- Is it important for the format to accommodate the inclusion of more than x, y, z data, including for example, intensity, spectral information, data from other sensors, etc.?
- Can the format natively accommodate rich metadata, or is a metadata sidecar file required?
- Are there any characteristics that pose stewardship challenges?
- What are the constraints posed by the format?
- Are the format contents easily updated with new data (ability to reindex)?
- Do the formats lend themselves to user metric collection in the cloud?
- Do storage APIs (e.g., for s3://, azure://, gs://) present any barriers for clients to access data in these formats?